Ralph Bergmann   Sebastian Görg   Gilbert Müller (Eds.)

Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB

Trier, Germany, October 7− 9, 2015

Published at http://ceur-ws.org

Preface

LWA 2015 is a joint conference of four special interest groups of the German Computer Science Society (GI), addressing research in the areas of knowledge discovery and machine learning, information retrieval, database systems, and knowledge management. The German acronym LWA stands for “Lernen, Wissen, Adaption” (Learning, Knowledge, Adaptation). Following the tradition of the last years, LWA 2015 provides a joint forum for experienced and young researchers, to bring insights to recent trends, technologies and applications and to promote interaction among the special interest groups. The following special interest groups participate at LWA 2015:

- KDML (Knowledge Discovery, Data Mining and Machine Learning)
- FGWM (Knowledge Management)
- IR (Information Retrieval)
- FGDB (Database Systems)

The papers have been selected by independent program committees from the respective domains.

The program consists of several joint sessions which include contributions of interest for all conference participants. In addition, there are parallel sessions for individual workshops focussing on more specific topics. A poster session gives all presenters the opportunity to discuss their work in a broader context. Recent trends in the corresponding research areas are highlighted by four distinguished keynote speakers:

- Steffen Staab, University of Koblenz-Landau
- Thomas Seidl, RWTH Aachen
- Norbert Ritter, University of Hamburg
- Andreas Henrich, University of Bamberg

The accompanying social programme includes a guided city tour in the city centre of Trier followed by a conference dinner in the “Bitburger Wirtshaus”.

The organizers would like to thank the workshop chairs and their programme committees for their excellent work as well as the keynote speakers for their contribution to the success of LWA 2015. Finally, we gratefully acknowledge the generous support of our sponsors, in particular EMPOLIS, IBM, and SER.

We hope that LWA 2015 will be an inspiring event for all participants with lots of scientific exchange and discussions.

September 2015

Ralph Bergmann
Sebastian Görg
Gilbert Müller
(Editors, LWA'15)

Organization

LWA 2015 is hosted and organized by the research group “Business Information Systems II” at the University of Trier.

LWA 2015 General Organization

Ralph Bergmann, University of Trier (General Chair)
Maria Gindorf, University of Trier (Conference Secretary)
Gilbert Müller, University of Trier
Sebastian Görg, University of Trier

KDML 2015 Workshop Organization

Robert Jäschke, Leibniz Universität Hannover
Emmanuel Müller, Hasso Plattner Institut

FGWM 2015 Workshop Organization

Michael Leyer, University of Rostock
Christian Sauer, University of West London

IR 2015 Workshop Organization

Claus-Peter Klas, GESIS – Leibniz Institute for the Social Sciences
Ingo Frommholz, University of Bedfordshire

FGDB 2015 Workshop Organization

Thomas Ruf, Universität Erlangen-Nürnberg
Alfons Kemper, TU München

KDML 2015 Programme Committee

Martin Atzmueller	University of Kassel
Christian Bauckhage	Fraunhofer IAIS
Martin Becker	University of Würzburg
Daniel Bengs	German Institute for International Educational Research
Alexander Dallmann	University of Würzburg
Ernesto Diaz-Aviles	IBM Research
Stephan Doerfel	University of Kassel
Wouter Duivesteijn	TU Dortmund
Stephan Günnemann	Carnegie Mellon University
Matthias Hagen	Bauhaus University Weimar

Marwan Hassani	RWTH Aachen University
Alexander Hinneburg	Martin-Luther-University Halle-Wittenberg
Andreas Hotho	University of Würzburg
Frederik Janssen	TU Darmstadt
Robert Jäschke	Leibniz Universität Hannover
Kristian Kersting	TU Dortmund University
Marius Kloft	TU Berlin
Ralf Krestel	Hasso Plattner Institute
Florian Lemmerich	University of Würzburg
Ulf Leser	Humboldt-Universität zu Berlin
Eneldo Loza Mencía	TU Darmstadt
Hannes Mühleisen	Centrum Wiskunde & Informatica (CWI)
Emmanuel Müller	Hasso Plattner Institut
Thomas Niebler	University of Würzburg
Nico Piatkowski	TU Dortmund University
Achim Rettinger	Karlsruhe Institute of Technology
Elena Sapozhnikova	University of Konstanz
Ansgar Scherp	Leibniz Information Center for Economics (ZBW)
Ute Schmid	University of Bamberg
Erich Schubert	Ludwig-Maximilians-Universität München
Robin Senge	University of Marburg
Arthur Zimek	Ludwig-Maximilians-Universität München

FGWM 2015 Programme Committee

Klaus-Dieter Althoff	DFKI / University of Hildesheim
Kerstin Bach	Norwegian University of Science and Technology
Joachim Baumeister	denkbares GmbH
Axel Benjamins	Universität Osnabrück
Mareike Dornhöfer	Universität Siegen
Susanne Durst	University of Skövde
Michael Fellmann	University of Rostock
Dimitris Karagiannis	University of Vienna
Andrea Kohlhase	University of Applied Sciences Neu-Ulm
Michael Leyer	University of Rostock
Ronald Maier	University of Innsbruck
Alke Martens	University of Rostock
Mirjam Minor	Goethe University Frankfurt
Miltos Petridis	Brighton University
Ulrich Reimer	University of Applied Sciences St. Gallen
Jochen Reutelshöfer	denkbares GmbH
Bodo Rieger	Universität Osnabrück
Peter Rossbach	Frankfurt School of Finance & Management
Thomas Roth-Berghofer	University of West London
Kurt Sandkuhl	University of Rostock

Christian Sauer	University of West London
Sahar Vahdati	University of Bonn

IR 2015 Programme Committee

Reginald Ferber	Hochschule Darmstadt
Ingo Frommholz	University of Bedfordshire
Joachim Griesbaum	University of Hildesheim
Andreas Henrich	University of Bamberg
Daniel Hienert	GESIS – Leibniz Institute for the Social Sciences
Frank Hopfgartner	University of Glasgow
Claus-Peter Klas	GESIS – Leibniz Institute for the Social Sciences
Udo Kruschwitz	University of Essex
Johannes Leveling	Elsevier
Thomas Mandl	University of Hildesheim
Philipp Mayr	GESIS – Leibniz Institute for the Social Sciences
Peter Mutschke	GESIS – Leibniz Institute for the Social Sciences
Henning Müller	HES-SO
Ralf Schenkel	University of Passau
Christian Wolff	University of Regensburg
David Zellhoefer	Berlin State Library

FGDB 2015 Programme Committee

Tilo Balke	TU Braunschweig
Silke Eckstein	TU Braunschweig
Vera Kamp	PLATH GmbH
Alfons Kemper	TU München
Richard Lenz	University of Erlangen
Daniela Nicklas	University of Bamberg
Thomas Ruf	Universität Erlangen-Nürnberg
Knut Stolze	IBM Germany

Keynote Talks

Bias in the Social Web

Steffen Staab

University of Koblenz-Landau,
Campus Koblenz, Universitätsstraße 1
56070 Koblenz, Germany
staab@uni-koblenz.de
http://west.uni-koblenz.de/de/ueber-uns/team/

Abstract. An assumption commonly unchallenged in Social Media is that its open nature leads to a representative view of the world. There are (at least) two issues with this assumption. The first issue is that such representativeness may be harmful and may contradict social principles, e.g. non-discrimination against women or minorities. The second issue is that algorithms that work on such representation may be harmful and may introduce bias misrepresenting people or peoples preferences.
In this talk we want to overview the issue of bias occurring in the Social Web. We will consider a case study of liquid feedback, a direct democracy platform of the German pirate party as well as models of (non-)discriminating systems. As a conclusion of this talk we stipulate the need of Social Media systems to bias their working according to social norms and to publish the bias they introduce.

Fast Multimedia Stream Data Mining

Thomas Seidl

RWTH Aachen,
Templergraben 55, 52056 Aachen, Germany
seidl@informatik.rwth-aachen.de
http://dme.rwth-aachen.de/de/team/seidl

Abstract. In our days, huge and still increasing amounts of data are collected from scientific experiments, sensor and communication networks, technical processes, business operations and many other domains. Database and data mining techniques aim at efficiently analyzing these large and complex data to support new insights and decision making based on the extraction of regular or irregular patterns hidden in the data. Current research trends in data analytics are driven by the high volume, velocity, and variety of Big Data.

The talk discusses some challenges in the field. In recent developments for dynamic stream data mining, anytime algorithms play an important role. Novel hierarchical, statistical indexing structures including BayesTree and ClusTree allow for obtaining high quality results at any time while adapting themselves to varying stream velocities. Particular challenges occur when supervised and unsupervised mining tasks are faced with multimodal streams of complex multimedia objects.

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

Scalable Cloud Data Management with Polyglot Persistence

Norbert Ritter

University of Hamburg,
Mittelweg 177, 20148 Hamburg, Germany
ritter@informatik.uni-hamburg.de
https://vsis-www.informatik.uni-hamburg.de/vsis/

Abstract. The combination of database systems and cloud computing is extremely attractive: unlimited storage capacities, elastic scalability and as-a-service models seem to be within reach. This talk first gives a brief survey of existing solutions for cloud databases that evolved in the last years and provides classification and comparison. In practice however, several severe problems remain unsolved. Latency, scalable transactions, SLAs, multi-tenancy, abstract data modelling, elastic scalability and polyglot persistence pose daunting tasks for many applications. Therefore, we introduce Orestes, a database-as-a-service middleware which aims at tackling all these problems through an integrative approach. To this end, Orestes incorporates intelligent web caching, autonomous management of polyglot storage systems, and realtime processing of continuous queries in order to provide a comprehensive and effective infrastructure for cloud data management.

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

Digital Research Infrastructures for the Humanities Approaches and Challenges for Retrieval Applications

Andreas Henrich

University of Bamberg,
An der Weberei 5, 96047 Bamberg, Germany
andreas.henrich@uni-bamberg.de
http://www.uni-bamberg.de/minf/team/henrich/

Abstract. Digital research infrastructures for the humanities try to provide a working environment for scholars in the arts and humanities. One main concern is to facilitate access to existing collections in order to foster the reuse of these collections. The wide spectrum of available collections and the diverse information needs in this context brings up the requirement for retrieval applications supporting the discovery of interesting collections, a combined view over diverse collections, and in depth search in specific collections. Obviously, this has to be supported on a heterogeneous set of collections specifically tailored to the needs of scholars in the arts and humanities. The talk describes the scenario, presents approaches and prototypes, and discusses the challenges and limitations.

Joint Sessions

PubRec: Recommending Publications Based On Publicly Available Meta-Data

Anas Alzoghbi, Victor Anthony Arrascue Ayala,
Peter M. Fischer, and Georg Lausen

Department of Computer Science, University of Freiburg
Georges-Köhler-Allee 051, 79110 Freiburg, Germany
{alzoghba,arrascue,peter.fischer,lausen}@informatik.uni-freiburg.de

Abstract. In recent years we can observe a steady growth of scientific publications in increasingly diverse scientific fields. Current digital libraries and retrieval systems make searching these publications easy, but determining which of these are relevant for a specific person remains a challenge. This becomes even harder if we constrain ourselves to publicly available meta-data, as complete information (in particular the fulltext) is rarely accessible due to licensing issues. In this paper we propose to model researcher profile as a multivariate linear regression problem leveraging meta-data like abstracts and titles in order to achieve effective publication recommendation. We also evaluate the proposed approach and show its effectiveness compared with competing approaches.

Keywords: Recommender System, Scientific Paper Recommendation, Content-based Filtering, Multivariate Linear Regression, User Modelling

1 Introduction

Modern research is remarkably boosted by contemporary research-supporting tools. Thanks to digital libraries, researchers support their work by accessing a large part of the complete human knowledge with little effort. However, the sheer amount of rapidly published scientific publications overwhelms researchers with a large number of potentially relevant pieces of information. Recommender systems have been introduced as an effective tool in pointing researchers to important publications [5, 9, 10]. An approach that gained a lot of interest [2] extracts the interests of a user from the text of his/her publication list. In order to do so in an effective manner, full access to the textual content of research papers is needed. Yet, digital libraries typically provide only meta-data for publications including the publication date, title, keywords list and abstract. Although the availability of such information facilitates the problem, the usefulness of such a limited amount of information for paper recommendation is still unclear.

In this work we explore an approach to effectively perform paper recommendation utilizing such limited information. We present an adaptive factor to measure the interest extent of the active researcher in each of her/his previous publications; we apply a learning algorithm to fit a user model which in turn can be used to calculate the possible interest in a potential paper. Our contributions can be summarized as follows:

- An effective approach for modeling researchers interest that does not require access to the fulltext of the publication, but only freely available meta-data.
- An adaptive *anti-aging* factor that defines, for each researcher and publication, a personalized interest extent, so that older contributions have less impact.
- Preliminary results of comparing our approach against two state of the art recommendation techniques that considers full textual content.

The rest of this paper is organized as follows. In Section 2 we review work related to our approach. Section 3 presents the problem definition and outlines the presented approach. Section 4 demonstrates the profile building model employing the *anti-aging* factor. In Section 5 we explain the conducted experiments and discuss the results. Finally we conclude the paper in Section 6

2 Related Work

Research paper recommendation has been a hot topic for more than a decade. Several works addressed this problem proposing ideas from different recommendation directions [2]. Publication title and abstract were employed in [10] to build a user model using collaborative topic regression combining ideas from Collaborative Filtering and content analysis, but results were of varying quality. Nascimento et al. [5] use titles and abstracts as well. Users provide a *representative* paper that fits their interests, out of which keywords are extracted from the title and abstract. These keywords are then used to retrieve similar papers from digital libraries. We believe this is a limited approach as keywords from one publication are not enough to capture user interests. Sugiyama and Kan in [8, 9] employ a simplified variation of the Rocchio algorithm [7] to build a user profile utilizing all terms which appear in the fulltext of the user's authored publications, while they also incorporate terms from the citing and referenced papers. However, this approach suffers from the poor quality of the terms used and from the dependency on tools to extract text from pdf files which have well-known limitations. Above all, the authors assumed the availability of the full text of the publications which is rarely the case. In this work we optimize the use of the publicly available meta-data rather than relying on the full text of the publication. Moreover, we build a researcher interest model that can depict different affinity models of researchers.

3 PubRec Model

We propose a content-based research publications recommender (PubRec) that models both the active user (the researcher) and the candidate publications in terms of domain-related keywords. This section introduces the basic concepts of PubRec along with the formal problem definition.

3.1 Research Publication Profile

Digital libraries like ACM, IEEE, Springer, etc. publish meta-data about research publications publicly. Out of this meta-data, we are interested in title, abstract, keyword list and publication year. The first three can be effectively exploited to build a profile for each publication p as a keyword vector, which represents p in terms of domain-related keywords: $\overrightarrow{V_p} = \langle w_{p,k_1}, w_{p,k_2}, ..., w_{pk_n} \rangle$, where k_i is a domain-related keyword from the set of all keywords K, and w_{p,k_i} is the weight of k_i in p with range of $[0, 1]$.
All keywords from the keyword list are added to $\overrightarrow{V_p}$ with the maximum weight value of 1 by virtue of their source. As they are assigned to publications explicitly by the authors, we consider them the most precise domain description for the underlying publication. This list, however, contains usually up to 10 keywords, which is a small number for modeling a publication, thus, we aim to extend this list. Titles and abstracts hold a great essence of the ideas presented in publications. Therefore, we treat them as the second source of keywords and for each publication we apply keyword extraction from the concatenation of its title and abstract with weights correspond to the TF-IDF weighting scheme.

3.2 Researcher Profile

Given a researcher r with a set of her/his publications, we construct a researcher profile $\overrightarrow{V_r} = \langle s_{r,k_1}, s_{r,k_2}, ..., s_{r,k_n} \rangle$ such that $k_i \in K$ is a domain-related keyword, and s_{r,k_i} is the importance of k_i to r. Our proposed profile construction method ensures that r's *Interest Extent* (IE) in a publication p is achieved by computing the dot product between the researcher's vector and the publication's vector:

$$IE(\overrightarrow{V_r}, \overrightarrow{V_p}) = \overrightarrow{V_r} \cdot \overrightarrow{V_p} \tag{1}$$

3.3 Problem Definition

Our problem can be formally defined as:
Given a researcher r along with the corresponding set of publications P_r and a candidate set of publications P_{cand}, find k publications from P_{cand} with the maximum IE. The presented approach can be summarized in the following steps:

- First, we build the researcher profile $\overrightarrow{V_r}$ using previous publications by modeling the problem as a multivariate linear regression problem (Section 4)
- Each candidate publication $p \in P_{cand}$ is modeled as a keyword-vector $\overrightarrow{V_p}$
- We use Formula 1 to calculate $IE(\overrightarrow{V_r}, \overrightarrow{V_p})$ for candidate publication $p \in P_{cand}$
- Candidate publications are ordered by their Interest Extents and the top k are recommended to r.

4 Modeling Researcher Interest

We utilize researchers' publications to draw conclusions about their interests. A key aspect of PubRec consists in considering the different interest researchers have in their publications. After all, this interest might vary from paper to paper depending on several factors. Moreover, the importance of these factors vary among researchers. Thus, we believe that the publication age is an important factor in this regard since a five years old publication, for example, might not reflect the author's current interest as much a publication of the current year. Based on that, we introduce a scoring function for estimating the affinity of a researcher r towards one of her publications $p \in P_r$ by engaging the publication's age, which is expressed by the number of years elapsed after the publication's date and represented by σ in the following function:

$$IE_{r,p} = e^{\frac{-(\sigma)^2}{\lambda}}. \tag{2}$$

Here, λ is the researcher-specific *anti-aging factor*. As depicted in Figure 1, the curve of IE is plotted for three different values of λ: 4, 20 and 50. There we can see how λ regulates the steepness of this curve. As the values of λ increase, the curve becomes less steep and results in higher IE values for older publications. For example consider researcher r', the Interest Extent of r' for p', a 3 years old publication, can be modeled in three different ways upon three different values of λ: $IE_{r',p'} = 0.1$ for $\lambda = 4$, $IE_{r',p'} = 0.63$ for $\lambda = 20$ and $IE_{r',p'} = 0.83$ for $\lambda = 50$. This behavior helps in modeling different types of researchers based on their affinity model. Such that, researchers who tend to stick to the same research topics longer time are modeled using larger λ values compared to other researchers who tend to change their topics of interest more rapidly. Choosing the best λ for each researcher is done empirically in this work, but further investigations about the correlation between researcher characteristics and the optimal λ value are left for future work.

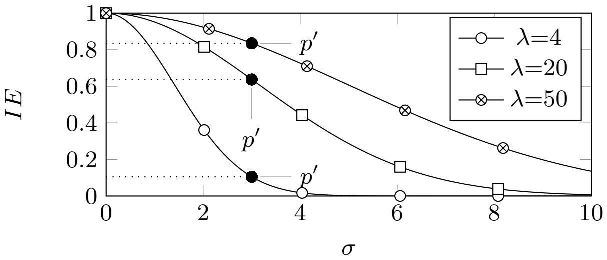


Fig. 1: *anti-aging factor* (*lambda*) impact on Interest Extent IE of researcher r'

4.1 Learning Researcher Profile

The second contribution in this work is to model the problem of measuring the importance of domain related keywords for a researcher r as a multivariate linear regression problem as follows: Given the set of r's publications P_r, for each publication $p_i \in P_r$ we build the underlying publication profile as described in section 3.1: $\overrightarrow{V_{p_i}} = \langle w_{p_i,k_1}, w_{p_i,k_2}, ..., w_{p_i,k_n} \rangle$. Furthermore, the Interest Extent IE_{r,p_i} is calculated using Formula 2 as shown in Figure 2. Let the set of keywords' weights of the paper p_i: $w_{p_i,k_1}, w_{p_i,k_2}, ..., w_{p_i,k_n}$ be the set of predictors related to the response variable IE_{r,p_i}, then the multivariate linear regression model [6] for p_i is defined as: $IE_{r,p_i} = \overrightarrow{\theta} \cdot \overrightarrow{V_{p_i}} = \theta_0 + \theta_1 w_{k_1} + ... + \theta_n w_{k_n}$.

$$\begin{array}{cccccccc|c}
 & & k_1 & k_2 & k_3 & & k_n & IE \\
\cline{3-8}
p_1 & \dashrightarrow & w_{1,1} & w_{1,2} & w_{1,3} & \dots & w_{1,n} & IE_{r,p_1} \\
p_2 & \dashrightarrow & w_{2,1} & w_{2,2} & w_{2,3} & \dots & w_{2,n} & IE_{r,p_1} \\
 & & & & \dots & & & \\
p_m & \dashrightarrow & w_{m,1} & w_{m,2} & w_{m,3} & \dots & w_{m,n} & IE_{r,p_m} \\
\cline{3-8}
\theta & \dashrightarrow & \theta_1 & \theta_2 & \theta_3 & \dots & \theta_n &
\end{array}$$

Fig. 2: Publications keyword vectors and Interest Extents for one researcher

Where $\overrightarrow{\theta}$ is the regression coefficient vector and $\theta_0, \theta_1, \ldots, \theta_n$ are the regression coefficients. Each coefficient value $\theta_j, j \in 1, \ldots n$ defines the relation between the researcher r and the keyword k_j, or in other words the importance of k_j for r. Consequently, the user profile is modeled by means of $\overrightarrow{\theta}$. Meaning, that in order to find the user profile $\overrightarrow{V_r}$, we should solve the previously mentioned regression problem and find the vector $\overrightarrow{\theta}$. This problem is solved by minimizing the cost function:

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (\overrightarrow{\theta} \cdot \overrightarrow{V_{p_i}} - IE_{r,p_i})^2$$

This is a well known optimization problem and there exist a couple of algorithms such as gradient descent or Normal equation to solve it [1]. We use an algorithm known for its efficiency, namely the L-BFGS algorithm [4].

5 Experiments

We conducted experiments to validate our approach and compared it against some state-of-the-art approaches. In the following we describe the used dataset along with the used evaluation metrics. Finally, we show and discuss the results.

5.1 Dataset

To evaluate the presented approach, we used the Scholarly publication Recommendation dataset[1]. It covers information about 50 anonymous researchers, enclosing their publication set, in addition to a set of publications of interest for each researcher. The interest lists are subsets of a larger collection of 100,531 publications called the candidate publications which is also provided.

[1] `https://www.comp.nus.edu.sg/~sugiyama/dataset2.html`

To the best of our knowledge, this is the only available dataset which provides the interest list for such a number of researchers. However, we had to resolve a major obstacle before we could use the dataset. That is, publications in the dataset are named by unique IDs without titles or author names, hence they cannot be identified and no meta-data was provided.

In order to make the dataset usable for our evaluation, we needed to identify the publications to be able to retrieve their meta-data. This was achieved by the following steps: (a) requesting and obtaining original pdf files from the dataset authors; (b) extracting publications' titles from the pdf files and using them to find publication identities within the DBLP[2] register; and finally (c) having the electronic edition pointer (ee) from DBLP publication's attributes, we retrieved needed information from corresponding publisher web site[3]. The result is a rich dataset that contains meta-data for 69,762 candidate publications, and more importantly the full publications and interest sets for 49 researchers. Lastly, for all publications in this dataset we applied the keyword extraction.

Keywords extraction and weighting. We use *Topia's Term Extractor*[4] because of its efficiency and usability. It is a tool that uses Parts-Of-Speech (POS) and statistical analysis to determine the terms and their strength in a given text. Yet we extended this tool in order to extract keywords with higher quality and make the best use out of the limited available resources. Our extensions to *Topia* are: (a) we apply post filtering on the resulting terms by choosing only those terms which appear in a white list of computer science terms; (b) the weights of extracted terms is calculated based on the normalized TF-IDF weighting scheme.

5.2 Evaluation metrics

We report the quality of our method with two important and widely adopted metrics for evaluating ranking algorithms in information retrieval. For the following metrics r is a researcher from the set of researchers R:

Mean Reciprocal Rank (MRR). MRR measures the method's quality by checking the first correct answer's position in the ranked result. For each researcher r, let p_r be the position of the first interesting publication from the recommended list, then MRR is calculated as $MRR = \frac{1}{|R|}\sum_{r \in R}\frac{1}{p_r}$

Normalized Discounted Cumulative Gain (nDCG)[3]. DCG@k indicates how good are the top k results of the ranked list. Typically in recommender systems DCG is measured for $k \in \{5, 10\}$ as users don't usually check recommended items beyond the 10th position. The DCG for a researcher r is calculated as $DCG_r@k = \sum_{i=1}^{k}\frac{2^{rel(i)}-1}{log_{10}(1+i)}$ where $rel(i)$ indicates the relevance of the item at position i: $rel(i) = 1$ if the i^{th} item is relevant and $rel(i) = 0$ otherwise. nDCG is the normalized score which takes values between 0 and 1, it is calculated as: $nDCG@k = \frac{DCG@k}{IDCG@k}$, where $IDCG@k$ is the DCG@k score of the ideal ranking, in which the top k items are relevant. In our case we report on the average $nDCG@k$ over all researchers for $k \in 5, 10$

[2] `http://dblp.uni-trier.de/`

[3] We received the ACM publications' meta-data from ACM as XML.

[4] `http://pypi.python.org/pypi/topia.termextract`

5.3 Experimental results

Using the previously described dataset and evaluation metrics, we conducted quality evaluations for our method with the following setup: given a set of candidate publications, and a set of researchers with their publications set, the system should correctly predict the interesting publications for each researcher. The results are demonstrated in the first row of Table 1. It shows that PubRec manages to achieve a high MRR score of 0.717. Looking deeper into the details of this metric by examining results for individual researchers gives more insights: for 29 out of 49 researchers the first relevant publication appeared at the first position of the recommended list, and at the second position for 7 researchers. We compared our approach with two state-of-the-art publication recommender systems [5, 9]. The work presented in [9] models each publication p using terms from p, from publications referenced by p and from publications that cite p. Additionally, the authors extended the set of citing publications by predicting potential citing publications. As our key contribution lies in utilizing only publicly available data, we implemented their core method[5] (Sogiyama) for modeling scientific publications considering only the terms which appear in the underlying publication. We compared PubRec against Sogiyama on two different setups: (a) Sogiyama using all terms appear in the full text of the publication[6]; (b) Sogiyama using our domain-related keywords. The results are shown in second and third row of the Table 1 respectively. In both setups PubRec outperforms Sogiyama in the three measured metrics. Furthermore, comparing our results with the results of Sogiyama as appeared in [9] where they assume the availability of the full text of the citing and referenced publications (5th row in Table 1) in addition to the potentially citing publications (4th row in Table 1), we find that our approach with such a limited available information is competitive and exhibits a reasonable trade-off between data-availability and recommendation quality. The last row in the table shows the scores of [5][7], where publications are modeled using N-grams extracted from titles and abstracts. Each user identifies a representative publication and the recommendation process turns into finding similar publications to the representative one by means of the cosine similarity.

6 Conclusion

We have proposed a novel approach on recommending scientific publications. By exploiting only publicly available meta-data from digital libraries the quality of the predictions is superior to state-of-the-art approaches, which require access to the full text of the paper. The focus is primarily on the user profiling, where a strategy to determine the trend of interests of a user in her own publications over time is integrated into a multivariate linear regression problem. The efficacy

[5] This method applies a light-weight variation of the Rocchio algorithm [7]

[6] The dataset provided by the authors of [9] contains all terms (not only domain-related terms) that appear in the full text of the publication.

[7] Values are taken from [9]

	MRR	nDCG@5	nDCG@10
PubRec	0.717	0.445	0.382
Sogiyama On their dataset	0.550	0.395	0.358
Sogiyama On PubRec dataset	0.577	0.345	0.285
Sugiyama and Kan [9]	0.793	0.579	0.577
Sugiyama and Kan [8]	0.751	0.525	0.479
Nascimento et al. [5]	0.438	0.336	0.308

Table 1: Recommendation accuracy comparison with other methods

of our approach is demonstrated by experiments on the Scholarly Paper Recommendation dataset. As future work, we plan to investigate the relationship between the *anti-aging* factor λ and researchers. Furthermore, we are interested in investigating the effects of enriching our modeling with meta-data from citing and referenced publications.

Acknowledgments. Work by Anas Alzoghbi was partially supported by the German Federal Ministry of Economics and Technology (BMWi) (KF2067905BZ). We thank Kazunari Sugiyama for his efforts in providing us with the complete Scholarly dataset. We also thank ACM for providing meta-data for our dataset.

References

1. Alpaydin, E.: Introduction to Machine Learning. Adaptive Computation and Machine Learning Series, MIT Press (2014)
2. Beel, J., Langer, S., Genzmehr, M., Gipp, B., Breitinger, C., Nürnberger, A.: Research paper recommender system evaluation: A quantitative literature survey. In: Proceedings of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation. RepSys '13 (2013)
3. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of ir techniques. ACM Trans. Inf. Syst. 20(4), 422–446 (Oct 2002)
4. Liu, D., Nocedal, J.: On the limited memory bfgs method for large scale optimization. Mathematical Programming 45(1-3), 503–528 (1989)
5. Nascimento, C., Laender, A.H., da Silva, A.S., Gonçalves, M.A.: A source independent framework for research paper recommendation. In: Proceedings of the 11th Annual International ACM/IEEE Joint Conference on Digital Libraries. JCDL '11, ACM (2011)
6. Rencher, A., Christensen, W.: Methods of Multivariate Analysis. Wiley Series in Probability and Statistics, Wiley (2012)
7. Rocchio, J.J.: Relevance feedback in information retrieval (1971)
8. Sugiyama, K., Kan, M.Y.: Scholarly paper recommendation via user's recent research interests. In: Proceedings of the 10th Annual Joint Conference on Digital Libraries. JCDL '10, ACM (2010)
9. Sugiyama, K., Kan, M.Y.: Exploiting potential citation papers in scholarly paper recommendation. In: Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries. ACM (2013)
10. Wang, C., Blei, D.M.: Collaborative topic modeling for recommending scientific articles. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '11 (2011)

Reality is not a game!

Extracting Semantics from Unconstrained Navigation on Wikipedia

Thomas Niebler, Daniel Schlör, Martin Becker, and Andreas Hotho

University of Wuerzburg
Tel.: +49-931-31-89094
{niebler,schloer,becker,hotho}@informatik.uni-wuerzburg.de

Abstract. Semantic relatedness between words has been successfully extracted from navigation on Wikipedia pages. However, the navigational data used in the corresponding works are sparse and expected to be biased since they have been collected in the context of games. In this paper, we raise this limitation and explore if semantic relatedness can also be extracted from unconstrained navigation. To this end, we first highlight structural differences between unconstrained navigation and game data. Then, we adapt a state of the art approach to extract semantic relatedness on Wikipedia paths. We apply this approach to transitions derived from two unconstrained navigation datasets as well as transitions from WikiGame and compare the results based on two common gold standards. We confirm expected structural differences when comparing unconstrained navigation with the paths collected by WikiGame. In line with this result, the mentioned state of the art approach for semantic extraction on navigation data does not yield good results for unconstrained navigation. Yet, we are able to derive a relatedness measure that performs well on both, unconstrained navigation data as well as game data. Overall, we show that unconstrained navigation data on Wikipedia is suited for extracting semantics.
The original paper is currently under review [Niebler et al(2015)].

References

Niebler et al(2015). Niebler T, Schlör D, Becker M, Hotho A (2015) Extracting semantics from unconstrained navigation on wikipedia. Künstliche Intelligenz - Special Issue: Semantic Web *currently under review*

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

Formalization and Preliminary Evaluation of a Pipeline for Text Extraction from Infographics

Falk Böschen[1] and Ansgar Scherp[1,2]

[1] Kiel University, Kiel, Germany
[2] ZBW - Leibniz Information Centre for Economics, Kiel, Germany
{fboe,asc}@informatik.uni-kiel.de

Abstract. We propose a pipeline for text extraction from infographics that makes use of a novel combination of data mining and computer vision techniques. The pipeline defines a sequence of steps to identify characters, cluster them into text lines, determine their rotation angle, and apply state-of-the-art OCR to recognize the text. In this paper, we formally define the pipeline and present its current implementation. In addition, we have conducted preliminary evaluations over a data corpus of 121 manually annotated infographics from a broad range of illustration types such as bar charts, pie charts, and line charts, maps, and others. We assess the results of our text extraction pipeline by comparing it with two baselines. Finally, we sketch an outline for future work and possibilities for improving the pipeline.

Keywords: infographics · OCR · multi-oriented text extraction · formalization

1 Introduction

Information graphics (short: *infographics*) are widely used to visualize core information like statistics, survey data or research results of scientific publications in a comprehensible manner. They contain information that is *frequently not present in the surrounding text* [3]. Current (web) retrieval systems do not consider this additional text information encoded in infographics. One reason might be the varying properties of text elements in infographics that makes it difficult to apply automated extraction techniques. First, information graphics contain text elements at various orientations. Second, text in infographics varies in font, size and emphasis and it comes in a wide range of colors on varying background colors.

Therefore, we propose a novel infographic processing pipeline that makes use of an improved combination of methods from data mining and computer vision to find and recognize text in information graphics. We evaluate on 121

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

infographics extracted from an open access corpus of scientific publications to demonstrate the effectiveness of our approach. It significantly outperforms two baselines based on the open source OCR engine Tesseract[3].

Subsequently, we discuss the related work. Section 3 presents our pipeline for text extraction and Section 4 specifies the experiment set-up and dataset used. The results regarding our OCR accuracy are presented in Section 5 and discussed in Section 6.

2 Related Work

Research on analyzing infographics is commonly conducted on classifying the information graphics into their diagram type [27] or separating the text from graphical elements [1], [6], [21]. Information graphics show a variety in appearance, which makes such classifications challenging. Thus, many researchers focus on specific types of infographics, e. g., extracting text and graphics from 2D plots using layout information [14]. Other works intend to extract the conveyed message (category) of an infographic [16]. Many research works focus on bar charts, pie charts and line charts when extracting text and graphical symbols [5], reengineer the original data [7], [22], or determine the infographic's core-message [4] to render it in a different modality or make it accessible to visually impaired users.

In any case, one requires clean and accurate OCR results for more complex processing steps, e. g. determining a message. Therefore, they use manually entered text. A different approach [13], [15] to make infographics available to sight impaired users is to translate infographics into Braille, the tactile language, which requires text extraction and layout analysis. This research is similar to our approach but relies on a semi-automatic approach which requires several minutes of human interaction per infographic. Furthermore their approach is challenged by image noise and their supervised character detection algorithm works under the assumption that the text has a unified style, i. e., font, size, and others. Another more specialized approach for mathematical figures [25] describes a pipeline for (mathematical-)text and graphic separation, but only for line graphs and the evaluation corpus is very small and they do not conduct any kind of OCR to verify the results. The assumption to automatically generate high-quality OCR on infographics with today's tools is certainly far-fetched.

3 TX Processing Pipeline

Our Text eXtraction from infographics (short: TX) pipeline consists of five steps plus a final evaluation step as shown in Figure 1. It combines certain ideas from related research [11], [13], [21] to build an automated pipeline which takes an infographic as input and returns all contained text. An initial version of our pipeline was briefly presented in [2]. Here we elaborate in detail on the steps of the pipeline, formalize it, and extend our evaluation. Given the heterogeneous

[3] `https://github.com/tesseract-ocr`, last access: Sep 07, 2015

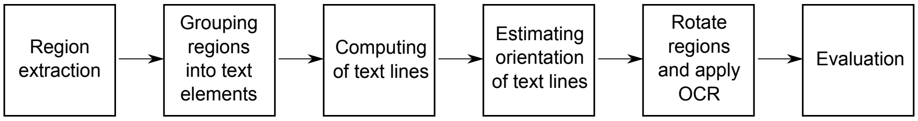

Region extraction
Grouping regions into text elements
Computing of text lines
Estimating orientation of text lines
Rotate regions and apply OCR
Evaluation

Fig. 1: Novel processing pipeline for text extraction from infographics

research field, a formalization is required to map the related work for a thorough comparison and assessment. In our pipeline, an information graphic I is defined as a set of pixels P with $p = (x, y) \in P \wedge x \in \{1 \,.\,. \, width(I)\} \wedge y \in \{1 \,.\,. \, height(I)\}$ where the latter two are integer arrays. The color information of each pixel p is defined by a function $\varPsi : P \rightarrow S$, where S is a color space. We use this information implicitly during our pipeline and use multiple $\varPsi$ functions to map to certain color spaces (e.g. RGB, grey scale,...). A set of text elements T is generated from P by applying the text extraction function $\varUpsilon$:

$$\varUpsilon : P, \varPsi \rightarrow T \tag{1}$$

Each text element $\tau \in T$ is a sequence of regular expressions ω_i specified as $\tau =< \omega_1, ..., \omega_n >$, separated by blank space characters, and with $\omega =$ [A-Za-z0-9!"§$%&/()=?´°{[]}\‘+-*,.;:|’#@_~<>€é£©®¥¢]*. In the following, we break down the formalization of $\varUpsilon$ into five sub-functions υ_j, one function for each step in our pipeline. We define $\varUpsilon$ as a composition:

$$\varUpsilon := \upsilon_5 \circ \upsilon_4 \circ \upsilon_3 \circ \upsilon_2 \circ \upsilon_1 \tag{2}$$

An overview of the notation used in this paper can be found in Table 1.

Table 1: Symbol notation used in this paper to formalize the TX pipeline

$\varUpsilon$, υ_j	text extraction function $\varUpsilon$ and its sub-functions υ_j
P , p	set of pixels P and individual pixel $p \in P$
R , r	set of regions R and individual region $r \in R$
C , c	a clustering C and individual cluster $c \in C$
C' , c'	a set of text lines C' and individual text line $c' \in C'$
$\varOmega$, ω	a set of words $\varOmega$ and individual word $\omega \in \varOmega$
A , α	set of text line orientations A and individual orientation $\alpha \in A$
T , τ	set of text elements T and individual text element $\tau \in T$

(1)Region extraction: The first step is to compute a set of disjoint regions R from the infographic's pixel set P using adaptive binarization and Connected Component Labeling [20]. This step is formally defined as:

$$\upsilon_1 : P \rightarrow R, R := \{r | r \subset P \wedge r \neq \emptyset \wedge \forall i, j, \;\; i \neq j : r_i \cap r_j = \emptyset\} \tag{3}$$

Each region $r \in R$ is a set of pixels forming a connected space, i.e. each region has a single outer boundary, but may contain multiple inner boundaries (holes).

Furthermore, the constraints in equation 3 ensure that all regions are non-empty and disjoint. First, we perform a newly-developed hierarchical, adaptive binarization that splits the infographic into tiles. The novelty of this approach is that it computes individual local thresholds to preserve the contours of all elements. This is based on the assumption that the relevant elements of an infographic are distinguishable through their edges. We start with a subdivision of the original image into four tiles by halving its height and width. For each tile, we apply the popular Sobel operator [24] to determine the edges. We compute the Hausdorff distance [9] over the edges of the current tiles and their parent tile. We further subdivide a tile, by halving its height and width, if a certain empirical value is not reached. A threshold for each tile is computed with Otsu's method [18] and the final threshold per pixel is the average of all thresholds for that pixel. This procedure appeared to be more noise tolerant and outperformed the usual methods, e. g., fixed threshold or histogram, during preliminary tests. The resulting binary image is labeled using the Connected Component Labeling method. This method iterates over a binary image and computes regions based on the pixel neighborhood giving each region a unique label. From the binary image, we compute for each region r the relevant image moments [10] m_{pq} as defined by:

$$m_{pq} = \sum_{x}\sum_{y} x^p y^q \Psi \qquad \text{with} \quad p, q = 0, 1, 2, \ldots \tag{4}$$

Please note that p, q hereby denote the p, q^{th} moment and may not be mistaken with the notation used in the remaining paper. For binary images, Ψ takes the values 0 or 1 and therefore only pixels contained in a region are considered for the computation of the moments. Using the first-order moments, we can compute each regions center of mass. Afterwards, we apply simple heuristics to perform an initial filtering. We discard all regions that fulfill the following constraints: (a) Either width or height of the region's bounding box are above average width/height plus 3 times standard deviation (e.g. axes) or (b) bounding box is smaller than 0.001% of the infographic's size (noise) as well as (c) elements occupying more than 80% of their bounding box (e.g. legend symbols). The function υ_1 generates a set of regions R, which can be categorized into "text elements" and "graphic symbols", the two types of elements in an infographic. Thus, in a next step we need to separate good candidates for text elements from other graphical symbols.

(2) Grouping regions to text elements: The second step computes a clustering C from the set of regions R by using DBSCAN [26] on the regions' features:

$$\upsilon_2 : R \rightarrow C, \quad C := \{c \subseteq R | c \neq \emptyset \wedge \forall i, j,\ i \neq j : c_i \cap c_j = \emptyset\} \tag{5}$$

Each cluster $c \in C$ is a subset of the regions R and all cluster are disjoint. For each region, the calculated feature vector comprises the x/y-coordinates of the region's center of mass, the width and height of its bounding box, and its mass-to-area ratio. Due to the huge variety of infographics, we apply the density-based hard clustering algorithm DBSCAN to categorize regions into text elements or noise (graphic symbols and others). This step outputs a clustering C where each cluster is a set of regions representing a candidate text element. We assume that these cluster contain only text while all graphical symbols are classified as noise.

(3) Computing of text lines: In this step, we generate a set of text lines C' on the clustering C by further subdividing each cluster $c \in C$. A text line c' is a set of regions that forms a single line, i.e. the OCR output for these regions is a single line of text. Each clustering c instead may generate multiple lines of text when processed by an OCR engine and therefore may implicitly contain other white space characters. To this end, we apply a second clustering based on a Minimum Spanning Tree (MST) [26] on top of the DBSCAN results, since clusters created by DBSCAN do not necessarily represent text lines. We compute a forest of Minimum Spanning Trees, one MST for each DBSCAN cluster. By splitting up the MST, a set of text lines for each cluster will be built. The rationale is that regions belonging to the same text lines a) tend to be closer together (than other regions) and b) the edges between those regions are of similar orientation. This is defined as:

$$\upsilon_3 : C \to C', \quad C' := \{c' \subseteq c | c \in C \wedge c' \neq \emptyset \wedge \forall i, j,\ i \neq j : c'_i \cap c'_j = \emptyset\} \tag{6}$$

Each text line $c' \in C'$ contains a subset of the regions of a specific cluster $c \in C$. Again, all text lines are non-empty and disjoint. For each cluster, the MST is built using the regions' center of mass coordinates which are the first two elements of the feature vectors computed in Step 2. We compute a histogram over the angles between the edges in the tree and discard those edges that differ from the main orientation. The orientation outliers are estimated from the angle histogram by finding the maximal occurring orientation and defining an empirical estimated range of ±60 degrees, where everything outside is an outlier.

(4) Estimating the orientation of text lines: In Step 4, we compute an orientation $\alpha \in A$ for each text line $c' \in C'$ so that we can rotate each line into horizontal orientation for OCR. This can be formalized as:

$$\upsilon_4 : C' \to C' \times A, \quad A := \mathbb{Z} \cap [-90, 90] \tag{7}$$

Every orientation angle $\alpha \in A$ for a text line c' can have an integer value from -90 to 90 degree. While the MST used in the previous step can well produce potential text lines, it is not well suited for estimating the orientation of text lines as it is constructed on the center of mass coordinates which differ from region to region. Thus, we apply a standard Hough line transformation [12] to estimate the actual text orientation. During the Hough transformation, the coordinates of the center of mass of each element are transformed into a line in Hough space, which is defined by angle and distance to origin, creating a maximal intersection at the lines' orientation. This computation is robust with regard to a small number of outliers that are not part of the main orientation.

(5) Rotate regions and apply OCR: The final step rotates the text lines along an angle of $-\alpha$ in order to apply a standard OCR tool. It is defined as:

$$\upsilon_5 : C' \times A \to T \tag{8}$$

We cut sub-images from the original graphic using the text lines C' from v_3, rotate them based on their orientation A from v_4 and finally apply OCR.

Step 6, the evaluation of the results, is described in detail below.

4 Evaluation Setup

We assess the results of our pipeline TX by comparing it with two baselines based on Tesseract, a state-of-the-art OCR engine. In our evaluation, we compute the performance over 1-,2- and 3-grams as well as words. During the evaluation, we match the results of TX and the baselines with some gold standard. Both, the position of the text elements as well as their orientation are considered in this process. We use different evaluation metrics as described in Section 4.4.

4.1 Dataset and Gold Standard

Our initial corpus for evaluating our pipeline consists of 121 infographics, which are manually labeled to create our gold standard. Those 121 infographics were randomly retrieved from an open access corpus of 288,000 economics publications. 200,000 candidates for infographics were extracted from these publications. All selected candidates have a width and height between 500 and 2000 pixel, since images below 500 most likely do not contain text of sufficient size and images above 2000 pixel appear to be full page scans in many cases. From the candidate set, we randomly picked images - one at a time - and presented them to a human viewer to confirm that it is an infographic. We developed a labeling tool to manually define text elements in infographics for the generation of our gold standard. For each text element we recorded its position, dimension, rotation and its alpha-numeric content. Please note that we considered using existing datasets like the 880 infographics from the University of Delaware[4], but they were incomplete or of poor quality.

4.2 Baselines

Today's tools are incapable of extracting text from arbitrary infographics. Even approaches from recent research works, as presented in Section 2, are too restrictive to be applicable on information graphics in general. This holds also for specialized research like rotation-invariant OCR [17], [19]. Since no specialized tools exist that could be used as a baseline, we rely on Tesseract, the state-of-the-art OCR engine, as our initial baseline (BL-1). It is reasonable to use this baseline, since Tesseract supports a rotation margin of ±15° [23] and is capable of detecting text rotated at ±90° due to its integrated layout analysis. Since infographics often contain text at specific orientations (0°,±45°,±90°), we also apply a second baseline. This second baseline (BL-2) consists of multiple runs of Tesseract with the rotated infographic at the above specified angles. We combine the five results from the different orientations by merging the results between those sets and in case of overlaps we take the element with greatest width.

[4] `http://ir.cis.udel.edu/~moraes/udgraphs/`, last access: Sep 07, 2015

4.3 Mapping to Gold Standard

The most accurate approach to compare OCR results with the gold standard would be to evaluate the results on the level of individual characters. Our pipeline, the baselines and the gold standard generate their output on varying levels. Only our pipeline supports the output of individual character regions. Tesseract supports only words, as specified in the hOCR standard[5], on the lowest level. Thus, we transform the gold standard and pipeline output to word level under the assumption of equality in line height and character width. Each text element is defined by its position, i.e. x/y coordinates of the upper left corner of the bounding box , its dimensions determined by width and height of the bounding box and its orientation in terms of a rotation angle around its center. We subdivide each text element τ into words by splitting at blank spaces and carriage returns. The new position and dimensions for each word $\omega \in \Omega$ are computed while retaining the text element's orientation. This is defined by:

$$\Phi: \quad T \times C' \times A \rightarrow \Omega \times C'' \times A \tag{9}$$

$$\Omega := \{\omega \in \tau | \tau \in T\} \tag{10}$$

$$C'' := \{c'' \subseteq c' | c' \in C' \wedge c'' \neq \emptyset \wedge \forall i,j,\ i \neq j : c''_i \cap c''_j = \emptyset\} \tag{11}$$

The bounding boxes of the individual words are matched between TX and gold standard as well as baselines and gold standard for evaluation. For each word $\omega \in \Omega$ we compute the contained n-grams for further evaluation.

4.4 Evaluation Metrics

As previously mentioned, we are evaluating our pipeline over n-grams and words. Since infographics often contain sparse and short text as well as short numbers, we only use 1-,2-, and 3-grams. We use standard metrics precision (PR), recall (RE), and F$_1$-measure (F_1) for our n-grams evaluation as defined by:

$$PR = \frac{|Extr \cap Rel|}{|Extr|},\ RE = \frac{|Extr \cap Rel|}{|Rel|},\ F_1 = \frac{2 \cdot PR \cdot RE}{PR + RE} \tag{12}$$

Here, *Extr* refers to the n-grams as they are computed from text elements that are extracted from an infographic by TX and the baseline, respectively. *Rel* refers to the relevant n-grams from the gold standard. For comparing individual words (i. e. sequences of alpha-numeric characters separated by blank or carriage return), we use standard Levenshtein distance. The same n-gram can appear multiple times in both the extractions result from TX, the baselines, as well as the gold standard. Thus, we have to deal with multisets when computing our evaluation metrics. In order to accommodate this, we have to slightly modify the standard definitions of PR and RE, respectively. To properly account for the number of times an n-gram can appear in *Extr* or *Rel*, we define the counter

[5] The hOCR Embedded OCR Workflow and Output Format: `http://tinyurl.com/hOCRFormat`, last access: Sep 07, 2015

function $\mathbf{C}_M(x) := |\{x|x \in M\}|$ (as an extension of a set indicator function) over a multiset M. For an intersection of multisets M and N, the counter function is formally defined by:

$$\mathbf{C}_{M\cap N}(x) := \min\{\mathbf{C}_M(x), \mathbf{C}_N(x)\} \tag{13}$$

Based on $\mathbf{C}_{M\cap N}(x)$, we define PR and RE for multisets:

$$PR = \frac{\sum_{x\in Extr\cup Rel} \mathbf{C}_{Extr\cap Rel}(x)}{\sum_{x\in Extr} \mathbf{C}_{Extr}(x)} \tag{14}$$

$$RE = \frac{\sum_{x\in Extr\cup Rel} \mathbf{C}_{Extr\cap Rel}(x)}{\sum_{x\in Rel} \mathbf{C}_{Rel}(x)} \tag{15}$$

Specific cases may happen when either one of the sets $Extr$ or Rel is empty. One case is that our pipeline TX or the baselines do not extract text where they should, i. e., $Extr = \emptyset$ and $Rel \neq \emptyset$. When such a false negative happens, we define $PR := 0$ and $RE := 0$ following Groot et al. [8]. For the second situation, when the approaches we compare find something where they shouldn't (false positives), i. e., $Extr \neq \emptyset$ and $Rel = \emptyset$, we define $PR := 0$ and $RE := 1$.

5 Results

This section presents the results of our initial evaluation to assess the quality of the OCR results using our pipeline. We start with a descriptive statistics of the gold standard and the extraction results over the infographics. Subsequently, we present the evaluation results in terms of precision, recall and F_1-measure for infographic and word-level evaluation of TX and the two baselines as well as the Levenshtein distances computed for the extracted text and the gold standard.

Data Characteristics: Table 2 presents the average numbers and standard deviation (in brackets) with regard to n-grams, words and word length for our extraction pipeline (TX), both baselines (BL-1/-2), and gold standard (GS). Table 2 clearly shows that our novel pipeline detects at least 1.5 as many n-grams and words as BL-1 and still some more than BL-2. Compared with the gold standard, TX extracts more n-grams and words. In addition TX and the baselines extract words shorter than the gold standard. Overall, we observe high standard deviations in the gold standard and the extraction results.

Evaluation results on word-level n-grams: The average precision (PR), recall (RE) and F_1-measures for n-grams in Table 3 (standard deviation in brackets) show a relative improvement (Diff.) of TX over BL-1 of about 30% on average. The differences are computed by setting the pipeline results into relation with the baselines. We verified the improvement using significance tests, i.e., if the two distributions obtained from TX and BL-1/2 significantly differ. We checked whether the data follows a normal distribution and has equal variances. Subsequently, we have applied Student's t-tests or the non-parametric Wilcoxon

Table 2: Average number of n-grams and words of the 121 infographics and average word length for GS/TX/BL-1/BL-2

	1-grams	2-grams	3-grams	Words	Length
GS	150.65 (122.28)	115.93 (103.09)	84.95 (85.61)	35.46 (22.24)	4.22 (1.48)
TX	177.21 (128.21)	127.34 (100.51)	89.34 (79.35)	50.07 (31.95)	3.63 (2.69)
BL-1	106.30 (87.71)	80.17 (69.12)	60.79 (54.54)	25.21 (22.12)	4.15 (2.25)
BL-2	135.08 (125.56)	100.20 (98.20)	75.08 (78.10)	35.25 (33.94)	4.08 (1.95)

signed rank test. For all statistical tests, we apply a standard significance level of $\alpha = 5\%$. All TX/BL-1 comparison results are significant with $p < .01$ except for the recall over trigrams which has $p < 0.046$. The test statistics for t-tests are between -7.5 and -3.1 and for the Wilcoxon tests between 1808 and 2619. The second part of Table 3 reports the comparison between TX and BL-2. The results are similar to the previous comparison, but for recall over unigrams and F_1-measure over trigrams the improvement is smaller. Here, all differences are significant with a p-value of $p < .01$ except for the recall and F_1-measure over trigrams with $p < 0.049$ and $p < 0.027$, respectively. The test statistics for t-tests are between -6.8 and -3.1 and between 1652 and 2626 for non-parametric tests. Finally, we observe a smaller performance increase when comparing the results from 1-grams to 3-grams as well as overall high standard deviations.

Table 3: Average PR, RE, F_1 measures for TX and BL-1/BL-2

		word level			infographic level		
	n-gram	PR	RE	F_1	PR	RE	F_1
TX	1	.50 (0.41)	.68 (0.36)	.47 (0.39)	.67 (0.23)	.79 (0.20)	.71 (0.21)
	2	.58 (0.39)	.54 (0.38)	.54 (0.34)	.60 (0.27)	.67 (0.25)	.62 (0.25)
	3	.52 (0.39)	.48 (0.37)	.49 (0.37)	.57 (0.29)	.60 (0.29)	.57 (0.28)
BL-1	1	.37 (0.36)	.48 (0.36)	.36 (0.35)	.67 (0.29)	.54 (0.31)	.58 (0.30)
	2	.42 (0.33)	.42 (0.34)	.42 (0.33)	.60 (0.33)	.50 (0.33)	.53 (0.32)
	3	.42 (0.31)	.42 (0.31)	.36 (0.33)	.55 (0.35)	.48 (0.34)	.49 (0.34)
Diff.	1	35.14%	41.67%	30.06%	0.00%	46.30%	22.41%
	2	38.10%	28.57%	28.57%	0.00%	34.00%	16.98%
	3	23.81%	14.29%	36.11%	3.64%	25.00%	16.33%
BL-2	1	.37 (0.37)	.51 (0.38)	.36 (0.36)	.65 (0.25)	.59 (0.29)	.60 (0.26)
	2	.42 (0.34)	.42 (0.35)	.42 (0.34)	.57 (0.31)	.52 (0.31)	.53 (0.30)
	3	.42 (0.32)	.42 (0.32)	.42 (0.32)	.51 (0.33)	.50 (0.34)	.49 (0.32)
Diff.	1	35.14%	33.33%	30.06%	3.08%	33.90%	18.33%
	2	38.10%	28.57%	28.57%	5.26%	28.85%	16.98%
	3	23.81%	14.29%	16.67%	11.76%	20.00%	16.33%

Evaluation results on infographic level n-grams: We conducted another evaluation on infographic level where we did not consider the location mapping constraint between words and compared the n-grams for the whole infographic. The results are shown in Table 3 for both baselines BL-1 and BL-2. While having on average higher values for all metrics in both comparisons, the relative improvement for precision, recall, and F_1-measure compared with the word level evaluation decreases in most cases. The significance of the results is only given for recall and F_1-measure, but not for precision. For recall and F_1-measure we have $p < .04$ and the test statistics are between -9.2 and -2.4 for t-tests.

Evaluation on words (Levenshtein): For TX the Levenshtein distance is on average 2.23 (SD=1.29). Hence, for an exact match one has to alter about two characters. The average Levenshtein distance for BL-1 is 2.53 (SD=1.59) and we verified that they differ significantly ($t(120) = 2.10, p < .04$). The difference in Levenshtein from BL-2 to TX with an average distance of 2.54 (SD=1.51) is significant as well ($V(120) = 4713, p < .01$).

Special case evaluations: The number of special cases for TX are on average 12.94 (SD=17.88) false negatives and 49.87 (SD=31.52) false positives. For BL-1 we can instead report 17.01 (SD=17.40) false negatives and 5.67 (SD=9.42) false positives on average. BL-2 generates on average 9.03(SD=15.61) false negatives and 17.01(SD=17.40) false positives. Comparing TX pipeline with BL-1 shows that TX produces significantly less false negatives ($V(120) = 4503.5, p < .01$), but simultaneously generates significantly more false positives ($t(120) = -16.6, p < .001$). The second baseline is on average better than TX with regard to false negatives and false positives.

6 Discussion

Our novel pipeline shows promising results for the extraction of multi-oriented text from information graphics. The difference between word and infographic level evaluation can be explained by the constraints induced by the matching procedure on word-level. The main reason for the performance improvement is the increased recall, which is a result of finding text at non-horizontal angles. We define all elements as non-horizontal which have an orientation outside of Tesseract's tolerance range of ±15 degree. About 20% of the words in an infographic are on average at non-horizontal orientation, as specified by the gold standard. Our pipeline output consists to 37% of non-horizontal words while extracting 41% more words on average than actually present in the gold standard. On the other hand, the first baseline which extracts only about 77% as many words as actually contained, all of horizontal orientation. The second baseline is closest to the gold standard with respect to the number of extracted words and contains on average 31% non-horizontal words. In addition, we have improved precision and therefore an overall performance increase, collected through the F_1-measure, with TX. The standard deviation is in all cases quite high, which can be explained by the variance in the gold standard. Consequently, these are dataset characteristics and not issues of TX or the baselines.

The lower number of 3-grams, which are on average only half as many as 1-grams, is a potential negative influence on the results. As reported in Table 2, there is a high standard deviation of the number of n-grams in the gold standard. Thus, some graphic might not even contain 3-grams. However for most cases, there are on average 85 3-grams per infographic as denoted by the gold standard statistics in Table 2, which is enough for reasonable results.

Furthermore, TX produces less false negatives, i. e., it extracts more text elements from the gold standard than BL-1. But it still makes more mistakes with regard to extracting text elements where there are none in the gold standard. This is reflected in Table 2, where TX extracts on average more text elements than there are actually present in the gold standard. These false positives often consist of special characters such as colons, semicolons, dots, hyphens, and others. Removing them will be a future extension of our work.

7 Conclusion

We have presented our novel pipeline for multi-oriented text extraction from information graphics and proved its concept on a set of 121 infographics. Our text extraction shows a significant increase in F_1-measure over two baselines, which is explained by detecting text elements at non-horizontal angles. In our future work, we plan to add a merge step after the MST clustering to reduce the Levenshtein distance and to perform entity detection over the text extraction results. In addition, we want to apply our pipeline to a larger set of infographics for a more thorough evaluation. We will create the required gold standard using crowd-sourcing in the near future. Finally, we plan to include alternative OCR engines like Ocropus to find the best solution for our needs.

References

[1] P. Agrawal and R. Varma. Text extraction from images. *IJCSET*, 2(4):1083–1087, 2012.

[2] F. Böschen and A. Scherp. Multi-oriented text extraction from information graphics. In *ACM DocEng*, 2015.

[3] S. Carberry, S. Elzer, and S. Demir. Information graphics: an untapped resource for digital libraries. In *SIGIR*, pages 581–588. ACM, 2006.

[4] S. Carberry, S. E. Schwartz, K. F. McCoy, S. Demir, P. Wu, C. Greenbacker, D. Chester, E. Schwartz, D. Oliver, and P. Moraes. Access to Multimodal Articles for Individuals with Sight Impairments. *TiiS*, 2(4):21:1–21:49, 2013.

[5] D. Chester and S. Elzer. Getting Computers to See Information Graphics So User Do Not Have to. In *Foundations of Intelligent Systems*, volume 3488 of *LNCS*, pages 660–668. Springer, 2005.

[6] S. R. Choudhury and C. L. Giles. An architecture for information extraction from figures in digital libraries. In *WWW*, pages 667–672, 2015.

[7] J. Gao, Y. Zhou, and K. E. Barner. VIEW: Visual information extraction widget for improving chart images accessibility. In *ICIP*, pages 2865–2868. IEEE, 2012.

[8] P. Groot, F. van Harmelen, and A. ten Teije. Torture tests: A quantitative analysis for the robustness of knowledge-based systems. In *EKAW*, pages 403–418, 2000.

[9] F. Hausdorff. *Grundzüge der Mengenlehre*. AMS Chelsea Publishing Series. Chelsea Publishing Company, 1949.

[10] M. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179–187, 1962.

[11] W. Huang and C. L. Tan. A system for understanding imaged infographics and its applications. In *ACM DocEng*, pages 9–18, 2007.

[12] J. Illingworth and J. Kittler. A survey of the hough transform. *Computer Vision, Graphics, and Image Processing*, 44(1):87–116, 1988.

[13] C. Jayant, M. Renzelmann, D. Wen, S. Krisnandi, R. E. Ladner, and D. Comden. Automated tactile graphics translation: in the field. In *ASSETS*, pages 75–82, 2007.

[14] S. Kataria, W. Browuer, P. Mitra, and C. L. Giles. Automatic extraction of data points and text blocks from 2-dimensional plots in digital documents. In *Advancement of Artificial Intelligence*, pages 1169–1174. AAAI, 2008.

[15] R. E. Ladner, M. Y. Ivory, R. Rao, S. Burgstahler, D. Comden, S. Hahn, M. Renzelmann, S. Krisnandi, M. Ramasamy, B. Slabosky, A. Martin, A. Lacenski, S. Olsen, and D. Groce. Automating tactile graphics translation. In *ASSETS*, pages 150–157, 2005.

[16] Z. Li, M. Stagitis, S. Carberry, and K. F. McCoy. Towards retrieving relevant information graphics. In *SIGIR*, pages 789–792. ACM, 2013.

[17] R. Mariani, M. P. Deseilligny, J. Labiche, and R. Mullot. Algorithms for the hydrographic network names association on geographic maps. In *ICDAR*. IEEE, 1997.

[18] N. Otsu. A threshold selection method from gray-level histograms. *TSMC*, 9(1):62–66, 1979.

[19] P. M. Patil and T. R. Sontakke. Rotation, scale and translation invariant handwritten devanagari numeral character recognition using general fuzzy neural network. *Pattern Recogn.*, 40(7):2110–2117, 2007.

[20] H. Samet and M. Tamminen. Efficient component labeling of images of arbitrary dimension represented by linear bintrees. *IEEE TPAMI*, 10(4):579–586, 1988.

[21] J. Sas and A. Zolnierek. Three-Stage Method of Text Region Extraction from Diagram Raster Images. In *CORES*, pages 527–538, 2013.

[22] M. Savva, N. Kong, A. Chhajta, L. Fei-Fei, M. Agrawala, and J. Heer. ReVision: Automated Classification, Analysis and Redesign of Chart Images. In *UIST*, pages 393–402. ACM, 2011.

[23] R. Smith. A simple and efficient skew detection algorithm via text row accumulation. In *ICDAR*, volume 2, pages 1145–1148, 1995.

[24] I. Sobel. History and definition of the so-called "sobel operator", more appropriately named the sobel-feldman operator. Sobel, I., Feldman, G., "A 3x3 Isotropic Gradient Operator for Image Processing", presented at the Stanford Artificial Intelligence Project (SAIL) in 1968., 2015.

[25] N. Takagi. Mathematical figure recognition for automating production of tactile graphics. In *ICSMC*, pages 4651–4656, 2009.

[26] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., 2005.

[27] F. Wang and M.-Y. Kan. NPIC: Hierarchical synthetic image classification using image search and generic features. In *CIVR*, volume 4071 of *LNCS*, pages 473–482. Springer, 2006.

Multiple-Resolution Stream Clustering Using Graph Maintenance

Marwan Hassani, Pascal Spaus, and Thomas Seidl

Data Management and Data Exploration Group
RWTH Aachen University, Germany
{hassani,spaus,seidl}@cs.rwth-aachen.de

Abstract. Challenges for clustering streaming data are getting continuously more sophisticated. They are driven by stream properties such as the continuous data arrival, the time-critical processing of objects, the evolution of the data streams, the presence of outliers and the varying densities of the data. Due to the continuously evolving nature of the stream, it is crucial that stream clustering algorithms autonomously detect clusters whose number, shapes and densities vary as the stream flows. We present the first hierarchical density-based stream clustering algorithm based on cluster stability, called *HASTREAM* [2] which is able to meet the above mentioned requirements.

We show additionally, that HASTREAM inherited efficiency issues as the main drawback of density-based hierarchical clustering algorithms, as these were not the scope of its contribution. We present then *I-HASTREAM* [1], a first density-based hierarchical clustering algorithm that has considerably less computational time compared to the first presented algorithm. I-HASTREAM utilizes and introduces techniques from the graph theory domain to devise an incremental update of the underlying model instead of repeatedly performing the expensive calculations of the huge graph. Specifically the Prim's algorithm for constructing the minimal spanning tree is adopted by introducing novel, incremental maintenance of the tree by vertex and edge insertion and deletion. The extensive experimental evaluation study on real world datasets shows that I-HASTREAM is considerably faster than HASTREAM while delivering almost the same clustering quality.

References

1. Marwan Hassani, Pascal Spaus, Alfredo Cuzzocrea, and Thomas Seidl. Adaptive stream clustering using incremental graph maintenance. In *BigMine 2015 at KDD'15*, pages 49–64, 2015.
2. Marwan Hassani, Pascal Spaus, and Thomas Seidl. Adaptive multiple-resolution stream clustering. In *MLDM '14*, pages 134–148, 2014.

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

Faceted Search for Mathematics

Radu Hambasan and Michael Kohlhase

Jacobs University Bremen

Abstract. Faceted search represents one of the most practical ways to browse a large corpus of information. Information is categorized automatically for a given query and the user is given the opportunity to further refine his/her query. Many search engines offer a powerful faceted search engine, but only on the textual level. Faceted Search in the context of Math Search is still unexplored territory.

In this paper, we describe one way of solving the faceted search problem in math: by extracting recognizable formula schemata from a given set of formulae and using these schemata to divide the initial set into formula classes. Also, we provide a direct application by integrating this solution with existing services.

1 Introduction

The size of digital data has been growing tremendously since the invention of the Internet. Today, the ability to quickly search for relevant information in the vast amount of knowledge available is essential in all domains. As a consequence, search engines have become the prevalent tool for exploring digital data.

Although text search engines (e.g. Google or DuckDuckGo [3]) seem to be sufficient for the average user, they are limited when it comes to finding scientific content. The limitation arises because STEM[1] documents are also relevant for the mathematical formulae they contain and math cannot be properly indexed by a textual search engine. Math comprises of tokens that are expressed as structural markup (fractions, square-roots, subscripts and superscripts), which are not captured by simply indexing the text content of a page.

A good math search engine is therefore needed in several applications. For example, a large airline manufacturer may have many ongoing research projects and could significantly improve efficiency if they had a way of searching for formulae in a corpus containing all their previous work in the fields of physics and mathematics. The same holds for all large physics-oriented research centers, such as CERN. Valuable time would be saved if scientists would have a fast, reliable and powerful math search engine to analyse previous related work. As a third application, university students should be mentioned. Their homework,

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

[1] Science, Technology, Engineering and Mathematics

research and overall study process would be facilitated once they are provided with more than textual search. For all these applications, we first need a strong math search engine and second, a large corpus of math to index.

The Cornell e-Print Archive, arXiv, is an example of such a corpus, containing over a million STEM documents from various scientific fields (Physics, Mathematics, Computer Science, Quantitative Biology, Quantitative Finance and Statistics) [1]. Given such a high number of documents, with several million formulae, the search engine must provide an expressive query language and query-refining options to be able to retrieve useful information. One service that provides both of these is the Zentralblatt Math service [14].

Zentralblatt Math now employs formula search for access to mathematical reviews [7]. Their database contains over 3 million abstract reviews spanning all areas of mathematics. To explore this database they provide a powerful search engine called "structured search". This engine is also capable of faceted search. Figure 1 shows a typical situation: a user searched for a keyword (here an author name) and the faceted search generated links for search refinements (the **facets**) on the right. Currently, facets for the primary search dimensions are generated – authors, journals, MSC[2], but not for formulae. In this way, the user is given the ability to further explore the result space, without knowing in advance the specifics of what he/she is looking for. Recently, formula search has been added as a component to the structured search facility. However, there is still no possibility of faceted search on the math content of the documents.

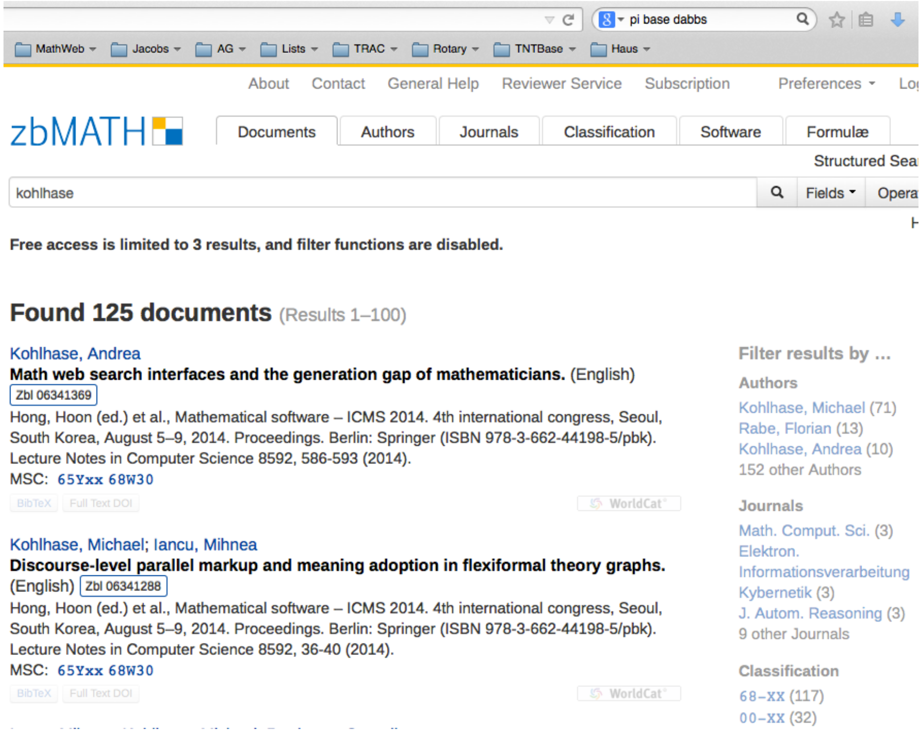
There are multiple ways in which we could understand a "math facet". One way would be through the MSC classification [10]. However, this would be rather vague because it will only provide information about the field of mathematics to which an article belongs. If the authors use formulae from another field in their paper, the results will suffer a drop in relevance.

We are attempting to solve this problem by extracting formula schemata from the query hits, as formula facets. A math facet consists of a set of formula schemata generated to further disambiguate the query by refining it in a new dimension. For instance, for the query above we could have the formulae in Figure 2, which allows the user to drill in on *i*) variation theory and minimal surfaces, *ii*) higher-order unification, and *iii*) type theory. Following the MWS (see 2.1) tradition, the red identifiers stand for query variables, their presence making the results **formula schemata**.

These formula schemata were manually created to judge the feasibility of using schemata as recognizable user interface entities, but for an application we need to generate them automatically from the query. Moreover, each schema should further expand to show the formula class it represents. Formula classes would consist of all formulae sharing the same schema. This is the algorithmic problem we explore in this paper.

Acknowledgements This work has been supported by the Leibniz Association under grant SAW-2012-FIZ_KA-2 (Project MathSearch). The authors gratefully

[2] Mathematics Subject Classification



Free access is limited to 3 results, and filter functions are disabled.

Found 125 documents (Results 1–100)

Kohlhase, Andrea
Math web search interfaces and the generation gap of mathematicians. (English) Zbl 06341369
Hong, Hoon (ed.) et al., Mathematical software – ICMS 2014. 4th international congress, Seoul, South Korea, August 5–9, 2014. Proceedings. Berlin: Springer (ISBN 978-3-662-44198-5/pbk). Lecture Notes in Computer Science 8592, 586-593 (2014).
MSC: 65Yxx 68W30

Kohlhase, Michael; Iancu, Mihnea
Discourse-level parallel markup and meaning adoption in flexiformal theory graphs. (English) Zbl 06341288
Hong, Hoon (ed.) et al., Mathematical software – ICMS 2014. 4th international congress, Seoul, South Korea, August 5–9, 2014. Proceedings. Berlin: Springer (ISBN 978-3-662-44198-5/pbk). Lecture Notes in Computer Science 8592, 36-40 (2014).
MSC: 65Yxx 68W30

Filter results by ...

Authors
Kohlhase, Michael (71)
Rabe, Florian (13)
Kohlhase, Andrea (10)
152 other Authors

Journals
Math. Comput. Sci. (3)
Elektron. Informationsverarbeitung Kybernetik (3)
J. Autom. Reasoning (3)
9 other Journals

Classification
68-XX (117)
00-XX (32)

Fig. 1: Faceted Search in ZBMath

$$\int_M \Phi(d_p f)dvol$$

$$\lambda X.h(H^1 X)\cdots H^n X$$

$$\frac{\Gamma\vdash A\gg\alpha}{D}$$

Fig. 2: formula facets

acknowledge fruitful discussions with Fabian Müller, Wolfram Sperber, and Olaf Teschke in the MathSearch Project, which led to this research (the ZBMath information service uses faceted search on the non-formula dimensions very successfully) and clarified the requirements from an application point of view.

2 Preliminaries

In this section we describe the existent systems on which our work will be based, with the intention of making this paper self-contained.

2.1 MathWebSearch

At its core, the MathWebSearch [12] system (MWS) is a content-based search engine for mathematical formulae. It indexes MathML [9] formulae, using a technique derived from automated theorem proving: Substitution Tree Indexing [6]. Recently, it was augmented with full-text search capabilities, combining keyword queries with unification-based formula search. The engine serving text queries is Elasticsearch 2.2. From now on, in order to avoid confusion, we will refer to the core system (providing just formula query capability) as MWS and to the complete service (MWS + Elasticsearch) as TeMaSearch (Text + Math Search).

Internal to MWS, each mathematical expression is encoded as a set of substitutions based on a depth-first traversal of its Content MathML tree. Furthermore, each tag from the Content MathML tree is encoded as a TokenID, to lower the size of the resulting index. The (bijective) mapping is also stored together with the index and is needed to reconstruct the original formula. The index itself is an in-memory trie of substitution paths.

To facilitate fast retrieval, MWS stores FormulaIDs in the leaves of the substitution tree. These are integers uniquely associated with formulae, and they are used to store the context in which the respective expressions occurred. These identifiers are stored in a separate LevelDB [8] database.

MathWebSearch exposes a RESTful HTTP API which accepts XML queries. A valid query must obey the Content MathML format, potentially augmented with *qvar* variables which match any subterms. A *qvar* variable acts as a wildcard in a query, with the restriction that if two *qvar*s have the same name, they must be substituted in the same way.

2.2 Elasticsearch

Elasticsearch [4] is a powerful and efficient full text search and analytics engine, built on top of Lucene. It can scale massively, because it partitions data in shards and is also fault tolerant, because it replicates data. It indexes schema-free JSON documents and the search engine exposes a RESTful web interface. The query is also structured as JSON and supports a multitude of features via its domain specific language: nested queries, filters, ranking, scoring, searching using wildcards/ranges and faceted search.

2.3 LaTeXML

An overwhelming majority of the digital scientific content is written using LaTeX or TeX, due to its usability and popularity among STEM researchers. However, formulae in these formats are not good candidates for searching because they do not display the mathematical structure of the underlying idea. For this purpose, conversion engines have been developed to convert LaTeX expressions to more organized formats such as MathML.

An open source example of such a conversion engine is LaTeXML [11]. The MathWebSearch project relies heavily on it, to convert arXiv documents from LaTeX to XHTML which is later indexed by MWS. It exposes a powerful API, accepting custom definition files which relate TeX elements to corresponding XML fragments that should be generated. For the scope of this project, we are more interested in another feature of LaTeXML: cross-referencing between Presentation MathML and Content MathML. While converting TeX entities to Presentation MathML trees, LaTeXML assigns each PMML element a unique identifier which is later referenced from the corresponding Content MathML element. In this manner, we can modify the Content MathML tree and reflect the changes in the Presentation MathML tree which can be displayed to the user.

3 Schematization of Formula Sets & Implementation

In this section, we provide a theoretical description of the problem of generating formula schemata and a practical implementation.

3.1 Formalizing the Problem

Let us now formulate the problem at hand more carefully.

Definition 1. *Given a set $\mathcal{D}$ of documents (fragments) – e.g. generated by a search query, a* ***coverage*** $0 < r \leq 1$*, and a* ***width*** n*, the* ***Formula Schemata Generation*** *(FSG) problem requires generating a set $\mathcal{F}$ of at most n formula schemata (content MathML expressions with* qvar *elements for query variables), such that $\mathcal{F}$ covers $\mathcal{D}$ with coverage r.*

Definition 2. *We say that a set $\mathcal{F}$ of formula schemata* ***covers*** *a set $\mathcal{D}$ of document fragments, with* ***coverage*** r*, iff at least $r \cdot |\mathcal{D}|$ formulae from $\mathcal{D}$ are an instance $\sigma(f)$ of some $f \in \mathcal{F}$ for a substitution σ.*

3.2 Defining a Cutoff Heuristic

To generate formula schemata, we must define a “cutoff heuristic”, which tells the program when two formulae belong to the same schema class. If there is no heuristic, two formulae would belong to the same class, only if they were identical. However, we want formulae that have something in common to be grouped together, even if they are not perfectly identical.

We experimented with several possibilities for the heuristic and found out that a dynamic cutoff which preserves the operators is optimal. We can identify the operators by looking at the first child of the apply token in the CMML tree. The user is given the option to have an absolute (fixed) or relative (depending on the depth of the CMML tree) cutoff for the operands.

Figure 3 illustrates this heuristic at depth 1. The divide element was kept, because it was the first child of apply, while the other children were removed. If we were to use a depth of 2, the plus element would also be included in the schema.

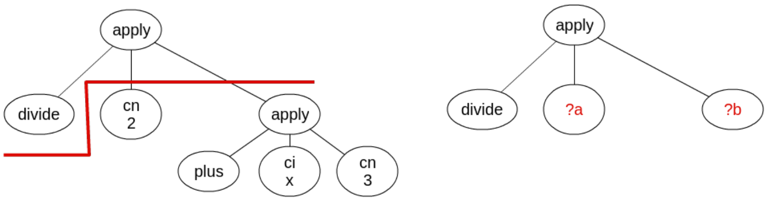


Fig. 3: Dynamic Cutoff

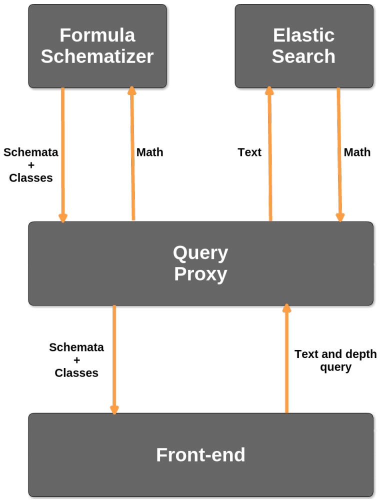


Fig. 4: FS Engine Architecture

3.3 Design Overview

The full faceted search system comprises of the following components: the Formula Schematizer 3.4, Elasticsearch, a proxy to mediate communication between the Schematizer and Elasticsearch and a Web front-end. The architecture of the system is shown in Figure 4.

Once the user enters a query (which consists of keywords and a depth), the front-end forwards the request to a back-end proxy. The proxy sends the text component of the query to Elasticsearch and receives back math contained in matching documents. Afterwards, it sends the retrieved math and the depth parameter (from the original query) to the Schematizer. The Schematizer will respond with a classification of the math in formula classes, as well as the corresponding schema for each class. Finally, the proxy forwards the result to the front-end which displays it to the user.

3.4 The Formula Schematizer

The Schematizer is the core part of our system. It receives a set of formulae in their Content MathML representation, generates corresponding formula schemata and classifies the formulae according to the generated schemata. It provides an HTTP endpoint and is therefore self-contained, i.e. it can be queried independently, not only as part of the faceted search system. As a consequence, the Schematizer displays a high degree of versatility, and can be integrated seamlessly with other applications.

The central idea behind the schematization process is to generate signatures from formulae which can be used to identify formula classes. We use the MathWebSearch encoding for MathML nodes, where each node is assigned an integer ID based on its tag and text content. If the node is not a leaf, then only the tag is considered. The signature will be a vector of integer IDs, corresponding to the pre-order traversal of the Content MathML tree.

Naturally, the signature depends on the depth chosen for the cutoff heuristic. At depth 0, the signature consists only of the root token of the Content MathML expression. At full depth (the maximum depth of the expression), the signature is the same as the depth-first traversal of the Content MathML tree.

Based on these computed signatures, we divide the input set of formulae into formula classes, i.e. all formulae with the same signature belong to the same class. For this operation we keep an in-memory hash table, where the keys are given by the signatures and the values are sets of formulae which have the signature key. After filling the hash table, we sort it according to the number of formulae in a given class, since the signatures which cover the most formulae should come at the beginning of the reported result.

The Schematizer caller can place an optional limit on the maximum number of schemata to be returned. If such a limit was specified, we apply it to our sorted list of signatures and take only the top ones.

As a last step, we need to construct Content MathML trees from the signatures, to be able to show the schemata as formulae to the user. We are able to do this because we know the arity of each token and the depth used for cutoff. The tree obtained after the reconstruction might be incomplete, so we insert query variables in place of missing subtrees. We finally return these Content MathML trees with query variables (the formula schemata), together with the formulae which they cover.

3.5 The Front-End

To show the capabilities of the Schematizer we have prepared two demos. The first one is a text-only search engine which returns the math from the matching documents, after running it through the Schematizer. This is the demo for showcasing the schematization process. The second one is a direct application of the Schematizer into a Math Search Engine which is capable of mathematical faceted search.

SchemaSearch

The SchemaSearch front-end provides just a textual search input field. It is intended for users who want an overview of the formulae contained in a corpus.

The user can enter a set of keywords for the query, as well as a schema depth, which defaults to 3. The maximum result size is not accessible to the user, to prevent abuses and reduce server load. There is also an "R" checkbox which specifies if the cutoff depth should be absolute or relative. If relative, the depth should be given in percentages.

TemaV2

The TemaV2 front-end extends TeMaSearch to be able to perform mathematical faceted search. It is intended for users who want to filter query results based on a given facet (formula schema in this case). The look and feel is similar to the previous version of TeMaSearch, where the first input field is used to specify keywords and the second one is used to specify LaTeX-style formulae for the query. When returning results, a "Math Facets" menu will be presented to the user. We discuss this in Section 4.2.

3.6 Presentation by Replacement

After obtaining the schemata and formula classes, we need to be able to display the result to the user. One possibility would be to have the Schematizer return Content MathML expressions for the schemata and use an XSL stylesheet [13] to convert them to Presentation MathML. This approach would unfortunately generate unrecognizable schemata due to the inherent ambiguity of CMML. For instance, a csymbol element can be represented in several different ways depending on the notation being used. Additionally, we cannot reliably foresee all possible rules that should be implemented in the stylesheet and as a consequence some formulae will be wrongly converted.

Since the XSL conversion is unreliable, we will make use of the cross reference system provided by LaTeXML, as discussed in Section 2.3. Instead of returning Content MathML expressions, the Schematizer will use the first formula in each class as a template and "punch holes into it", effectively returning the ID of the nodes that are to be substituted with query variables. We will use this IDs to replace the referenced PMML nodes with <mi> nodes representing the qvars.

Figure 5 shows the presentation by replacement technique for a given schema. The Schematizer returned a schema which was checked against the first formula in its class ($\frac{2}{x+3}$) to generate two substitutions, marked with red on the left side. Due to the cross-reference system provided by LaTeXML, we are able to find the corresponding PMML elements and substitute them with <mi> tokens. The result will be displayed to the user as $\frac{?x}{?y}$.

4 Evaluation

4.1 SchemaSearch Front-end

Figure 6a shows the formula schemata at depth 3, over the arXiv corpus, for a query containing the keyword "Kohlhase". By default, the top 40 schemata are

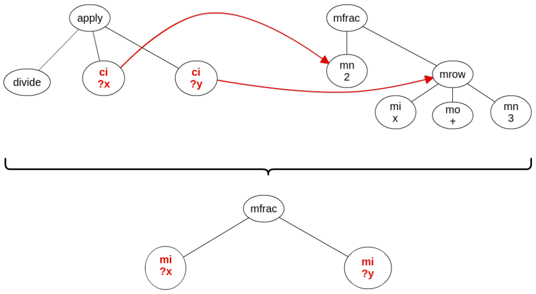
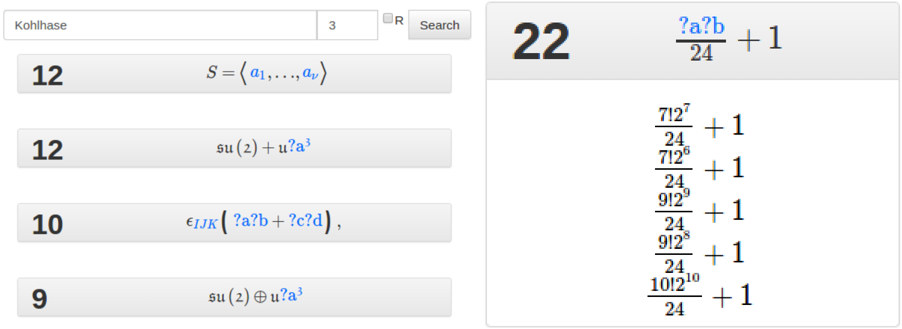


Fig. 5: Presentation by Replacement



Kohlhase | 3 | □R | Search

12 $S = \langle a_1, \ldots, a_\nu \rangle$

12 $\mathfrak{su}(2) + \mathfrak{u}?a^3$

10 $\epsilon_{IJK}(?a?b + ?c?d)$,

9 $\mathfrak{su}(2) \oplus \mathfrak{u}?a^3$

22 $\frac{?a?b}{24} + 1$

$\frac{7!2^7}{24} + 1$
$\frac{7!2^6}{24} + 1$
$\frac{9!2^9}{24} + 1$
$\frac{9!2^8}{24} + 1$
$\frac{10!2^{10}}{24} + 1$

(a) Faceted Results at depth 3

(b) Expansion of a Formula Class

shown, but the results are truncated for brevity. The bold number on the left side of each result item indicates how many formulae are present in each formula class. For instance, the third schema represents a formula class containing 10 formulae. The entities marked in blue are query variables (qvars).

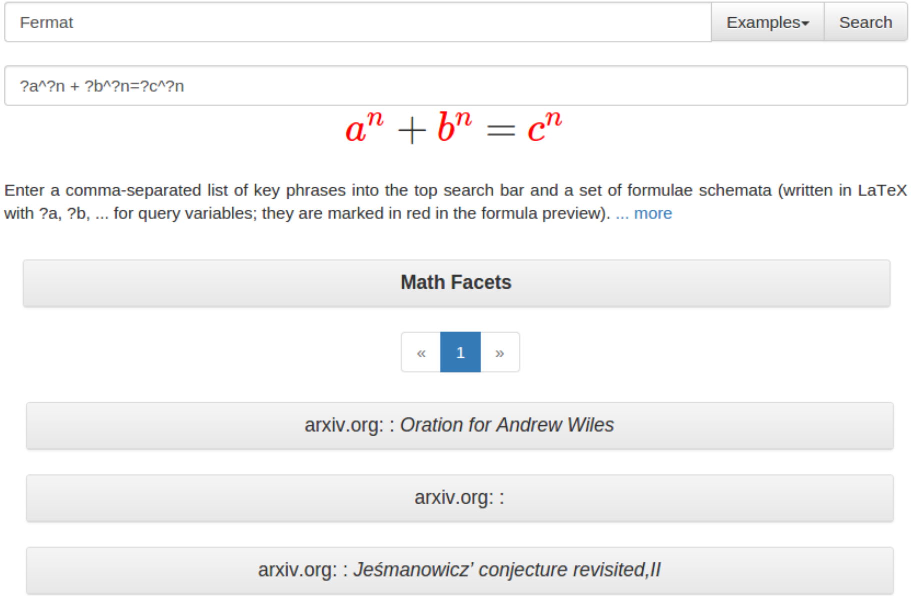
Figure 6b shows the expansion of a formula class. There are 22 formulae in the class given by this particular math schema, as indicated by the count on the left upper side, out of which only ten are shown to the user (for brevity the class is truncated to 5 formulae).

We can see 2 unnamed query variables marked with blue as $?a$ and $?b$. By seeing the schema, the user can form an impression about the general structure of the formulae from that class. After expanding the class, the listing of concrete formulae appears. If the user clicks on one of them, he is redirected to the source document from which that expression was extracted.

4.2 TemaV2 Front-end

Figure 7 shows the results of a query for “Fermat” and $?a^{?n} + ?b^{?n} = ?c^{?n}$. Besides the regular TeMaSearch results, the user is also presented with a “Math Facets” section.

When the “Math Facets” section is expanded the user can see the top 10 schemata (ranked with respect to their coverage), as shown in Figure 8 (results



Fermat

?a^?n + ?b^?n=?c^?n

$$a^n + b^n = c^n$$

Enter a comma-separated list of key phrases into the top search bar and a set of formulae schemata (written in LaTeX with ?a, ?b, ... for query variables; they are marked in red in the formula preview). ... more

arxiv.org: : *Oration for Andrew Wiles*

arxiv.org: :

arxiv.org: : *Jeśmanowicz' conjecture revisited,II*

Fig. 7: TeMa v2 Query Results

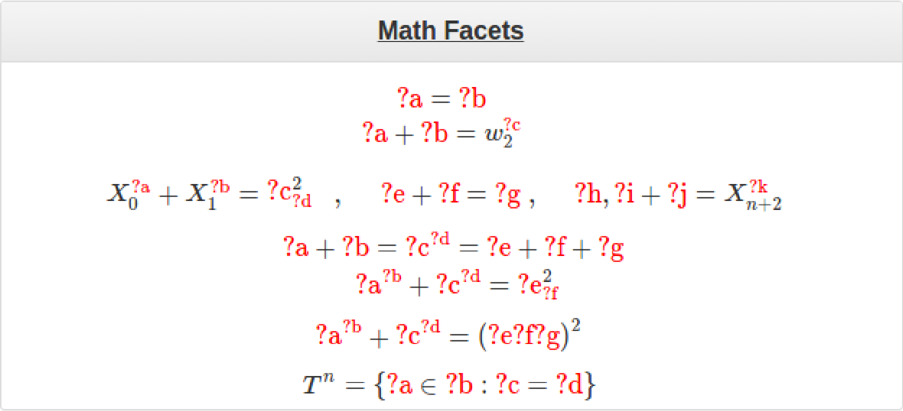
truncated for brevity). We have also implemented a "search-on-click" functionality that allows the user the do a fresh search using the clicked schema and the initial keyword, which effectively filters the current results.

4.3 Performance of the Schematizer

We designed the Schematizer to be a very lightweight daemon, both as memory requirements and as CPU usage. To test if we achieved this goal, we benchmarked it on a server running Linux 3.2.0, with 10 cores (Intel Xeon CPU E5-2650 2.00GHz) and 80 GB of RAM.

We obtained the 1123 expressions to be schematized by querying Elasticsearch with the keyword "Fermat". While the overall time taken by the faceted search engine was around 5 seconds, less than a second was spent in the Schematizer. Also, the CPU utilized by the Schematizer never rose higher than 15% (as indicated by the `top` utility). Asymptotically, the algorithm would run in $O(N)$ time, where N is the number of input formulae. We are able to reach linear time performance, because each formula is processed exactly once and the signature is stored in a hash table, as discussed in Section 3.4.

Due to its implementation, the Schematizer is indefinitely scalable, because it does not require shared state between formulae and can therefore be implemented as a MapReduce [2] job, where mappers compute the signature of assigned formulae and reducers assemble the signature hash table.



Math Facets

$$?a = ?b$$
$$?a + ?b = w_2^{?c}$$
$$X_0^{?a} + X_1^{?b} = ?c^2_{?d} \quad , \quad ?e + ?f = ?g \, , \quad ?h, ?i + ?j = X^{?k}_{n+2}$$
$$?a + ?b = ?c^{?d} = ?e + ?f + ?g$$
$$?a^{?b} + ?c^{?d} = ?e^2_{?f}$$
$$?a^{?b} + ?c^{?d} = (?e?f?g)^2$$
$$T^n = \{?a \in ?b : ?c = ?d\}$$

Fig. 8: Math Facets in TeMa v2

5 Future Work

One application of the faceted search engine can be providing mathematical definitions with the help of NNexus [5]. NNexus is an auto-linker for mathematical concepts from several encyclopedias, e.g. PlanetMath, Wikipedia. Assuming we are able to generate relevant schemata in response to keyword queries, we can target the faceted search engine with all the concepts stored by NNexus and store a schema for each such concept. Afterwards, for a given query, we can obtain the schema and check it against our stored set of schemata. If we find it, we can link the given expression to its mathematical definition. Given a large number of stored concepts and a high schemata relevance, the user should be able to see the definition of any encountered formulae on the Web. For example, hovering over $a^2 + b^2 = c^2$ will show the definition of the Pythagorean theorem.

Another, more direct, application of the Schematizer would be *Similarity Search*. One could create a MathWebSearch based search engine, which accepts an input formula and a similarity degree (between 0% and 100%). The engine would then create a formula schema at a relative depth corresponding to the similarity degree and use this schema to search the corpus. This approach defines the similarity between two formulae as the percentage of the CMML tree depth that they share.

6 Conclusion

We have presented the design and implementation of a system capable of mathematical faceted search. Moreover, we have described a general purpose scalable Schematizer which can generate intuitive and recognizable formula schemata and divide expressions into formula classes according to said schemata.

Although the Schematizer provides recognizable formulae, some queries to SchemaSearch (e.g. using an author as keyword) provide hits with a very low relevance. This is because we cannot distinguish between the work of the author and work where the author is cited at the textual level. As a consequence, searching for “Fermat” would also show formulae from papers where Fermat was cited and if these papers are numerous, as it happens with known authors, would provide the user with misleading results. This suggests that a better source of mathematical expressions might be required for the SchemaSearch demo.

References

[1] *ArXiv Online*. Dec. 21, 2014. URL: `http://arxiv.org/` (visited on 12/21/2014).

[2] Jeffrey Dean and Sanjay Ghemawat. *MapReduce: Simplified Data Processing on Large Clusters*. 2004.

[3] *DuckDuckGo Website*. May 8, 2015. URL: `https://duckduckgo.com` (visited on 05/08/2015).

[4] *Elastic Search*. Dec. 7, 2014. URL: `http://www.elasticsearch.org/` (visited on 12/07/2014).

[5] Deyan Ginev and Joseph Corneli. “NNexus Reloaded”. In: *Intelligent Computer Mathematics*. Conferences on Intelligent Computer Mathematics (Coimbra, Portugal, July 7–11, 2014). Ed. by Stephan Watt et al. Lecture Notes in Computer Science. Springer, 2014, pp. 423–426. URL: `http://arxiv.org/abs/1404.6548`.

[6] Peter Graf. *Substitution Tree Indexing*. 1994.

[7] Michael Kohlhase et al. “Zentralblatt Column: Mathematical Formula Search”. In: *EMS Newsletter* (Sept. 2013), pp. 56–57. URL: `http://www.ems-ph.org/journals/newsletter/pdf/2013-09-89.pdf`.

[8] *LevelDB*. Dec. 21, 2014. URL: `http://leveldb.org/` (visited on 12/21/2014).

[9] *Mathematical Markup Language*. URL: `http://www.w3.org/TR/MathML3/`.

[10] *Mathematics Subject Classification (MSC) SKOS*. 2012. URL: `http://msc2010.org/resources/MSC/2010/info/` (visited on 08/31/2012).

[11] Bruce Miller. *LaTeXML: A LaTeX to XML Converter*. URL: `http://dlmf.nist.gov/LaTeXML/` (visited on 03/12/2013).

[12] Corneliu C. Prodescu and Michael Kohlhase. “MathWebSearch 0.5 - Open Formula Search Engine”. In: *Wissens- und Erfahrungsmanagement LWA (Lernen, Wissensentdeckung und Adaptivität) Conference Proceedings*. Sept. 2011. URL: `https://svn.mathweb.org/repos/mws/doc/2011/newmws/main.pdf`.

[13] *XSLT for Presentation MathML in a Browser*. Dec. 20, 2000. URL: `http://dpcarlisle.blogspot.de/2009/12/xslt-for-presentation-mathml-in-browser.html#uds-search-results` (visited on 04/04/2015).

[14] *Zentralblatt Math Website*. Dec. 7, 2014. URL: `http://zbmath.org/` (visited on 12/07/2014).

Effiziente Integration von Data- und Graph-Mining-Algorithmen in relationale Datenbanksysteme

Manuel Then, Linnea Passing, Nina Hubig,
Stephan Günnemann, Alfons Kemper und Thomas Neumann

TU München, Lehrstuhl für Datenbanksysteme,
Boltzmannstraße 3, 85748 Garching bei München
{then,passing,hubig,guennemann,kemper,neumann}@in.tum.de

Zusammenfassung. Die Nutzung komplexer Algorithmen zur Analyse oft hochdimensionaler Datensätze gerät im Kontext von „Big Data“ immer mehr in das Zentrum der Aufmerksamkeit. Um diese komplexen Datenanalysen effizient zu ermöglichen liegt es nahe, sie in die am weitesten verbreiteten Datenspeicher zu integrieren – in relationale Datenbanksysteme. Dies führt zu interessanten Fragestellungen nicht nur im Bereich der technischen Integration sondern besonders auch in der Anfragespezifikation und -auswertung. In diesem Kurzbeitrag beschreiben wir, wie Algorithmen zur Datenanalyse effizient und nutzerfreundlich in das relationale Hauptspeicherdatenbanksystem HyPer integriert werden können. Wir evaluieren unseren Ansatz anhand eines Vergleichs mit zwei verbreiteten Datenanalysesystemen auf Graph- und Vektordaten.

Schlüsselwörter: Data Mining, Graph, SQL, HyPer, RDBMS

1 Motivation

Die gegenwärtige Datenexplosion stellt stand-alone Data-Mining-Programme vor Schwierigkeiten: zur Analyse großer Datenmengen sind sie durch ihre eingeschränkte Datenverwaltungsfunktionalität kaum geeignet. Besonders da die zu analysierenden Daten in die Applikationen kopiert werden müssen, sind diese für sich ändernde Daten ineffizient. Im Gegensatz hierzu bieten (relationale) Datenbanksysteme eine effiziente und update-freundliche Datenspeicherung. Laut Aggarwal et al. [1] ist die nahtlose Integration von Data-Mining-Technologien in Datenbanksysteme daher eine der momentan wichtigsten Herausforderungen. Einige Datenbanksysteme, etwa SAP HANA [3] und HyPer [4], integrieren bereits die verschiedenen Workloads OLAP und OLTP in ein einzelnes System, sodass die Datenbasis nur einmal vorgehalten werden muss und ETL-Zyklen entfallen. Durch das Paradigma „Data Mining in the database“ [6] entsteht so eine Datenbasis, die für sämtliche Anfragen genutzt werden kann.

1.1 Stand der Technik

SAP HANAs *Predictive Analytics Library* [3] und Oracle *Data Miner* [6] erlauben es Data-Mining-Algorithmen ähnlich zu SQL-Anfragen einzeln auszuführen. Die Ergebnisse der Algorithmen werden jeweils in zu spezifizierenden Tabellen abgelegt und können damit in separaten SQL-Anfragen genutzt werden. Eine interaktive Weiterverarbeitung der Ergebnisse in derselben Anfrage ist somit nicht möglich. Oracle Data Miner legt zudem den Fokus auf *supervised* Machine-Learning-Algorithmen. Es wird hier zunächst ein Modell mit Trainingsdaten angelegt, das anschließend mithilfe von SQL-Funktionen auf Testdaten angewandt. Für *unsupervised* Algorithmen erscheint dies umständlich, da ebenfalls ein persistentes Modell angelegt werden muss. Beide Produkte benennen als Vorteil, dass die Daten nicht mehr kopiert werden müssen, sondern innerhalb der Datenbank analysiert werden können. Wie in diesem Abschnitt gezeigt, ist die Integration in SQL-Anfragen bei beiden Lösungen jedoch nur oberflächlich gegeben.

Im Gegensatz hierzu streben wir eine tiefere Integration mit SQL an, sodass SQL- und Data-Mining-Anfragen nahtlos miteinander verwendet und kompiliert werden können.

1.2 Wissenschaftlicher Beitrag

In diesem Kurzbeitrag stellen wir am Beispiel von HyPer dar, wie Data-Mining-Algorithmen effizient in relationale Datenbanksystemen integriert werden können. Die wichtigsten Kontributionen unseres Beitrags sind zusammengefasst:

- Mehrschichtiges Spezifikationsmodell zur Erstellung und Nutzung der Algorithmen, bestehend aus Laien-, Domänenexperten- und Programmierer-Sicht
- SQL-Erweiterung zur Spezifikation von effizienten iterativen Algorithmen
- Laufzeitvergleiche mit Data-Mining-Anwendungen am Beispiel der Algorithmen *PageRank* (für Graphdaten) und *K-Means* (für Vektordaten)

2 Arten der Integration von Algorithmen in HyPer

Existierende Datenanalysesysteme nutzen häufig eigene, meist proprietäre, Sprachen oder APIs, um Analysen zu spezifizieren. Dies hat diverse Nachteile. Unübliche Anfragesprachen machen es nötig, die Nutzer – häufig Datenanalysten aus der Anwendungsdomäne – aufwändig zu schulen. Werden Hochsprachen-APIs – z.B. in Java – verwendet, gibt es zwar viele erfahrene Programmierer, jedoch haben diese selten das nötige Domänenwissen. Bei in Hochsprachen spezifizierten Anfragen ist es zudem für das Datenanalysesystem sehr schwierig, die Anfrage zu optimieren, um eine effiziente Ausführung zu ermöglichen.

Wir wählen daher einen neuartigen, mehrstufigen Ansatz für die Integration von Data Mining in HyPer. Unser Ziel ist es, Domänenspezialisten auf einfache Art und Weise effiziente Anfragen spezifizieren zu lassen, während Spezialisten alle Freiheitsgrade behalten. Die vier im folgenden vorgestellten Stufen der Integration unterscheiden sich daher sowohl in der Mächtigkeit der Spezifikation, als auch in den Möglichkeiten des DBMS, die Anfragen zu optimieren.

2.1 Externer Zugriff auf die Datenbank

Um allgemeine Data-Mining-Funktionalität anzubinden, bei der das DBMS nur als Datenspeicher verwendet wird, bietet HyPer PostgreSQL-kompatible Datenbankschnittstellen, u.a. JDBC. Zwar ermöglichen diese beliebige Berechnungen, jedoch verhindert der Zugriff über sie umfassende Anfrageoptimierungen und führt potentiell zu teurem Datenaustausch zwischen den beteiligten Systemen.

2.2 Programmausführung in der Datenbank

Als tiefergehende Integration von Data Mining erlaubt HyPer die Ausführung von Nutzercode als *User-defined Functions (UDFs).* Wie bei anderen Datenbanksystemen können berechtigte Nutzer dabei beliebige Funktionalität hinzufügen. Diese wird dann entweder direkt innerhalb des Datenbanksystems (*unfenced*) oder in einer Sandbox (*fenced*) ausgeführt. Dadurch ist es nicht mehr nötig, Daten in externe Systeme zu kopieren.

2.3 SQL-Spracherweiterungen

Oft lassen sich Data-Mining-Algorithmen nur umständlich in SQL ausdrücken. Dies liegt unter anderem daran, dass viele Verfahren iterativ sind. Um diese in SQL abzubilden kommen häufig rekursive *Common Table Expressions* (`WITH`-Statements) zum Einsatz, die eine monoton wachsende Relation berechnen. Da iterative Algorithmen im Normalfall jedoch nur auf die Daten der vorherigen Iteration zugreifen, um die aktuelle Iteration zu berechnen, wird bei diesem Vorgehen viel Speicher unnütz belegt. Dies ist vor allem für Hauptspeicherdatenbanksysteme ein Problem, da Speicher hier eine besonders wertvolle Ressource ist. Als Lösung für dieses Problem schlagen wir ein Iterationskonzept für SQL vor. Syntaktisch ist dieses an `WITH` angelehnt:

```
with recursive [Algo] as ([Initialization] iterate [Step]
                          until [Condition])
                         select * from [Algo]
```

Es wird hier eine temporäre Relation *Algo* erstellt, die anfangs das Resultat der Unteranfrage *Initialization* enthält und auf die iterativ *Step* angewendet wird, bis der boolsche Ausdruck *Condition* wahr ist. Diese Spracherweiterung erlaubt es uns, iterative Data-Mining-Verfahren auf einfache Weise direkt in SQL auszudrücken. Dies ermöglicht nicht nur die direkte Verwendung des ausgereiften state-of-the-art relationalen Anfrageoptimierers von HyPer, sondern auch die Nutzung der hochoptimierten parallelen Codegenerierungs- und Ausführungsengine [4].

2.4 Data Mining im Datenbankkern

Im Gegensatz zu anderen Datenbanksystemen integriert HyPer wichtige Data-Mining-Funktionalität direkt im Datenbankkern. Für den Nutzer sind diese syntaktisch nicht von den zuvor beschriebenen UDFs zu unterscheiden. So berechnet folgende Anfrage für jeden Knoten des durch die Kanten in *edges* gebildeten Graphen die parametrisierte PageRank-Metrik:[1]

```
select * from pagerank((select src,dest from edges), 0.85, 0.001)
```

Intern wird die Berechnung jedoch von spezialisierten Operatoren ausgeführt, in diesem Fall durch einen Sort- gefolgt von einem PageRank-Operator.

HyPer wählt dabei u.a. eine effiziente interne Graphrepräsentation und führt weitere Vorverarbeitungsschritte durch, um die Metrik zu berechnen. Des Weiteren kennt der Anfrageoptimierer die genauen Eigenschaften des PageRank-Operators und kann somit den optimalen Ausführungsplan wählen.

Lambda-Ausdrücke Vordefinierte Funktionen allein decken jedoch nicht alle Einsatzzwecke ab. HyPer erlaubt daher die Verwendung von Lambda-Ausdrücken in SQL-Anfragen. Dies ermöglicht etwa im K-Means-Algorithmus den Einsatz benutzerdefinierter Distanzfunktionen, wobei die volle Optimierbarkeit der Anfrage erhalten bleibt. Bei entsprechend gewählter Distanzfunktion können dabei numerische und kategorische Daten kombiniert analysiert werden, was unverzichtbar für Datenbanksysteme mit ihren verschiedenen Datentypen ist.

3 Experimentelle Evaluierung

In diesem Abschnitt evaluieren wir unsere Ansätze aus den Abschnitten 2.3, nachfolgend *HyPer SQL*, und 2.4, im Folgenden *HyPer Op*. Wir implementieren dazu jeweils einen Graph- und Vektoralgorithmus. *PageRank* wählen wir als bekannten Vertreter der Graphverfahren und als Basis weiterer iterativer Algorithmen. Für Vektordaten verwenden wir *K-Means*, ein häufig genutztes Clusteringverfahren [7].

Als Vergleichssysteme verwenden wir Apache Spark 1.4.0 [8] und MATLAB R2015 [5] mit litekmeans [2]. Alle Tests wurden auf einem Intel Core i7-5820K (6x3,3 GHz) mit 32 GB Hauptspeicher unter Ubuntu Linux 15.04, Kernel 3.19 durchgeführt. Die Datensätze, Tabelle 1, passen auch mit zusätzlichen programmspezifischen Datenstrukturen noch in den Hauptspeicher. Die LDBC-Graphdatensets wurden mit dem gleichnamigen Datengenerator[2] erstellt.

In unseren Tests, Tabelle 2, zeigt MATLAB die längsten Laufzeiten und das schlechtere Skalierungsverhalten[3], weshalb wir uns in der weiteren Auswertung auf den Vergleich mit Spark fokussieren. Im Bezug auf die Vektordatensätze zeigen HyPer und Spark ein ähnliches Skalierungsverhalten, wobei HyPer im Datenbankkern jedoch um Faktor 1–2 schneller ist. Für die PageRank-Graphanalyse zeigt sich, dass HyPer sowohl besser skaliert als Spark, als auch 1–2 Größenordnungen schneller ist. HyPer mit in SQL spezifizierten Data-Mining-Anfragen zeigt Potential, erzeugt aber im Fall von K-Means noch keine optimalen Ausführungspläne.

[1] Die explizite Klammerung der Unteranfrage ist nötig, da wir dort beliebige Anfragen erlauben; einfache Kommatrennung führt zu Mehrdeutigkeiten in der Grammatik.

[2] Siehe `https://github.com/ldbc/ldbc_snb_datagen`.

[3] MATLAB führt die Algorithmen sequentiell aus. Jedoch wäre die Laufzeit auch bei perfekter Skalierung über die sechs verfügbaren CPU-Kerne noch immer unterlegen.

Tabelle 1. Datensätze zur Evaluierung der gewählten Verfahren

Datensatz	**Datenmodell**	**# Tupel**	**# Dimensionen**	**Größe in GB**
Syn 1	Vektordaten	50 000	50	0,01
Syn 2	Vektordaten	15 000 000	4	0,22
Syn 3	Vektordaten	15 000 000	50	2,79
		# Knoten	**# Kanten**	
LDBC SF 1	Graph/Kantenliste	10 993	451 522	0,03
LDBC SF 10	Graph/Kantenliste	72 949	4 641 430	0,26

Tabelle 2. Laufzeiten in Sekunden, OOM = Out of Memory

	Datensatz	**Spark**	**MATLAB**	**HyPer Op**	**HyPer SQL**
K-Means,	Syn 1	0,287	1,504	0,110	1,040
$k = 3$,	Syn 2	5,347	582,139	4,643	87,588
3 Iterationen	Syn 3	51,632	OOM	19,496	369,438
PageRank,	LDBC SF 1	2,329	4,506	0,16	0,30
$d = 0.85$,	LDBC SF 10	72,391	OOM	1,69	4,76
$e = 0.0001$					

Zusammenfassung Wir haben eine vierschichtige Integration von Data Mining in unser relationales Hauptspeicherdatenbanksystem HyPer vorgestellt. Wichtige Algorithmen sind hochoptimiert im Datenbankkern integriert und können direkt per SQL aufgerufen werden, wo sie auch Laien leicht zugänglich sind. Zusätzliche Algorithmen können durch unsere Spracherweiterung direkt in SQL spezifiziert werden und nutzen somit HyPers effiziente Codegenerierung und Laufzeitumgebung. Unsere Testergebnisse zeigen, dass Data Mining auf Vektor- und Graphdaten in HyPer performanter ist und besser skaliert als in vergleichbaren state-of-the-art Datenanalysesystemen. Zeitaufwändige ETL-Zyklen entfallen.

Literatur

1. N. Aggarwal, A. Kumar, H. Khatter, and V. Aggarwal. Analysis the effect of data mining techniques on database. *Advances in Engineering Software*, 47(1), 2012.
2. D. Cai. Litekmeans: the fastest matlab implementation of kmeans. *Available at:* *http://www.zjucadcg.cn/dengcai/Data/Clustering.html*, 2011.
3. F. Färber, N. May, W. Lehner, P. Große, I. Müller, H. Rauhe, and J. Dees. The SAP HANA database – an architecture overview. *IEEE Data Eng. Bull.*, 35, 2012.
4. A. Kemper and T. Neumann. HyPer: A hybrid OLTP & OLAP main memory database system based on virtual memory snapshots. In *ICDE*, April 2011.
5. MATLAB. *Version 8.5 (R2015a).* The MathWorks Inc., 2015.
6. P. Tamayo et al. Oracle data mining. In O. Maimon and L. Rokach, editors, *Data Mining and Knowledge Discovery Handbook*, pages 1315–1329. Springer US, 2005.
7. Xindongi Wu et al. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37, 2008.
8. M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica. Spark: Cluster computing with working sets. In *HotCloud'10.* USENIX Association, 2010.

KDML: Workshop on Knowledge Discovery, Data Mining and Machine Learning

Fast Description-Oriented Community Detection using Subgroup Discovery
(Abstract)

Martin Atzmueller, Stephan Doerfel, and Folke Mitzlaff

University of Kassel, ITeG Research Center, KDE Group
Wilhelmshöher Allee 73, 34121 Kassel, Germany

{atzmueller, doerfel, mitzlaff}@cs.uni-kassel.de

Abstract

Communities can intuitively be defined as subsets of nodes of a graph with a dense structure. However, for mining such communities usually only structural aspects are taken into account. Typically, no concise and easily interpretable community description is provided. For tackling this issue, we focus on fast description-oriented community detection using subgroup discovery, cf. [1, 2]. In order to provide both structurally valid and interpretable communities we utilize the graph structure as well as additional descriptive features of the contained nodes. A descriptive community pattern built upon these features then describes and identifies a community given by a set of nodes, and vice versa. Essentially, we mine for patterns in the "description space" characterizing interesting sets of nodes in the "graph/community space"; the interestingness of a community is then evaluated by a selectable quality measure.

We aim at identifying communities according to standard community quality measures, while providing characteristic descriptions of the respective communities at the same time. In order to implement an efficient approach, we propose several optimistic estimates of standard community quality functions. Together with the proposed exhaustive branch-and-bound algorithm, these estimates enable fast description-oriented community detection. This is demonstrated in an evaluation using five real-world data sets, obtained from three different social media applications.

References

1. Atzmueller, M., Doerfel, S., Mitzlaff, F.: Description-Oriented Community Detection using Exhaustive Subgroup Discovery. Information Sciences (2015), http://dx.doi.org/10.1016/j.ins.2015.05.008
2. Atzmueller, M., Mitzlaff, F.: Efficient Descriptive Community Mining. In: Proc. 24th International FLAIRS Conference. pp. 459 – 464. AAAI Press, Palo Alto, CA, USA (2011)

This abstract summarizes the paper [1].

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

Automatic Threshold Calculation for the Categorical Distance Measure ConDist

Markus Ring[1], Dieter Landes[1], and Andreas Hotho[2]

[1] Faculty of Electrical Engineering and Informatics, Coburg University of Applied Sciences and Arts, 96450 Coburg, Germany,
{markus.ring,dieter.landes}@hs-coburg.de,
[2] Data Mining and Information Retrieval Group, University of Würzburg, 97074 Würzburg, Germany
{hotho}@informatik.uni-wuerzburg.de

Abstract. The measurement of distances between objects described by categorical attributes is a key challenge in data mining. The unsupervised distance measure *ConDist* approaches this challenge based on the idea that categorical values within an attribute are similar if they occur with similar value distributions on correlated context attributes. An impact function controls the influence of the correlated context attributes in *ConDist's* distance calculation process.
ConDist requires a user-defined threshold to purge context attributes whose correlations are caused by noisy, non-representative or small data sets. In this work, we propose an automatic threshold calculation method for each pair of attributes based on their value distributions and the number of objects in the data set. Further, these thresholds are also considered when applying *ConDist's* impact function. Experiments show that this approach is competitive with respect to well selected user-defined thresholds and superior to poorly selected user-defined thresholds.

Keywords: categorical data, distance measure, unsupervised learning

1 Introduction

Distance calculation between objects is a key requirement for many data mining tasks like clustering, classification or outlier detection [15]. Objects are described by a set of attributes which can be divided into continuous and categorical attributes. For continuous attributes, distance calculation is well understood and mostly uses the Minkowski distance [2]. For categorical attributes, defining meaningful distance measures is more challenging since the values within such attributes have no inherent order [4]. However, several methods exist to address this issue. A comprehensive overview of categorical distance measures is given

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

in [4]. Yet, more sophisticated categorical distance measures incorporate statistical information like correlations about the data [8,9,11,14]. *ConDist* (Context based Categorical Distance Measure) [14] is such an unsupervised categorical distance measure. For distance calculation, *ConDist* extracts available information from correlations between the target attribute (the attribute for which distances shall be calculated) and the correlated context attributes. *ConDist* uses a correlation measure based on the information gain. Each context attribute whose correlation exceeds a user-defined threshold θ is used for distance calculation. This threshold θ must be large enough to ensure that context attributes are purged whose correlations are caused by noisy, non-representative or too small data sets. Simultaneously, the threshold θ must be small enough to retain context attributes with significant correlations.

In this paper, we propose a data-driven method for calculating *ConDist's* threshold. In [14], the user has to define a single threshold for all attributes. In contrast to this approach, the proposed method calculates an individual threshold $\theta_{X|Y}$ for each combination of target attribute X and context attribute Y. These thresholds $\theta_{X|Y}$ can be better adapted to the specific correlation requirements of two concrete attributes than a single threshold θ. We consider the number of objects in the data set and the value distributions of target attribute X and context attribute Y when calculating the individual thresholds $\theta_{X|Y}$. The calculated thresholds $\theta_{X|Y}$ are also taken into account when applying *ConDist's* impact function. The impact function controls the influence of the correlated context attributes in *ConDist's* distance calculation process and considers the varying amount of information that can be extracted from a correlated context attribute. The proposed method for the automatic threshold calculation makes *ConDist* parameterless and simplifies the application of the distance measure.

The rest of the paper is organized as follows: Related work on categorical distances measures and their approaches for identifying correlated context attributes are discussed in Section 2. Section 3 gives a short description of the categorical distance measure *ConDist*. Section 4 introduces the proposed method for the automatic threshold calculation. Section 5 gives an experimental evaluation of the proposed automatic threshold calculation method and the results are discussed in Section 6. The last section summarizes the paper.

2 Related Work

Unsupervised categorical distance measures may be divided into distance calculation (I) without considering context attributes and (II) considering context attributes.

Boriah et al. [4] give a comprehensive overview of distances measures from category (I). These distance measures ignore information that could be extracted from context attributes. For example, the distance measure *Eskin* only uses the cardinality of the target attribute domain to calculate distances.

Distance measures from category (II) consider context attributes in the distance calculation process [1,8,9,10,11,14]. For example, the distance measures proposed in [1] and [11] use all context attributes for distance calculation without distinguishing between correlated and uncorrelated. Conversely, the proposed distance measures in [8] and [9] use only a subset of context attributes for distance calculation. Jia and Cheung [9] use a normalized version of the mutual information (NMI) [3], whereas DILCA [8] relies on Symmetric Uncertainty (SU) [17] to determine the correlation between two attributes. For both, NMI and SU, the user has to define a threshold for the selection of correlated context attributes. The distance measure *CBDL* [10] uses the *Pearson's chi-squared test* χ^2 [12] for identifying correlated context attributes. Yet, the user needs to provide a significance level alpha for the *Pearson's chi-squared test* χ^2.

Like [9], *ConDist* [14] only uses correlated context attributes for distance calculation. It measures the correlation between attributes based on the information theoretical concept of entropy. In [14], the user has to define a threshold θ for the selection of correlated context attributes. In this work, we propose an automatic threshold calculation method for *ConDist*, which is based on the value distribution of the attributes and the number of objects in the data set.

3 The Distance Measure ConDist

In this section, we give a short description of the categorical distance measure *ConDist* [14]. The core idea is presented in Section 3.1. Since *ConDist* uses correlated context attributes in the distance calculation process, we explain in Section 3.2 how the set of correlated context attributes is derived. Section 3.3 describes the impact function of *ConDist* which accounts for the varying amount of information that can be extracted from a correlated context attribute.

3.1 ConDist

The distance between two objects A and B is calculated as the sum of distances in each attribute and defined as follows:

$$ConDist(A,B) = \sum_X w_X \cdot \frac{d_X(A,B)}{d_{X,max}}, \tag{1}$$

where w_X denotes a weighting factor assigned to attribute X. Since w_X is not relevant for threshold calculation, the reader is referred to [14] for further details on w_X. The function $d_X(A,B)$ denotes the distance of the values A_X and B_X of the objects A and B in attribute X. The maximum distance between any two values $x, u \in dom(X)$ of attribute X is given by $d_{X,max}$ and is used to normalize all attribute distances to the interval $[0,1]$.

The distance $d_X(A,B)$ between two values A_X and B_X within an attribute X is calculated according to the following formula:

$$d_X(A,B) = \sum_{Y \in context_X} impact_X(Y) \sqrt{\sum_{y \in dom(Y)} \Big(p(y|A_X) - p(y|B_X)\Big)^2}, \tag{2}$$

where $dom(Y)$ is the domain of attribute Y, and $p(y|A_X) = p(y|X = A_X)$ denotes the probability that value y of context attribute Y is observed under the condition that value A_X of attribute X is observed in data set D. The set of correlated context attributes for a specific target attribute X is given by $context_X$ (see Section 3.2). The function $impact_X(Y)$ controls the influence of context attribute Y on target attribute X and is described in Section 3.3.

3.2 Selection of Context Attributes

ConDist uses an asymmetric function $cor(X|Y)$ to measure the correlation between a target attribute X and a context attribute Y. The function $cor(X|Y)$ is defined as follows:

$$cor(X|Y) = \frac{IG(X|Y)}{H(X)}, \tag{3}$$

where $H(X)$ is the entropy of the target attribute X and $IG(X|Y)$ is the information gain of target attribute X given context attribute Y. The information gain $IG(X|Y)$ is the difference between the entropy $H(X)$ of attribute X and the conditional entropy $H(X|Y)$ of attribute X given attribute Y:

$$IG(X|Y) = H(X) - H(X|Y) \tag{4}$$

Consequently, the function $cor(X|Y)$ is normalized to the interval $[0, 1]$. The higher the value of the correlation function $cor(X|Y)$, the higher the correlation between the two attributes. In [14], all context attributes whose correlations exceed a user-defined threshold θ are added to the set of correlated context attributes $context_X$ for target attribute X:

$$context_X = \{Y \mid cor(X|Y) \geq \theta\} \tag{5}$$

Note that the target attribute X itself is always in the set of correlated context attributes $context_X$ since $cor(X|X) = 1$.

3.3 The Impact of Context Attributes

ConDist uses an impact function $impact_X(Y)$ to control the influence of a correlated context attribute Y on target attribute X in the distance calculation process. This function accounts for the fact that the varying amount of extractable information depends on the degree of correlation between the attributes X and Y. In general, the quality of the extracted information grows with the strength of the correlation. However, for highly correlated attributes, the amount of extractable information decreases. In the extreme case of a perfectly correlated context attribute Y, no further information about distinct values in target attribute X can be extracted since Y predicts the values of X. To be precise, *ConDist* uses the impact function as defined as:

$$impact_X(Y) = cor(X|Y)\Big(1 - \frac{1}{2}cor(X|Y)\Big)^2, \tag{6}$$

where $cor(X|Y)$ is the correlation function introduced in Section 3.2.

4 Automatic Threshold Calculation Method

In this section, we propose a data-driven approach to replace the user-defined threshold θ of Section 3.2. In principle, *ConDist's* impact function should control automatically the influence of the context attributes without additional thresholds. However, the experiments in [14] showed that an additional threshold θ is necessary, especially for non-correlated data sets.

In Section 4.1, we use an example to explain the reason why an additional threshold is necessary. Based on that example, we propose a way how this threshold could be calculated from the data set in Section 4.2. The proposed automatic threshold calculation method involves an additional adjustment of *ConDist's* impact function which is described in Section 4.3.

4.1 Problem Description by Example

The impact function $impact_X(Y)$ (Section 3.3) controls the influence of context attributes in the distance calculation process and depends on the value of the correlation function $cor(X|Y)$ (Section 3.2). We give an example when these two functions fail to control the influence of context attributes without additional threshold θ.

Table 1: Example data set which describes eight people with three categorical attributes *sex*, *height* and *haircolor*.

#	sex	haircolor	height
1	male	brown	tall
2	male	blond	tall
3	male	black	medium
4	male	brown	medium
5	female	blond	medium
6	female	black	small
7	female	brown	small
8	female	blond	small

Consider the example data set in Table 1. Let us assume, we want to calculate distances for the attribute *height*. In this case, *sex* and *haircolor* are the context attributes. Further, we may assume that in the considered population attributes *haircolor* and *height* are independent of each other, while attributes *height* and *sex* are correlated. When applying *ConDist's* correlation function $cor(X|Y)$ and impact function $impact_X(Y)$, we achieve the following results:

$$cor(height|sex) = \frac{IG(height|sex)}{H(height)} \approx \frac{1.561 - 0.906}{1.561} \approx 0.420 \qquad (7)$$

$$cor(height|haircolor) = \frac{IG(height|haircolor)}{H(height)} \approx \frac{1.561 - 1.439}{1.561} \approx 0.122 \qquad (8)$$

$$impact_{height}(sex) \approx 0.262 \tag{9}$$

$$impact_{height}(haircolor) \approx 0.108 \tag{10}$$

As expected, the context attribute *sex* has higher impact on the target attribute *height* than context attribute *haircolor*. However, the context attribute *haircolor* has also a small impact on the target attribute *height*. Since we have also a highly correlated context attribute *sex*, the small impact of context attribute *haircolor* is almost negligible.

However, if we would have only the context attribute *haircolor*, the small impact factor would lead to small differences for distinct values in target attribute *height*. These small differences originate from the fact that the estimated probability density functions used in $cor(X|Y)$ are not representative due to the small training data set. Consequently, the differences are conceptually not intended since, given the particular population of our example, the context attribute *haircolor* is independent from *height*. In this case, it would be preferable to use only the target attribute itself for distance calculation. Therefore, a threshold θ is necessary to purge such context attributes.

4.2 Data-Driven Threshold Calculation

The example in Section 4.1 shows that too small data sets are problematic for the correlation function $cor(X|Y)$. This follows from the fact that $cor(X|Y)$ requires the information gain $IG(X|Y)$, which in turn requires the entropy of attribute X and the conditional entropy of attribute X given attribute Y. The entropy $H(X)$ and the conditional entropy $H(X|Y)$ are defined as follows:

$$H(X) = -\sum_{x \in dom(X)} p(x) \log_2 \big(p(x)\big) \text{ and} \tag{11}$$

$$H(X|Y) = -\sum_{y \in dom(Y)} p(y) \sum_{x \in dom(X)} p(x|y) \log_2 \big(p(x|y)\big), \tag{12}$$

where $p(x)$ is the probability of value x and $p(x|y)$ is the conditional probability of value x given value y in data set D. Consequently, the probability density functions $p(X)$ and $p(Y)$ of the attributes X and Y are necessary for calculating $H(X)$ and $H(X|Y)$. These two functions can be estimated more accurately if the data set is large. Consequently, the smaller the data set, the higher the possibility of errors in the results delivered by the correlation function.

Further, two attributes X and Y are non-correlated in *ConDist's* correlation function $cor(X|Y)$, if and only if the following equation holds:

$$H(X) = H(X|Y) \tag{13}$$

Equation (13) requires that the conditional probability density functions of attribute X given a value $y \in dom(Y)$ are all identical and equal to the probability density function of attribute X. The larger the cardinality of $dom(X)$ and $dom(Y)$, the more objects are necessary to fulfill this requirement in the case of non-correlated attributes since the value distributions and conditional value distributions must be estimated from the data set. Consequently, the cardinality and the distribution of $dom(X)$ and $dom(Y)$ should be considered in the threshold calculation process as well. Both factors are reflected in the entropy of an attribute.

Therefore, we calculate the threshold $\theta_{X|Y}$ based on these two aspects:

$$\theta_{X|Y} = \frac{H(X) \cdot H(Y)}{n}, \tag{14}$$

where n is the number of objects in the data set. This threshold decreases with an increasing number of objects and increases with increasing attribute entropies $H(X)$ and $H(Y)$. The threshold $\theta_{X|Y}$ may be viewed as an estimate of the portion of correlation that is due to estimating the probability density functions $p(X)$ and $p(Y)$ from the data set. The calculation of $\theta_{X|Y}$ is easy and no user-defined parameter is necessary.

If we apply the automatic calculation of the threshold $\theta_{X|Y}$ to the example in the Section 4.1, we can observe the following results:

$$\theta_{height|sex} = \frac{H(height) \cdot H(sex)}{n} \approx \frac{1.561 \cdot 1}{8} \approx 0.195 \text{ and} \tag{15}$$

$$\theta_{height|haircolor} = \frac{H(height) \cdot H(haircolor)}{n} \approx \frac{1.561 \cdot 1.561}{8} \approx 0.305. \tag{16}$$

The correlation value of attribute *sex* (see Equation (7)) exceeds the calculated threshold $\theta_{height|sex}$, whereas the correlation value of attribute *haircolor* (see Equation (8)) does not exceed the threshold $\theta_{height|haircolor}$. Applying the proposed context-sensitive threshold $\theta_{X|Y}$ would imply that only the attribute *sex* would be added to the set of correlated context attributes $context_{height}$ for target attribute *height*.

4.3 Adjustment of the Impact Function

In Section 4.2, we interpreted the threshold $\theta_{X|Y}$ as the amount of correlation which is caused by estimating probability density functions from the data set. Consequently, this amount of correlation should also be considered in the impact function $impact_X(Y)$. To that end, we adjust *ConDist's* impact function $impact_X(Y)$ as follows:

$$impact_X(Y) = \begin{cases} 0 & \text{if } cor(X|Y) \leq \theta_{X|Y} \\ cor_\theta(X|Y)\Big(1 - \frac{1}{2}cor_\theta(X|Y)\Big)^2 & \text{if } cor(X|Y) > \theta_{X|Y} \end{cases}, \tag{17}$$

where $cor_\theta(X|Y)$ is the adjusted correlation value rescaled to the interval $[0, 1]$:

$$cor_\theta(X|Y) = \frac{cor(X|Y) - \theta_{X|Y}}{1 - \theta_{X|Y}}. \tag{18}$$

If we apply the new impact function to the example in the Section 4.1, we can observe the following results:

$$cor_\theta(height|sex) \approx \frac{0.420 - 0.195}{1 - 0.195} \approx 0.280 \tag{19}$$

$$impact_{height}(sex) \approx 0.280\left(1 - \frac{1}{2} \cdot 0.280\right)^2 \approx 0.207 \tag{20}$$

$$impact_{height}(haircolor) = 0 \tag{21}$$

Using the proposed approach, only the context attribute *sex* has an impact on target attribute *height*.

5 Experimental Evaluation

This section presents an experimental evaluation of the automatic calculation of the threshold $\theta_{X|Y}$ (Section 4). We compare our new approach with the user-defined threshold method presented in [14] and with the categorical distance measure *DILCA* [8], which is the most serious competitor in [14]. For *DILCA*, we used the non-parametric approach $DILCA_{RR}$ as described in [8].

5.1 Evaluation Methodology

We evaluate the different threshold calculation methods for *ConDist* in the context of classification. A k-Nearest-Neighbor classifier is used to compare the different categorical distance measures (*DILCA* and *ConDist*) and the different methods for threshold calculation in *ConDist*. For simplification, we do not try to optimize the selection of the parameter k of the k-Nearest-Neighbor classifier. Instead we fix the number of neighbors $k = 7$ in all tests in order to create an equal base for the different configurations. We evaluate by 10-fold-cross validation and use the classification accuracy as evaluation measure. To reduce confounding effects of the generated subsets, 10-fold cross-validation is repeated 100 times with different subsets for each data set.

For evaluation, the *multivariate categorical data sets for classification* from the UCI machine learning repository [13] are chosen. We exclude data sets with less than 25 objects (e.g., *Balloons*) or mainly binary attributes (e.g., *Chess*). Furthermore, we include some *multivariate mixed data sets for classification* from [13] which mainly consist of categorical attributes and some integer attributes with a small set of distinct values (e.g. an integer attribute that contains the number of students in a course): *Teaching Assistant Evaluation, Breast Cancer Wisconsin, Dermatology* and *Post-Operative Patient*. All integer attributes are treated as categorical. The final set of data sets is given in Table 2. The column *Correlation* contains the average correlation between each distinct pair of attributes, calculated by the function $cor(X|Y)$, see Equation (3). The value ranges from 0 if no correlation exists to 1 if all attributes are perfectly correlated. The data sets are separated in two groups: correlated (Correlation > 0) and non-correlated (Correlation $= 0$).

Table 2: Characteristics of the data sets.

Data Sets	Instances	Attributes	Classes	Correlation
Teaching Assistant Evaluation	151	5	3	0.336
Soybean Large	307	35	19	0.263
Breast Cancer Wisconsin	699	10	2	0.216
Dermatology	366	34	6	0.098
Lymphography	148	18	4	0.070
Audiology-Standard	226	69	24	0.044
Hayes-Roth	160	4	3	0.045
Post-Operative Patient	90	8	3	0.031
TicTacToe	958	9	2	0.012
Monks	432	6	2	0.000
Balance-Scale	625	4	3	0.000
Car	1728	6	4	0.000
Nursey	12960	8	5	0.000

5.2 Experimental Setup and Results

This experiment compares the automatic calculated threshold $\theta_{X|Y}$ with various user-defined thresholds θ in *ConDist* and with the categorical distance measure *DILCA*. The threshold θ expresses the minimum value of the function $cor(X|Y)$ that a context attribute Y has to achieve in order to be selected as correlated context attribute for the target attribute X. The higher the threshold θ, the fewer context attributes are used. In the extreme case of $\theta = 0$, all context attributes are used for distance calculation. The automatic calculated threshold $\theta_{X|Y}$ follows the approach of Section 4. The results of this experiment are summarized in Table 3, where each column contains the average classification accuracies for a particular threshold.

Table 3 shows that the automatic calculation of the threshold $\theta_{X|Y}$ achieves the best average classification accuracy. The user-defined thresholds $\theta = 0.01$ and $\theta = 0.02$ achieve similar good results. Without any threshold $\theta = 0$, a decreasing classification accuracy can be observed for non-correlated data sets. For too high user-defined thresholds θ, the average classification accuracies decrease. Compared with *DILCA*, the proposed approach $\theta_{X|Y}$ is comparable for highly correlated data sets and superior for weakly- and non-correlated data sets.

Statistical Significance Test. This test aims at examining if the differences in Table 3 are statistically significant. Demšar [5] deals with the statistical comparison of classifiers over multiple data sets. They recommend the Wilcoxon Signed-Ranks Test [16] for the comparison of two classifiers and the Friedman-Test [6,7] for the comparison of multiple classifiers. Following this line, we use the Friedman-Test to compare all different configurations and the Wilcoxon Signed-Ranks Test for post-hoc tests. The Friedman-Test is significant for $p < 0.05$; thus we can reject the null hypothesis that all threshold calculation methods in *ConDist* and

Table 3: Classification accuracies for the proposed automatic threshold calculation (column $\theta_{X|Y}$), various user-defined thresholds and *DILCA*. Each column contains the results for a specific threshold, e.g. the column 0.02 contains the results for $\theta = 0.02$.

	ConDist								*DILCA*
Data Set	$\theta_{X\|Y}$	0	0.01	0.02	0.05	0.1	0.2	1.0	$DILCA_{RR}$
Teaching A. E.	49.93	49.85	49.85	49.85	49.71	48.74	48.74	45.84	**50.86**
Soybean Large	91.76	91.74	91.74	91.79	**91.82**	89.75	89.36	91.30	91.48
B. C. Wisconsin	96.17	96.13	96.13	96.13	96.13	96.15	**96.25**	95.25	95.55
Dermatology	96.70	96.74	96.74	96.76	96.81	96.35	96.23	95.90	**97.97**
Lymphography	**83.36**	**83.36**	**83.36**	83.30	83.01	81.99	82.01	81.26	82.09
Hayes-Roth	68.59	68.11	68.36	68.50	**69.21**	64.47	64.47	61.74	67.59
Audiology-Std.	66.22	66.33	66.27	66.27	**66.56**	65.41	61.81	61.35	62.31
Postoperative P.	69.71	**69.83**	69.81	69.62	**69.83**	68.27	68.58	68.59	68.22
TicTacToe	**99.99**	**99.99**	**99.99**	**99.99**	94.74	94.74	94.74	94.74	90.65
Car	**90.56**	88.98	**90.56**	**90.56**	**90.56**	**90.56**	**90.56**	**90.56**	90.25
Monks	**97.32**	95.16	**97.32**	**97.32**	**97.32**	**97.32**	**97.32**	**97.32**	92.06
Balance-Scale	**78.66**	77.35	**78.66**	**78.66**	**78.66**	**78.66**	**78.66**	**78.66**	78.43
Nursey	**94.94**	94.43	**94.94**	**94.94**	**94.94**	**94.94**	**94.94**	**94.94**	92.61
Average	**83.38**	82.92	83.36	83.36	83.02	82.10	81.82	81.34	81.53

DILCA are equivalent. Subsequently, we applied the Wilcoxon Signed-Ranks Test with $\alpha = 0.05$ on the classification accuracies of Table 3.

Table 4 shows significant differences between $\theta_{X|Y}$ and *DILCA* and between $\theta_{X|Y}$ and the user-defined thresholds $\theta = 0.1$, $\theta = 0.2$ and $\theta = 1.0$. For the remaining user-defined thresholds θ, the Wilcoxon Signed-Ranks Test shows no statistically significant differences.

6 Discussion

For correlated data sets, high user-defined thresholds θ lead to decreasing results, e.g. $\theta = 0.1$, $\theta = 0.2$ or $\theta = 1.0$ for the data sets *Teaching Assistant Evaluation* and *Lymphography*. For these thresholds, many useful correlated context attributes are discarded. The same observation can be made for weakly-correlated data sets at lower thresholds. Consider the decreasing classification accuracy for the data set *TicTacToe* at threshold $\theta = 0.05$. For non-correlated data sets, nearly all threshold methods achieve the same results. Only the absence of any threshold ($\theta = 0$) leads to inferior results. In this case, non-correlated context attributes are added to the set of context attributes $context_X$, which may contribute noise to the distance calculation process.

The proposed automatic calculation of the threshold $\theta_{X|Y}$ achieves good results for correlated and non-correlated data sets. As a consequence, the proposed method achieves the best average classification accuracy. The average classification accuracies for user-defined thresholds $\theta = 0.01$, $\theta = 0.02$ and $\theta = 0.05$ are

Table 4: Results of the Wilcoxon Signed-Ranks Test comparing the classification accuracies of the automatic calculation of the threshold $\theta_{X|Y}$ with various user-defined thresholds θ and with *DILCA*. The first row contains the calculated p-values, the second row contains the result of the Wilcoxon Signed-Ranks Test: *yes*, if $\theta_{X|Y}$ performs significantly different, *no* otherwise.

	$\theta = 0$	$\theta = 0.01$	$\theta = 0.02$	$\theta = 0.05$	$\theta = 0.1$	$\theta = 0.2$	$\theta = 1$	$DILCA$
p-value	0.0830	0.7998	0.2070	1	0.0092	0.0113	0.0092	0.0231
significant	no	no	no	no	yes	yes	yes	yes

marginally worse. The Wilcoxon-Signed Ranks Test confirms that there are no statistical significant differences between them. In contrast to this, statistically significant differences can be observed for too high user-defined thresholds.

These observations indicate that the proposed automatic calculation of the threshold $\theta_{X|Y}$ is superior to poorly selected user-defined thresholds and competitive to well selected user-defined thresholds. Consequently, $\theta_{X|Y}$ is preferable to the user-defined approach in [14], since the user-defined parameter θ is omitted and the quality of results does not deteriorate.

For highly correlated data sets, the results of the proposed approach $\theta_{X|Y}$ and *DILCA* are comparable. For weakly- and non-correlated data sets, *DILCA* achieves inferior results in comparison to *ConDist*. This is because *DILCA* uses only context attributes for distance calculation which results in random distances if all context attributes are non-correlated.

7 Summary

Categorical distance calculation is a key requirement for many data mining tasks. In this paper, we propose an extension for the unsupervised categorical distance measure *ConDist* [14]. *ConDist* uses the correlation between attributes to extract available information for distance calculation. In [14], the user has to define a threshold θ for the selection of correlated context attributes. This threshold θ has to purge context attributes whose correlations are caused by noisy, non-representative or too small data sets.

In this work, we proposed an automatic threshold calculation method for the distance measure *ConDist*. This approach calculates for each pair of target attribute X and context attribute Y an individual threshold instead of using a single user-defined threshold θ. The calculated thresholds $\theta_{X|Y}$ depend on the number of objects in the data set and the entropies of the attributes. Consequently, these individual thresholds can be better adapted to the specific correlation requirements of each pair of attributes. Further, additional adjustments were made to *ConDist's* impact function $impact_X(Y)$.

The proposed extension makes *ConDist* parameterless and simplifies the application of the distance measure. Our experiments show that the automatic threshold calculation method is competitive to well selected user-defined thresholds θ and superior to poorly selected user-defined thresholds θ. For these two reasons, the proposed approach is preferable to the user-defined approach in [14].

Acknowledgements This work is funded by the Bavarian Ministry for Economic Affairs through the WISENT project (grant no. IUK 452/002) and by the DFG through the PoSTs II project (grant no. HO 2586/2-2).

References

1. Ahmad, A., Dey, L.: A method to compute distance between two categorical values of same attribute in unsupervised learning for categorical data set. Pattern Recognition Letters 28(1), 110–118 (2007)
2. Alamuri, M., Surampudi, B.R., Negi, A.: A survey of distance/similarity measures for categorical data. In: Proc. of IJCNN. pp. 1907–1914. IEEE (2014)
3. Au, W.H., Chan, K.C., Wong, A.K., Wang, Y.: Attribute clustering for grouping, selection, and classification of gene expression data. IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) 2(2), 83–101 (2005)
4. Boriah, S., Chandola, V., Kumar, V.: Similarity measures for categorical data: A comparative evaluation. In: Proc. SIAM Int. Conference on Data Mining. pp. 243–254 (2008)
5. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. The Journal of Machine Learning Research 7, 1–30 (2006)
6. Friedman, M.: The use of ranks to avoid the assumption of normality implicit in the analysis of variance. Journal of the American Statistical Association 32(200), 675–701 (1937)
7. Friedman, M.: A comparison of alternative tests of significance for the problem of m rankings. The Annals of Mathematical Statistics 11(1), 86–92 (1940)
8. Ienco, D., Pensa, R.G., Meo, R.: Context-based distance learning for categorical data clustering. In: Advances in Intelligent Data Analysis VIII, pp. 83–94. Springer (2009)
9. Jia, H., Cheung, Y.M.: A new distance metric for unsupervised learning of categorical data. In: Proc. of IJCNN. pp. 1893–1899. IEEE (2014)
10. Khorshidpour, Z., Hashemi, S., Hamzeh, A.: Cbdl: Context-based distance learning for categorical attributes. Int. J. Intell. Syst. 26(11), 1076–1100 (2011)
11. Le, S.Q., Ho, T.B.: An association-based dissimilarity measure for categorical data. Pattern Recognition Letters 26(16), 2549–2557 (2005)
12. Lehmann, E., Romano, J.: Testing Statistical Hypotheses. Springer Texts in Statistics, Springer (2005)
13. M. Lichman: Uci machine learning repository (2013), `http://archive.ics.uci.edu/ml`
14. Ring, M., Otto, F., Becker, M., Niebler, T., Landes, D., Hotho, A.: Condist: A context-driven categorical distance measure. In: Machine Learning and Knowledge Discovery in Databases. pp. 251–266. Springer (2015)
15. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to data mining. Pearson Addison Wesley Boston (2006)
16. Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics bulletin 1(6), 80–83 (1945)
17. Yu, L., Liu, H.: Feature selection for high-dimensional data: A fast correlation-based filter solution. In: ICML. vol. 3, pp. 856–863 (2003)

Media Bias in German Online Newspapers

Alexander Dallmann[1], Florian Lemmerich[2], Daniel Zoller[1], and Andreas Hotho[1,3]

[1] Data Mining and Information Retrieval Group, University of Würzburg (Germany)
{dallmann, zoller, hotho}@informatik.uni-wuerzburg.de
[2] Computational Social Science Group, GESIS - Leibniz Institute for the Social Sciences (Germany)
florian.lemmerich@gesis.org
[3] L3S Research Center (Germany)

Online newspapers have been established as a crucial information source, at least partially replacing traditional media like television or print media. As all other media, online newspapers are potentially affected by media bias. This describes non-neutral reporting of journalists and other news producers, e.g., with respect to specific opinions or political parties. Analysis of media bias has a long tradition in political science. However, traditional techniques rely heavily on manual annotation and are thus often limited to the analysis of small sets of articles.

In [1] we investigate a dataset that covers all political and economical news over a four-year period from four leading German online newspapers, namely *faz.net*, *spiegel.de*, *taz.de*, and *zeit.de*. We perform a comparative analysis of party coverage by analyzing the occurrences of both acronyms and parliament members in title, text and meta-information. The comparative analysis shows significant differences in coverage between different parties. For example, it can be observed that *faz.net* favors the conservative parties CDU and CSU over the left and green parties Linke and Grüne.

We also investigate a relation in ideology by comparing the usage of ideological terms (e.g., freedom, solidarity) in online newspapers and party manifestos, by counting occurrences and computing the cosine-similarity. Results show that a higher similarity in the usage of key vocabulary can be observed for some parties and online newspapers. For example *taz.de* tends to favor a key vocabulary similar to the left party Linke over other parties.

Finally, we analyze the expression of sentiment towards parties but the results are inconclusive.

References

1. Dallmann, A., Lemmerich, F., Zoller, D., Hotho, A.: Media bias in german online newspapers. In: Proceedings of the 26th ACM Conference on Hypertext & Social Media. pp. 133–137. ACM (2015)

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

Development of an Automatic Pollen Classification System Using Shape, Texture and Aperture Features

Celeste Chudyk[1], Hugo Castaneda[2], Romain Leger[2], Islem Yahiaoui[2], Frank Boochs[1]

[1] i3mainz, University of Applied Sciences Mainz, Germany
{celeste.chudyk,boochs}@hs-mainz.de
[2] Dijon Institute of Technology, Burgundy University, France
{hugo.castaneda,romain.leger,islem.yahiaoui}@iut-dijon.u-bourgogne.fr

Abstract. Automatic detection and classification of pollen species has value for use inside of palynologic allergen studies. Traditional labeling of different pollen species requires an expert biologist to classify particles by sight, and is therefore time-consuming and expensive. Here, an automatic process is developed which segments the particle contour and uses the extracted features for the classification process. We consider shape features, texture features and aperture features and analyze which are useful. The texture features analyzed include: Gabor Filters, Fast Fourier Transform, Local Binary Patterns, Histogram of Oriented Gradients, and Haralick features. We have streamlined the process into one code base, and developed multithreading functionality to decrease the processing time for large datasets.

Keywords: Image processing, Machine learning, Pollen, Texture classification

1 Introduction

Currently, pollen count information is usually limited to generalizing all pollen types with no access to information regarding particular species. In order to differentiate species, typically a trained palynologist would have to manually count samples using a microscope. Advances in image processing and machine learning enable the development of an automatic system that, given a digital image from a bright-field microscope, can automatically detect and describe the species of pollen particles present.

We build upon previous work from within our lab which has planned the structure for a complete personal pollen tracker [6]. For image classification,

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

preliminary results have shown that extraction of both shape features and aperture features lead to useful results [5]. To expand on this research, we have built a software process that not only considers shape and aperture features, but also adds multiple texture features. The range of tested image types has also been greatly expanded in order to build a model capable of classifying a highly variable dataset.

2 Overview

The steps for our process are as follows: 1. Image acquisition and particle segmentation, 2. Feature extraction, and 3. Classification.

Our process begins with scanning glass slides of the various pollen species with a digital microscope, then segmenting these images to gather samples of individual pollen particles. These images are then further segmented to identify the pollen boundary, and the area within this boundary is used for feature extraction. 18 shape features, texture features including the Fast Fourier Transform, Local Binary Patterns, Histogram of Oriented Gradients, and Haralick features, as well as aperture features are used. These features are then trained using supervised learning to build a model for the 5 pollen species sampled. The model is then tested with ten-fold cross validation. The process is illustrated in figure 1.

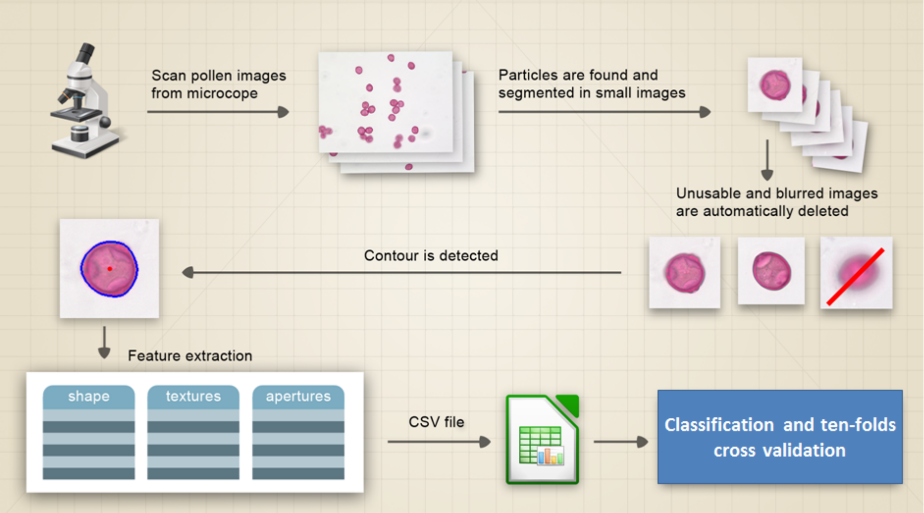

Scan pollen images from microcope
Particles are found and segmented in small images
Unusable and blurred images are automatically deleted
Contour is detected
Feature extraction
shape
textures
apertures
CSV file
Classification and ten-folds cross validation

Fig. 1. Pollen image acquisition and classification process

3 Image Acquisition and Particle Segmentation

Five different species (Alder, Birch, Hazel, Mugwort, and Sweet Grass) have been stained and prepared on glass slides for use with a common digital bright-field microscope. In order to build a robust model, all species had sample images derived from three distinct laboratory slides (using a total of 600 sample images obtained from 15 different slides).

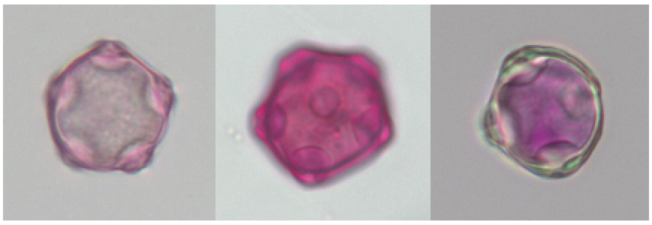


Fig. 2. Diverse image types, all example data used for training the Alder class of pollen. Access to the complete dataset can be found at http://dx.doi.org/10.5072/dans-zpr-rjm6.

For particle segmentation, each digital image is processed in order to locate and segment out a confining square surrounding a pollen particle. First, a median blur and Gauassian blur are applied to a negative of the image in order to remove smaller particles that are background noise (often dirt or imperfections on the background). Next, a threshold is applied to the image, using the OTSU algorithm to automatically detect the histogram peak. The returned image is an optimized binary image. A second set of filters is then applied using morphological operators (iterations of erosions and dilations) to fill in the particle area. Finally, the image is converted to have a white background in preparation for further processing steps.

A blob detection algorithm is now applied in order to extract a small image surrounding each particle. This algorithm is based on four attributes – Area, Circularity, Convexity and Inertia Ratio, with parameters for “minimum” and “maximum” values for each. By setting the parameters for the expected characteristics of pollen grains, the smaller images are then found and extracted.

The last filter used on the resulting images of particles is depicted in Figure 3. Because the pollen grains settle into the slide adhesive at different depths, some particles will be out of focus. These blurry images will provide insufficient data especially concerning texture features, therefore we remove them from our analysis. A blur detection algorithm was developed and applied to each image: a Laplacian filter set to a manually determined threshold value determines which images are too blurry and removed from further processing steps.

Lastly, the contour surrounding each pollen particle is identified, using OpenCV’s `findContours()` method.

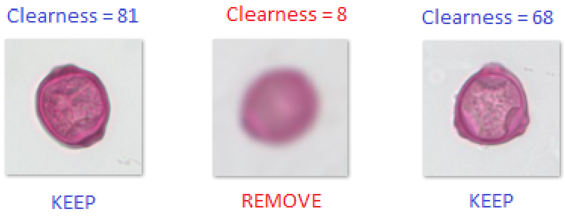


Fig. 3. Blur detection example

4 Feature Extraction

4.1 Shape features

We have used 18 shape features already identified to be useful through previous iterations of our research [5]. The 18 selected were based on the research of developing an identification process for the Urticaceae family of pollen [11], as well as research into developing universal shape descriptors [1].

Shape features used:

Perimeter (P) Length of contour given by OpenCV's `arcLength()` function
Area (A) Number of pixels contained inside the contour
Roundness (R) $\frac{4\pi A}{P^2}$
Compactness $\frac{1}{R}$
Roundness/Circularity Ratio (RC) Another measure of roundness, see [9] $\frac{P-\sqrt{P^2-4\pi A}}{P+\sqrt{P^2-4\pi A}}$
Mean Distance ($\bar{S}$) Average of the distance between the center of gravity and the contour
Minimum Distance (S_{min}) Smallest distance between the center of gravity and the contour
Maximum Distance (S_{max}) Longest distance between the center of gravity and the contour
Ratio1 (R_1) Ratio of maximum distance to minimal distance S_{max}/S_{min}
Ratio2 (R_2) Ratio of maximum distance to mean distance $S_{max}/\bar{S}$
Ratio3 (R_3) Ratio of minimum distance to mean distance $S_{min}/\bar{S}$
Diameter (D) Longest distance between any two points along the contour
Radius Dispersion (RD) Standard deviation of the distances between the center of gravity and the contour
Holes (H) Sum of differences between the Maximum Distance and the distance between center of gravity and the contour
Euclidean Norm (EN_2) Second Euclidean Norm
RMS Mean RMS mean size
Mean Distance to Boundary Average distance between every point within the area and the contour

Complexity (F) Shape complexity measure based on the ratio of the area and the mean distance to boundary

4.2 Texture feature extraction

A variety of texture features were selected due to their performance in prior research [11,10,7,8]. The texture features extracted included: Gabor Filters (GF), the Fast Fourier Transform (FFT), the Local Binary Pattern (LBP), the Histogram of Oriented Gradients (HOG), and Haralick features.

Gabor Filters Gabor filters have been proven useful in image segmentation and texture analysis [12]. The Gabor Filter function consists of the application of 5 different size masks and 8 orientation masks (See Figure 4) in order to produce output images. For each of the 40 resulting images, we calculate the local energy over the entire image (the sum of the square of the gray-level pixel intensity), and the mean amplitude (the sum of the amplitudes divided by the total number of images). In addition to these 80 values, we also store the total local energy for each of the 8 directions as well as the direction where the local energy is at the maximum.

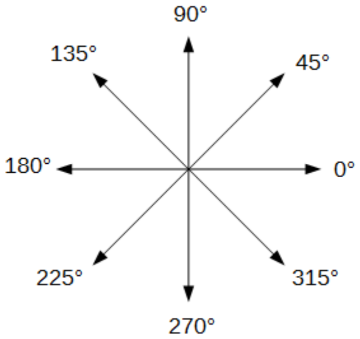


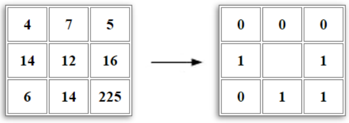
Fig. 4. The 8 directions of the mask for the Gabor Filters

Fourier Transform Fourier Transforms translate an image from the spatial domain into the frequency domain, and are useful because lower frequencies represent an area of an image with consistent intensity (relatively featureless areas) and higher frequencies represent areas of change [2]. Just as in spatial analysis, we cannot compare images directly, but first need to extract features. In the frequency domain, we likewise extract useful information through analysis of frequency peaks. Here, we apply a Fast Fourier Transform to the image, apply a logarithmic transformation, and create a graph of the resulting frequency domain. After taking the highest 10 frequency peaks, we compute the differences between the peaks and store these values, as well as the mean of the differences and the variance of the differences.

Haralick Features Haralick features [3] are determined by computations over the GLCM (Grey-Level Co-Occurence Matrix). Here, we use: the *angular second moment*, *contrast*, *correlation*, *sum of squares: variance*, *inverse difference moment*, *sum average*, *sum variance*, *sum entropy*, *entropy*, *difference variance*, *difference entropy*, *measure of correlation 1*, and *measure of correlation 2*. These are 13 out of the 14 original features developed by Haralick: the 14th is typically left out of computations due to uncertainty in the metric's stability.

Histogram oriented gradient (HOG) The Histogram of Oriented Gradients is calculated by first determining gradient values over a 3 by 3 Sobel mask. Next, bins are created for the creation of cell histograms; here, 10 bins were used. The gradient angles are divided into these bins, and the gradient magnitudes of the pixel values are used to determine orientation. After normalization, the values are flattened into one feature vector.

Local Binary Pattern (LBP) To obtain local binary patterns, a 3 by 3 pixel window is moved over the image, and the value of the central pixel is compared to the value of its neighbors. In the case that the neighbor is of lower value, it is assigned a zero, and in the case of a higher value, a one. This string of eight numbers ("00011101" for instance) is the determined local pattern. The frequency of the occurrence of each pattern is used as the texture description.



4	7	5
14	12	16
6	14	225

→

0	0	0
1		1
0	1	1

Fig. 5. Example output for Local Binary Patterns

Aperture Detection The number and type of apertures present on the pollen surface is a typical feature used by palynologists in order to determine the pollen species. Therefore, it seems useful to also build an automatic aperture detection function in order to identify and count apertures as an addition feature set. Preliminary work identifying apertures [4] has shown potential for this analysis. First, a moving window segments the pollen image into smaller areas. Each

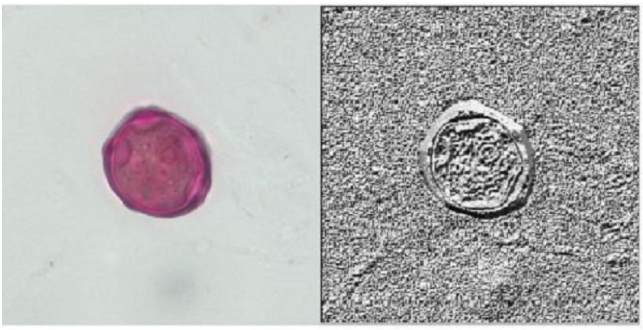


Fig. 6. Local Binary Pattern function applied to pollen image

smaller image is manually labeled as an aperture or not an aperture. Texture features are extracted from these smaller images, including those through a Fast Fourier Transform (FFT), Gabor Filters (GF), Local Binary Pattern (LBP), Histogram of Oriented Gradients (HOG), and Haralick features. A supervised learning process (through the use of support vector machines) then creates a model for each of the four species expected to include apertures on the surface. Once an unlabeled pollen image is given to be classified, the system again uses a moving window to break up the image into subsections. These smaller sections are then loaded into the generated model, and four values are returned for each detected aperture, corresponding to the probability that the aperture is of type Alder, Birch, Hazel, and Mugwort.

5 Classification

Once the shape, texture, and aperture features have been calculated, they are added together into a csv file. A data set of 5 species with 40 sample pollen images from 3 separate sample slides led to a total of 600 samples, each with 252 extracted features. A supervised learning process used this data for model creation, which was then tested using ten-fold cross validation. Both support vector machines and a random forest classifier showed promising (and very similar results); for the results reported here a random forest classifier was used due to faster processing on the larger data sets. The n-estimators parameter for this method was set to a typical size of 100 (increasing this number did lead to slightly improved results yet also dramatically increased processing times).

6 Results

Using a random forest classifier on a total of 600 samples (120 each for each species) and 252 features, a model was generated with an accuracy of 87% ± 2%.

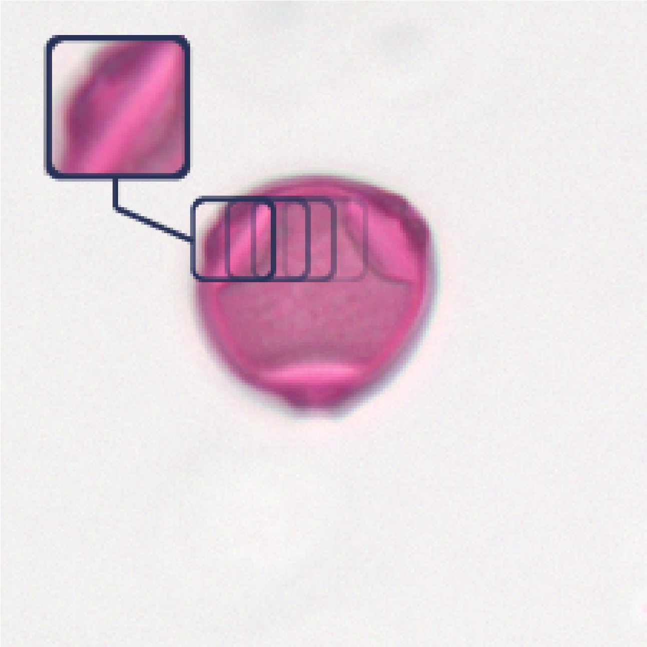


Fig. 7. Window moving all over the pollen

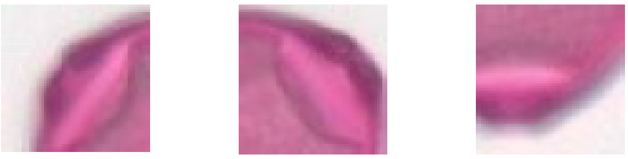


Fig. 8. Apertures detected by the program on a Birch pollen

Considering that the samples were intentionally selected for variability in their appearance and background (See Figure 2), this is an indication of a robust, reliable model that shows promise for expansion in the future to also include datasets collected from an outdoor environment.

The dataset was further modified into different versions in order to test the results using only subsets of the features available.

The above table shows the accuracies of the trained models. Using only the 18 shape features, an accuracy of 64% ± 3% was achieved, and adding texture information either through Gabor Filters or Haralick features substantially improved the results.

Features	Accuracy
Shape features	64% ± 3%
Shape and Gabor	76% ± 2%
Shape and FFT	65% ± 2%
Shape and LBP	65% ± 3%
Shape and HOG	67% ± 2%
Shape and Haralick	87% ± 3%
Shape and Aperture	67% ± 2%

7 Conclusion

Through this research, we have tested an expanded sample set of 5 species of pollen particles and used shape, texture and aperture features for use in classification. Use of all features led to an accuracy of 87% ± 2%. Through testing of individual texture features in combination with shape features, it was found that using only the shape and Haralick features resulted in an accuracy of 87% ± 3%. Gabor Filters also proved to be a useful feature as seen through the improved accuracy compared to using just the shape features alone. Surprisingly, the other texture features as well as the aperture features did not result in significant accuracy gains. One next step of research would be to investigate under which exact conditions certain texture features prove useful. In the case of the aperture features, one known limitation is that the aperture types were trained on a more limited dataset. Because the aperture detection process technique developed did have positive results in determining correct aperture positions, it would be interesting to retrain the aperture type on a wider dataset and see if this results in a more useful set of extracted features. Furthermore, extending the dataset not only beyond 600 images but especially to include more than three microscope slides per species would test against possible overfitting to particular slide conditions. Future research would also include application of this process to data collected outside of a laboratory environment, as well as expansion to include more pollen species.

References

1. da Fontoura Costa, L., Cesar Jr., R.M.: Shape Classification and Analysis: Theory and Practice. CRC Press, Inc., Boca Raton, FL, USA, 2nd edn. (2009)
2. Haas, N.Q.: Automated Pollen Image Classification. Master's thesis, University of Tennessee (2011), http://trace.tennessee.edu/utk_gradthes/1113
3. Haralick, R., Shanmugam, K., Dinstein, I.: Textural features for image classification. Systems, Man and Cybernetics, IEEE Transactions on SMC-3(6), 610–621 (Nov 1973)
4. Lozano-Vega, G., Benezeth, Y., Marzani, F., Boochs, F.: Classification of pollen apertures using bag of words. In: Petrosino, A. (ed.) Image Analysis and Processing – ICIAP 2013, Lecture Notes in Computer Science, vol. 8156, pp. 712–721. Springer Berlin Heidelberg (2013), http://dx.doi.org/10.1007/978-3-642-41181-6_72

5. Lozano-Vega, G., Benezeth, Y., Marzani, F., Boochs, F.: Analysis of relevant features for pollen classification. In: Iliadis, L., Maglogiannis, I., Papadopoulos, H. (eds.) Artificial Intelligence Applications and Innovations, IFIP Advances in Information and Communication Technology, vol. 436, pp. 395–404. Springer Berlin Heidelberg (2014), http://dx.doi.org/10.1007/978-3-662-44654-6_39
6. Lozano Vega, G., Benezeth, Y., Uhler, M., Boochs, F., Marzani, F.: Sketch of an automatic image based pollen detection system. In: 32. Wissenschaftlich-Technische Jahrestagung der DGPF. vol. 21, pp. 202–209. Potsdam, Germany (Mar 2012), https://hal.archives-ouvertes.fr/hal-00824014
7. Maillard, P.: Comparing texture analysis methods through classification. Photogrammetric Engineering & Remote Sensing 69(4), 357–367 (2003), http://www.ingentaconnect.com/content/asprs/pers/2003/00000069/00000004/art00003
8. Marcos, J.V., Nava, R., Cristóbal, G., Redondo, R., Escalante-Ramírez, B., Bueno, G., Déniz, O., González-Porto, A., Pardo, C., Chung, F.e.a.: Automated pollen identification using microscopic imaging and texture analysis. Micron 68, 36–46 (2015)
9. O'Higgins, P.: Methodological issues in the description of forms. In: Lestrel, P.E. (ed.) Fourier Descriptors and their Applications in Biology, pp. 74–105. Cambridge University Press (1997), http://dx.doi.org/10.1017/CBO9780511529870.005, cambridge Books Online
10. Redondo, R., Bueno, G., Chung, F., Nava, R., Marcos, J.V., Cristóbal, G., Rodríguez, T., Gonzalez-Porto, A., Pardo, C., Déniz, O., Escalante-Ramírez, B.: Pollen segmentation and feature evaluation for automatic classification in bright-field microscopy. Computers and Electronics in Agriculture 110(0), 56 – 69 (2015), http://www.sciencedirect.com/science/article/pii/S0168169914002348
11. Rodriguez-Damian, M., Cernadas, E., Formella, A., Fernandez-Delgado, M., Sa-Otero, P.D.: Automatic detection and classification of grains of pollen based on shape and texture. IEEE Trans. Syst., Man, Cybern. C 36(4), 531–542 (jul 2006), http://dx.doi.org/10.1109/TSMCC.2005.855426
12. Zheng, D., Zhao, Y., Wang, J.: Features extraction using a gabor filter family. In: Hamza, M.H. (ed.) Proceedings of the 6th IASTED International Conference. pp. 139–144. Signal and Image Processing, Acta Press (2004), www.paper.edu.cn/scholar/downpaper/wangjiaxin-13

Automatic Identification of Multipage News: A Machine Learning Approach

Pashutan Modaresi

Heinrich-Heine-University of Düsseldorf
Institute of Computer Science, Düsseldorf, Germany
modaresi@cs.uni-duesseldorf.de

Online news contain valuable information that can be utilized for private or commercial purposes. In the commercial context, online media monitoring services provide other companies or individuals with their required information in a systematic manner. This is accomplished by crawling plenty of news websites. Numerous news websites follow the strategy of pagination to split the stories into multiple pages. Given that, to identify multipage stories, manual rules have to be defined. On the other hand, the dynamic nature of the HTML pages requires a tremendous amount of effort in maintaining these rules. With this in mind, in this work we propose an automatic approach to identify multipage news stories.

We collected a list of web-pages in which the news were splitted in multiple pages and manually annotated them. To each link on the page a label has been assigned. That is, a link either points to the next page of the news or not. As the number of links which do not point to the next pages significantly dominates the number of link pointing to the next page of a news, the data set is highly imbalanced. Moreover, in order to design a language independent algorithm, news pages originating from different countries have been considered.

For each link, the *class* and *id* attributes of the corresponding anchor element, together with the text content of the anchor have been concatenated and fed into a Naive Bayes classifier. The same set of features extracted from the parent elements of an underlying link has been fed into another Naive Bayes classifier. Moreover, the relative position of a link on the news page (calculated by means of a heuristic) has been used to train a regression model. Additionally, some other features such as the structure of the *href* attribute of an anchor or the length of its text content have been integrated. Intentionally, the similarity between the content of the base page and the one of the target page has be ignored, as the calculation of this feature requires network availability that is not always given.

By cause of various learning algorithms being used, the final binary decision has to be performed by combining the results of the single constructed models. For this we use a *stacking* technique where we train a learning algorithm to combine the predictions of the constructed models.

Our first experimental results have revealed very high precision and recall values (≥ 0.9) for both labels under analysis.

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

Big Data Science Architecture for Continuous Technology Transfer from Research to Industry Operations

Richard A. E. Leibrandt

WidasConcepts Unternehmensberatung GmbH,
Maybachstrae 2, 71299 Wimsheim, Deutschland
richard.leibrandt@widas.de
http://www.widas.de

Abstract. Big Data without analysis is hardly anything but dead weight. But how to analyse it? Finding algorithms to do so is one of the Data Scientist's jobs. However, we would like to not only explore our data, but also automatise the process by building systems that analyse our data for us. A solution should enable research, meet industry demands and enable continuous delivery of technology transfer.

For this we need a Big Data Science Architecture. Why? Because in Big Data Science (BDS) projects, Big Data (BD) and Data Science (DS) – influencing each other – can't be handled separately. Thus, their complexities (and gain) multiply: $BDS \neq BD + DS$, $BDS = BD \cdot DS$.

This complexity boost increases further by the clash of the two different worlds of scientific research programming (DS) and enterprise software engineering (BD). The former thrives on explorative experiments which are often messy, ad hoc and uncertain in their findings. The later requires code quality and fail-safe operation, achieved by well defined processes with access control and automated testing and deployment.

We present a blue print for a Big Data Science Architecture. It includes data cleaning, feature derivation and machine learning, using Batch and Real-time engines. It spans the entire lifecycle with three environments: Experiments, close-to-life-tests, life-operations, enabling creativity while ensuring fail-safe operation. It takes the needs of data scientist, software engineers and operation administrators into account.

Data can be creatively explored in the experimental environment. Thanks to strict read governance no critical systems are endangered. After algorithms are developed, a technology transfer to the test environment takes place, which is build the same as the life-operations environment. There the algorithm is adapted to run in automated operations and tested thoroughly. On acceptance the algorithms are deployed to life-operations.

Keywords: Big Data, Data Science, Architecture, Industrial Challenges, Technology Transfer, Continuous Delivery, Batch- and Real-Time-Processing

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

IbmdbPy: Accelerating Python Analytics by In-Database Processing

Edouard Fouché and Michael Wurst

IBM Deutschland Research & Development GmbH

Abstract. The Python programming language is becoming widely used in data science and machine learning. Thus, Python ecosystem is very rich and provides intuitive tools for data analysis. However, most Python libraries require the data to be extracted from the database to working memory and ressources are limited by computational power and memory. Analyzing a large amount of data is often impractical or even impossible. IbmdbPy is an open-source python package, developed by IBM, which provides a Python interface for data manipulation and machine learning algorithms such as Kmeans or Linear Regression to make working with databases more efficient by seamlessly pushing operations written in Python into the underlying database for execution. This does not only lift the memory limit of Python, but also allows users to profit from performance-enhancing features of the underlying database management system. IbmdbPy is designed for IBM dashDB, a database system available on IBM BlueMix, the IBM cloud application development and analytics platform. Via remote connection, user operations can benefit from dashDB specific features, such as columnar technology and parallel processing, without having to interact with the database explicitly. Some in-database functions additionally use lazy loading to load only parts of the data that are actually required to further increase efficiency. Keeping the data in the database also avoids security issues that are associated with extracting data and ensures that the data that is being analyzed is as current as possible. IbmdbPy can be used by Python developers with very little additional knowledge, since it imitates the well-known interface of Pandas library for data manipulation and Scikit-learn library for machine learning algorithms. The project is still at an early stage, but several experiments have already been conducted to measure the advantage of using IbmdbPy over the corresponding in-memory implementation. The results show that it provides a great runtime advantage for operations on medium to large dataset, i.e. on tables that have 1 million rows or more. The project aims to extend the BlueMix ecosystem, by providing a Python interface for dashDB, bridging the gap between the analytics platform and end-user environment, so that developers can benefit both from the expressivity of Python and from the speed-up provided by SQL execution in dashDB, which can be run on a cluster.

Deploying Machine Learning at Web Scale

Christoph Schmitz

1&1 Mail & Media Development & Technology GmbH

Presentation Abstract

1&1 uses machine learning on some of the largest German web portals with practical challenges which are underrepresented in the academic literature. In our presentation, we will discuss these and some of our solutions in practice.

Data. Data quality is a major concern when integrating data sources within the company. The hardest problems in our production environment are gradual degradations in data quality. The root causes are hard to find, often occuring several steps upstream in the data pipeline. Organizational constraints can impede the collection of good quality data. Data sets from questionnaires can be skewed and thus require considerable preprocessing to be usable.

Modeling. Machine learning tools are usually targeted at an exploratory, interactive work flow. Building and maintaining hundreds of models at the same time leads to other requirements, though. We treat models like code, using versioning, continuous integration, and deployment strategies from software development. Much of the training work flow is automated, allowing a small team of data scientists to maintain models for a large number of target groups.

Constraints. In our applications, constraints are important when assessing the quality of models. One major example is the joint distribution of target variables with the age and gender of customers. Thus, measuring and visualizing these additional constraints is part of our modeling work flow. We are also looking into including these constraints directly in the training of models itself.

Processing. To be able to efficiently score more than 300 models, we use custom planning logic to split data flows into a minimal number of MapReduce jobs. Again, the common machine learning tools are not made for this. We will discuss challenges and solutions embedding Weka into a Hadoop application, e. g., schema handling, missing values, and dealing with errors.

Keeping Up. Since big data technology evolves at a breakneck pace, we need to trade off missing the latest features or the newest frameworks against the considerable cost of updating dozens of machines. While vendors promise hassle-free rolling upgrades, in practice upgrades are much more involved and entail considerable risk, effort, and organizational overhead.

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

Random tournaments: who plays with whom and how many times?

Adil Paul, Róbert Busa-Fekete, and and Eyke Hüllermeier

Department of Computer Science, University of Paderborn, Warburger Str. 100, 33098
Paderborn, Germany
{adil.paul,busarobi,eyke}@upb.de

Abstract. The weighted feedback arc set problem on tournaments (WFAS-T) is defined by a weighted tournament graph whose nodes represents the teams/items, and the goal is to find a ranking over the teams that minimizes the sum of the weights of the feedback arcs. We consider the probabilistic version of WFAS-T, in which the weights of the directed edges between every pair of teams sum up to one. A WFAS-T with this probabilistic constraint naturally determines a distribution over the tournament graphs. In this study, we investigate an online learning problem where the learner can observe tournament graphs drawn from this distribution. The goal of the learner is to approximate the solution ranking of the underlying probabilistic WFAS-T problem. We also investigate the partial information case, known from the multi-armed bandit problem, where the learner is allowed to select single edges and observe the corresponding value. Since the probabilistic WFAS-T problem is in general NP-hard [1], our learning algorithm relies on some recent approximation results for WFAS-T [2, 3]. We also consider some interesting special cases where the learner is able to estimate the exact solution, for example where the weights of the tournament graph satisfy the Bradley-Terry assumption [4].

Keywords: weighted feedback arc set problem, tournaments, online learning, bandit feedback

References

1. Noga Alon. Ranking tournaments. *SIAM J. Discrete Math.*, 20(1):137–142, 2006.
2. Don Coppersmith, Lisa K. Fleischer, and Atri Rurda. Ordering by weighted number of wins gives a good ranking for weighted tournaments. *ACM Trans. Algorithms*, 6(3):55:1–55:13, 2010.
3. Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: Ranking and clustering. In *Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing*, pages 684–693, 2005.
4. Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

Improving Multi-label Classification by Means of Cross-Ontology Association Rules

Fernando Benites and Elena Sapozhnikova

Department of Computer and Information Science,
{Fernando.Benites,Elena.Sapozhnikova}@uni-konstanz.de

Abstract. Recently several methods were proposed for the improvement of multi-label classification performance by using constraints on labels. Such constraints are based on dependencies between classes often present in multi-label data and can be mined as association rules from training data. The rules are then applied in a post-processing step to correct the classifier predictions. Due to properties of association rule mining these improvement methods often achieve low improvement expressed mostly in the better prediction of large classes. In the presence of class ontologies this is undesirable because larger classes correspond to higher levels in hierarchies presenting general concepts and can thus be trivial. In this paper we overcome the problem by focusing on improving multi-label classification performance on small classes. We present a new method of improvement based on mining cross-ontology association rules which is best suited for classification with multiple class ontologies, but can also be applied to multi-label classification with one class taxonomy.

1 Introduction

The increasing popularity of ontologies in different areas has led to the availability of data that can be annotated with multiple classes coming from different class taxonomies. This is a special case of multi-label classification. Generally, combining information from the ontologies providing different insights into a domain can be helpful in discovering new cross-ontology associations not evident from only one ontology. For example, if a film is classified by its genre in a genre ontology and by the producing company in an ontology of producers, one can find a possible interesting relation between a certain genre and a producing company, specialized in this genre. Recently, data mining techniques such as association analysis were applied to finding valuable cross-ontology Association Rules (ARs) between multiple ontologies corresponding to distinct categorizations of genes in bioinformatics [2, 9].

On the other hand, useful information from such cross-ontology rules can be successfully employed to improve performance in multi-label classification: ARs

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

found among classes of multiple ontologies can be used to correct predicted labels because the presence of a certain class or classes can be helpful for predicting another one. For example, a proper application of the association between a certain genre and a producing company specializing in this genre, as discussed above, can increase the probability of correctly predicting a genre, providing the corresponding company has been already correctly predicted. Thus it should lead to an improvement in classification performance. This is especially important with respect to very large ontologies with many thousands of classes, which are usually difficult to deal with.

Another problem with class ontologies that has not yet been dealt with sufficiently in recent research is that classifier performance in such a case is largely dominated by more general classes higher in the hierarchy because they are more present in the data and hence simpler to predict for a classifier. On the other hand, such general classes do not often provide interesting information and are sometimes trivial. For this reason, in mining cross-ontology rules rare association rules [14] are preferred, especially in large ontologies [2]. Similarly, in the improvement of multi-label classification performance, prediction improvement is more interesting for small and more specific classes in comparison to larger ones. As existing methods have not yet addressed this problem, our paper will focus on mining rare cross-ontology ARs and applying them to the multi-label classification improvement on small classes. For this purpose, a special interestingness measure well-suited for mining rare rules is utilized.

An important difference between our approach and several state-of-the-art methods discussed in the next section is that they use constraints for labels of one labelset and not two different class ontologies. Further, our approach focuses on rare labels, i.e. the ones with low support, since they are normally the greater part of the labelset.

The rest of the paper is organized as follows. A brief overview of the approaches to multi-label improvement with ARs is given in Section 2. Afterwards, our approach is explained in Section 3, followed by the experiments of Section 4. In Section 5 we conclude the paper.

2 Related Work

Recently several approaches to improving multi-label classification performance were proposed that dealt with dependencies between classes present in multi-label data. Some of them belong to the field of multi-label classification with constraints and apply constraints on labels to performance improvement usually in a post-processing step. The constraints are often mined in form of ARs from training data [8, 10].

The initial work was devoted to prediction corrections within the ranking by pairwise comparison framework [10]. The constraints were in the form of many-to-one ARs *labelset→label*, i.e. implications from a labelset to a single label. They could be positive or negative which involves either setting or removing a consequent label as a result of the presence of an antecedent label combination. The constraint rules were extracted using a standard support-confidence AR mining framework in order to change the predicted rankings. The results of the method obtained on real-world datasets were negative: no improvement in comparison to the baseline performance was observed. For this reason, this method will not be used for comparison with the proposed method below.

A more recent approach of [8] used only one-to-one ARs $label_i \rightarrow label_j$ in order to improve SVM performance in the Binary Relevance (BR) setting. The rules to apply were chosen by minimizing the Ranking Loss performance measure through a cross-validation process on the training data. The selected rules were then applied to predicted label rankings in the test phase, if an antecedent label was set, boosting the score of the corresponding consequent label. The improved results were obtained for two real-world datasets Yeast and Reuters. AR mining was based on the standard support-confidence framework. A subsequently extended approach [6] differs in that it uses subsets of labels gathered by clustering, and also extracts negative and many-to-one ARs. Still the evaluation of the extended approach was restricted to smaller datasets than in the earlier paper (e.g. the Reuters dataset was not included), perhaps pointing to a higher complexity of the algorithm, making it probably inapt to be applied to large datasets. Taking this into account, we selected only the initial method of [8] for comparison and will refer to it as Label Constraints for SVMs (LCS).

In contrast to the discussed post-processing methods evaluated in a certain multi-label classification setting (either pairwise or BR), a more general approach, Label Reduction with Association Rules (LRwAR), was proposed in [5]. It includes pre- and post-processing for the reduction of the label dimensionality. First, ARs are extracted and those labels that are only in the consequents of rules are removed from the data to be learned. Then a multi-label classifier is applied to the classification problem with a reduced labelset. After classification, the rules are applied to recover missing labels. An advantage of this approach is a shorter time needed to train a classifier on a smaller labelset. It also used the standard support-confidence framework to mine ARs, although its recent extension [4] proposed Conviction instead of Confidence. However it was not shown to provide significantly better results. The base method was evaluated on different multi-label classifiers including ML-kNN, BP-MLL and C4.5 (the latter in BR and label powerset settings) as well as six datasets. On several of them it showed either minimal (e.g. 0.6% relative improvement on the Yeast data) or no improvement at all. The performance of the extended method measured in terms of two performance measures was lower on the Yeast data in comparison to the baseline classifiers and it was generally inferior or equal to them in more than half of all experiments (79 from 140).

3 Improvement with Rare Association Rules

Besides relatively low improvement demonstrated by the existing methods, they have the problem of using the standard support-confidence framework for AR mining, which normally extracts high support rules that often exist between large classes. So they ignore small classes as a potential source for improvement because minimum support filtering can remove not only noise but also rare classes. The greatest problem of Confidence in such a setup is that associations of small classes to large ones are normally ranked very high. In the case of a class ontology these ARs simply show hierarchical parent-child relations, i.e. that one label a is more specific than another b. The rule extracted would be $a \rightarrow b$, which means that if label a appears, then label b should appear too. Such obvious relations can be derived from an extracted hierarchy, on the one hand, as in [3] and are misleading for classification improvement, on the other. The reason being that applying such rules in the case of LRwAR [5] for removing classes with a high support and setting them based on predictions of classes with a lower support, is prone to error. An example would be if class A appeared only 10 times, class B appeared 100 times and both appeared together 10 times, so a rule A→B would be extracted. LRwAR would then imply that B should be removed from the labelset and only in the post-processing step reinserted based on the prediction of A. Although in [4] Conviction was used instead of Confidence, it is still closely related to Confidence and behaves similarly.

Another problem with the standard AR framework is choosing the thresholds for Support and Confidence which is done manually and can therefore be suboptimal. So, the first issue to be dealt with improving multi-label classification performance by constraints is the acquisition of high-quality rules. In the proposed approach we solve this problem by omitting the minimum support threshold and using a special interestingness measure which is well-suited for rare rules. Additionally we tune its threshold automatically depending on the range of values for extracted rules. The idea is to use rare ARs between classes that is from a small class to another small one and that classes belong to two or more ontologies describing different aspects of a dataset. In this case hierarchical relations between the classes of one ontology will not be taken into account. In such a setting, training data are annotated with categories of both ontologies. Then mined rules are used to improve class predictions for one of both ontologies. So, rare cross-ontology rules can be helpful in order to solve the described problems.

The second important issue is deciding when to apply a rule. Is the predicted label trusty enough to insert an additional label based on its presence? The worst case scenario would be that the rule is applied on the basis of a false positive inserting an additional false positive. Another undesirable outcome would be that the antecedent label is a true positive but applying a rule would create a false positive, i.e. the prediction of the classifier should not be overruled. The desirable decision is only to use a true positive to add another true positive, i.e. that a rule corrects the missclassification of a label. In [8] the rules are applied to all labels in the rules, assuming that all antecedents were reliably predicted. Although the rules were used before to optimize the Ranking Loss, it was not clear if the antecedent's ranking was high enough to be predicted. In order to solve this problem, the control of the quality of classifier predictions is proposed to create a basis for application of a rule. The detailed discussion of the proposed approach is presented in the next two subsections.

3.1 Selection of Rules

To extract pairwise ARs, we performed experiments with the interestingness measures well-suited for mining rare rules [14]. Such measures should possess the important property of null-transaction invariance [1]. Due to the lack of space we will focus here only on the Kulczynski measure (*Kulc*) which showed good results:

$$Kulc(A,B) = \frac{P_{AB}}{2} * (\frac{1}{P_A} + \frac{1}{P_B}).$$

In order to select only the best rules, adaptive thresholding without a predefined value was applied as follows: After calculating *Kulc* of all rules, the values are sorted in descending order as a curve C. We assume that there will be a slope between a few high scored rules and the rest. In order to select these rules the curve C is smoothed into S and only the part with a relatively low variance is analyzed. Thus we need first to determine whether the variance of the curve S is high:

$$CV = \frac{MEAN(S) - VAR(S)}{MEAN(S)} > \rho_m \tag{1}$$

where $MEAN(S)$ is the average value of the curve S and $VAR(S)$ its variance. If the condition of Eq. (1) is true, we use only the values in the slope of the curve and calculate the median that defines a Threshold Value TV for the most interesting rules:

$$TV = C_{MEDIAN(\{i|DIFF(S_i) > MEAN(DIFF(S))\})}$$

where $DIFF(S)$ is the difference between two neighbor values in the curve S. Since the step size between two values is 1, $DIFF(S)$ can also be seen as the derivative of S. Otherwise, i.e. if the variance is high, the average of the values not much lower than the mean of the entire curve is taken:

$$TV = \underset{j \in \{i|S_i > MEAN(S) * \rho_t\}}{MEAN}(S_j) \quad .$$

Defining the threshold in this manner, we select only those rules that have *Kulc* values above TV as good enough to be applied to prediction improvement.

ρ_m and ρ_t should be set so that the changes are significant and the only high valued rules are selected, respectively.

3.2 Application of Rules

As discussed above, applying a rule for insertion of a label without taking the corresponding classifier's judgment into account can lead to no improvement or even poorer prediction performance. A better way would be to use rankings produced by the classifier. An attempt was proposed in [8] where the scores provided by the classifier for each label were used to optimize a parameter w varied from 0 to 1 for each pair of labels i and j. For a rule $i \rightarrow j$, new rankings of label j were calculated for each sample x as: $p_j(x) = w * p_j(x) + (1 - w) * p_i(x)$, where $p_i(x)$ is the score assigned to a label i, analogously for j. by its respective BR classifier for that sample. These new rankings were used to minimize Ranking Loss by varying w during cross-validation on a validation set. However the label i was chosen in the test phase, only if its score was above a threshold t used to turn predicted rankings into classes (also called decision boundary later on). Thus the rule $i \rightarrow j$ could not be applied otherwise.

A drawback of this method is that it relies on individual parameter optimization for each rule through a cost-intensive calculation of the Ranking Loss. This is not viable for large datasets as the later work [6] shows by using a fixed parameter value.

We propose a similar approach. The antecedent A of a rule $A \rightarrow B$ should be already positively predicted, i.e. should have a score greater than the threshold t, but an additional criterion should hold: $\frac{V_B}{V_A} > 0.5$, i.e. the score of the consequent should be at least 50% of the value of the antecedent in order to set the consequent.

As emphasized before, our approach was designed to work on classification problems with two different multi-label sets coming from two ontologies, but it can still be applied to the problems with only one class taxonomy in order to compare to other methods, as is shown in the next section.

4 Experiments

4.1 Data

We used two multi-label real-world datasets: Reuters and Yeast. The first one was used with two class ontologies "Topics" and "Industries" for mining cross-ontology ARs as well as in a simplified version with only "Topics" labels in order to compare our method to the results of other improvement methods published elsewhere.

The two-ontology Reuters dataset was formed by preprocessing with stopword removal and stemming the original data provided by `http://trec.nist.gov/data/reuters/reuters.html`. We used the 5000 most frequent terms in the training set and applied tf-idf weighting as well as column-wise normalization performed separately on training and test data. In the original 800k samples only 300k contained at least one "Industries" label. From these we selected random 30k samples and split them into a training and a test set with the ratio of 2:1, i.e. 20k training samples and 10k test samples. In total there were 103 "Topics" labels and 364 "Industries" labels. We will denote this dataset as Reuters 10k below.

The simplified version (Reuters 5k with only "Topics" classes) consisted of 5000 training and 5000 test samples chosen randomly from the original 23k training set. The data preprocessing was performed as described above.

For the sake of comparison, the Yeast dataset was also taken from the MEKA package [11]. It contains only 14 labels in one non-hierarchical label set and is therefore not very interesting for our experiments, but it is often used in the works on multi-label classification. From its 2417 samples, 1500 were selected randomly for training and the rest for test. We did not use cross-validation since certain aspects would be more difficult to analyze, for example, the graphs.

4.2 Experiment Setting

As a baseline classifier we used LIBLINEAR [7] in the BR setting, and on single class ontology datasets also ML-kNN as well as Classifier Chains (CC) based on LIBLINEAR. A crucial complication with LIBLINEAR is that the choice of the threshold t can be difficult. Normally, the value of 0.5 is recommended but in the work of [8] a different value and individual for each dataset (0.45 for Yeast and 0.47 for Reuters) was chosen. We also used different values for each dataset and additionally compared the results to the results of an adaptive method for selecting t, which automatically adjusts it to be close to the dataset label cardinality [13]. We will refer here to this method as Label Cardinality Approach (LCA). ML-kNN and CC were not used with two class ontologies because they do not scale well on large datasets, specially with LCS.

For LRwAR we also implemented a variation using rankings for all classes and only inserting new labels. We use the acronym OF (only fill) for this variation. The original method foresees deleting rankings of classes in the consequent of a rule as well.

Parameters of LRwAR and LCS were set as in their original works [5, 8], whereas the Confidence threshold was used to obtain the best results for both methods. In particular, we changed it for Reuters 5k so that the h-loss performance measure was comparable to the value of the baseline classifier.

Parameters for adaptive thresholding were set to $\rho_m = 0.2$ and $\rho_t = 0.75$. These values were obtained by the manual experimentation on the Reuters dataset and are, in our opinion, general enough to be used for all datasets.

4.3 Performance measures

We used the F-1 measure, which is the harmonic mean of Recall and Precision. It can be calculated in several ways depending on averaging [15]. First, we used instance-based averaging, i.e. we calculated F-1 for every single instance and then took the mean value (denoted as IF1). Additionally we used label-based F-1 both in micro-averaged version mF1 and in macro-averaged one LF1= $\frac{1}{Q}\sum_{i=1\ldots Q}\frac{2*tp_i}{2*tp_i+fn_i+fp_i}$. Here Q is the number of labels and tp_i, fp_i and fn_i are, respectively, the number of true, false positives and false negatives for a label i. Micro-averaged mF1 is known to be dominated by the performance on large classes. Also Hamming Loss (h-loss) was used: $HL = \frac{fp+fn}{Q*N}$, where N is the number of test samples.

4.4 Results

Datasets with one class taxonomy: Yeast and Reuters 5k First we compared our approach to LRwAR,LCS, and LCA on the datasets used in other studies. In Table 1 the results for Yeast and Reuters 5k are depicted.

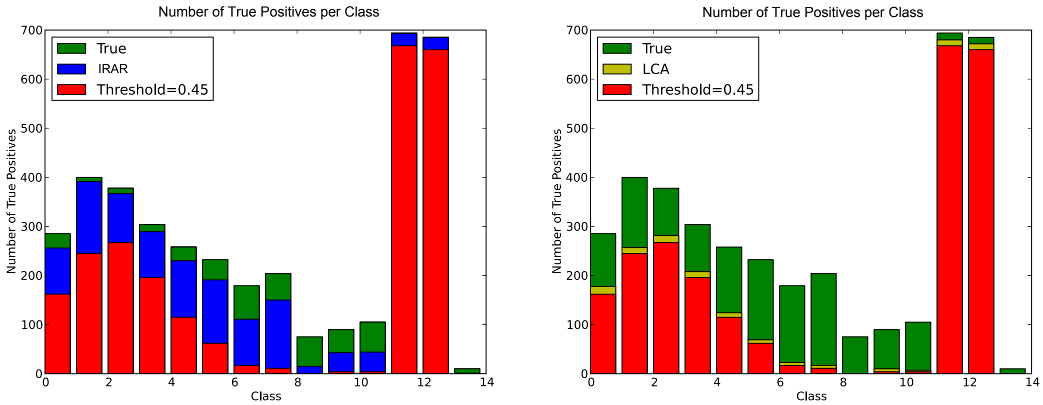
On the Yeast data, IRAR, LCS, and LRwAR OF could improve the results of BR and ML-kNN classifiers in terms of mF1, IF1, and LF1. The improvement achieved by IRAR was the highest. H-loss for this dataset could not be increased by any of the compared methods. Among them LRwAR was the worst because its results were even worse as those of the baseline classifiers in terms of all performance measures. In contrast to the other methods, IRAR was also better than LCA for BR and comparable to it for ML-kNN. This shows that a powerful thresholding strategy can outperform many improvement methods based on label constraints. IRAR was the only improvement methods that could increase the CC results. This was the highest LF1 value by far on this dataset.

This is consistent with the important fact that IRAR could increase the LF1 value significantly more than the other methods in almost all configurations. The only exception was for Reuters 5k and BR where it improved second best. This can be explained by the better improvement of the classification performance on small classes. Indeeed, as Figure 1a shows, the number of true positives on small and middle-size classes obtained by IRAR was higher than that of LCA (Figure 1b). This difference is even more pronounced if we compare F-1 values for each class obtained by all improvement methods and presented in Figure 2a. One can see, for example, that IRAR achieves a significant improvement in F-1 for the last class where the other methods show no improvement at all or that it has much more improvement on the classes 5-10.

Analyzing the curves of mF1 and LF1 in dependence on the threshold t one can see a trade-off between them (Figure 2b). IRAR is able to achieve both high LF1 and mF1 values near their crossing point.

On the Reuters 5k dataset, IRAR had again the highest improvement against the baseline classifier as compared to the other improvement methods in terms of all performance measures, except for h-loss. The largest performance difference was again in LF1. IRAR performance was comparable to that of LCA. LCS and LRwAR achieved a very small improvement against the baseline classifier and were worse than LCA in terms of all performance measures, except for h-loss. Here we can see that CC had was the second best classifier, but no improvement could beat LCA method. Again, the exception remains IRAR with LF1, having a 18% value increase over the baseline performance and 3% over LCA.

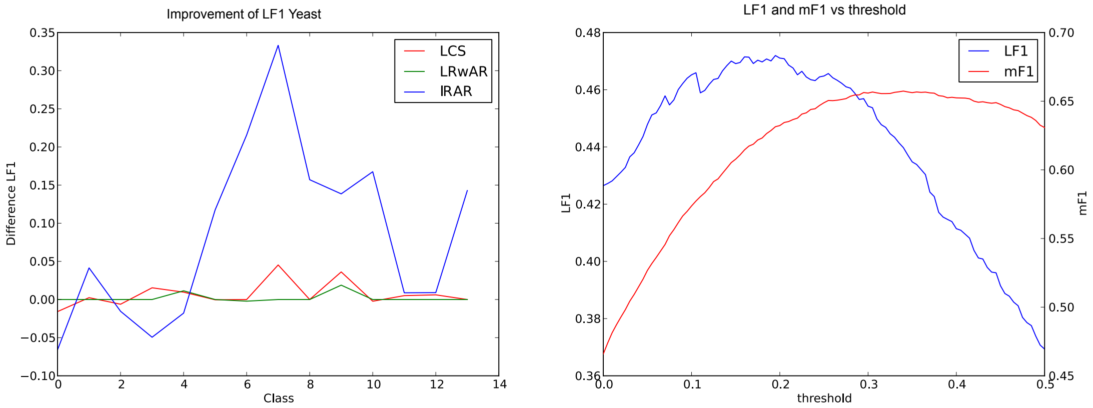
CC did not outperform BR in the experiments, although CC does consider the connections between the labels in a certain way. A solution would be to use Ensembles of CC (ECC) [12], since the order of the labels can be taken into account. However for ECC, the issue of larger label sets will be even much severe, since the label order must be permutated when creating a new CC to exhaust all alternatives at best.



(a) Yeast, IRAR, number of true positives

(b) Yeast, LCA, number of true positives

Fig. 1. Distributions of true positives on Yeast data.



(a) Yeast, improvement of LF1 per class for LCS, LRwAR, and IRAR

(b) Yeast, LF1 and mF1 calculated using the respective threshold on the rankings

Fig. 2. Improvement comparison on Yeast data.

Table 1. LCS, LRwAR, and IRAR applied to Yeast and Reuters 5k, OF=Only Filling, t = threshold, bold values mark the best values per dataset and column.

	BR				ML-kNN				CC			
Metrics	h-loss	mF1	IF1	LF1	h-loss	mF1	IF1	LF1	h-loss	mF1	IF1	LF1
Yeast												
LCA	0.2121	0.6498	0.6362	0.4088	0.2066	**0.6583**	**0.6467**	0.4042	0.2148	**0.6455**	**0.6326**	0.4053
	t=0.45				t=0.5				t=0.45			
baseline	**0.2043**	0.6477	0.6288	0.3915	**0.1990**	0.6221	0.5978	0.3496	**0.2097**	0.6370	0.6184	0.3854
LCS Cnf=0.6	0.2071	0.6516	0.6326	0.3983	0.1996	0.6263	0.6016	0.3596	0.2141	0.6313	0.6144	0.3540
LRwAR Cnf=0.6	0.2071	0.6328	0.6160	0.3479	0.2047	0.6034	0.5838	0.2982	0.2121	0.6223	0.6057	0.3401
LRwAR OF Cnf=0.6	0.2047	0.6480	0.6293	0.3935	**0.1990**	0.6221	0.5978	0.3496	0.2105	0.6367	0.6183	0.3857
IRAR	0.2269	**0.6544**	**0.6400**	**0.4453**	0.2169	0.6560	0.6446	**0.4101**	0.3553	0.6031	0.5992	**0.4760**
Reuters 5k												
LCA	0.0131	0.7893	**0.7955**	**0.4177**	**0.0164**	**0.7361**	**0.7446**	0.4306	0.0148	**0.7608**	**0.7716**	0.3910
	t=0.3											
baseline	**0.0122**	0.7849	0.7817	0.3690	**0.0164**	0.7339	0.7376	0.4303	0.0133	0.7567	0.7492	0.3302
LCS n=6,Cnf=0.8	**0.0122**	0.7856	0.7823	0.3696	0.0165	0.7335	0.7377	0.4311	0.0140	0.7382	0.7305	0.3246
LCS n=6, Cnf=.85	**0.0122**	0.7856	0.7823	0.3696	0.0165	0.7335	0.7377	0.4311	0.0140	0.7383	0.7307	0.3249
LRwAR Cnf=0.8	0.0138	0.7465	0.7442	0.3576	0.0177	0.7002	0.7015	0.4191	0.0148	0.7170	0.7119	0.3194
LRwAR Cnf=0.85	0.0130	0.7667	0.7633	0.3614	0.0169	0.7189	0.7204	0.4231	0.0140	0.7377	0.7302	0.3231
LRwAR OF Cnf=0.8	**0.0122**	0.7851	0.7819	0.3690	**0.0164**	0.7340	0.7378	0.4303	**0.0132**	0.7584	0.7508	0.3313
LRwAR OF Cnf=0.85	**0.0122**	0.7849	0.7817	0.3690	**0.0164**	0.7339	0.7376	0.4303	**0.0132**	0.7584	0.7508	0.3313
IRAR	0.0125	**0.7900**	0.7895	0.3958	0.0187	0.7174	0.7347	**0.4433**	0.0164	0.7452	0.7490	**0.4001**

Dataset with two class ontologies: Reuters 10k Table 2 depicts the results of classification improvement for the Reuters 10k dataset, first classified separately in "Topics" and "Industries" and then with improved "Industries" predictions, by using cross-ontological ARs. In general, the classification performance for "Topics" was higher than for "Industries" classes. The results of the improvement methods LCS and IRAR for this class ontology were better than those of the baseline classifier, except that h-loss of IRAR was lower. At the same time, LRwAR showed negative improvement and LRwAR OF only improvement at the fourth place after the decimal point. In contrast, IRAR was able to achieve the overall best LF1. Its results were also somewhat similar to the results of LCA. It is interesting to note that LCS outperformed LCA in terms of mF1 and IRAR in terms of LF1. So, we can conclude that LCS is more effective for classes with large support and IRAR for those with small support. This will be due to the use of confidence to extract the rules.

The results for Reuters 10k "Industries" are similar to those obtained for "Topics". Here LRwAR had even more negative improvement in terms of all performance measures and LRwAR OF showed again only marginal improvement. LCS was equal or better than the baseline and achieved again the highest mF1 value. This time both LCA and IRAR were worse than the baseline in terms of h-loss and mF1, but improved IF1 and LF1. However IRAR was better than LCA in three out of four performance measures and had again the best LF1.

Using cross-ontology ARs for the improvement of “Industries” predictions revealed an interesting fact: the results of LCS and both LRwAR variants worsened in comparison with those shown in the previous experiment while IRAR could improve its h-loss and mF1 values. Here, LCS uses the thresholds of different classifiers trained with different labelsets that may obstruct its performance. Also the low occurrence of labels in the labelsets may lead to poor results of the Confidence-based methods.

Table 2. LCS, LRwAR, and IRAR applied to Reuters BR’s predictions for “Topics” and “Industries” 10k, OF=Only Filling, t = threshold, bold values mark the best values per dataset and column.

Metrics	h-loss	mF1	IF1	LF1	h-loss	mF1	IF1	LF1
	Reuters 10k “Topics” t=0.45				Reuters 10k “Industries” t=0.3			
LCA	0.0123	0.8257	**0.8335**	0.4237	0.0070	0.6462	**0.6466**	0.2884
baseline	**0.0116**	0.8258	0.8282	0.3938	**0.0061**	0.6589	0.6060	0.2772
LCS k=5,n=6,Cnf=0.7	**0.0116**	**0.8264**	0.8287	0.3942	**0.0061**	**0.6605**	0.6077	0.2778
LCS k=5,n=6,Cnf=0.85	**0.0116**	**0.8264**	0.8287	0.3942	**0.0061**	0.6603	0.6076	0.2776
LRwAR Cnf=0.7	0.0134	0.7912	0.7890	0.3819	0.0071	0.5491	0.4649	0.2672
LRwAR Cnf=0.85	0.0122	0.8143	0.8145	0.3871	0.0067	0.5936	0.5138	0.2698
LRwAR OF Cnf=0.7	**0.0116**	0.8259	0.8284	0.3940	**0.0061**	0.6592	0.6068	0.2773
LRwAR OF Cnf=0.85	**0.0116**	0.8260	0.8285	0.3940	**0.0061**	0.6592	0.6067	0.2773
IRAR	0.0132	0.8187	0.8312	**0.4298**	0.0067	0.6539	0.6120	**0.2918**
Reuters 10k “Topics”→“Industries”, t=0.3								
LCS Cnf=0.7,	**0.0061**	0.6589	0.6060	0.2772				
LCS k=5,n=6,Cnf=0.85,	**0.0061**	0.6589	0.6060	0.2772				
LRwAR Cnf=0.7	0.0087	0.3482	0.2880	0.2293				
LRwAR OF Cnf=0.7	**0.0061**	0.6585	0.6056	0.2771				
IRAR	0.0062	**0.6590**	**0.6092**	**0.2825**				

5 Conclusion

In this paper we proposed a novel method of classification improvement in multi-label classification IRAR. It uses cross-ontology association rules and focuses on the improvement of predictions for small classes. Additionally, we compared it with state-of-the-art methods developed to correct predicted rankings by using constraints on labels in a post-processing step. One of the methods, LRwAR, showed negative improvement in most of the experiments and its variation only marginal improvement. LCS scored better in terms of improvement, but a better thresholding strategy such as LCA often achieved even more improvement. IRAR could outperform LCA in three out of four performance measures on the Yeast and Reuters 10k datasets. More importantly, it boosted the LF1 value significantly and showed the best LF1 result in three of five experiments. This means that IRAR is well suited for improving performance on small classes. This method is also able to achieve the trade-off between LF1 and mF1, i.e. it was able to achieve a high LF1 at a relatively low number of false positives. This points to the fact that the method can be used effectively with datasets exhibiting highly skewed label distributions as, for example, in the case of class ontologies.

References

1. Benites, F., Sapozhnikova, E.: Evaluation of hierarchical interestingness measures for mining pairwise generalized association rules. IEEE Trans. Knowl. Data Eng. 26(12), 3012–3025 (2014)
2. Benites, F., Simon, S., Sapozhnikova, E.: Mining rare associations between biological ontologies. PLoS ONE 9, e84475 (2014)
3. Brucker, F., Benites, F., Sapozhnikova, E.P.: Multi-label classification and extracting predicted class hierarchies. Pattern Recognition 44(3), 724–738 (2011)
4. Charte, F., Rivera, A., del Jesus, M., Herrera, F.: LI-MLC: A label inference methodology for addressing high dimensionality in the label space for multilabel classification. IEEE Trans. Neural Netw. Learn. Syst. 25(10), 1842–1854 (2014)
5. Charte, F., Rivera, A., del Jesus, M., Herrera, F.: Improving multi-label classifiers via label reduction with association rules. In: Hybrid Artificial Intelligent Systems, LNCS, vol. 7209, pp. 188–199 (2012)
6. Chen, B., Hong, X., Duan, L., Hu, J.: Improving multi-label classification performance by label constraints. In: IJCNN 2013. pp. 1–5 (Aug 2013)
7. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. JMLR 9, 1871–1874 (2008)
8. Gu, W., Chen, B., Hu, J.: Combining binary-svm and pairwise label constraints for multi-label classification. In: Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on. pp. 4176–4181 (2010)
9. Manda, P., McCarthy, F.M., Bridges, S.M.: Interestingness measures and strategies for mining multi-ontology multi-level association rules from gene ontology annotations for the discovery of new go relationships. J. of Biomedical Informatics 46(5), 849–856 (2013)
10. Park, S.H., Fürnkranz, J.: Multi-label classification with label constraints. In: ECML PKDD 2008 Workshop on Preference Learning. pp. 157–171 (2008)
11. Read, J., Reutemann., P.: Meka multi-label dataset repository, http://meka.sourceforge.net/, May 20 2015
12. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. In: European Conference on Machine Learning and Knowledge Discovery in Databases: Part II. pp. 254–269. ECML PKDD '09, Springer-Verlag, Berlin, Heidelberg (2009), http://dx.doi.org/10.1007/978-3-642-04174-7_17
13. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. Machine learning 85(3), 333–359 (2011)
14. Surana, A., Kiran, U., Reddy, P.K.: Selecting a right interestingness measure for rare association rules. In: 16th Int. Conf. on Management of Data (COMAD). pp. 115–124 (2010)
15. Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining multi-label data. In: In Data Mining and Knowledge Discovery Handbook. pp. 667–685 (2010)

Impact of Warping vs Smoothing for Time Series Similarity

Frank Höppner

Ostfalia University of Applied Sciences
Dept. of Computer Science, D-38302 Wolfenbüttel, Germany

Introduction. When dealing with time series, the application of a smoothing filter (to get rid of random fluctuations and better recognise the relevant structure) is usually one of the first steps. In the literature on time series similarity measures, however, the impact of smoothing is not explicitly or systematically considered – despite extensive experiments in, e.g., [2]. Instead, complex similarity measures are frequently applied (e.g. dynamic time warping (DTW)), which implicitly deal with noise, but mainly with temporal dilation and translation effects. So up to now it is unclear, to what extent the good performance of DTW is due to its smoothing or warping capabilities.

Optimal Filter. In this work we consider a simple Euclidean distance applied to preprocessed (smoothed) time series. It is unlikely that one similarity measure fits all problem types (or data sets), so by choosing an appropriate filter, we may adopt to the problem at hand. The filter is automatically determined given a training set of classified series, such that distances between series of the same (different) class are minimised (maximised). The obtained similarity measure is then tested in cross-validated 1-NN classification for various data sets (as in [2]) and compared against the DTW performance. Starting from Euclidean distance (without any preprocessing) as a baseline, it turns out that for many data sets a substantial fraction of the performance improvement obtained with DTW is also obtained by choosing the appropriate filter. In some cases, the performance is even better than with DTW, which is due to the fact that a filter is a versatile tool: for some problems it may be advantageous to distinguish time series by their derivative rather than the original series and in such cases a filter that estimates the derivative may be retrieved. For further details the reader is referred to [1].

References

1. F. Höppner. Optimal filtering for time series classification. In *Proc. 16th Int. Conf. Intelligent Data Engineering and Automated Learning*, 2015.
2. X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh. Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery*, 26(2):275–309, Feb. 2012.

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

In-stream Frequent Itemset Mining with Output Proportional Memory Footprint

Daniel Trabold[1], Mario Boley[2], Michael Mock[1], and Tamas Horváth[2,1]

[1]Fraunhofer IAIS, Schloss Birlinghoven, 53754 St. Augustin, Germany
[2]Dept. of Computer Science, University of Bonn, Germany
{daniel.trabold,michael.mock,tamas.horvat}@iais.fraunhofer.de
mario.boley@gmail.com

Abstract. We propose an online partial counting algorithm based on statistical inference that approximates itemset frequencies from data streams. The space complexity of our algorithm is proportional to the number of frequent itemsets in the stream at any time. Furthermore, the longer an itemset is frequent the closer is the approximation to its frequency, implying that the results become more precise as the stream evolves. We empirically compare our approach in terms of correctness and memory footprint to CARMA and Lossy Counting. Though our algorithm outperforms only CARMA in correctness, it requires much less space than both of these algorithms providing an alternative to Lossy Counting when the memory available is limited.

1 Introduction

Mining frequent itemsets from data streams with small memory footprint is an important data mining task with many applications such as, for example, credit card payment monitoring or phone call processing A stream setting requires algorithms that provide aggregated results on the current status of the stream in low latency at any time. Furthermore, the memory consumption must remain small, ideally proportional to the number of frequent itemsets in the input. A small overhead per transaction may add up to a huge amount for long streams.

The goal of itemset frequency estimation is to report the frequencies for all itemsets which occur more frequent than some threshold in a stream of transactions. A number of papers consider the simpler problem of counting single items or 1-itemsets from streams (see, e.g., [1–3]). All known solutions for the task of itemset frequency estimation fall in one of the following two categories: Algorithms without buffers and algorithms with some form of transaction buffering. The former include Carma [4] and hMiner [5].

Carma determines a superset of all frequent itemsets in a first pass and needs a second pass to compute the exact family of frequent itemsets. Requiring a

In: R. Bergmann, S. Gorg, G. Muller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

second pass is, however, unrealistic for large data streams. In contrast to Carma, our algorithm does not require a second pass. Regarding hMiner, it has the drawback that the maximum number of possible items in the stream must be known for the algorithm in advance. In contrast to hMiner, we do not assume any such prior knowledge.

One of the representative algorithms belonging to the second category is the Lossy Counting algorithm [2], which has a frequency threshold θ and an error parameter ϵ with $\epsilon < \theta$. This algorithm stores not only the itemsets with frequency at least θ, but also all itemsets with frequency above ϵ, implying that Lossy Counting counts also a large number of infrequent itemsets with frequency between ϵ and θ and keeps those counts in the memory.

This paper presents an algorithm that, at any point in time, approximates the family of frequent itemsets and their support counts from a data stream. The central idea of our algorithm is to combine partial counts with a simple, yet sound statistical model of inference. Our algorithm starts counting an itemset when all of its non-empty proper subsets have reached the minimal frequency threshold and prunes them immediately when they are no longer frequent. Because counting starts only when all subsets are frequent, we may have only partial counts even for the frequent itemsets. To obtain the frequency for the entire stream, our algorithm uses statistical inference to explicitly model the probabilities of itemsets observed from a certain time point t and estimates the frequency for the unobserved interval $[1, t-1]$. Each transaction contributes evidence to this model, making our approach more accurate. In fact, one can show that for infinite data streams, our algorithm is always correct in the limit.

We have compared our algorithm to the CARMA [4] and Lossy Counting [2] algorithms. In our experimental evaluations we have measured the accuracy and the memory footprint for all of the three algorithms, where the accuracy has been measured by calculating the Jaccard distance between the set of correct frequent itemsets and that of the output of the algorithms. While our algorithm has outperformed CARMA both in accuracy and memory, Lossy Counting has generated the most accurate output. However, regarding the memory consumption, our algorithm has required much less space than Lossy Counting. Thus, once the space available is limited, our algorithm can be regarded as an alternative to Lossy Counting.

The rest of the paper is organized as follows. We first collect the necessary preliminaries in Section 2 and then present our algorithm in Section 3. In Section 4 we discuss some estimation strategies natural for our algorithm. Section 5 provides some important details about the implementation, while Section 6 presents our empirical results. We conclude in Section 7 with some interesting problems for future works.

2 Preliminaries

We follow the standard terminology used in frequent itemset mining (see, e.g., [6, 7]). In particular, for a set I of items, a subset $X \subseteq I$ will be referred to as *itemset*.

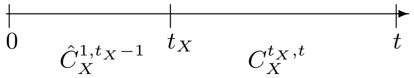


Fig. 1. Transaction stream: The occurrences of itemset X are counted from t_X. $C_X^{t_X,t}$ denotes this observed count for X from t_X to t. As the support count of X is not available for the period 1 to $t_X - 1$, it must be estimated for this time interval; $\hat{C}_X^{1,t_X-1}$ denotes this estimated count.

If for the cardinality of X we have $|X| = k$ then X is a *k-itemset*. We consider a potentially infinite stream of transactions, i.e., a potentially unbounded sequence of subsets of I, that are observed consecutively at discrete time points. Moreover, we assume that each element of this stream is generated independently according to some *fixed*[1], but *unknown* distribution $\mathcal{D} : 2^I \to [0, 1]$. Formally, we have an input sequence $D_1, D_2, ...$ with $D_t \subseteq I$ and $D_t \sim \mathcal{D}$ for each point in time $t \in \mathbb{N}$.

For an itemset $X \subseteq I$ we define the *support count* of X from time i to j by

$$C_X^{i,j} = |\{t \in \mathbb{N} : i \leq t \leq j, X \subseteq D_t\}| \enspace ,$$

i.e., the support count of X is equal to the number of transactions from D_i to D_j that contain X. We define the *frequency* of X at time t as the relative support count

$$\mathrm{freq}_t(X) = \frac{C_X^{1,t}}{t} \enspace .$$

Finally, using these concepts, the family of frequent sets at time t with respect to a frequency threshold $\theta \in [0, 1]$ is given by

$$\mathcal{F}_{t,\theta} = \{X \subseteq I : \mathrm{freq}_t(X) > \theta\} \enspace .$$

Our goal in this paper is to design an algorithm that produces a sequence $\mathcal{C}_1, \mathcal{C}_2, ...$ of families of itemsets such that at each point in time $t \in \mathbb{N}$, the family $\mathcal{C}_t$ approximates $\mathcal{F}_{t,\theta}$ closely, while maintaining a small memory footprint. To evaluate the approximation performance of our and other algorithms, we will use the Jaccard distance as loss function defined for all $A, B \subseteq 2^I$ by

$$l(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|} \enspace .$$

3 Estimating support counts based on partial counts

In this section we present our Partial Counting algorithm approximating the frequency of frequent itemsets from a data stream of transactions at any point of time. The inputs to the algorithm are a minimum frequency threshold $\theta \in [0, 1]$ and, for each time point $t \in \mathbb{N}$, a transaction $D_t \subseteq I$. For each

[1] In the long version of this paper we will show how to get rid of this assumption.

$t \in \mathbb{N}$, the algorithm estimates the frequency of all frequent itemsets with respect to the transaction database $\{D_1, \ldots, D_t\}$, in one pass and without storing the transactions $D_1, \ldots, D_{t-1}$ seen earlier.

For all $t \in \mathbb{N}$, the algorithm keeps only those itemsets in the memory that have been estimated as frequent after having processed D_t; all other itemsets, except for the 1-itemsets, are removed from the memory. Thus, in contrast to e.g. the LOSSY COUNTING algorithm [2], the space complexity of our algorithm is only $O(\hat{\mathcal{F}}_{t,\theta})$, where $\hat{\mathcal{F}}_{t,\theta}$ is the family of itemsets estimated as frequent with respect to $D_1, \ldots, D_t$.

The algorithm is depicted in Algorithm 1. To process the next transaction D_t arriving at time t, the algorithm takes in a first step (Lines 4–12) all non-empty subsets X of D_t (Line 4) and increments the counter for X if X is already in the memory (Lines 5–6). Otherwise, if X is a singleton or it is a k-itemset for some $k > 1$ and all of its $k-1$-subsets are already in the memory, it stores X with some additional auxiliary information. We utilize the Apriori property, i.e., that a k-itemset cannot be frequent if at least one of its (k-1)-subsets is infrequent. More precisely, for X we store a quadruple $(x.set, x.s, x.t, x.count)$ that will be used to estimate the frequency of X for the data stream $D_1, \ldots, D_{t-1}$ (see Lines 8–12 for the definitions of entries in the quadruple). It is important to note that the subsets of D_t are processed in increasing cardinalities, as otherwise potential new frequent itemsets can be lost (see Lines 4,8, and 9).

In a second step (Lines 13–16) the algorithm then prunes all quadruples corresponding to itemsets X from the memory that satisfy $|X| > 1$ and are estimated as infrequent at point t. In Line 14, we first calculate an estimation of the support count of X; we present different strategies for this estimation step in Section 4 by noting that all these strategies estimate the support count from the counts (i.e., *x.count*) maintained by the algorithm using some statistical inference. Figure 1 illustrates the general setting: For an itemset X (re)counted from time t_X, its support count for the period from 1 to $t_X - 1$ must be estimated from the information available at time t. If the frequency derived from this estimation is below the threshold θ, X is removed from the memory. Thus, when an itemset becomes frequent it is stored and counted as long as it is estimated as frequent. When it becomes infrequent, it is immediately pruned.

For a query after time point t, we output all itemsets with their estimated support counts from $\mathcal{F}$ that meet the minimum frequency condition (Line 18–19). According to the construction, all itemsets X in $\mathcal{F}$ with $|X| > 1$ will automatically be part of the output. In summary, we have a true online algorithm that returns a family $\hat{\mathcal{F}}_{t,\theta}$ of itemsets predicted as frequent from the stream of transactions from the beginning of the stream up to time t.

4 Support count estimators

In this section we describe a generic framework for support count estimation and present different strategies for this problem. Except for one, all of the strategies

Algorithm 1 Partial Counting

```
Intitalization
$\mathcal{F} \leftarrow \emptyset$                // current set of frequent patterns with auxiliary information
Processing of transaction $D_t$
for $X \subseteq D_t$ in increasing cardinality do                                  // counting
    if $\exists x \in \mathcal{F}$ with $x.set = X$ then
        increment $x.count$
    else
        if $|X| = 1 \vee (\forall Y \subset X : \exists y \in \mathcal{F}$ with $y.set = Y \wedge |Y| = |X| - 1)$ then
            $x.s \leftarrow \{(y.set, y.count) : y \in \mathcal{F} \wedge y.set \subset x.set \wedge |y.set| = |x.set| - 1)\}$
            $x.t = t$
            $x.count = 1$
            $\mathcal{F} \leftarrow \mathcal{F} \cup \{x\}$
for $x \in \mathcal{F}$ with $|x.set| > 1$ do                                        // pruning
    compute $\hat{C}_x^{1,t}$                                                     // see Section 4
    if $\hat{C}_x^{1,t}/t < \theta$ then
        delete $x$ from $\mathcal{F}$
Output after timepoint $t$
for $x \in \mathcal{F}$ do
    if $|x.set| > 1 \vee x.count/t > \theta$ then output (x.set,$\hat{C}_x^{1,t}$)
```

in this section are based on some careful combination of the observed counts with the estimated ones that are derived from conditional probabilities.

We write $\hat{C}$ for estimated support counts. As illustrated in Figure 1, whenever there is some observed support count for an itemset X, the estimated support count for the entire period of all transactions is given by the *observed* support count $C_X^{t_X,t}$ plus the *estimation* $\hat{C}_X^{1,t_X-1}$ of the support count for the time period $[1, t_X - 1]$ for which the support count of X is not available at point t. As long as no observed count for X exists, its estimated frequency is 0, i.e.,

$$\hat{C}_X^{1,t} = \begin{cases} \hat{C}_X^{1,t_X-1} + C_X^{t_X,t} & \text{if } t \geq t_X \\ 0 & \text{o/w} . \end{cases}$$

We now present different strategies to compute $\hat{C}_X^{1,t_X-1}$. We first recall an estimation from [4] and then propose two natural strategies based on conditional probabilities.

4.1 Upper bound estimation (ube)

This estimation strategy is used in Carma [4]. It takes the minimum count of all $(k-1)$-subitemsets Y of a k-itemset X, i.e.,

$$\hat{C}_X^{1,t_X-1} = \min_{\substack{Y \subset X, \\ |Y|=|X|-1}} \hat{C}_Y^{1,t_X-1} .$$

Clearly, this formula gives an upper bound on the true support count of X. Notice that this is a static strategy in the sense that it does not improve the estimation $\hat{C}_X^{1,t_X-1}$ as further transactions arrive.

4.2 Estimation based on conditional probabilities

We now turn to more complex, dynamic estimation strategies. They are based on the probabilistic view of frequencies that for any itemset X and $t \in \mathbb{N}$

$$p(X)^{1,t} = p(Y)^{1,t} \cdot p(X|Y)^{1,t}$$

for any $Y \subset X$. To estimate $p(X)^{1,t}$, we need to estimate $p(X)^{1,t_X-1}$, as all information about X is available for the interval $[t_X, t]$. We estimate $p(X)^{1,t_x-1}$ by estimating (i) $p(Y)^{1,t_X-1}$ and (ii) $p(X|Y)^{1,t_X-1}$.

(i) Regarding $p(Y)^{1,t_X-1}$, we estimate it recursively by

$$p(Y)^{1,t_X-1} \approx \frac{\hat{C}_Y^{1,t_X-1}}{t_X - 1} \tag{1}$$

by noting that the support counts are stored from the very beginning for all 1-itemsets.

(ii) Regarding $p(X|Y)^{1,t_X-1}$, we make use of the fact that $[t_X, t]$ is a common observation period for both X and Y and the assumption that the underlying distribution $\mathcal{D}$ is stationary and estimate $p(X|Y)^{1,t_X-1}$ by

$$p(X|Y)^{1,t_X-1} \approx p(X|Y)^{t_X,t}, \tag{2}$$

which, in turn, can be calculated by

$$p(X|Y)^{t_X,t} = \frac{C_X^{t_X,t}}{C_Y^{t_X,t}} \quad . \tag{3}$$

One can show that for sufficiently large t_X and t, (2) gives a close estimation with high probability.

Putting together, from (1), (2), and (3) it follows that $p(X)^{1,t_X-1}$ can be estimated by

$$p(X)^{1,t_X-1} \approx \frac{\hat{C}_Y^{1,t_X-1}}{t_X - 1} \cdot \frac{C_X^{t_X,t}}{C_Y^{t_X,t}} \quad . \tag{4}$$

As the frequency of X in $[1, t_X - 1]$ is identical to the probability $p(X)^{1,t_X-1}$, by (4) the support count $\hat{C}_X^{1,t_X-1}$ can be estimated by

$$\begin{aligned} \hat{C}_X^{1,t_X-1} &= p(X)^{1,t_X-1} \cdot (t_X - 1) \\ &\approx \frac{\hat{C}_Y^{1,t_X-1} \cdot C_X^{t_X,t}}{C_Y^{t_X,t}} \quad . \end{aligned}$$

The two strategies presented below build upon the idea discussed above. They differ from each other only by the particular choice of Y. All strategies have in common that they estimate the support counts only for itemsets X with $|X| \geq 2$.

transactions	a	b	a	ab	a	b	a	ab
t	1	2	3	4	5	6	7	8
freq(a)	1	0.5	0.66	0.75	0.8	0.6	0.71	0.75
freq(b)	0	0.5	0.33	0.5	0.4	0.5	0.43	0.5
freq(ab)	0	0	0	0.25	0.2	0.17	0.14	0.25
ube	0	0	0	0.5	0.4	0.33	0.29	0.38
me	0	0	0	0.5	0.4	0.25	0.21	0.33
ae	0	0	0	0.5	0.4	0.29	0.23	0.35

Table 1. A data stream of transactions illustrating the different estimation strategies ube, me, and ae. The first row shows the transactions, the second row t, the next three rows the frequencies for itemsets a, b and ab respectively. The last three rows show the estimated frequencies for the itemsets a, b, and ab for the three strategies presented.

(a) Minimum estimation (me): This strategy uses the single subset Y that results in the minimum estimated count for X, i.e.,

$$\hat{C}_X^{1,t_X-1} = \begin{cases} \min\limits_{\substack{Y \subset X, \\ |Y|=|X|-1}} \dfrac{\hat{C}_Y^{1,t_X-1} \cdot C_X^{t_X,t}}{C_Y^{t_X,t}} & \text{if } |X| > 1 \\ 0 & \text{o/w (i.e., if } |X| = 1) \ . \end{cases}$$

(b) Average estimation (ae): Averaging is a standard technique to combine several uncertain predictors for obtaining a more robust result than any individual predictor would give. This strategy averages over all y, i.e.,

$$\hat{C}_X^{1,t_X-1} = \begin{cases} \dfrac{1}{|X|} \cdot \sum\limits_{\substack{Y \subset X, \\ |Y|=|X|-1}} \dfrac{\hat{C}_Y^{1,t_X-1} \cdot C_X^{t_X,t}}{C_Y^{t_X,t}} & \text{if } |X| > 1 \\ 0 & \text{o/w (i.e., if } |X| = 1) \ . \end{cases}$$

4.3 An Illustrative Example

To illustrate the three different strategies presented above, we use the small example given in Table 1. It shows a total of 8 transactions in the first row, t in the second row, and the frequencies of a, b, and ab in rows three to five. The last three rows show the estimated frequencies for the three strategies.

In the example the first estimation occurs for transaction 4 as the set ab occurs for the first time in transaction 4 and is counted from this transaction onwards. Up to and including the third transaction the set a occurs twice, the set b once. All estimations rely on the counts of a and b for the first three transactions. The strategies start to differ from the 6th transaction onwards. We will therefore illustrate the different strategies for the 6th transaction. $C_{ab}^{4,6} = 1$ for all strategies.

(i) ube estimates $\hat{C}_{ab}^{1,3} = min(2, 1) = 1$, which gives

$$\hat{C}_{ab}^{1,6} = \hat{C}_{ab}^{1,3} + C_{ab}^{4,6} = 1 + 1 = 2$$

and thus, a frequency of $\frac{2}{6} = 0.33$.

(ii) me estimates $\hat{C}_{ab}^{1,3} = min(1 \cdot \frac{1}{2}, 2 \cdot \frac{1}{2}) = 0.5$, which gives

$$\hat{C}_{ab}^{1,6} = \hat{C}_{ab}^{1,3} + C_{ab}^{4,6} = 0.5 + 1 = 1.5$$

and a frequency of $\frac{1.5}{6} = 0.25$.

(iii) ae estimates $\hat{C}_{ab}^{1,3} = \frac{1}{2}(1 \cdot \frac{1}{2} + 2 \cdot \frac{1}{2}) = 0.75$, which gives

$$\hat{C}_{ab}^{1,6} = \hat{C}_{ab}^{1,3} + C_{ab}^{4,6} = 0.75 + 1 = 1.75$$

and a frequency of $\frac{1.75}{6} = 0.29$.

All other estimations are computed accordingly.

5 Implementation

In this section we briefly discuss some important implementation issues of the Partial Counting algorithm, in particular, the data structure, insertion into the data structure, and pruning.

The data structure $\mathcal{F}$ can be stored as a prefix tree. This allows for a compact representation of the family of itemsets, as it suffices to store for each itemset X only a single item. Indeed, the itemset corresponding to a node can uniquely be recovered by concatenating the items on the path from the root to the node at hand. Furthermore, it also allows for pruning entire branches, if a node has to be deleted which has further children.

The set $x.s$ can be stored compactly as an array of integers. For an itemset X with $|X| = k$, there are k different $(k-1)$-subsets. We sort all these subsets Y lexicographically and take only the index of the missing item i (i.e., which satisfies $Y \cup \{i\} = X$) to store the count for itemset Y. Thus, in this way there are k counters for the subsets and one for the itemset X.

Theoretically, we may start observing an itemset X as soon as the estimated frequencies for all subsets $Y \subset X$ have reached the minimum frequency threshold θ. X may not occur in the transaction, when this condition is met. However, X may become frequent only when it occurs in the current transaction. That is, we start counting X, with the transaction containing X after all Y have already reached the minimum frequency threshold. The price we pay for this is that the first estimation of $p(X|Y)$ is 1 and as such, very inaccurate.

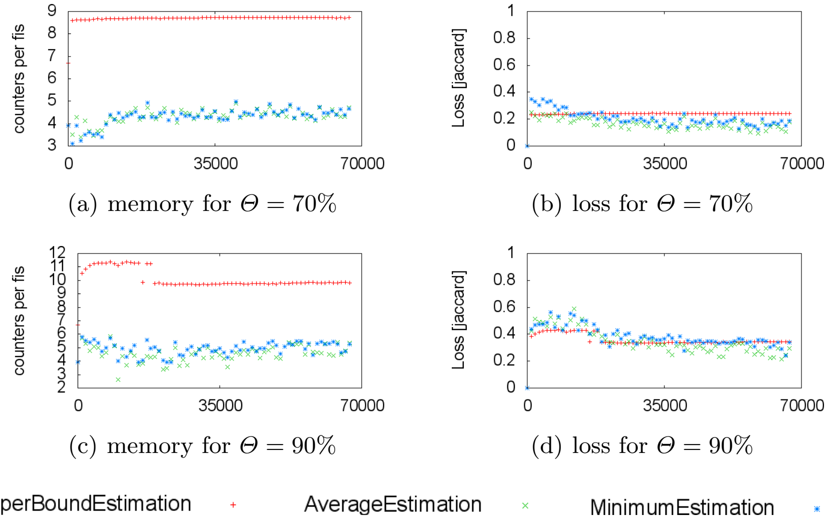
An itemset which is inserted in $\mathcal{F}$ at time t is never pruned in t, otherwise it would not have been inserted. We exploit this by skipping the pruning test for itemsets which were inserted in the same transaction.

6 Experimental Evaluation

To assess the performance of our algorithm in practice, we analyzed the number of counters in memory and the empirical loss for real-world datasets taken from the UCI machine learning repository [8]. The empirical evaluations focus on the following three main aspects:

- Comparing the memory and loss of the different estimation strategies.
- Comparing Partial Counting with Lossy Counting and Carma in terms of memory requirements and loss.
- Evaluating the memory and loss of Partial Counting with decreasing frequency threshold.

For each dataset we shuffle the transactions, add them one by one and mine at regular intervals. The ground truth is obtained with Apriori [6]. We compute the Jaccard distance based loss for each algorithm as defined in Section 2. This loss accounts for both over and underestimation of itemset frequencies without further distinguishing between the two.



(a) memory for $\Theta = 70\%$

(b) loss for $\Theta = 70\%$

(c) memory for $\Theta = 90\%$

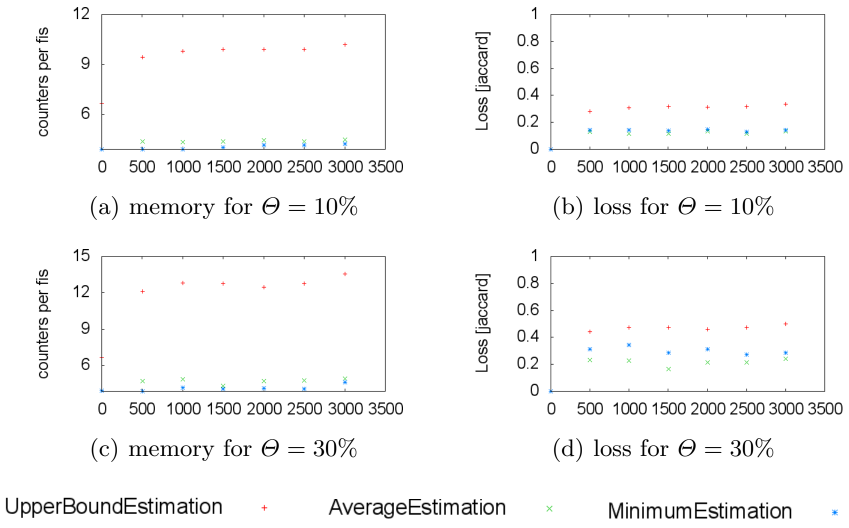
(d) loss for $\Theta = 90\%$

Fig. 2. Memory and loss of Partial Counting with different estimation strategies for dataset connect-4 at 70% (top) and 90% (bottom) frequency threshold.

Overall, the results show that

(i) our algorithm requires less memory than CARMA and Lossy Counting (see Figure 4),
(ii) mining at lower thresholds results in a better approximation (see Figures 2, 3, and 4),
(iii) the two strategies (me and ae) based on statistical inference outperform the simpler strategy (ube) (see Figures 3(b) and 3(d)), and
(iv) the loss decreases as the stream evolves (Figures 2(d) and 4(b)).

We report the memory requirements and loss for the dataset connect-4 with a maximal itemset length of 3 and a minimum frequency threshold of 70% and 90% in Figure 2 and for the chess dataset at 10% and 30% frequency threshold in Figure 3.



(a) memory for $\Theta = 10\%$

(b) loss for $\Theta = 10\%$

(c) memory for $\Theta = 30\%$

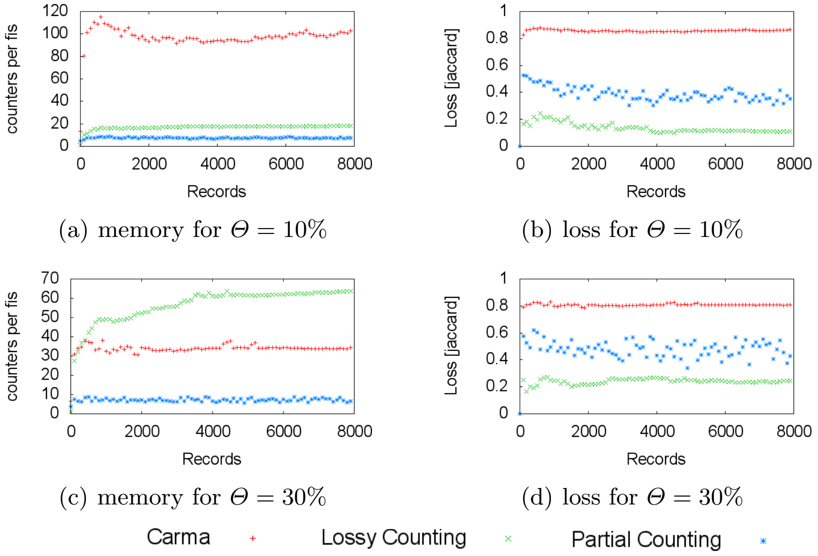
(d) loss for $\Theta = 30\%$

Fig. 3. Memory and loss of Partial Counting with different estimation strategies for dataset chess at 10% (top) and 30% (bottom) frequency threshold.

We first compare the memory and loss of the different estimation strategies. The memory is measured in counters per truly frequent itemset. The average and minimum estimation strategies require consistently fewer counters per frequent itemset than the upper bound estimation strategy (Figures 2(a), 2(c), 3(a), and 3(c)). In terms of loss the inference based estimation strategies are the best. The average estimation strategy has the overall lowest loss, the upper bound strategy the highest (Figures 2(b), 2(d), 3(b), and 3(d)).

To compare our approach with state of the art algorithms, we reimplemented Lossy Counting and Carma. In Figure 4 we compare the counters in memory per truly frequent itemset 4(c) and the loss 4(d) of Carma, Lossy Counting, and Partial Counting. The number of counters per truly frequent itemset is a fair mode of comparison, as it is an implementation independent measure, whereas Megabytes depend more on the internal data representation. Our Partial Counting algorithm clearly outperforms both algorithms in terms of memory and Carma also in loss. Note that the number of counters in memory of Lossy Counting increases as the stream evolves (see Figure 4(c)). The memory required by our Partial Counting algorithm remains low, as the stream evolves. It is not surprising that Lossy Counting performs better in terms of loss for the 10% frequency threshold, as it stores more information yielding typically better results.

We now take a look at the effect of the frequency threshold upon our algorithm. The experiments show that a lower frequency threshold results in a lower loss for our algorithm (Figures 2(b) vs 2(d), 3(b) vs 3(d), and 4(d) vs 4(b)). The effect on memory is the same but less prominent (Figure 2(a) vs 2(c)).



(a) memory for $\Theta = 10\%$

(b) loss for $\Theta = 10\%$

(c) memory for $\Theta = 30\%$

(d) loss for $\Theta = 30\%$

Fig. 4. Memory and loss of Carma, Lossy Counting, and Partial Counting at 10% (top) and 30% (bottom) frequency threshold for the dataset mushroom.

Finally we observe that the loss decreases as the stream evolves (Figures 2(b), 2(d), 4(b) and 4(d)). This trend may be not observed for the short stream in Figure 3.

7 Conclusion

We have presented the Partial Counting algorithm for mining frequent itemsets from data streams with a space complexity proportional to the number of frequent itemsets generated from the stream at any point in time. The algorithm starts counting the frequency of an itemset once all of its subsets are estimated to be frequent. The support count of itemsets is estimated by statistical inference.

We have evaluated our algorithm empirically and compared its memory consumption and accuracy with that of CARMA and Lossy Counting. While our algorithm outperforms CARMA in both of these aspects, Lossy Counting turned out to have a better accuracy (measured with Jaccard distance). This is, however, not surprising because Lossy Counting, as our experiments clearly demonstrate, uses significantly more space than Partial Counting. Thus, our algorithm is an alternative to Lossy Counting when memory is limited.

In its current status, Partial Counting is outperformed in terms of accuracy by Lossy Counting. As for future work, we would like to reduce this gap. We are going to follow two ideas to achieve this goal. We will experiment with a confidence parameter to reduce the effect of overestimation when we first observe itemsets X and Y together. We believe that we can further improve our probabilistic inference mechanism based on the observation that an itemset which is not counted, while all of its subsets are counted, must be infrequent.

Another important issue is that, as we will show in the long version of this paper, our algorithm is correct for any infinite data stream. We note that this holds even if we relax the assumption that the underlying distribution is stationary. It is an interesting question that we are investigating, whether concept drift can be handled for finite data streams as well.

Last but not least, we are going to analyse our algorithm for distributed data streams as well. One of the most challenging questions towards this direction is to find a distributed algorithm that generates frequent itemsets of good approximation performance with as few as possible communication.

8 Acknowledgment

This research has been supported by the EU FP7-ICT-2013-11 under grant 619491 (FERARI).

References

1. Demaine, E.D., López-Ortiz, A., Munro, J.I.: Frequency estimation of internet packet streams with limited space. In: Proceedings of the 10th Annual European Symposium on Algorithms. ESA '02, London, UK, UK, Springer-Verlag (2002) 348–360
2. Manku, G.S., Motwani, R.: Approximate frequency counts over data streams. In: Proceedings of the 28th International Conference on Very Large Data Bases. VLDB '02, VLDB Endowment (2002) 346–357
3. Cormode, G., Muthukrishnan, S.: An improved data stream summary: the count-min sketch and its applications. Journal of Algorithms **55**(1) (2005) 58 – 75
4. Hidber, C.: Online association rule mining. In Delis, A., Faloutsos, C., Ghandeharizadeh, S., eds.: SIGMOD 1999, Proceedings ACM SIGMOD International Conference on Management of Data, June 1-3, 1999, Philadephia, Pennsylvania, USA, ACM Press (1999) 145–156
5. Wang, E., Chen, A.: A novel hash-based approach for mining frequent itemsets over data streams requiring less memory space. Data Mining and Knowledge Discovery **19**(1) (2009) 132–172
6. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. In: SIGMOD '93: Proceedings of the 1993 ACM SIGMOD international conference on Management of data, New York, NY, USA, ACM (1993) 207–216
7. Mannila, H., Toivonen, H., Verkamo, A.I.: Efficient algorithms for discovering association rules. In: Knowledge Discovery in Databases: Papers from the 1994 AAAI Workshop, Seattle, Washington, July 1994. Technical Report WS-94-03. (1994) 181–192
8. Bache, K., Lichman, M.: UCI machine learning repository (2013)

Linearizing Belief Propagation for Efficient Label Propagation

Stephan Günnemann

Carnegie Mellon University, USA
sguennem@cs.cmu.edu

Technische Universität München, Germany
guennemann@in.tum.de

Abstract. How can we tell when accounts are fake or real in a social network? And how can we tell which accounts belong to liberal, conservative or centrist users? Often, we can answer such questions and label nodes in a network based on the labels of their neighbors and appropriate assumptions of homophily (”birds of a feather flock together”) or heterophily (”opposites attract”). One of the most widely used methods for this kind of inference is Belief Propagation (BP), which can effectively been used as a principle for label propagation in partially labeled networks. One main problem with BP, however, is that the convergence in graphs with loops is not guaranteed.

In this talk, I will present two principles for efficient label propagation that are based on the idea of linearizing Belief Propagation [1]. First, I will introduce ’Linearized Belief Propagation’ (LinBP), a linearization of BP that allows a closed-form solution via intuitive matrix equations and, thus, comes with convergence guarantees. It handles homophily, heterophily, and more general cases that arise in multi-class settings. In the second part, I will present ’Single-pass Belief Propagation’ (SBP), a ”localized” version of LinBP that propagates information across every edge at most once and for which the final class assignments depend only on the nearest labeled neighbors. In addition, SBP allows fast incremental updates in dynamic networks. Runtime experiments show that LinBP and SBP are orders of magnitude faster than standard BP, while leading to almost identical node labels.

References

1. W. Gatterbauer, S. Günnemann, D. Koutra, and C. Faloutsos. Linearized and single-pass belief propagation. *PVLDB*, 8(5):581–592, 2015.

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

Online F-Measure Optimization

Róbert Busa-Fekete[1], Balázs Szörényi[2], Krzysztof Dembczyński[3], and Eyke Hüllermeier[1]

[1] Department of Computer Science, University of Paderborn, Warburger Str. 100,
33098 Paderborn, Germany
{busarobi,eyke}@upb.de

[2] MTA-SZTE Research Group on Artificial Intelligence, Tisza Lajos krt. 103.,
H-6720 Szeged, Hungary
szorenyi@inf.u-szeged.hu

[3] Institute of Computing Science, Poznań University of Technology, Piotrowo 2,
60-965 Poznań, Poland Krzysztof.Dembczynski@cs.put.poznan.pl

Abstract. [4] The F-measure is an important and commonly used performance metric for binary prediction tasks. By combining precision and recall into a single score, it avoids disadvantages of simple metrics like the error rate, especially in cases of imbalanced class distributions. The problem of optimizing the F-measure, that is, of developing learning algorithms that perform optimally in the sense of this measure, has recently been tackled by several authors. In this paper, we study the problem of F-measure maximization in the setting of online learning. We propose an efficient online algorithm and provide a formal analysis of its convergence properties. Moreover, first experimental results are presented, showing that our method performs well in practice.

Keywords: classification, learning theory, F-measure, structured output prediction

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

[4] This work is accepted at the Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS).

Probabilistic Frequent Subtree Kernels

Pascal Welke[1], Tamás Horváth[1,2], and Stefan Wrobel[2,1]

[1] Dept. of Computer Science, University of Bonn, Germany
[2] Fraunhofer IAIS, Schloss Birlinghoven, Sankt Augustin, Germany

Abstract. Graph kernels have become a well-established approach in graph mining. One of the early graph kernels, the *frequent subgraph kernel*, is based on embedding the graphs into a feature space spanned by the set of all frequent connected subgraphs in the input graph database. A drawback of this graph kernel is that the preprocessing step of generating *all* frequent connected subgraphs is computationally intractable. Many practical approaches ignore this limitation, implying that such systems can be infeasible even for small datasets. Approaches that do not disregard this aspect either restrict the feature space or restrict the class of the input graphs to guarantee correctness and efficiency.
We propose a frequent subgraph kernel that is not restricted to any particular graph class, but still efficiently computable. All such kernels can only be achieved by relaxing the correctness condition on mining frequent connected subgraphs. We give up the demand on completeness and represent each input graph by a polynomial size random sample of its spanning trees. Such a random sample is a forest and can be generated in polynomial time. Thus, as frequent subtrees in forests can be listed with polynomial delay, we arrive at an efficient frequent subgraph mining algorithm. Our approach is sound, but incomplete: (i) it is only able to identify frequent subtrees, and not arbitrary graph patterns, and (ii) even if a tree pattern is frequent, it might not be identified as such. Calculating a representation in this feature space for any unseeng graph is done by the same incomplete procedure.
Our empirical evaluation on two chemical datasets shows that a considerable fraction of all frequent subtrees can be recovered even from *one* random spanning tree per graph. Regarding the expressive power of probabilistic frequent subtrees, we have observed a marginal loss in predictive performance. However, we have achieved a three time speed-up against the ordinary frequent subgraph kernel. Furthermore, our method is able to process significantly larger datasets and generates a much smaller feature set than the original algorithm.

A long version of this extended abstract appeared in [1].

[1] P. Welke, T. Horváth, and S. Wrobel. Probabilistic Subtree Kernels. To appear in: New Frontiers in Mining Complex Patterns, Springer, 2016.

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

Simulated Annealing with Parameter Tuning for Wind Turbine Placement Optimization

Daniel Lückehe[1], Oliver Kramer[2], and Manfred Weisensee[3]

[1] Department of Geoinformation, Jade University of Applied Sciences, Oldenburg, Germany daniel.lueckehe@uni-oldenburg.de
[2] Department of Computing Science, University of Oldenburg, Oldenburg, Germany oliver.kramer@uni-oldenburg.de
[3] Department of Geoinformation, Jade University of Applied Sciences, Oldenburg, Germany weisensee@jade-hs.de

Abstract. Because of wake effects and geographical constraints, the search for optimal positions of wind turbines has an important part to play for their efficiency. The determination of their positions can be treated as optimization problem and can be solved by various methods. In this paper, we propose optimization approaches based on Simulated Annealing (SA) to improve solutions for the wind turbines placement problem. We define neighborhoods of solutions and analyze the influence of specific parameters, e.g., neighborhood distance and SA temperature in experimental studies. The experiments are based on a real-world scenario with a wind model, wind data from a meteorological service, and geographical constraints. Inspired by adaptive step size control applied in Evolutionary Strategies (ES), we propose an approach using an adaptive neighborhood distance and compare the results to optimization runs with a constant neighborhood distance. Also the best and worst optimization run and the corresponding placement results are shown and compared.

1 Introduction

Planning and optimization of renewable energy resources is an important part of today's activities towards an ecologically friendly smart grid. As the environment of wind turbines is significant for their efficiency, the determination of their locations should be handled carefully and with consideration of various aspects. In this paper, we use a wind model based on wind distributions using data from the German Weather Service and take geographical constraints into account. We apply different optimization approaches using SA to the turbine placement problem with the objective to maximize the power output. SA is often used for combinatorial problems, but can also be applied to continuous solution spaces. For this, we define neighborhoods in the continuous solution space of turbine

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

positions. We also propose an adaptive variant of neighborhoods for SA inspired by an adaptive method from the field of ES, as the definition of neighborhood corresponds to the step size of ES.

This paper is structured as follows. In Section 2, we give an overview to related work. The wind model is explained in Section 3 and includes the definition of the employed scenario. In Section 4, we introduce the optimzation approaches, followed by the experimental results in Section 5. In Section 6, conclusions are drawn.

2 Related Work

The wind turbine placement problem is widely known and there are a lot of different models and approaches to solve it [5]. We observe a trend towards more realistic representations of the optimization problem, e.g., Kusiak and Song [8] are using Weibull [18] distribution and the Jensen [13] wake model to describe the behavior of the wind. Their model is able to compute the power output of turbines on a continuous map considering wake effects. They solved the optimization problem with a simple ES. Further works exploit more complex approaches like the covariance matrix adaptation evolution strategy (CMA-ES) [4] to solve the optimization problem [17]. There are also approaches that include geographical information from map services to the turbine placement problem [9].

To solve the turbine placement problem, we employ stochastic search algorithms [12]. In particular, we aim for an approach that exploits SA. For a comprehensive overview of SA, we refer to [6]. In this work, the basic idea of SA is explained, critically analyzed, and different variants of SA are experimentally considered. Nourani and Andresen [14] put a focus on the cooling schedules for SA. In their work, constant thermodynamic speed, exponential, logarithmic, and linear cooling schedules are analyzed. Rivas et al. [15] published an approach to solve the turbine placement problem for large offshore wind farms by SA. Their algorithm employs three types of local search operations: add, move, and remove. The operations are performed recursively and each has its own temperature. Based on the experimental results, in the conclusion of this work SA is called a suitable method for the wind turbine placement optimization problem.

3 Wind Setting

In this section, we introduce the wind turbine model that we use in the experimental part of this work. This model computes the produced energy of a wind farm. The description is followed by the specification of the real-world scenario used in this paper.

3.1 Wind Turbine Model

As the objective in this work is to maximize the power output of a wind farm, we apply a wind turbine model to calculate the produced energy of the turbines.

The wind turbine model f exploits a scenario that consists of wind turbines and their power curves based on the Enercon E101, wind data from the German Weather Service, wake effects computation using the Jensen wake model [13], and geographical constraints based on data from OpenStreetMap [3]. The COSMO-DE [2] wind data from the German Weather Service are used to calculate Weibull distributions [18] for every position and wind direction. With the model from Kusiak and Song [8] the power output E is calculated. For one wind turbine with position $\mathbf{t}_i$, it applies:

$$E(\mathbf{t}_i) = \int_0^{360} p_\theta(\mathbf{t}_i, \theta) \cdot E_\theta(\mathbf{t}_i, \theta) d\theta \tag{1}$$

with the power output $E_\theta(\mathbf{t}_i, \theta)$ for one wind direction:

$$E_\theta(\mathbf{t}_i, \theta) = \int_0^{\infty} \beta_i(v) \cdot p_v(v, k(\mathbf{t}_i, \theta), c(\mathbf{t}_i, \theta)) dv. \tag{2}$$

The distribution of wind angles is described in $p_\theta(\mathbf{t}_i, \theta)$, the function $\beta_i(v)$ specifies the power curve of the used wind turbine, and the Weibull distribution $p_v(v, k(\mathbf{t}_i, \theta), c(\mathbf{t}_i, \theta))$ represents the wind speed distribution. In the evolutionary optimization process, the geographical constraints are modeled by a variant of death penalty that is similar to the approach used by Morales and Quezada [11]. For a detailed description of the wind model, we refer to our depiction in [10].

The model f computes the power output of a solution $\mathbf{x}$ that describes the positions of multiple wind turbines for a defined scenario, i.e., a *scenario* specifies the map section and the wind distributions. Thereby, the optimization objective is to maximize the sum of the power output E of all turbines $\mathbf{t}$: $f(\mathbf{x}) = \sum_{i=1}^{N/2} E(\mathbf{t}_i)$. The solution $\mathbf{x}$ is a vector of elements $\mathbf{x} = (x_1, x_2, \ldots, x_N)^T$ with the length N coding the x- and y-coordinate of every turbine x^t and y^t, i.e.:

$$\mathbf{x} = (x_1^t, y_1^t, x_2^t, y_2^t, \ldots, x_{N/2}^t, y_{N/2}^t).$$

With $\mathbf{t}_i = (x_i^t, y_i^t)$ the solution vector can be written as:

$$\mathbf{x} = (\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_{N/2}).$$

We denote the position $\mathbf{t}_i$ of a specific turbines of solution $\mathbf{x}_j$ as $\mathbf{t}_i^{\mathbf{x}_j}$. This notation is required for the definition of neighborhoods in Section 4.1.

3.2 Scenario

To specify a scenario, we keep in mind the objective to model realistic settings for Lower Saxony, Germany. There are more than $5,000$ wind turbines in Lower Saxony. Most of them are grouped in wind farms smaller than 30 turbines [16]. We define an onshore scenario with 22 turbines in an area of 5 km x 5 km, which leads to a 44-dimensional solution space. To take into account the constraints outside the feasible area, we also consider the geographical information within

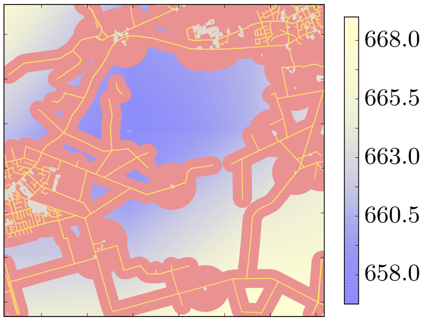


Fig. 1. Visualization of the scenario.

a distance of 1 km beyond each border. The coordinates of the scenario are 53.3925° - 53.45648°, 7.7395° - 7.84304° in decimal degrees. Figure 1 shows the scenario with red constrains and yellow to blue potential map. It consists of 311 buildings and 355 streets consisting of 1987 parts modeled in OpenStreetMap. The potential for a turbine without wake effect by other turbines is about 660 kW becoming lower in the middle of the map. Figure 2 shows an exemplary wind rose in our scenario. Most wind is coming from south-west in this scenario.

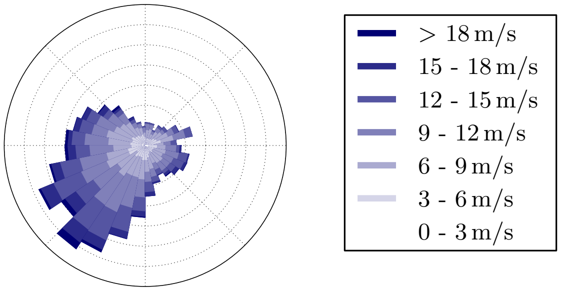


Fig. 2. Wind rose at location of the scenario.

4 Simulated Annealing

In the following, we introduce the SA variants that are compared in the experimental study. First, the concept of SA is introduced. Then, we define the neighborhood of two different solutions $\mathbf{x}_i$ and $\mathbf{x}_j$ using a neighborhood distance d_n, followed by the introduction of two optimization algorithms. The first one is an SA approach with a deterministic cooling schedule. In the second approach, we propose an adaptive control of the neighborhood distance d_n, which is inspired by the step size control used by ES.

SA is based on the cooling process of metallic elements that reduce defects in crystals. The algorithm starts with an initial solution $\mathbf{x}_0$. In our work, we start with a feasible randomly created solution. Then, the optimization algorithm creates a neighbor $\mathbf{x}'$ of the actual solution $\mathbf{x}$. The neighboring solution $\mathbf{x}'$ is analyzed w. r. t. fitness function f. If $f(\mathbf{x}') > f(\mathbf{x})$ the solution $\mathbf{x}'$ replaces $\mathbf{x}$ in the following iteration. For a better chance to leave local optima, SA can also accept worse solutions $\mathbf{x}'$ with a fitness function value $f(\mathbf{x}') < f(\mathbf{x})$. The probability M that describes the chance to accept a worse solution is depending on the difference of the fitness function values $f(\mathbf{x}')$ and $f(\mathbf{x})$ and on a parameter T that is called temperature, i.e.:

$$M(\mathbf{x}, \mathbf{x}', T) = e^{-\frac{f(\mathbf{x}) - f(\mathbf{x}')}{T}} \tag{3}$$

The temperature T is variable during an optimization run with SA. Equation 3 shows how the temperature T affects the probability M to accept a worse solution: The higher the temperature T is, the higher is the probability M.

4.1 Turbine-Oriented Neighborhood

To define the neighborhood between two solutions $\mathbf{x}_i$ and $\mathbf{x}_j$, we first specify a neighborhood distance d_n. Solution $\mathbf{x}_j$ is in the neighborhood of solution $\mathbf{x}_i$, if one turbine $\mathbf{t}_k$ with $k \in \{1, \ldots, N/2\}$ is different in both solutions but within the distance d_n:

$$\max\left(\mathbf{t}_k^{\mathbf{x}_i} - \mathbf{t}_k^{\mathbf{x}_j}\right) \leq d_n \tag{4}$$

and all other turbine positions from both solutions are equal:

$$\forall \mathbf{t}_l : \mathbf{t}_l^{\mathbf{x}_i} = \mathbf{t}_l^{\mathbf{x}_j} \tag{5}$$

with $l \in \{1, \ldots, N/2\}$ and $l \neq k$.

4.2 Deterministic Cooling Schedule

In SA, the cooling process starts with a high probability M at the beginning of the optimization process, while M is reduced in the course of the optimization. Various options to reduce M have been introduced in the past, mainly by decreasing temperature T. A common rule to cool down is a deterministic temperature control with $T' = \alpha \cdot T$. Defining i as iteration number and T_0 as starting temperature, the cooling process can be described by:

$$T_i = \alpha^i \cdot T_0 \tag{6}$$

with $0 < \alpha < 1$.

4.3 Approach with Constant Neighborhood Distance

To implement an optimization algorithm using SA with a deterministic cooling schedule, we have to determine the neighborhood distance d_n, the initial temperature T_0, and the cooling factor α. In this work, we use a small and a large neighborhood distance d_n. The small distance d_n^- is set to 50 m, which means a turbine $\mathbf{t}$ can be shifted up to 50 m per iteration in both dimensions. With this distance the solution can reach the next local optimum, but is not able to shift a turbine $\mathbf{t}$ over a constraint like a street. Therefore, we define the large distance $d_n^+ = 500\,\mathrm{m}$ which makes it possible to shift a turbine over a street but the fine-tuning is more difficult.

As we can see in Equation 3, the temperature T must be chosen depending on the scale of the difference $\Delta f = f(\mathbf{x}) - f(\mathbf{x}')$. Preliminary experiments show that the difference Δf depends on the neighborhood distance d_n. Using d_n^-, in the first 100 iterations with a new feasible solution $\mathbf{x}'$, it applies for the difference Δf that its mean value and standard deviation are $\Delta f \approx 0 \pm 6$ with $\max(\Delta f) \approx 20$. It means better and worse solutions are equally distributed, in about 70% of the iterations the difference is smaller than 6, and the maximum value is approximately 20. Using d_n^+, it applies for the first 100 iterations with a new feasible solution $\mathbf{x}'$: $\Delta f \approx 10 \pm 30$ with $\max(\Delta f) \approx 120$. We define two different initial temperatures using this information. A high temperature T_0^h has the objective that at the beginning of the optimization process in 10% of the cases a worse solution is accepted. As we are not focusing on extreme values, we use the standard deviation. It applies for d_n^-:

$$M = 0.1/0.7 = e^{-\frac{6}{T_0^h}} \Rightarrow T_0^h = -\frac{6}{\ln(0.1/0.7)} \approx 3.08 \tag{7}$$

And a low temperature T_0^l with the objective to accept 1% of the worse solutions at the beginning of the optimization process:

$$T_0^l = -\frac{6}{\ln(0.01/0.7)} \approx 0.71 \tag{8}$$

For d_n^+, we are using $T_0^h = 15.4$ and $T_0^l = 3.55$. To determine α, we define that the temperature T should be decreased by 10% every 100 iterations. So it applies:

$$\alpha = \sqrt[100]{0.9} \approx 0.99895 \tag{9}$$

4.4 Approach with Adaptive Neighborhood Distance

In this section, we propose an approach based on the algorithm using the deterministic cooling schedule from the last section extending it with an adaptive technique. In the field of ES, an adaptive step size control is a common tool to improve optimization results. The idea is that the solution space conditions change during an optimization run and therefore the optimal step size is not the same during the whole run. At the beginning, a large step size allows the exploration of the solution space, while at the end, a small step size allows the fine-tuning of solutions. A well-known example is Rechenberg's step size control [1]. In our approach using SA, the neighborhood distance d_n play a similar role like the step size, i.e., controlling the exploration characteristics of the algorithm. According to Rechenberg's step size control, we count the number of improved solutions $\mathbf{x}'$ with $f(\mathbf{x}') > f(\mathbf{x})$ and compute ratio of improved solutions w.r.t. all solutions. If more than 1/5th of the new solutions have been improved, the neighborhood distance d_n is increased. Otherwise, it is decreased. To become independent from short-term fluctuations, we evaluate the ratio after 100 new solutions. To increase or decrease, the neighborhood distance d_n is modified with factor $\tau = 1.1$ representing a change of 10%.

Algorithm 1 Adaptive Neighborhood Distance

```
Require: $d_n$, $T$, $\alpha$
  $\mathbf{x} \leftarrow \mathbf{x}_0$, $i \leftarrow 1$, $o \leftarrow 0$
  while $i \leq I$ do
    Create neighbor $\mathbf{x}'$ from $\mathbf{x}$
    if $f(\mathbf{x}') > f(\mathbf{x})$ then
      $\mathbf{x} \leftarrow \mathbf{x}'$
      $o \leftarrow o + 1$
    else if $U \sim \mathcal{U}[0,1] > e^{-\frac{f(\mathbf{x})-f(\mathbf{x}')}{T}}$ then
      $\mathbf{x} \leftarrow \mathbf{x}'$
      $d_n \leftarrow d_n \cdot 2.0$
    end if
    if $i \mod 100 = 0$ then
      if $o \geq 20$ then
        $d_n \leftarrow d_n \cdot 1.1$
      else
        $d_n \leftarrow d_n / 1.1$
      end if
      $o \leftarrow 0$
    end if
    $T \leftarrow \alpha \cdot T$, $i \leftarrow i + 1$
  end while
  return $\mathbf{x}$
```

Preliminary experiments show that the acceptance of a worse solution can change the optimization process. The adaptive control may result in an inappropriate neighborhood distance d_n. To prevent this, we implement a neighborhood distance boost b_{d_n} in case of accepting a worse solution. Hence, the optimization process can explore a larger area after taking a worse solution and thus it is less sensitive to changes in the solution space. We set this neighborhood distance boost to $b_{d_n} = 2.0$, which turned out to be reasonable in our experimental studies. The neighborhood distance d_n is doubled after accepting a worse solution.

Algorithm 1 shows the pseudocode of the optimization approach with the total number of iterations I and a random value $U \sim \mathcal{U}[0,1]$.

5 Experimental Results

Every optimization is run for $10,000$ iterations. As we use SA which is a heuristic optimization approach and also apply random initializations, we repeat every experiment 100 times and interpret the mean value and standard deviation. Additionally, we test the significance of the experimental results with a Wilcoxon signed rank-sum test [7].

5.1 Comparison of the Configurations

We use eight different configurations to test the capability of SA for the wind turbine placement problem with geographical constraints. The optimization runs are separated into two categories. First, runs that use the deterministic cooling schedule with a constant neighborhood distance and second, runs that use the adaptive neighborhood distance. In each category, we test the initial neighborhood distances d_n^- and d_n^+ and the starting temperatures T_0^h and T_0^l.

Table 1. Experimental comparison between constant and adaptive neighborhood distance control.

Algorithm	Mean $\pm$ Std	Max
	P in kW $\pm$ P in kW	P in kW
Without Optimization	12 923.81 $\pm$ 186.64	13 397.48
Constant Neighborhood Distance	13 233.57 $\pm$ 183.49	13 598.25
Adaptive Neighborhood Control	13 384.33 $\pm$ 121.81	13 654.61

Table 1 shows the comparison of the experimental results for the two different categories including various configurations. The values specify the average power production from the wind turbines in kilowatts. Both approaches are able to clearly improve the initial solution. The approach with adaptive neighborhood control performs significantly better than the approach with a constant neighborhood distance, confirmed by a Wilcoxon signed rank-sum test with a p-value $= 6.39 \cdot 10^{-30}$. It should also be noted, that the approach with the adaptive neighborhood control is able to reduce the standard deviation considerably, which means that the approach is more reliable. Also the best solution is created by the approach with adaptive neighborhood control.

In Table 2 concentrates on a detailed comparison between different configurations for start temperatures and neighborhood sizes. First, we can observe that the random initialization leads to slightly different initial values. But clearly confirmed by a Wilcoxon signed rank-sum test with a p-value $= 0.761$, there is no significant difference between the initial solutions. With a constant neighborhood distance, the approaches using T_0^l perform better than the approaches using

Table 2. Experimental comparison with various start temperatures and neighborhood sizes.

	Constant Neighborhood Distance			Adaptive Neighborhood Distance		
Algorithm	Mean ±	Std	Max	Mean ±	Std	Max
	P in kW ±	P in kW	P in kW	P in kW ±	P in kW	P in kW
Without optimization	12 925.07 ±	182.41	13 309.67	12 922.55 ±	190.78	13 397.48
SA with T_0^l & d_n^-	13 340.32 ±	123.44	13 598.25	13 401.58 ±	132.44	13 624.78
SA with T_0^h & d_n^-	13 173.90 ±	140.71	13 418.70	13 402.24 ±	101.37	13 633.63
SA with T_0^l & d_n^+	13 370.38 ±	111.74	13 565.51	13 395.63 ±	123.92	13 643.72
SA with T_0^h & d_n^+	13 049.68 ±	140.22	13 306.36	13 337.86 ±	115.20	13 654.61

T_0^h. This is probably due to the fact, that a higher temperature increases the number of accepted worse solutions. As we have a highly complex 44-dimensional solution space, too many accepted worse solutions can reduce the quality of the optimization. This effect is increased by the use of a larger neighborhood distance, as we can see comparing the results of SA with T_0^h & d_n^- and SA with T_0^h & d_n^+ using a constant neighborhood distance.

The adaptive neighborhood distance adapts itself to the solution space and can counteract this effect. But also here, we can observe that the worst mean value is achieved by SA with T_0^h & d_n^+. Interestingly, this configuration created the best overall solution. Although the higher acceptance rate of worse solutions decreases the optimization results in mean, there is a small chance that the algorithm chooses exactly the worst solutions which helps to leave local optima. A run with *good* choices is able to create the best solution with this configuration. This indicates that it might be promising future research to further analyze the highly complex solution space. This may allow better predictions, if accepting worse solutions may help to leave local optima. Between the approaches with the best mean values SA with T_0^l & d_n^-, SA with T_0^h & d_n^-, and SA with T_0^l & d_n^+ using the adaptive neighborhood distance is not significant difference. With carefully chosen parameters, the adaptive neighborhood control is able to operate well.

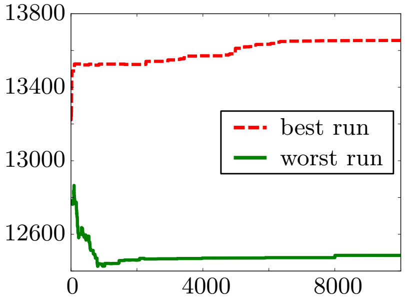
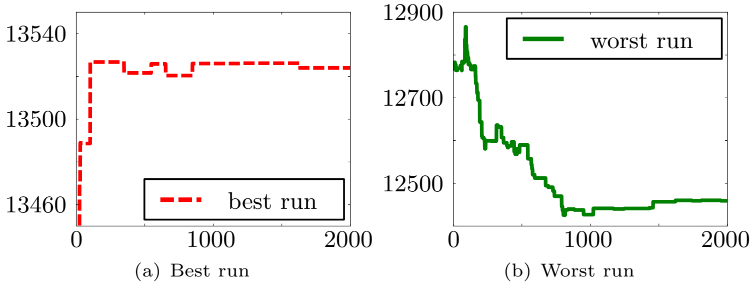


Fig. 3. Comparison of the best and worst optimization run.

5.2 Optimization Runs

Figure 3 shows the dynamic of the best with and the worst run. Both runs are achieved by SA with T_0^h & d_n^+ but the best run uses the adaptive neighborhood distance while the worst run uses the constant neighborhood distance. The important parts of the fitness development are enlarged and visualized in Figure 4. The y-scale in Figure 4(a) and 4(b) is different by factor 5 because the changes in the worst run are larger than the changes in the best run. In the best run, we can see the acceptance of worse solutions, but only with few deteriorations of the fitness function. This observation is different in case of the worst solution. Clearly worse solutions are accepted and the optimization process is unable to improve the fitness function to the starting level. This also explains, why it is possible that the best solution optimized with SA with T_0^h & d_n^+ is worse than the best initial solution, see Table 2.

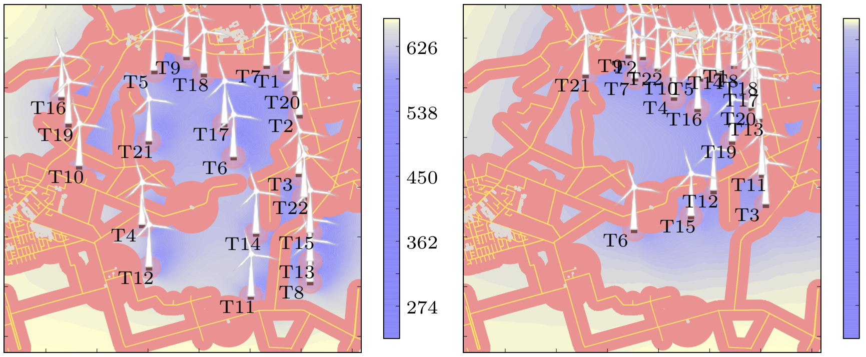


(a) Best run (b) Worst run

Fig. 4. Zoomed visualization of best and worst optimization run.

5.3 Placement Result

In the last experimental section, we show the placement results of the best solution, see Figure 5(a), and the worst solution, see Figure 5(b). The best solution maximizes the distances between the turbines w. r. t. the wind rose, see Figure 2. We can observe four lines of turbines. The largest line has been placed on the right, because most of the wind comes from direction south-west. This line of turbines does no cause wake effects for other turbines. The second line on the right is curved, also reducing the wake effects, e.g., for Turbine $T9$ and $T18$, as the wind comes from direction south-west. The geographical constraints are considered, e.g., on the left of the area with Turbines $T10$, $T16$, $T19$ placed in the free areas between the constraints. The placement of the worst solution is interesting. The turbines are pulled to the upper right corner. The distances between the turbines are small, so major wake effects reduce the power output. We can also see this behavior in other solutions created with a high temperature. Accepting to many worse solutions can lead to deadlock situations in the



(a) Best placement (b) Worst placement

Fig. 5. Placement results.

optimization process, where multiple turbines blockade each other, i.e., increasing the distance between two turbines will decrease the distance to a different turbine. An aggravation of this effect can be the geographical constraints. SA has serious difficulties to resolve deadlock situations. Again, this confirms the need to analyze in detail, if accepting worse solutions in highly complex solution spaces may help to leave local optima and avoiding deadlock situations.

6 Conclusions

Finding optimal turbine locations is important for the power output of wind turbines. The optimization process to improve the positions of turbines induces a highly complex solution space. SA is able to optimize the solutions, but the choice of appropriate parameters for neighborhood distance and temperature is important. Our approach using an adaptive neighborhood distance is more reliable than the variant with constant neighborhood distance and can significantly improve the results.

Our experiments show that an intelligent choice of accepted worse solutions can be a promising field for further research. Further, the integration of additional techniques from ES to SA may be an interesting research direction, e.g., the employment of a population like in parallel SA. The success of ES in turbine placement is an indicator that a smart combination of both fields could improve the results.

References

1. H. Beyer and H. Schwefel. Evolution strategies - A comprehensive introduction. *Natural Computing*, 1(1):3–52, 2002.
2. German Weather Service. COSMO-DE: numerical weather prediction model for Germany, 2012. http://tinyurl.com/dwd-cosmo-de.

3. M. M. Haklay and P. Weber. Openstreetmap: User-generated street maps. *IEEE Pervasive Computing*, 7(4):12–18, Oct. 2008.
4. N. Hansen. The CMA evolution strategy: a comparing review. In *Towards a new evolutionary computation. Advances in estimation of distribution algorithms*, pages 75–102. Springer, 2006.
5. J. F. Herbert-Acero, O. Probst, P.-E. Rethore, G. C. Larsen, and K. K. Castillo-Villar. A review of methodological approaches for the design and optimization of wind farms. *Energies*, 7(11):6930–7016, 2014.
6. L. Ingber. Simulated annealing: Practice versus theory. *Mathl. Comput. Modelling*, 18:29–57, 1993.
7. G. Kanji. *100 Statistical Tests*. SAGE Publications, London, 1993.
8. A. Kusiak and Z. Song. Design of wind farm layout for maximum wind energy capture. *Renewable Energy*, 35(3):685–694, 2010.
9. D. Lückehe, O. Kramer, and M. Weisensee. An evolutionary approach to geoplanning of renewable energies. In *28th International Conference on Informatics for Environmental Protection: ICT for Energy Effieciency (EnviroInfo)*, pages 501–508, 2014.
10. D. Lückehe, M. Wagner, and O. Kramer. On evolutionary approaches to wind turbine placement with geo-constraints. In *Genetic and Evolutionary Computation Conference, GECCO '15, Madrid, Spain, July 11-15, 2015*, pages 1223–1230, 2015.
11. A. K. Morales and C. V. Quezada. A universal eclectic genetic algorithm for constrained optimization. In *In: Proceedings 6th European Congress on Intelligent Techniques and Soft Computing (EUFIT)*, pages 518–522. Verlag Mainz, 1998.
12. F. Neumann and C. Witt. *Bioinspired Computation in Combinatorial Optimization: Algorithms and Their Computational Complexity*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010.
13. H. Neustadter. Method for evaluating wind turbine wake effects on wind farm performance. *Journal of Solar Energy Engineering*, pages 107–240, 1985.
14. Y. Nourani and B. Andresen. A comparison of simulated annealing cooling strategies. *Journal of Physics A: Mathematical and General*, 31:8373, 1998.
15. R. Rivas, J. Clausen, K. Hansen, and L. Jensen. Solving the turbine positioning problem for large offshore wind farms by simulated annealing. *Wind Engineering*, 33:287–297, 2009.
16. The Wind Power. Wind farms in Lower Saxony, Germany, 2015. http://tinyurl.com/parks-lower-saxony.
17. M. Wagner, K. Veeramachaneni, F. Neumann, and U.-M. O'Reilly. Optimizing the layout of 1000 wind turbines. In *European Wind Energy Association Annual Event*, 2011.
18. W. Weibull. A statistical distribution function of wide applicability. *Journal Applied Mechanics - Transactions of ASME*, 3(18):293–297, 1951.

Analyzing Geo-Spatial Trails: Visualizing and Comparing Movement Hypotheses

Martin Becker[1], Philipp Singer[2] Florian Lemmerich[2],
Andreas Hotho[1,3], Denis Helic[4], and Markus Strohmaier[2,5]

[1] University of Würzburg, Germany
{becker,hotho}@informatik.uni-wuerzburg.de
[2] GESIS, Cologne, Germany
{philipp.singer,florian.lemmerich,markus.strohmaier}@gesis.org
[3] L3S Research Center, Hannover, Germany
{philipp.singer,markus.strohmaier}@gesis.org
[4] Graz University of Technology, Graz, Austria
dhelic@tugraz.at
[5] University of Koblenz-Landau, Mainz, Germany

Abstract. Understanding the way people move through urban areas is an important problem that has implications for a range of societal challenges such as city planning, public transportation, or crime analysis. We present a visualization tool called VizTrails for exploring and understanding such human movement [2]. For the explored movement, we utilize the Bayesian approach HypTrails to formulate and compare different hypotheses explaining the underlying processes [1].

VizTrails features aggregated statistics of trails for geographic areas on a map, e.g., the number of users passing through or the locations commonly visited next. Amongst other tools, VizTrails also allows to visualize the results of SPARQL queries in order to relate the observed statistics with its geo-spatial context, e.g., considering a city's points of interest.

The insights from exploring the corresponding trajectories and features can be directly applied to modelling hypotheses about how the observed patterns can be explained. Then, the Bayesian approach HypTrails allows to compare the plausibility of such hypotheses with each other.

References

1. Becker, M., Singer, P., Lemmerich, F., Hotho, A., Helic, D., Strohmaier, M.: Photowalking the city: Comparing hypotheses about urban photo trails on flickr, http://dmir.org/pub/2015/photowalking-socinfo.pdf, under review
2. Becker, M., Singer, P., Lemmerich, F., Hotho, A., Helic, D., Strohmaier, M.: Viztrails: An information visualization tool for exploring geographic movement trajectories. In: Proceedings of the 26th ACM Conference on Hypertext & Social Media. pp. 319–320. HT '15, ACM, New York, NY, USA (2015)

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

On the Challenges of Real World Data in Predictive Maintenance Scenarios: A Railway Application

Sebastian Kauschke[1], Frederik Janssen[2] and Immanuel Schweizer[3]

[1] kauschke@ke.tu-darmstadt.de
[2] janssen@ke.tu-darmstadt.de
Knowledge Engineering Group & Telecooperation Group
Technische Universität Darmstadt, Germany
[3] schweizer@cs.tu-darmstadt.de
Telecooperation Group
Technische Universität Darmstadt, Germany

Abstract. Predictive maintenance is a challenging task, which aims at forecasting failure of a machine or one of its components. It allows companies to utilize just-in-time maintenance procedures instead of corrective or fixed-schedule ones. In order to achieve this goal, a complex and potentially error-prone process has to be completed successfully.

Based on a real-world failure prediction example originated in the railway domain, we discuss a summary of the required processing steps in order to create a sound prediction process.

Starting with the initial data acquisition and data fusion of three heterogeneous sources, the train diagnostic data, the workshop records and the failure report data, we identify and elaborate on the difficulties of finding a valid ground truth for the prediction of a compressor failure, caused by the integration of manually entered and potentially erroneous data.

In further steps we point out the challenges of processing event-based diagnostic data to create useful features in order to train a classifier for the prediction task. Finally, we give an outlook on the tasks we yet have to accomplish and summarize the work we have done.

1 Introduction

Predictive maintenance (PM) scenarios usually evolve around big machinery. This is mainly caused by those machines being both expensive and important for production processes of the company they are used in. A successful predictive maintenance process for a machine can help at preventing this, aid in planning for resources and material, and reduce maintenance cost and production downtime. In order to benefit from PM, a constant monitoring and recording of the machine status data is required.

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

Usually, historical data is used to train a model of either the standard behaviour of the machine, or - if enough example cases have been recorded - a model of the deviant behaviour right before the failure. These models are then used on live data to determine whether the machine is operating within standard parameters, or - in the second case - if the operating characteristics are similar to the failure scenario. If the model is trained correctly, it will give an alarm in due time. An overview of various PM methods is given in [4].

In our example case, the machines are cargo trains. These trains are pulling up to 3000 tons of cargo, so a lot of parts are prone to deterioration effects. In this paper we will present a workflow which will help us discover the necessary information needed to use a classifier to predict a specific failure case, the main air compressor failure. It is relevant for pressured air that is used in many pneumatic applications in a train, e.g. for braking or opening/closing doors and occurred quite often in the two year period of the historical data we received. This allows us to have a fair amount of example instances to train the classifier upon.

This paper is organized as follows. Section 2 will give an introduction to the problems and challenges. In Sect. 3 we will introduce the data sources in detail, followed by the challenge of finding a suitable ground truth therein (Sect. 4). From there on we will elaborate on how to extract meaningful features (Sect. 5) and propose a labelling process (Sect. 6) until we conclude our findings in Sect. 7 and give an outlook on further work.

2 Problem Definition and Challenges

We want to predict the failure of the main air compressor in a complex system containing many components. Therefore we will build a predictive model using a supervised learning approach on historical data, and apply it to new(er) data.

We were supplied by *DB Schenker Rail AG*[4] with data from three distinct datasets, from which we are going to extract the exact points in time when the failures have happened: (i) the diagnostic data recorded on the trains, (ii) the maintenance activity data gathered in the workshops and (iii) the failure report data with information from the hotline regarding the time and cause of the failure. Furthermore, we will create features that are descriptive and discriminative for the compressor failure, label the instances and train a classifier that will give a warning before the actual failure happens. We will have to face the following challenges:

1. Deal with large amount of diagnostic data that has unusual properties, i.e., inhomogeneous data distribution[5].
2. Extract a valid ground truth from three given datasets to find out exactly when the compressor failures happened by searching for indications for that type of failure and recognizing unnecessary workshop layovers (i.e. through premature indication of failure by the diagnostic system).

[4] www.rail.dbschenker.de

[5] Most often failures cases of machines are extremely rare. However, extracting instances that describe the regular operation of the machine comes more or less for free, in predictive maintenance, we have to deal with a very skewed distribution of the classes.

3. Recognize errors, incompleteness and imprecision in the data and derive methods to deal with them.
4. Create meaningful features that emphasize the underlying effects that indicate an upcoming failure.
5. Set up a labelling process for time-series data that enables classifiers to anticipate impending failures early enough to allow for corrective reactions.
6. Define an evaluation procedure that maximises the outcome with respect to given criteria in order to achieve optimal prediction.

3 Data

In this section, we will give a short introduction to the three information sources that were available, and make assumptions about the quality and the completeness of the data. In comparison to our effort in 2014 ([1]), we were supplied with more data (span of two years) from different databases, which enables us to gather more precise information and tackle a variety of new issues. With this information we will then filter the occurrences of the compressor failure scenario and determine the exact date of the failures (Sect. 4).

3.1 Diagnostics Data

The diagnostics data is recorded directly on the train in the form of a logfile. It shows all events that happened on the train, from the manual triggering of a switch by the train driver to warning and error messages issued by the trains many systems.

We have access to data from a complete model range of train's, the class *BR185*. This class was built in the 1990s, so it has internal processing and logging installed, but the storage capacity is rather limited. Back then, predictive maintenance was not anticipated, and therefore the logging processes were not engineered for this application. This becomes abundantly clear, when we consider the steps necessary to retrieve the logfiles from the train. It has to be done manually by a mechanic, each time the train is in the workshop for scheduled or irregular maintenance tasks.

In the two years we are using as historic data, around 21 Million data instances have been recorded in a fleet of 400 trains.

Diagnostic Codes and System Variables. As already mentioned, the diagnostic data is in the form of a logfile. It consists of separate messages, each having a code to identify its type. When we refer to a diagnostic code, we usually mean a message with this specific code. In total there are 6909 diagnostic codes that can occur.

To each diagnostic code, a set of system variables is attached. Those are encoded as Strings. Since the whole system is event based, the variables are not monitored periodically, but only recorded when a diagnostic message occurs. Which variables are encoded is depending on the diagnostic message that was recorded. This implies that some variables will be recorded rarely and sometimes not for hours or days. Overall, there are 2291 system variables available: simple booleans, numeric values like temperature or pressure and system states of certain components.

The event-based nature makes handling values as a time series difficult, for some sort of binning has to be done in order to achieve regularly spaced entities. In our case, a bin size of one day was used. Especially when relying on attributes that involve temperature or pressure measurements it is impossible to create a complete and fine-grained time-series of their values, because they appear too sparsely.

There is one speciality about these diagnostic messages: They have two timestamps, one for when the code occurred first (*coming*), and one for when it went away (*going*). This is designed for status reporting messages that last a certain time. Most codes only occur once, so they do not have a timespan, others can last up to days. For example there is code 8703 (*Battery Voltage On*) which is always recorded when the train has been switched on, and lasts until the train is switched off again.

Because of the two entries *coming* and *going*, there are usually two measurements of a variable for each diagnostic code entry. To handle these variables correctly, we separate them from the diagnostic messages and use both values as a separate measurement.

3.2 Failure Report Data

The Failure Report Data contains information when a failure has been reported by a train driver. In the driving cabin there is the Multi-Function-Display (MFD), which shows warnings or error messages. When a critical problem occurs, the information in conjunction with a problem description is shown to the driver. If he is unable to solve the issue, he will call a hotline. The hotline operator will try to help find an immediate solution. In the case of an unsolvable or safety-critical problem, the hotline operator will schedule a maintenance slot for the train and organise the towing if necessary.

The information recorded here is the date of the hotline call, the stated reason of the caller, and the information if the issue had an impact on the overall train schedule, i.e. the train was delayed more than ten minutes.

The start and end date of the train being in the workshop for repairs will be added to this database afterwards (manually). In general, the textual description given by the train driver and hotline operator are free text inputs and not consistent. The easiest possible way of finding out instances of a failure type is to search these text descriptions for certain keywords. In our case, 159 compressor failures on 95 trains were recorded.

Since this data is added manually and the dates are filled in afterwards, there is no guarantee that it is correct in any way.

3.3 Workshop Data

Compared to the Failure Report Data, the Workshop data is gathered in a much more controlled environment. Every maintenance activity is recorded here, from the replacement of consumables up to the more complex repair activities. Each entry has a date stamp as well as an exact id to each activity predefined in a database. All activities are divided into systems, as well as tagged with a price. The information, if a certain action was ”corrective” or a ”scheduled replacement” is also available.

The correct tracking of the maintenance records is necessary for invoices, so it is plausible to assume that these are handled more carefully than the failure report data.

They are manually entered into the system and it can not be guaranteed that they resemble the exact activities that have been applied to the train.

3.4 Quality issues

All of the three datasets have an issue in common: they may not be complete (missing entries, descriptions or dates) or are filled with false values (wrong dates, typos that make it hard to find a keyword). Whether this is caused by human error, negligence or processes that do not cover enough details is not important. In any case, these problems need to be dealt with in a way that renders each dataset as useful as possible.

4 Finding the ground truth

Our primary goal is to reconstruct the ground truth, in order to be able to create good labels for a classification process. In the special case at hand, this means finding out when a failure has happened exactly. In this section, we will show how much information is contained in each of the datasets described in Sect. 3 w.r.t. the ground truth, and combine them in such a way that the optimal result based on our current state of knowledge is received:

- the time of failure given as an approximation of the day,
- information on when the train was in the workshop afterwards, and
- a list of layovers that were unnecessary.

4.1 Pure Diagnostic Data

When we look at the information we can retrieve from the pure diagnostic data, it seems reasonable to think that we can extract the point in time when the train driver received the precise error message that led him to report the failure.

In reality, however, the messages displayed on the MFD can not be retrieved from the diagnostic data. They are generated from it with a certain internal programming logic. Unfortunately, as it remains unknown how this is done, the combinations of codes needed to display a certain message also is not accessible. Given the original documentation, we would be able to identify the causes, which could help find the exact reasons the train driver called the hotline, and also evaluate which of those messages occur before real failures and which before unnecessary layovers (see Sec. 4.4).

When we take all diagnostic messages that explicitly state a malfunction of the main compressor into account, a result as depicted in Fig. 1 can be achieved, each square indicating one failure day. Hence, for the train in our example a total of six failure indications are present.

Because the underlying reasons that caused this messages are unclear, we proceeded by taking further knowledge into account and refine these findings in subsequent steps.

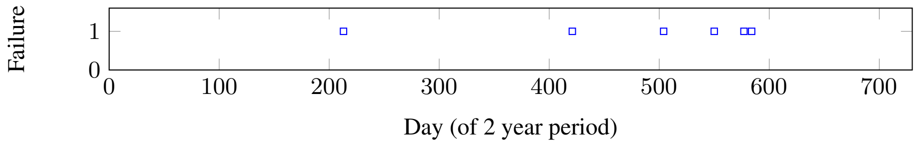


Fig. 1. Discovered failures for one exemplary train using only diagnostics data

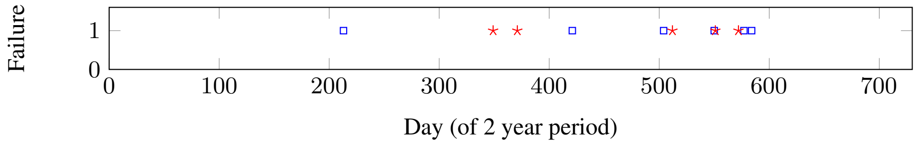


Fig. 2. Discovered failures using workshop data (red) compared to Fig. 1 (blue)

4.2 Workshop data

Using the workshop data, we can determine the point in time when a certain part has been replaced, and if the replacement was corrective or scheduled (mandatory). This greatly improved the identification of the true failures. Still, depending on how maintenance is handled, it only gives us a rough estimate of the point in time the failure actually took place. Note that maintenance procedures are not always carried out directly. Some types of failures require the train to be sent to a certain workshop, as not all workshops are equipped to handle every repair procedure. This may cause some days or even weeks in delay before the train finally is repaired. Therefore, the workshop date is not precise enough for a valid labelling.

A comparison of the extracted failure points can be seen in Fig. 2 depicted as red stars, showing a certain overlap with the findings from Fig. 1, but also completely unrelated entries. On average, red stars are 24 days away from blue boxes. If we only take pairs that are less than 21 days apart into account, the average distance is 6.5 days. But those are only 6 out of 10 instances, which leaves room for improvement and consequently leads us to the next step.

4.3 Failure Report Data

Utilizing failure report data, we are able to increase our understanding of when the actual breakdown has happened. The date of the reporting is noted here, and with high confidence it also states the correct day. We encountered some irregularities, for example the reporting date being behind the date the train was then brought into the workshop. We can still use this information to narrow down the exact day of the failure, but can not narrow it down to anything more fine grained than a day, because the precision of the timestamp that is recorded in the reporting system is not high enough. Therefore, we decided to take one day as the smallest unit a prediction can be done for. Since we expect to predict failures weeks in advance, this is not crucial. However, when failures have to be predicted that are appearing within minutes, the proposed method is not suitable any more.

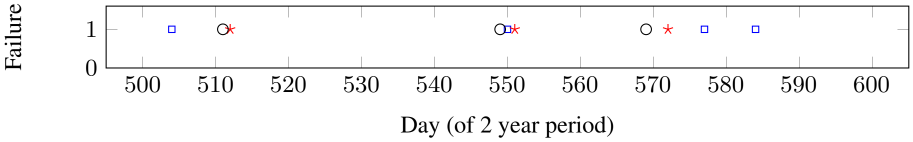


Fig. 3. Discovered failures using failure report data (black) compared to Figs. 1 (blue) and 2 (red)

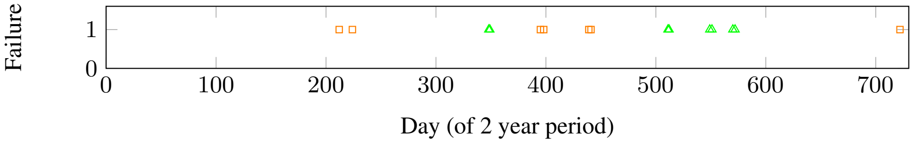


Fig. 4. Discovered failures (green) and unnecessary layovers (orange)

In Fig. 3 we now look at a smaller part of the timeline from day 500 to 620 (for better visibility). It is obvious that the failure report dates (black circles) are related to the workshop dates (red star), but not always to the diagnostic data (blue squares).

Therefore, we can conclude that only the combination of a failure report and a following repair is truly indicative of a failure. The diagnostic messages seem to indicate failures, but, surprisingly, after most of them the train is not affected negatively. Comparing the failure report dates with the repair dates an average distance of 1.6 days is yielded when events that are more than 21 days apart are discarded.

4.4 Unnecessary workshop layover

Unnecessary workshop layovers mostly happen because of the train drivers concern with safety, or them being overly cautious. As we were told by domain experts, the programming logic that drives the MFD's error and warning display is usually very sensitive, therefore generating a certain amount of false positives.

This may cause the train driver to trigger unnecessary maintenance actions. In the workshop the mechanics will then check the system, conclude that there is no failure and cancel the scheduled replacement. With the workshop data and the failure report data combined, we are able to differentiate the necessary from the unnecessary activities and exclude them from the pool of failures. This emphasizes the strong need for combining the different data sources by using expert knolwedge, as only then high-quality datasets can finally be built. In Fig. 4 the detected unnecessary layovers in comparison to the correct repairs are shown.

4.5 Missed and double repairs

Related to the unnecessary workshop activities are the missed repairs. Sometimes the train might arrive in the workshop with a given failure to check, and the repair crew may not be able to reproduce the symptoms, hence declaring this an unnecessary activity. A few days later the train might get submitted for the same reason again, and often only then the crew will actually repair or replace the component.

This effect has two implications, the first being that the time between those two layovers should not be used to train the model, because it may contain data where it is not certain if the failure is near or not. Second, it is also not clear whether the replacement that was made in the second attempt was actually well reasoned, or the maintenance crew decided to simply replace the part in order to eliminate the interference from happening again. These events are not documented as such, and we can only avoid negative influence on the training by removing the instances from the training set completely.

In the case of double repairs, we treat layovers caused by the same reasons that appear in a less than two weeks time as a single one, therefore assuming that the reasons to bring the train in were correct in the first place. Unfortunately, we can not prove if this assumption is always correct, but a discussion with the domain experts assured us that it is usually the case. With this 14 day range, the total number of compressor failure cases is reduced from 159 to 135.

5 Extracting meaningful features

In this section, we will describe the techniques we used to extract features from the given datasets. A thorough review of how to create features from various types of data can be found in [2]. We will give a short overview of the different types of attributes we extract and describe which features we create from them. As we have to do aggregation because of event based data, we chose the smallest possible bucket size of one day. As mentioned before (cf. Sect. 4.3), technical limitations apply. Furthermore, a failure should be predicted weeks in advance, and it is very likely that signs of deterioration are developed over a longer time, so one day seems to be appropriate.

5.1 Diagnostic messages and Status variables

As mentioned in Sect. 3.1, a diagnostic message has a pair of timestamps and values, one for *coming* and one for *going*, possibly spanning a certain duration. We use the values from one diagnostic code occurrence as source for multiple features. The information of a trains' runtime during one day is used to scale the attributes.

A status variable may have a certain amount of states, each defined by a number. For each of the states a feature is generated. The states behave like the diagnostic codes, the machine is in a certain state for a certain time span, therefore the features for both types are equal:

1. *Total duration of code/status*: All durations summed up for the whole day
2. *Frequency*: How often one diagnostic code/status occurs during one day
3. *Average duration*: Total duration divided by the frequency

These attributes cover the primary properties of the appearance of diagnostic codes and states. Other statistical values might be useful, e.g., variance of the average duration. It is planned to conduct further experiments including other statistics in the future, however, we are confident that these three statistics have the highest impact on the quality of the features.

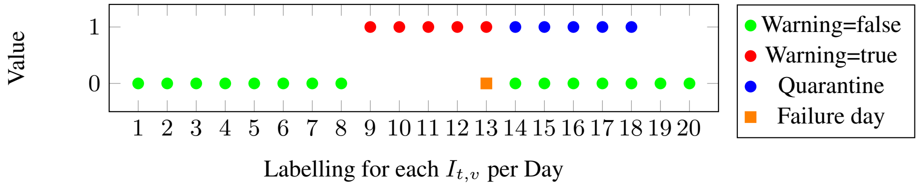


Fig. 5. The label $(warning = true|false)$ assigned to instances before and after the failure

5.2 Numeric values

Numeric values occur in a wide range of applications, for example the measurement of temperature values. For these variables we use standard statistical measurements:

1. *Average*: Arithmetic mean of all recorded values in one day
2. *Maximum*: Maximum recorded value in one day
3. *Minimum*: Minimum recorded value in one day
4. *Variance*: Variance of all recorded values in one day

These attributes cover important properties of numerical values, more complex ones may be evaluated in later experiments.

5.3 Time normalization

Since a train does not have the same runtime each day, we scale the time-based values that are absolute (duration, occurrences) to the total uptime per day, in order to increase comparability between days. As a result we achieve frequency (occurrences per hour) as well as average duration per hour.

6 Labelling

In this section we will describe the labelling process with emphasis on the preprocessing steps. We will describe why large amounts of the data were not used for training and evaluation because of inconsistencies and information gaps.

6.1 Aggregation

As proposed in related literature (e.g. [2, 5, 6]), we will use a sliding window approach to label the instances. The goal of this is to calculate not only a certain feature-vector A_t for a given day t, but instead calculate the trend leading towards this point in time. For this, we use linear regression with the window size v for each day t such as to create a vector $I_{t,v} = linreg(A_{t-v}, ..., A_{t-1}, A_t)$, in order to represent the behaviour of the system in the last v days. The gradient of the linear regression is then used as the attribute. Each of those vectors represents one instance, as can be seen in Fig. 5. The labels are then assigned as follows:

Step 1: Label all instances as $warning = false$

Step 2: For each failure on train B and on day S, label the instances $B_{S-w}...B_S$ as $warning = true$

The value w represents the "warning epoch". The optimal value of w will be determined experimentally, and depends on the specific type of failure. The optimal value for v will also be determined experimentally.

6.2 Quarantine area

Because of the nature of the sliding window, we need to assure that - right after a part has been repaired - we will not immediately create instances with $warning = false$. For example, given a window size of $v = 8$ and a failure/repair on day F: if we create $I_{(F+2),8}$ the window will date back to 6 days before the failure and incorporate the measurements from those days. The calculated features would be influenced by the behaviour before the maintenance. Therefore we introduce the quarantine interval, also of length v. All instances in this interval may be affected by the failure and have to be treated accordingly, in our case removed. The quarantine interval prevents instances that are influenced by the effects of the failure, but labelled as $warning = false$ (see Fig. 5).

6.3 Unnecessary layover area

In Sect. 4.4 we elaborated on how we detect unnecessary layovers. Apparently these result from values in the diagnostics system which caused it to issue a warning on the MFD. Thus, some sort of non-standard behaviour has been detected. Compared to our ground truth we can state that - although abnormal - the records do not correlate with the failure we are trying to detect. We do not want these to affect the training of the classifiers, so we create a buffer area around those dates. The buffer area affects all instances from $I_{t-v}...I_{t+v}$. The instances inside this area will not be used for training.

6.4 Removal of instances

As stated before, the diagnostic data we built the instances upon is not recorded continuously, but on an event-triggered basis. For example, data is not recorded when the train is switched off. To address this issue, the concept of *validity* was introduced. If no data was recorded on a given day, this day is regarded as *invalid*. The same applies, when no mileage was recorded on a day. It can happen that a train is switched on and records data, even when it does not actually drive. Most often this happens in situations where the train is moved to another rail, hence, we consider a mileage of less than 10 km per day as *invalid*, since driving less than 10 km definitely is no cargo delivery.

The last attribute that has an influence on the *validity* is the information, if a train was in the workshop at a given day. In workshop layovers, usually problem detection gear is attached and some diagnostic programs are executed, causing the train to emit more diagnostic messages than usual. In order to keep this artificially created information from influencing the process, workshop days are also handled as *invalid*.

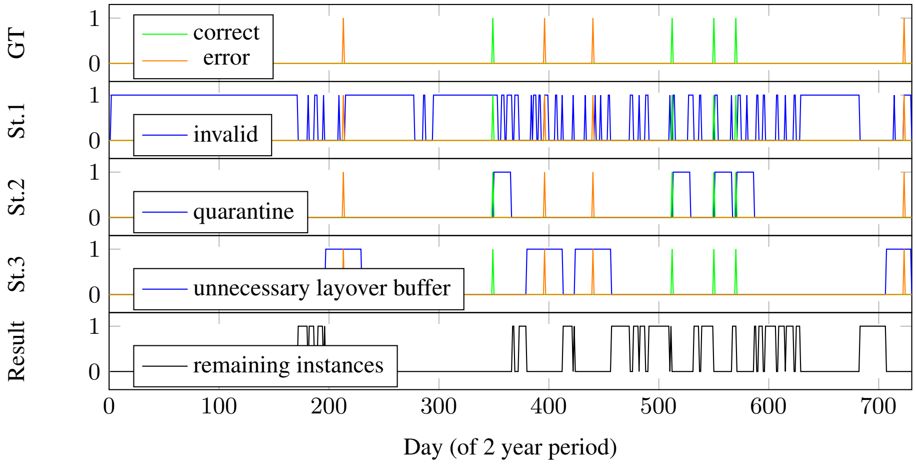


Fig. 6. Stepwise removal of invalid and unreliable values

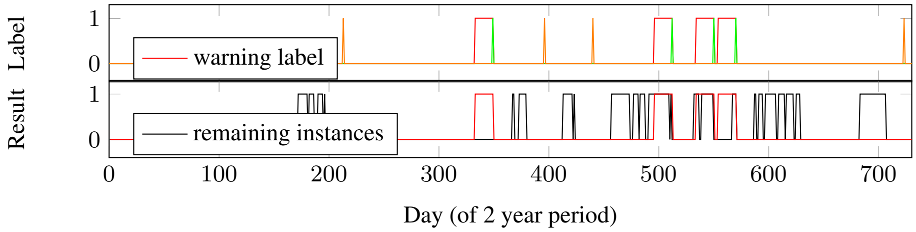


Fig. 7. Remaining instances compared to positive labels

In Fig. 6 the sequence of removal steps is displayed. In the first part of the figure, the ground-truth (GT) resulting from the process of Sect. 4 is shown. During a period of 2 years, we calculate the conditions for an instance for each day. In steps 1-3 those criteria are displayed, the status being true (1) when the condition applies.

The first step of the removal process eliminates all the invalid instances (St.1). In the second step, we remove all instances that appear in the quarantine period defined in Sect. 6.2. Finally, we remove data in the unnecessary layover buffer area from Sect. 6.3 in step 3. This is done in order to eliminate all negative training influences those instances might have.

At the end of this process we are left with a significantly smaller number of instances, as can be seen in the *Result* column of Fig. 6. In comparison to the actual labels we assign to those instances, we can see in Fig. 7 that a significant number of the “warning=true” instances was removed during the process. The quality of those remaining instances with respect to our labelling is highly increased when employing these steps, since potentially problematic, useless or erroneous instances are completely removed.

7 Conclusion

We have presented a preliminary process for converting a predictive maintenance scenario into a classification problem.

We dealt with domain-specific data, special characteristics of that data and presented preliminary solutions for the resulting challenges. We showed the necessity of incorporating domain expert knowledge in the process that proved to be successful for labelling the instances correctly. Several of the clean-up steps would not be possible without knowing the specific properties of the domain at hand.

Unfortunately, preliminary results in terms of classification accuracy were yet not promising. However, we are confident that with a further refinement of the presented procedures we will achieve better results soon. We will continue our work in the future with the following steps:

1. Discussion of validation methods and the implications they have (cf. [7])
2. Usage of sophisticated feature selection methods in order to improve classifier performance
3. Evaluation of classifier performance and parameter optimization
4. Solving the problem of of skewed class distribution
5. Evaluation of different approaches for converting predictive maintenance scenarios into classification problems (cf., e.g., [3])

References

1. Sebastian Kauschke, Immanuel Schweizer, Frederik Janssen, and Michael Fiebrig. Learning to predict component failures in trains. In *Proceedings of the LWA 2014 Workshops: KDML, IR and FGWM*, 2014.
2. Silvain Létourneau, Chunsheng Yang, Chris Drummond, Elizabeth Scarlett, Julio Valdes, and Marvin Zaluski. A domain independent data mining methodology for prognostics. In *Essential technologies for successful prognostics : proceedings of the 59th Meeting of the Society for Machinery Failure Prevention Technology*, 2005.
3. David Martinez-Rego, Oscar Fontenla-Romero, and Amparo Alonso-Betanzos. Power wind mill fault detection via one-class ny-svm vibration signal analysis. In *Proceedings of International Joint Conference on Neural Networks, San Jose, California, USA, July 31 - August 5, 2011*.
4. Ying Peng, Ming Dong, and MingJian Zuo. Current status of machine prognostics in condition-based maintenance: a review. *The International Journal of Advanced Manufacturing Technology*, 50(1-4):297–313, 2010.
5. Ruben Sipos, Dmitriy Fradkin, Fabian Moerchen, and Zhuang Wang. Log-based predictive maintenance. In *KDD '14 Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*.
6. Marvin Zaluski, Silvain Létourneau, Jeff Bird, and Chunsheng Yang. Developing data mining-based prognostic models for cf-18 aircraft. In *Journal of Engineering for Gas Turbines and Power*, volume 133, 2011.
7. Indre Zliobaite, Albert Bifet, Jesse Read, Bernhard Pfahringer, and Geoff Holmes. Evaluation methods and decision theory for classification of streaming data with temporal dependence. *Machine Learning*, 98(3):455–482, 2015.

k-Maxoids Clustering

Christian Bauckhage[1,2] and Rafet Sifa[2]

[1]B-IT, University of Bonn, Bonn, Germany
[2]Fraunhofer IAIS, Sankt Augustin, Germany
http://mmprec.iais.fraunhofer.de/bauckhage.html

Abstract. We explore the idea of clustering according to extremal rather than to central data points. To this end, we introduce the notion of the maxoid of a data set and present an algorithm for k-maxoids clustering which can be understood as a variant of classical k-means clustering. Exemplary results demonstrate that extremal cluster prototypes are more distinctive and hence more interpretable than central ones.

1 Introduction

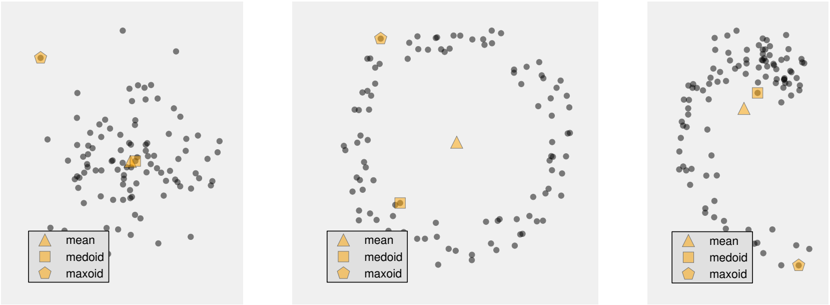
In this paper, we introduce a novel, prototype-based clustering algorithm. Since numerous such algorithms exist already [1, 12], our main goal is to fathom the potential of a paradigm that differs from existing prototype-based methods.

Whereas most prototype-based clustering algorithms produce prototypes that represent modes of a distribution of data (notable examples include the k-means procedure, the mean-shift algorithm, self organizing maps, or DBSCAN [7, 9, 11, 15]), our algorithm determines cluster prototypes that are extreme rather than central. They reside on the convex hull of their corresponding clusters and, in addition, are as far apart as possible.

The idea for this approach was motivated by research on efficient archetypal analysis, a matrix factorization technique that expresses a data set in terms of convex combinations of points on the data convex hull [5, 8, 13, 17]. The resulting representations are easily interpretable by human analysts [8, 19, 21], allow for clustering, and can facilitate classification. However, as their computation involves demanding optimization problems, the quest for more efficient methods and heuristics has become an active area of research [5, 6, 16, 18].

In the following, we first define the notion of the *maxoid* of a data set, prove that it will be furthest from the sample mean and necessarily coincides with a vertex of the data convex hull. We then introduce a simple and efficient clustering algorithm based on maxoids. It can be understood as a variant of the popular k-means procedure, however, whereas k-means determines cluster prototypes based on local information, our approach assumes a global view and selects the

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org



(a) Gaussian blob (b) ring (c) spiral

Fig. 1: Three data sets and their means, medoids, and maxoids.

prototype of a cluster with respect to those of other clusters. In experiments with synthetic and real world data, we illustrate the behavior of this algorithm and observe that it yields prototypes which are more distinct and hence more amenable to human interpretation than those produced by k-means.

2 Means, Medoids, and Maxoids

In this section, we briefly recall the concepts of the sample mean and sample medoid, introduce the idea of the sample maxoid, and review its characteristics.

Consider a finite set $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^{n} \subset \mathbb{R}^m$ of data points. The *sample mean*

$$\boldsymbol{\mu} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{x}_i. \tag{1}$$

is arguably the most popular summary statistic of such data. The closely related concept of the *sample medoid*, however, seems less well known. It is given by

$$\mathbf{m} = \operatorname*{argmin}_{\boldsymbol{x}_j} \frac{1}{n} \sum_{i=1}^{n} \left\| \boldsymbol{x}_j - \boldsymbol{x}_i \right\|^2 \tag{2}$$

and coincides with the data point $\boldsymbol{x}_j$ whose average distance to all other points is smallest which is to say that it is the data point closest to the mean [3, 14].

Yet, our focus in this paper is not on central tendencies but on extremal characteristics of a set of data. To make this notion precise, we introduce the idea of the *sample maxoid* and define

Definition 1. *The maxoid of a set* $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^{n} \subset \mathbb{R}^m$ *is given by*

$$\boldsymbol{m} = \operatorname*{argmax}_{\boldsymbol{x}_j} \frac{1}{n} \sum_{i=1}^{n} \left\| \boldsymbol{x}_j - \boldsymbol{x}_i \right\|^2. \tag{3}$$

Apparently, this definition reverses that of the sample medoid in that it replaces minimization by maximization. It is thus straightforward to prove

Lemma 1. *Given a set $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^n$ of real valued data vectors, let $\boldsymbol{\mu} = \frac{1}{n}\sum_i \boldsymbol{x}_i$ be the sample mean and $\|\cdot\|$ be the Euclidean norm. Then*

$$\frac{1}{n}\sum_i \left\|\boldsymbol{x}_j - \boldsymbol{x}_i\right\|^2 \geq \frac{1}{n}\sum_i \left\|\boldsymbol{x}_k - \boldsymbol{x}_i\right\|^2 \tag{4}$$

implies that

$$\left\|\boldsymbol{x}_j - \boldsymbol{\mu}\right\|^2 \geq \left\|\boldsymbol{x}_k - \boldsymbol{\mu}\right\|^2. \tag{5}$$

That is, the maxoid $\boldsymbol{m}$, i.e. the point $\boldsymbol{x}_j \in \mathcal{X}$ with the largest average distance to all other points in $\mathcal{X}$, is farthest from the sample mean $\boldsymbol{\mu}$.

Proof. Note that the left hand side of (4) can be written as

$$\frac{1}{n}\sum_i \left\|\boldsymbol{x}_j - \boldsymbol{x}_i\right\|^2 = \frac{1}{n}\sum_i \left\|(\boldsymbol{x}_j - \boldsymbol{\mu}) - (\boldsymbol{x}_i - \boldsymbol{\mu})\right\|^2.$$

Expanding the squared Euclidean distances in this sum, we have

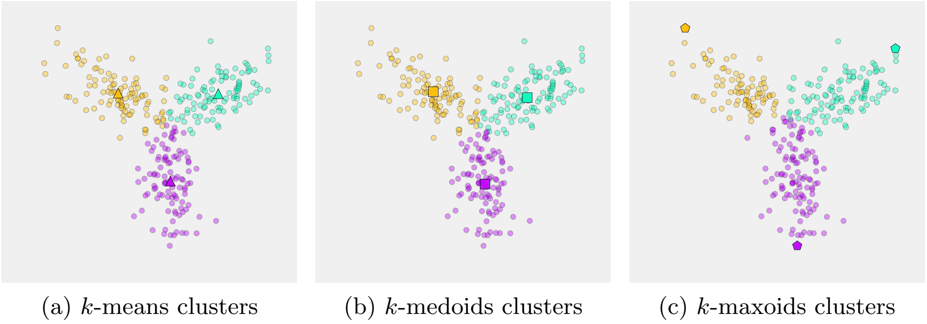
$$\begin{aligned}
&\frac{1}{n}\sum_i \Big(\left\|\boldsymbol{x}_j - \boldsymbol{\mu}\right\|^2 + \left\|\boldsymbol{x}_i - \boldsymbol{\mu}\right\|^2 - 2(\boldsymbol{x}_j - \boldsymbol{\mu})^T(\boldsymbol{x}_i - \boldsymbol{\mu})\Big) \\
&\quad = \left\|\boldsymbol{x}_j - \boldsymbol{\mu}\right\|^2 + \frac{1}{n}\sum_i \left\|\boldsymbol{x}_i - \boldsymbol{\mu}\right\|^2 - 2(\boldsymbol{x}_j - \boldsymbol{\mu})^T \frac{1}{n}\sum_i (\boldsymbol{x}_i - \boldsymbol{\mu}) \\
&\quad = \left\|\boldsymbol{x}_j - \boldsymbol{\mu}\right\|^2 + \frac{1}{n}\sum_i \left\|\boldsymbol{x}_i - \boldsymbol{\mu}\right\|^2 - 2(\boldsymbol{x}_j - \boldsymbol{\mu})^T(\boldsymbol{\mu} - \boldsymbol{\mu}) \\
&\quad = \left\|\boldsymbol{x}_j - \boldsymbol{\mu}\right\|^2 + \frac{1}{n}\sum_i \left\|\boldsymbol{x}_i - \boldsymbol{\mu}\right\|^2.
\end{aligned}$$

Since these arguments also apply to the right hand side of (4), the inequality in (4) can be cast as

$$\left\|\boldsymbol{x}_j - \boldsymbol{\mu}\right\|^2 + \frac{1}{n}\sum_i \left\|\boldsymbol{x}_i - \boldsymbol{\mu}\right\|^2 \geq \left\|\boldsymbol{x}_k - \boldsymbol{\mu}\right\|^2 + \frac{1}{n}\sum_i \left\|\boldsymbol{x}_i - \boldsymbol{\mu}\right\|^2$$

which is to say that $\left\|\boldsymbol{x}_j - \boldsymbol{\mu}\right\|^2 \geq \left\|\boldsymbol{x}_k - \boldsymbol{\mu}\right\|^2$. □

Given this result, it is easy to understand the behavior of the means, medoids, and maxoids in Fig. 1. In particular, we note a caveat for analysts working with centroid methods: the sample mean is always is located in the center of the data, yet, in cases where there is no clear mode, it is rather far from most data.The medoid may or may not be close to the mean but always coincides with a data point. The maxoid, too, always coincides with a data point but its behavior seems not to depend on whether or not there is a mode. In fact, we can prove the following



(a) k-means clusters (b) k-medoids clusters (c) k-maxoids clusters

Fig. 2: A simple data set consisting of three Gaussian blobs and results obtained from k-means, k-medoids, and k-maxoids clustering.

Lemma 2. *The maxoid of a finite set $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^n$ of real valued data vectors coincides with a vertex of the convex hull of $\mathcal{X}$.*

Proof. The maxoid of $\mathcal{X}$ is the maximizer of the convex function

$$f(\boldsymbol{x}) = \frac{1}{n} \sum_{\boldsymbol{x}_i \in \mathcal{X}} \left\| \boldsymbol{x} - \boldsymbol{x}_i \right\|^2. \tag{6}$$

The domain of f is given by the discrete set $\mathcal{X}$ which defines a polytope, that is, a convex set of finitely many vertices. By Jensen's inequality, the maximum of a convex function over a convex set is attained at a vertex. □

3 From k-Means Clustering to k-Maxoids Clustering

Having familiarized ourselves with means, medoids, and maxoids, we ever so briefly revisit k-means clustering and then present our idea for how to extend it towards k-maxoid clustering.

In the simplest setting, k-means clustering considers a set of n data points $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^n \subset \mathbb{R}^m$ and attempts to determine a set $\mathcal{C} = \{\mathcal{C}_\kappa\}_{\kappa=1}^k$ of k clusters where $\mathcal{C}_\kappa \subset \mathcal{X}$ such that data points within a cluster are similar. In order to assess similarity, the algorithm represents each cluster by its mean $\boldsymbol{\mu}_\kappa$ and assigns data point $\boldsymbol{x}_i$ to cluster $\mathcal{C}_\kappa$ if $\boldsymbol{\mu}_\kappa$ is the closest mean. This idea reduces clustering to the problem of finding appropriate means which can be formalized as solving

$$\underset{\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_k}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{\boldsymbol{x}_j \in \mathcal{C}_\kappa} \left\| \boldsymbol{x}_j - \boldsymbol{\mu}_\kappa \right\|^2. \tag{7}$$

Since this may prove surprisingly difficult [2], k-means clustering is typically realized using the following greedy optimization procedure:

1. initialize cluster means $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_k$
2. repeat until convergence
   (a) determine all clusters

$$\mathcal{C}_\kappa = \left\{ \boldsymbol{x}_i \;\middle|\; \left\lVert \boldsymbol{x}_i - \boldsymbol{\mu}_\kappa \right\rVert^2 \leq \left\lVert \boldsymbol{x}_i - \boldsymbol{\mu}_\lambda \right\rVert^2 \right\} \tag{8}$$

   (b) update all cluster means

$$\boldsymbol{\mu}_\kappa = \frac{1}{|\mathcal{C}_\kappa|} \sum_{\boldsymbol{x}_i \in \mathcal{C}_\kappa} \boldsymbol{x}_i \tag{9}$$

Looking at this procedure, its adaptation towards k-medoids clustering is obvious: we simply have to replace the computation of means by that of medoids and use cluster medoids instead of means in (8). The extension towards meaningful k-maxoids clustering is straightforward, too, but not quite as obvious.

Assuming that k data points have been randomly selected as initial maxoids, we may of course cluster the data with respect to their distance to the maxoids. This is again in direct analogy to (8). However, updating the maxoids only w.r.t. the data points in their corresponding clusters may fail to produce reasonable partitions of the data since initially selected maxoids may be close to each other so that one (or several) of them may dominate the others in the subsequent cluster assignment. Our idea is thus to update maxoids not only w.r.t. the data in their cluster but also w.r.t. to the maxoids. That is, for the update step, we propose to select the new maxoid of cluster C_κ as the data point in C_κ that is farthest from the maxoids in the other clusters. Formally, this idea amounts to solving the following constrained minimizing problem

$$\begin{aligned} \underset{\boldsymbol{m}_1,\ldots,\boldsymbol{m}_k}{\operatorname{argmin}} \; & \sum_{i=1}^{k} \sum_{\boldsymbol{x}_j \in \mathcal{C}_\kappa} \left\lVert \boldsymbol{x}_j - \boldsymbol{m}_\kappa \right\rVert^2 \\ \text{s.t.} \quad & \boldsymbol{m}_\kappa = \underset{\boldsymbol{x}_j \in \mathcal{C}_\kappa}{\operatorname{argmax}} \sum_{\lambda \neq \kappa} \left\lVert \boldsymbol{x}_j - \boldsymbol{m}_\lambda \right\rVert^2. \end{aligned} \tag{10}$$

which is easy to recognize as a variant of the problem in (7). The corresponding greedy optimization procedure is shown in algorithm 1.

Figure 2 shows how k-means, k-medoids, and k-maxoids clustering perform on a data set consisting of three blob-like components. Setting $k = 3$, all three methods reliably identify the latent structures in these data. Observable differences are miniscule and arguably negligible in practice.

However, an important question is how k-maxoids clustering will deal with situations where not all of the clusters contained in a data set are close to the data convex hull. To illustrate this problem and answer the question, Fig. 3 shows a set of 2D data consisting of five clusters where one of them is situated in between the others and does not contain any point on the the data convex hull. The figure illustrates how the updates in algorithm 1 cause five randomly selected maxoids to quickly move away from each other; in fact, in this example, the algorithm converged to a stable clustering within only four iterations.

Algorithm 1 k-maxoids clustering

```
Require: discrete set $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subset \mathbb{R}^m$ and parameter $k \in \mathbb{N}$
  initialize iteration counter $t \leftarrow 0$ and cluster maxoids $\boldsymbol{m}_1^{(0)}, \boldsymbol{m}_2^{(0)}, \ldots, \boldsymbol{m}_k^{(0)}$
  while not converged do
    determine all clusters
        $\mathcal{C}_\kappa^{(t)} = \left\{ \boldsymbol{x}_i \;\middle|\; \left\| \boldsymbol{x}_i - \boldsymbol{m}_\kappa^{(t)} \right\|^2 \leq \left\| \boldsymbol{x}_i - \boldsymbol{m}_\lambda^{(t)} \right\|^2 \right\}$
    update all cluster maxoids
        $\boldsymbol{m}_\kappa^{(t)} = \underset{\boldsymbol{x}_i \in \mathcal{C}_\kappa}{\operatorname{argmax}} \sum_{\lambda \neq \kappa} \left\| \boldsymbol{x}_i - \boldsymbol{m}_\lambda^{(t)} \right\|^2$
    increase iteration counter $t \leftarrow t + 1$
```

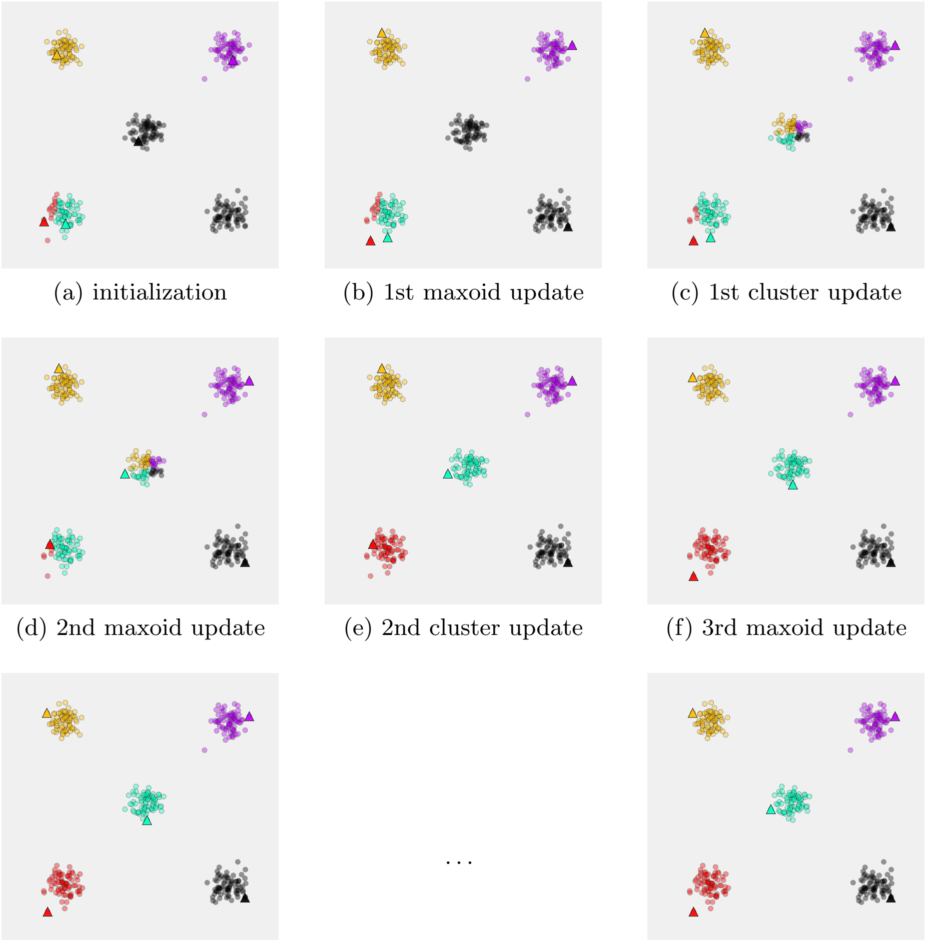
Although we observed this kind of efficiency in other experiments as well, the important point conveyed by this example is that the idea of clustering according to extremes works well even if there are substructures who cannot possibly be represented by prototypes on the data convex hull. This is, again, due to the fact that algorithm 1 inherently causes selected maxoids to be as far apart as possible.

In order to illustrate that extremal cluster prototypes may be more easily interpretable to human analysts than central ones, we conducted an experiment with the CBCL data set of face images[1] which contains 2429 portraits of people each of a resolution of 19×19 pixels. We turned each image into a 361 dimensional vector and applied k-means, k-medoids, and k-maxoids clustering where $k = 9$. The resulting prototypes in Fig. 4 clearly highlight the differences between the three approaches.

Figure 4(b) shows the prototypes returned by k-means clustering. They represent the average face of each cluster and, since each cluster contains several hundred images, are blurred to an extent that makes it difficult to assign distinctive characteristics to these prototypes. A similiar observation applies to the results produced by k-medoids clustering shown in Fig. 4(b). Here, the prototypes correspond to actual data points yet still appear rather similar. The prototypes in Fig. 4(c), on the other hand, resulted from k-maxoids clustering and show clearly distinguishable visual characteristics. Again each corresponding cluster contains several hundred images, yet their prototypes coincide with actual data points far from one another. It is rather easy to identify these faces as prototypes of pale or dark skinned people, of people wearing glasses, sporting mustaches, or having been photographed under varying illumination conditions.

In the next section, we will present and discuss an example of a real world application which further highlights this favorable property of clustering with extremes, namely the property of producing interpretable results.

[1] CBCL Face Database #1, MIT Center for Biological and Computation Learning, http://www.ai.mit.edu/projects/cbcl



(a) initialization

(b) 1st maxoid update

(c) 1st cluster update

(d) 2nd maxoid update

(e) 2nd cluster update

(f) 3rd maxoid update

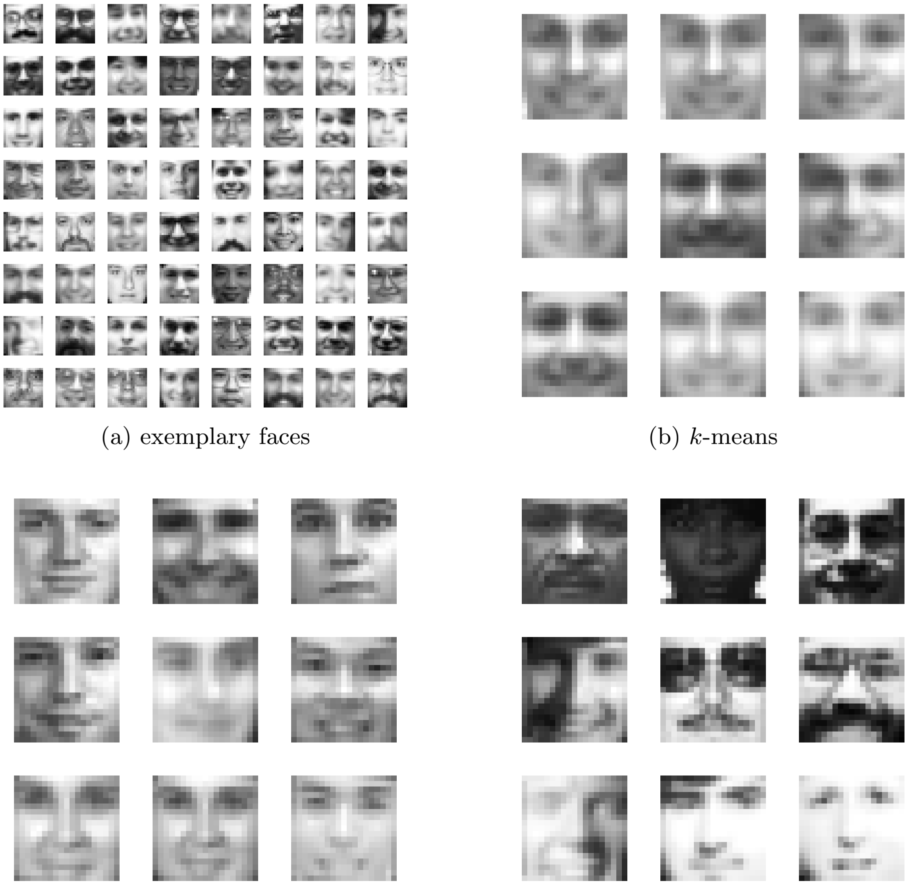
(g) 3rd cluster update

(h) final result

Fig. 3: Convergence behavior of k-maxoid clustering applied to a 2D data set containing five blob like clusters. Started with a random initialization of maxoids, the algorithm quickly moves them apart and converges within four iterations.

4 A Practical Application: Player Preference Profiling in the Online Game Battlefield 3

With the rise of mobile, console, and PC based games that operate on a so called freemium model, the problem of understanding how players interact with games has become a major aspect of the game development cycle [4, 10, 19, 20]. In this context, analytics provides actionable insights as to player behaviors and allows developers and publishers to quickly adjust their content with respect to the



(a) exemplary faces

(b) k-means

(c) k-medoids

(d) k-maxoids

Fig. 4: Clustering with $k = 9$ prototypes on the CBCL data base of face images. (a) examples of 64 face images in this data collection which illustrate the range of appearances. (b) k-means clustering produces cluster prototypes with are the means of the corresponding clusters. (c) k-medoids clustering determines prototypes that are actual data points closest to the local mean. (d) k-maxoids clustering yields cluster prototypes that are extremal data points and therefore appear more distinguishable to human observers than means or medoids.

outcomes they receive and thus to increase sales and monetization rates. In this section, we apply the k-maxoids algorithm to a game analytics task, namely the problem of deriving interpretable player profiles from analyzing vehicle usage data of Battlefield 3.

Battlefield 3 is a first person shooter military simulation game published by Electronic Arts in the Fall of 2011 as the eleventh installment in the Battlefield Series which, as of this writing, has a history of over 15 years. The game offers a single- and multi-player game-play experience where the former is composed of a storyline that allows the player to control variety of military characters in different real world locations and the latter puts the player in a imaginary war between the United States of America (USA) and the Russian Federation (RF). The combination of rich storyline, realistic graphics, flexibility through numerous manageable in-game components (such as vehicles and character customization), and the ability of supporting matches with large number of players has made the game one of the most played titles in its genre. Compared to its competitors, one of the most distinguishable features of the Battlefield series is the unique vehicle experience which allows the players to control air-, land-, water-based, and stationary vehicles.

The data we use in this study is a collection of vehicle usage logs of a random sample of 22,000 Battlefield 3 players which we obtained using a Web-based API for the Player Stats Network (https://p-stats.com/). In order to extract vehicle usage profiles from this data that can reveal how players interact with vehicle, we used accumulated activity statistics as to time-spent, number of character kills, and vehicle destroys made with the available 43 vehicles in the game.

Running the k-maxoids algorithm on our data set, we obtained interpretable player profiles that are semantically distinguishable from each other. In Fig. 5, we an example of k-maxoids cluster prototypes indicating different player preferences for vehicles in Battlefield 3. For each maxoid, we also indicate the percentage of players it represents.

Upon a closer look at the maxoids, we observe entirely distinct player profiles each representing different preferences for vehicles in the game. The first maxoid represents a pilot player behavior, that is, a behavior where players spend most of their vehicle time flying multirole fighter jets (F-18 and Su-35) and attack jets (A-10 Thunderbolt and Su 25) where the same vehicle ordering applies for both kills and destroys. Specifically for this particular maxoid the total flying time is 982 hours which is actually comparable to the average yearly flight time of experienced pilots in real life. It is important to note that the players in this cluster particularly chose to fly with the equivalent (*counterpart*) planes for American and Russian teams, which, during gameplay, creates a balance between two teams. In other words, the prototype indicates a habit of choosing a particular type of vehicle during a game. Indeed, behavioral patterns like this are also observed for the profiles represented by the other prototypes.

The second and the fourth most populated profiles represent a preference for land oriented vehicles. Again, counterpart-vehicle mastering is also observed where the players of the second and fourth profiles prefer to mostly use the counterpart heavy and light tanks the American M1 Abrahams and the Russian T-90 and the infantry fighting vehicles BMB and LAV respectively.

A more distinct tower defense behavior is observed for the third maxoid where players in the corresponding cluster spend 85% of their time on two counterpart

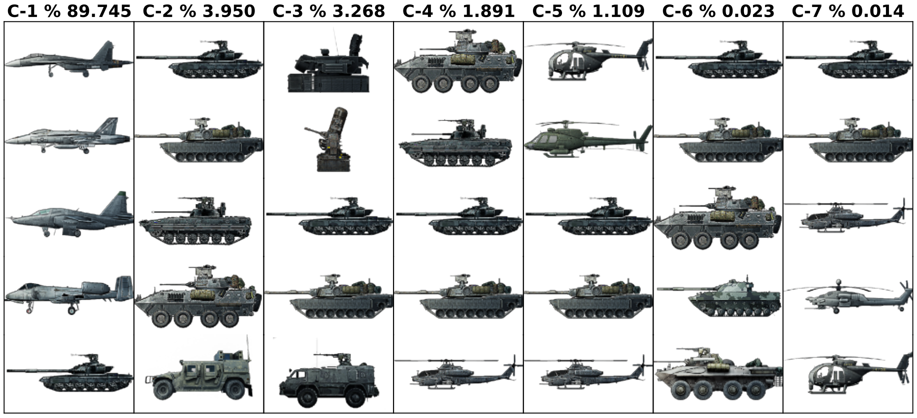


Fig. 5: Seven player vehicle usage profiles obtained from k-**maxoids** clustering. Each column visualizes a cluster prototype $\boldsymbol{m}_\kappa$ which indicates the most popular vehicles in the corresponding cluster of players. Note that prototypes are sorted according to the percentage of players they represent and that we show the top 5 elements of each prototype.

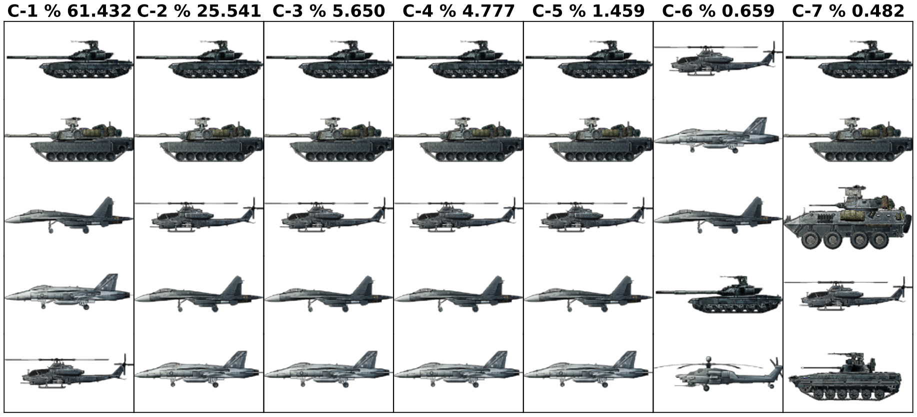


Fig. 6: Seven player vehicle usage profiles resulting from k-**means** clustering.

stationary anti-aircraft vehicles. Similar to the second profile we observe the use of two counterpart heavy tanks M1 Abrahams and T-90 for this profile as well.

The fifth profile, on the other hand, shows a helicopter pilot profile where the maxoid player in this cluster spends 68% of his time flying light fighter helicopters AH 6 and Z 11.

Finally, the least populated profile, represented by the right most maxoid in the figure, indicates that some players spend most of their time on the heavy counterpart tanks and the light fighting helicopters.

For comparison, we present results obtained from k-means clustering in Fig. 6. Similar to the face clustering example discussed in the previous section, we find k-means profiles to indicate general averages or mixed preferences for (counterpart) heavy tanks, jets, and light fighting helicopters where each of the vehicles in a prototype ranks high among the overall most frequently played vehicles in our data set. Hence, while k-means results represent average behavioral profiles (as already hinted at in [10]), the maxoids found by k-maxoids clustering represent more extreme or archetypal behavior that can help game developers to develop a deeper understanding of truly different types of user preferences and profiles that cannot be captured k-means clustering but are important w.r.t. balancing the game mechanics.

5 Conclusion

In this paper, we investigated the idea of clustering according to extremal rather average properties of data points. In particular, we defined the notion of the maxoid of a data set and presented an algorithm for k-maxoids clustering. This algorithm can be understood as a modification of the classical k-means procedure, where, in contrast to the classical approach, we determine cluster prototypes not only w.r.t. the data points in a cluster but w.r.t. the prototypes of other clusters. In a couple of didactic examples, we illustrated the behavior of this algorithm and then applied it to a practical problem in the area of game analytics.

In our didactic examples, as well as in our real world application, we observed our algorithm to produce cluster prototypes that are well distinguishable from one another and are thus more easily interpretable for human analysts. This property of clustering with extremes is particularly interesting for practitioners in game analytics for it allows them to quickly identify potentially imbalanced game mechanics.

In addition to these kinds of practical applications of k-maxoids clustering, we are currently investigating more theoretical aspects of its use. In particular, we examine its use as a mechanism to preselect archetypes for efficient archetypal analysis and hope to be able to report corresponding results soon.

References

1. Aggarwal, C., Reddy, C. (eds.): Data Clustering: Algorithms and Applications. Chapman & Hall/CRC (2013)
2. Aloise, D., Deshapande, A., Hansen, P., Popat, P.: NP-Hardness of Euclidean Sum-of-Squares Clustering. Machine Learning 75(2) (2009)
3. Bauckhage, C.: NumPy / SciPy Recipes for Data Science: k-Medoids Clustering. researchgate.net (Feb 2015), https://dx.doi.org/10.13140/2.1.4453.2009
4. Bauckhage, C., Kersting, K., Sifa, R., Thurau, C., Drachen, A., Canossa, A.: How Players Lose Interest in Playing a Game: An Empirical Study Based on Distributions of Total Playing Times. In: Proc. IEEE CIG (2012)

5. Bauckhage, C., Thurau, C.: Making Archetypal Analysis Practical. In: Denzler, J., Notni, G. (eds.) Pattern Recogntion. LNCS, vol. 5748. Springer (2009)
6. Chang, C.I., Wu, C.C., Liu, W.M., Ouyang, Y.C.: A New Growing Method for Simplex-Based Endmember Extraction Algorithm. IEEE Trans. on Geoscience and Remote Sensing 44(10) (2006)
7. Cheng, Y.: Mean Shift, Mode Seeking, and Clustering. IEEE Trans. on Pattern Analysis and Machine Intelligence 17(8) (1995)
8. Cutler, A., Breiman, L.: Archetypal Analysis. Technometrics 36(4) (1994)
9. Drachen, A., Canossa, A., Yannakakis, G.: Player Modeling using Self-Organization in Tomb Raider: Underworld. In: Proc. IEEE CIG (2009)
10. Drachen, A., Sifa, R., Bauckhage, C., Thurau, C.: Guns, Swords and Data: Clustering of Player Behavior in Computer Games in the Wild. In: Proc. IEEE CIG (2012)
11. Ester, M., H.-P.Kriegel, Sander, J., Xu, X.: A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In: Proc. ACM KDD (1996)
12. Estivill-Castro, V.: Why So Many Clustering Algorithms: A Position Paper. ACM SIGKDD Explorations Newsletter 4(1) (2002)
13. Eugster, M., Leisch, F.: Weighted and Robust Archetypal Analysis. Computational Statistics & Data Analysis 55(3) (2011)
14. Kaufman, L., Rousseeuv, P.: Clustering by Means of Medoids. In: Dodge, Y. (ed.) Statistical Data Analysis Based on the L1-Norm and Related Methods. Elsevier (1987)
15. MacQueen, J.: Some Methods for Classification and Analysis of Multivariate Observations. In: Proc. Berkeley Symp. on Mathematical Statistics and Probability (1967)
16. Miao, L., Qi, H.: Endmember Extraction From Highly Mixed Data Using Minimum Volume Constrained Nonnegative Matrix Factorization. IEEE Trans. on Geoscience and Remote Sensing 45(3) (2007)
17. Morup, M., Hansen, L.: Archetypal Analysis for Machine Learning and Data Mining. Neurocomputing 80 (2012)
18. Ostrouchov, G., Samatova, N.: On FastMap and the convex hull of multivariate data: toward fast and robust dimension reduction. IEEE Trans. on Pattern Analysis and Machine Intelligence 27(8) (2005)
19. Sifa, R., Bauckhage, C., Drachen, A.: Archetypal game recommender systems. In: Proc. LWA KDML (2014)
20. Sifa, R., Ojeda, C., Bauckhage, C.: User Churn Migration Analysis with DEDICOM. In: Proc. ACM RecSys (2015)
21. Thurau, C., Kersting, K., Wahabzada, M., Bauckhage, C.: Descriptive Matrix Factorization for Sustainability: Adopting the Principle of Opposites. Data Mining and Knowledge Discovery 24(2) (2012)

An Adaptive Grid Segmentation Algorithm for Mountain Silhouette Extraction from Images

Daniel Braun, Michael Singhof, and Stefan Conrad

Heinrich-Heine-Universität Düsseldorf, Institut für Informatik,
Universitätsstr. 1, 40225 Düsseldorf, Germany
{braun,singhof,conrad}@cs.uni-duesseldorf.de

Modern image sharing platforms such as instagram or flickr support an easy publication of photos to the internet, thus leading to great numbers of available photos. However, many of these images are not properly tagged so that there is no notion of what they are showing. For the example of mountain recognition, it is advisable to create reference silhouettes from digital elevation maps. Those are matched with the silhouette extracted from a given image in order to recognise the mountain. It is therefore necessary to obtain a very precise silhouette from the query image.

Our method utilises an adaptive grid segmentation algorithm that extracts the silhouette from a query image. This approach first overlays the image with a grid, with defined grid element spacing, and calculates, through a classification step, for every grid point a score for the probability to belong to the sky segment of the image. Afterwards, the algorithm segments the image with a seed growing algorithm, starting at the grid points with the highest score, which are additionally connected to an high score point in the top row of the image, due to the assumption, that the sky will be localised in the upper part of the image. Having the image binary segmented the algorithm extracts the transition between the two segments as initial silhouette.

The silhouette extracted by this approach may, however, include outliers that are either artefacts, for example as result of segmentation errors, or obstacles like trees in front of the mountain's silhouette. Our approach tries to find these outliers during an outlier detection step and afterwards to classify those into the mentioned classes. If an obstacle is detected, it is removed from the silhouette by replacing it by a straight. If an artefact is detected this gets reported to the segmentation step of our algorithm. There, with changed parameters, for the grid points located around the artefact, for edge detection, we try to find a better segmentation for the part of the silhouette the outlier appeared in. These steps are repeated until we end with a silhouette free of outliers and obstacles.

First experiments show that we reach a median average deviation of 1.51 pixels to the reference silhouettes. Hereby, we measure the deviation of each pixel of one silhouette extracted by our approach to the corresponding pixel of the reference silhouette.

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

An In-Database Rough Set Toolkit

Frank Beer and Ulrich Bühler

University of Applied Sciences Fulda
Leipziger Straße 123, 36037 Fulda, Germany
{frank.beer,u.buehler}@informatik.hs-fulda.de

Abstract. The Rough Set Theory is a common methodology to discover hidden patterns in data. Most software systems and libraries using methods of that theory originated in the mid 1990s and suffer from time-consuming operations or high communication costs. Today on the other hand there is a perceptible trend for in-database analytics allowing on-demand decision support. While data processing and predictive models remain in one system, data movement is eliminated and latency is reduced. In this paper we contribute to this trend by computing traditional rough sets solely inside relational databases. As such we leverage the efficient data structures and algorithms provided by that systems. Thereby we introduce a baseline framework for in-database mining supported by Rough Set Theory. Immediately, it can be utilized for common discovery tasks such as feature selection or reasoning under uncertainty and is applicable to most conventional databases as our experiments indicate.

Keywords: concept approximation, in-database analytics, knowledge discovery in databases, relational algebra, relational database systems, rough set theory

1 Introduction

Over the past decades, the huge quantities of data accumulating as a part of business operations or scientific research raised the necessity for managing and analyzing them effectively. As a result, Rough Set Theory (RST) became subject to these interdisciplinary areas as reliable instrument of extracting hidden knowledge from data. That trend is visible in the versatile existence of rough set-based software libraries and tools interfacing data from flat files [1–4]. The design of such libraries and tools, however, suffers when applying them to real-world data sets due to resource and time-consuming file operations. To overcome this technological drawback, researchers have made the effort to build more scalable rough set systems by utilizing relational databases which provide very efficient structures and algorithms designed to handle huge amounts of information [5–9].

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

However, the exploitation of database technology can be further extended. One can assess these relational systems to be expandable platforms capable of solving complex mining tasks independently. This design principle has been broadly established under the term *in-database analytics* [10]. It provides essential benefits, because hidden knowledge is stored in relational repositories predominantly either given through transactional data or warehouses. Thus, pattern extraction can be applied in a more data-centric fashion. As such, data transports to external mining frameworks are minimized and processing time can be reduced to a large extend. That given, one can observe database manufacturers continiously expand their engines for analytical models[1] such as association rule mining or data classification.

A full integration of rough sets inside relational systems is most favorable where both processing and data movement is costly. Unfortunately in-database processing and related applications are only covered partially in existing RST literature. Few practical attempts have been made to express the fundamental concept approximation based on existing database operations. In this paper we concentrate on that gap and present a concrete model to calculate rough sets inside relational databases. We redefine the traditional concept approximation and compute it by utilizing extended relational algebra. This model can be translated to various SQL dialects and thus enriches most conventional database systems. In line with ordinary RST our proposed model can be applied to common mining problems such as dimensionality reduction, pattern extraction or classification. Instrumenting SQL and its extensions enable us to cover further steps in the classic knowledge discovery process implicitly including selection and preprocessing. Combined, we obtain a baseline toolkit for in-database mining which relies on rough set methodology and database operations. It is natively applicable without the use of external software logic at low communication costs. Additionally, relational database engines have been significantly improved over the last decades, implementing both a high degree of parallelism for queries and physical operations based on hash algorithms which is a major factor for the efficiency of our model.

The remainder is structured as follows: First we present important aspects of the RST (Section 2). In Section 3 we review ideas and prototypes developed by other authors. Section 4 restructures the concept approximation. The resulting propositions are utilized to build a model based on database operations in Section 5. Then we briefly demonstrate how our model scales (Section 6). Based on that, we present future work (Section 7) and conclude in Section 8.

2 Rough Set Preliminaries

Proposed in the early 1980s by Zdzislaw Pawlak [11, 12], RST is a mathematical framework to analyze data under vagueness and uncertainty. In this section

[1] see Data Mining Extensions for Microsoft SQL Server: `https://msdn.microsoft.com/en-us/library/ms132058.aspx` (June, 2015) or Oracle Advanced Analytics: `http://oracle.com/technetwork/database/options/advanced-analytics` (June, 2015)

we outline principles of that theory: the basic data structures including the indiscernibility relation (Section 2.1) and the illustration of the concept approximation (Section 2.2).

2.1 Information Systems and Object Indiscernibility

Information in RST is structured in an Information System (IS) [13], i.e. a data table consisting of objects and attributes. Such an IS can thus be expressed in a tuple $\mathcal{A} = \langle \mathbb{U}, A \rangle$, where the universe of discourse $\mathbb{U} = \{x_1, ..., x_n\}, n \in \mathbb{N}$, is a set of objects characterized by the feature set $A = \{a_1, ..., a_m\}, m \in \mathbb{N}$, such that $a : \mathbb{U} \to V_a, \forall a \in A$, where V_a represents the value range of attribute a. An extension to an IS is the Decision System (DS). A DS even holds a set of attributes where some context-specific decision is represented. It consists of common condition features A and the decision attributes $d_i \in D$ with $d_i : \mathbb{U} \to V_{d_i}, 1 \leq i \leq |D|$ and $A \cap D = \emptyset$. A DS is denoted by $\mathcal{A}_D = \langle \mathbb{U}, A, D \rangle$. If we have for any $a \in A \cup D : a(x) = \perp$, i.e. a missing value, the underlying structure is called incomplete, otherwise we call it complete.

The indiscernibility relation classifies objects based on their characteristics. Formally, it is a parametrizable equivalence relation with respect to a specified attribute set and can be defined as follows: Let be an IS $\mathcal{A} = \langle \mathbb{U}, A \rangle$, $B \subseteq A$, then the indiscernibility relation $IND_{\mathcal{A}}(B) = \{(x, y) \in \mathbb{U}^2 \mid a(x) = a(y), \forall a \in B\}$ induces a partition $\mathbb{U}/IND_{\mathcal{A}}(B) = \{K_1, ..., K_p\}, p \in \mathbb{N}$ of disjoint equivalence classes over $\mathbb{U}$ with respect to B. Out of convenience we write IND_B or $\mathbb{U}/B$ to indicate the resulting partition.

2.2 Concept Approximation

To describe or predict an ordinary set of objects in the universe, RST provides an approximation of that target concept applying the indiscernibility relation. Let be $\mathcal{A} = \langle \mathbb{U}, A \rangle$, $B \subseteq A$ and a concept $X \subseteq \mathbb{U}$. Then, the B-lower approximation of the concept X can be specified through

$$\underline{X}_B = \bigcup \{K \in IND_B \mid K \subseteq X\} \tag{1}$$

while the B-upper approximation of X is defined as

$$\overline{X}_B = \bigcup \{K \in IND_B \mid K \cap X \neq \emptyset\} \ . \tag{2}$$

Traditionally, (1) and (2) can be expressed in a tuple $\langle \underline{X}_B, \overline{X}_B \rangle$, i.e. the rough set approximation of X with respect to the knowledge in B. In a rough set, we can assert objects in $\overline{X}_B$ to be fully or partly contained in X, while objects in $\underline{X}_B$ can be determined to be surely in the concept. Hence, there may be equivalence classes which describe X only in an uncertain fashion. This constitutes the B-boundary $\overline{\underline{X}}_B = \overline{X}_B - \underline{X}_B$. Depending on the characteristics of $\overline{\underline{X}}_B$ we get an indication of the roughness of $\langle \underline{X}_B, \overline{X}_B \rangle$. For $\overline{\underline{X}}_B = \emptyset$, we can classify X decisively, while for $\overline{\underline{X}}_B \neq \emptyset$, the information in B appears to be insufficient to describe X properly. The latter leads to an inconsistency in the data. The rest of objects not involved in $\langle \underline{X}_B, \overline{X}_B \rangle$ seems to be unimportant and thus can be disregarded. This set is called B-outside region and is the relative complement of $\overline{X}_B$ with respect to $\mathbb{U}$, i.e. $\mathbb{U} - \overline{X}_B$.

When we are focused in approximating all available concepts induced by the decision attributes, RST provides general notations consequently. Let be $\mathcal{A}_D = \langle \mathbb{U}, A, D \rangle$ and $B \subseteq A, E \subseteq D$, then all decision classes induced by IND_E can be expressed and analyzed by two sets, i.e. the B-positive region denoted as

$$POS_B(E) = \bigcup\nolimits_{X \in IND_E} \ \underline{X}_B \tag{3}$$

and the B-boundary region

$$BND_B(E) = \bigcup\nolimits_{X \in IND_E} \ \underline{\overline{X}}_B \ . \tag{4}$$

For $POS_B(E)$ and $BND_B(E)$ we get a complete indication whether the expressiveness of attributes B is sufficient in order to classify objects well in terms of the decisions given in E. Based on that, the concept approximation is suitable for a varity of data mining problems. Among others, it can be applied to quantify imprecision, rule induction or feature dependency analysis including core and reduct computation for dimensionality reduction [12].

3 Related Work

The amount of existing RST literature intersecting with databases theory increased continuously since the beginning. In this section we outline the most relevant concepts and systems introduced by other authors.

One of the first systems combining RST with database systems was introduced in [5]. The presented approach exploits database potentials only partially, because used SQL commands are embedded inside external programming logic. Porting this sort-based implementation for in-database applications implies the usage of procedural structures such as cursors, which is not favorable in processing enormous data. In [14], the authors modify relational algebra to calculate the concept approximation. Database internals need to be touched and hence a general employment is not given. The approaches in [6, 7] utilize efficient relational algebra for feature selection. The algorithms omit the usage of the concept approximation by other elaborated rough set properties. This factor limits the application to dimension reduction only. Sun et al. calculate rough sets based on extended equivalence matrices inside databases [9]. Once data is transformed into that matrix structure, the proposed methods apply but rely on procedural logic rather than scalable database operations. The work of Nguyen aims for a reduction of huge data loads in the knowledge discovery process [15]. Therefore appropriate methods are introduced using simpler SQL queries to minimize traffic in client-server architectures. The software design follows to the one in [5]. In [8], Chan transforms RST into a multiset decision table which allows to calculate the concept approximation with database queries. The initial construction of such a data table relies on the execution of dynamic queries, helper tables and row-by-row updates as stated in [16] and thus depends on inefficient preprocessing. The work of Naouali et al. implements α-RST in data warehouse environments [17]. The algorithm relies on iterative processing and insert commands to determine the final classification. Details about its efficiency are not presented. Another model is known as rough relational database [18]. These systems base on multi-valued relations designed to query data under uncertainty. Over the years, specific operations and properties of this theoretic model have been further extended. The authors in [19] try to port the rough relational data model to mature database systems. Details of migrating its algebra are not reported. Infobright is another database system that focuses on fast data processing towards ad-hoc querying [20]. This is achieve by a novel data retrieval strategy based on compression and inspired by RST. Data is organized underneath the knowledge grid. It is used to get estimated query results rather than seeking costly information from disk, which is valid to some domain of interest.

Most discussed approaches utilize inefficient procedural structures, external programs or leverage relational operations for very specific subjects. In contrast, we make use of existing, reliable and highly optimized database operations to compute the concept approximation not employing further procedural mechanisms. With this, we stretch the applicability of independent databases to a broader range of rough set mining problems.

4 Redefining the Concept Approximation

This section points out the formal ideas of transforming Pawlak's concept approximation to relational database systems by introducing a mapping of (1) and (2) to rewritten set-oriented expressions. Those propositions can then be applied to database algebra easily and enable us to transport both, the positive region and the boundary region in addition. We also show that these redefinings are no extensions to the traditional model, but equivalent terms.

Explained in Section 2.2, a rough set $\langle \underline{X}_B, \overline{X}_B \rangle$ can typically be extracted from a concept $X \subseteq \mathbb{U}$ of an IS $\mathcal{A} = \langle \mathbb{U}, A \rangle$ on a specific attribute set $B \subseteq A$, while the classification of each object is based on the induced partition $\mathbb{U}/B$. At this point, we make use of $X/B := X/IND_{\mathcal{A}}(B) = \{H_1, ..., H_q\}, q \in \mathbb{N}$, restructuring the concept approximation of X. Thus, we can deduce two relationships between classes $H \in X/B$ and $K \in \mathbb{U}/B$: $H \cap K \neq \emptyset$, $H \cap K = \emptyset$. This basic idea leads to two propositions, which we discuss in the remainder of this section:

$$\underline{X}_B = \bigcup \{H \in \mathbb{U}/B \mid H \in X/B\} \tag{5}$$

Proof. Considering the classes $H \in X/B$, the following two cases are of interest to form the B-lower approximation: (a) $\exists K \in \mathbb{U}/B : K = H \subseteq X$ and (b) $\exists K \in \mathbb{U}/B : K \neq H$ and $K \cap H \neq \emptyset$. Case (b) implies $\exists z \in K : z \notin X$ and thus $K \nsubseteq X$. As a result, only classes $K = H$ are relevant. Likewise, (1) only contains objects of classes $K \in \mathbb{U}/B$, where $K \subseteq X$. We consider X/B that induces classes $H \in \mathbb{U}/B$ and $H' \notin \mathbb{U}/B$, because $X \subseteq \mathbb{U}$. Combined, we immediately get to (5). □

$$\overline{X}_B = \bigcup \{K \in \mathbb{U}/B \mid \exists H \in X/B : H \subseteq K\} \tag{6}$$

Proof. On the one hand, the partition X/B can only produce equivalence classes $H, H' \subseteq X$ which satisfy $H \in \mathbb{U}/B$ and $H' \notin \mathbb{U}/B$. Obviously, those H are members of the B-lower approximation, whereas each class H' has a matching partner class K with $H' \subset K \in \mathbb{U}/B$ which build the B-boundary approximation. With these classes H, K, we directly receive: $\overline{X}_B = \underline{X}_B \cup \underline{\overline{X}}_B$. On the other hand, $\overline{X}_B$ holds objects of classes $K \in \mathbb{U}/B$ with $K \cap X \neq \emptyset$ (see (2)), i.e. each class $K \in X/B$ and $K \supset H \in X/B$. This is proposed by (6). □

Up to this point, the B-boundary approximation and the B-outside region remain untouched for further restructuring since both sets build on the B-lower and B-upper approximation. They have the same validity to the propositions in (5) and (6) as to the classical rough set model.

5 Combining RST and Database Systems

5.1 Information Systems and Database Tables

The IS is a specific way to organize data, similar to a data table in relational database terms. But there are essential differences in their scientific scopes [13]. While an IS is used to discover patterns in a snapshot fashion, the philosophy of databases concerns with long term data storing and retrieval respectively [21].

However, we try to overcome these gaps by simply assembling an IS or DS to the relational database domain considering the following: Let be $\mathcal{A}_D = \langle \mathbb{U}, A, D \rangle$ with the universe $\mathbb{U} = \{x_1, ..., x_n\}, n \in \mathbb{N}$, the features $A = \{a_1, ..., a_m\}, m \in \mathbb{N}$ and the decision $D = \{d_1, ..., d_p\}, p \in \mathbb{N}$, then we use the traditional notation of a $(m+p)$-ary database relation $R \subseteq V_{a_1} \times ... \times V_{a_m} \times V_{d_1} \times ... \times V_{d_p}$, where V_{a_i} and V_{d_j} are the attribute domains of $a_i, d_j, 1 \leq i, j, \leq m, p$.

In database theory, the order of attributes in a relation schema has significance to both semantics and operations. With this we simplify the employment of attributes to finite sets and write $A = \{a_1, ..., a_q\}, q \in \mathbb{N}$ for the ordered appearance in relation R. We notate R_A as shortform or R_{A+D} to identify a decision table. Furthermore modern databases permits duplicated tuples within its relational structure. We adopt this rudiment with practical relevance and designate these types of relations as *database relation* or *data table* respectively.

5.2 Indiscernibility and Relational Operations

Inspired by [5–7], we make use of extended relational algebra to calculate the partition of the indiscernibility relation. Using the *projection* operation $\pi_B(R_A)$ allows to project tuples $t \in R_A$ to a specified feature subset $B \subseteq A$ while eliminating duplicates. Thus, we get each class represented by a proxy tuple with schema B. A column reduction without duplicate elimination is indicated by $\pi^+_B(R_A)$. Additionally, we introduce the *selection* operation $\sigma_\phi(R_A)$ with filter property ϕ and output schema A. Given a geometric repository R_{A+D} (see Figure 1), we may query objects $x \in R_{A+D}$ that are colored *red* by $\sigma_{x.color=red}(R_{A+D})$.

Most relational database systems provide an extension to $\pi_B(R_A)$, i.e. the *grouping* operator γ. It groups tuples of a relation R_A if they share identical values entirely over an specified attribute set $G \subseteq A$, i.e. the grouping attributes.

Each group is only represented once in the resulting relation through a proxy tuple (see π-operator). In addition, γ can be enriched with aggregation functions $f_1, ..., f_n$ that may be applied to each group during the grouping phase. Generally, this operation can be defined by $\gamma_{F;G;B}(R_A)$, where B are the output features with $B \subseteq G \subseteq A$ and $F = \{f_1, ..., f_n\}, n \in \mathbb{N}_0$. For our purpose we simply count the number of members in each single group (class) of R_A, i.e. the cardinality expressed by the aggregate *count*$(*)$, and include it as new feature. Consolidated, we make use of the following notation

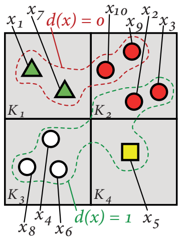
$$\mathcal{I}_B^G(R_A) := \rho_{card \leftarrow count(*)}(\gamma_{\{count(*)\};G;B}(R_A)) \tag{7}$$

where $\rho_{b \leftarrow a}(R_A)$ is the renaming operation of an arbitrary attribute $a \in A$ to its new name b in table R_A. Then $\mathcal{I}_B^B(R_A)$ is supposed to be noted as our compressed multiset representation of a given database table R_A considering feature set $B \subseteq A$. An illustration of this composed operation and its parallels to the RST is depicted in Figure 1 with $A = \{shape, color\}$, $D = \{d\}$ and $B = A$.

R_{A+D}

-	*shape*	*color*	*d*
x_1	triangle	green	0
x_2	circle	red	1
x_3	circle	red	1
x_4	circle	white	1
x_5	square	yellow	1
x_6	circle	white	1
x_7	triangle	green	0
x_8	circle	white	1
x_9	circle	red	0
x_{10}	circle	red	0

⇔



⇔

$\mathcal{I}_B^B(R_{A+D})$

card	*shape*	*color*
2	triangle	green
4	circle	red
3	circle	white
1	square	yellow

⇔

$\mathcal{I}_B^{B+D}(R_{A+D})$

card	*shape*	*color*
2	triangle	green
2	circle	red
2	circle	red
3	circle	white
1	square	yellow

Fig. 1. Mapping the object indiscernibility to relational data tables

5.3 Mapping the Concept Approximation

In practice, the extraction process of a single target concept may vary dependent on domain and underlying data model. In most cases an ordinary target concept can be modelled through decision attributes, i.e. a decision table. However there might be domains of interest, where the concept is located outside the original data table. Especially, this is the case in highly normalized environments. We support both approaches within the boundaries of the relational model. As simplification we assume the target concept C_A and the original data collection R_A to be given through either adequate relational operations or their native existence in a relation where C_A is a subset of R_A.

Taking this and the previous sections into account, we are now able to demonstrate the classical concept approximation in terms of relational algebra and its extensions: Let be R_A representing the universe and C_A our target concept to be examined with the feature subset $B \subseteq A$, then the B-lower approximation of the concept can be expressed by

$$\mathcal{L}_B(R_A, C_A) := \mathcal{I}_B^B(C_A) \cap \mathcal{I}_B^B(R_A)\ . \tag{8}$$

Initially, (8) establishes the partition for C_A and R_A independently. The intersection then only holds those kinds of equivalence classes included with their full cardinality in both induced partitions, i.e. the B-lower approximation in terms of the RST (see (5)). The B-upper approxiation contains all equivalence classes associated with the target concept. Thus, we simply can extract one representative of these classes from the induced partition of C_A applying the information in B. However, this information is not sufficient to get the correct cardinality of those classes involved. Hence we must consider the data space of R_A in order to find the number of all equivalences. That methodology can be expressed through

$$\mathcal{U}_B(R_A, C_A) := \pi_B(C_A) \bowtie \mathcal{I}_B^B(R_A) \tag{9}$$

whereas $\bowtie$ is the natural join operator, assembling two data tables S_G, T_H to a new relation R such that $s.b = t.b$ for all tuples $s \in S_G, t \in T_H$ and attributes $b \in G \cap H$. Note, R consists of all attributes in G, H, where overlapping attributes are shown only once. As a result, we get all equivalence classes with their cardinality, involved in the B-upper approximation (see (6)). Classically, the B-boundary consists of objects located in the set-difference of B-upper and B-lower approximation. Because of the structural unity of $\mathcal{L}_B(R_A, C_A)$ and $\mathcal{U}_B(R_A, C_A)$, it can be expressed by

$$\mathcal{B}_B(R_A, C_A) := \mathcal{U}_B(R_A, C_A) - \mathcal{L}_B(R_A, C_A)\ . \tag{10}$$

Equivalence classes outside the concept approximation can be found when searching for tuples not included in the B-upper approximation. With the support of both $\mathcal{I}_B^B(R_A)$ and $\mathcal{U}_B(R_A, C_A)$, we therefore get the B-outside region

$$\mathcal{O}_B(R_A, C_A) := \mathcal{I}_B^B(R_A) - \mathcal{U}_B(R_A, C_A)\ . \tag{11}$$

In order to present an equivalent relational mapping of (3) and (4), we first have to look at a methodology that allows us to query each target concept separately. Within a decision table R_{A+D}, let us assume the partition induced by the information in $E \subseteq D$ consists of n decision class. For each of these classes we can find an appropriate condition $\phi_i, 1 \leq i \leq n$ that assists in extracting the associate tuples $t \in R_{A+D}$ belonging to each concept $C_A^{\phi_i}$. One can simply think of a walk through $\pi_E(R_{A+D})$. In the i-th iteration we fetch the decision values, say $v_1, ..., v_m$, for the corresponding features in $E = \{d_1, ...d_m\}, m \in \mathbb{N}$ and build $\phi_i = \bigwedge_{1 \leq j \leq m} t.d_j = v_j$. Thus, we have access to each decision class $C_A^{\phi_i} = \pi_A^+(\sigma_{\phi_i}(R_{A+D}))$ produced by E. With this idea in mind and supported by (8) we are now able to introduce the B-positive region: In a decision table R_{A+D} and $B \subseteq A, E \subseteq D$, the B-positive region is the union of all B-lower approximations induced by the attributes in E. Those concepts can be retrieved by $C_A^{\phi_i}, 1 \leq i \leq n$ where n is the cardinality of $\pi_E(R_{A+D})$. As a consequence we get to

$$\mathcal{L}_B(R_{A+D}, C_A^{\phi_1}) \cup ... \cup \mathcal{L}_B(R_{A+D}, C_A^{\phi_n}) \tag{12}$$

which can be rewritten as

$$\bigcup_{i=1,...,n} \mathcal{I}_B^B(C_A^{\phi_i}) \cap \mathcal{I}_B^B(R_{A+D}) \tag{13}$$

such that we finally have the B-positive region in relational terms defined over a decision table

$$\mathcal{L}_B^E(R_{A+D}) := \pi_{B'}(\mathcal{I}_B^{B+E}(R_{A+D})) \cap \mathcal{I}_B^B(R_{A+D}) \tag{14}$$

with $B' = \{card, b_1, ..., b_k\}, b_j \in B, 1 \leq j \leq k$. Likewise, the B-boundary region consists of tuples in $\mathcal{U}_B(R_{A+D}, C_A^{\phi_1}) \cup ... \cup \mathcal{U}_B(R_{A+D}, C_A^{\phi_n})$ but not in $\mathcal{L}_B^E(R_{A+D})$, where $C_A^{\phi_i}, 1 \leq i \leq n \in \mathbb{N}$ are the separated target concepts induced by E. Hence, we can query these through

$$\bigcup_{i=1,...,n} \mathcal{U}_B(R_{A+D}, C_A^{\phi_i}) - \mathcal{L}_B^E(R_{A+D}) \tag{15}$$

which is equivalent to

$$\mathcal{I}_B^B(R_{A+D}) - (\pi_{B'}(\mathcal{I}_B^{B+E}(R_{A+D})) \cap \mathcal{I}_B^B(R_{A+D})) \tag{16}$$

in a complete decision table and immediately come to our definition of the B-boundary region

$$\mathcal{B}_B^E(R_{A+D}) := \mathcal{I}_B^B(R_{A+D}) - \pi_{B'}(\mathcal{I}_B^{B+E}(R_{A+D})) \tag{17}$$

where $B' = \{card, b_1, ..., b_k\}, b_j \in B, 1 \leq j \leq k$. Denote, we directly deduced $\mathcal{L}_B^E(R_{A+D})$ and $\mathcal{B}_B^E(R_{A+D})$ from (8) and (9). For practical reasons, further simplification can be applied by removing the π-operator. One may verify, this change still preserves the exact same result set, because both expressions rely on $\mathcal{I}_B^B(R_{A+D})$ initially.

6 Experimental Results

In this section, we present the initial experimental results applying the concluded expressions from Section 5.3 to some well-known data sets and two database systems. The objective of this experiment is to demonstrate the performance of our model in a conservative test environment not utilizing major optimization steps such as the application of indices, table partitioning or compression strategies. Thus, we get an impression of how the model behaves natively in different databases. We chose PostgreSQL (PSQL) and Microsoft SQL Server (MSSQL) as two prominent engines providing us with the required relational operations. The hardware profile[2] represents a standalone server environment commonly used in small and medium-sized organizations. Most of our benchmark data sets are extracted from [22] varying in data types and distribution. Table 1 states further details. Both, PSQL and MSSQL provide similar query plans based on hash

[2] OS: Microsoft Windows 2012 R2 (Standard edition x64); DBs: Microsoft SQL Server 2014 (Developer edition 12.0.2, 64-bit), PostgreSQL 9.4 (Compiled by Visual C++ build 1800, 64-bit); Memory: 24 GByte; CPU: 16x2.6 GHz Intel Xeon E312xx (Sandy Bridge); HDD: 500 GByte

Table 1. Summarized characteristics of the assessed data sets

Data set	*Records*	$\|A\|$	$\|D\|$	$\|IND_A\|$	$\|IND_D\|$	$\|C_A\|$
HIGGS [24]	11.000.000	28	1	10.721.302	2	5.829.123
RLCP [25]	5.749.132	11	1	5.749.132	2	5.728.201
SUSY [24]	5.000.000	18	1	5.000.000	2	2.712.173
KDD99	4.898.431	41	1	1.074.974	23	2.807.886
KDD99_M	4.898.431	42	1	1.075.016	23	2.807.886
PAMAP2 [26]	3.850.505	53	1	3.850.505	19	1.125.552
Poker Hand	1.025.010	10	1	1.022.771	10	511.308
Covertype [23]	581.012	54	1	581.012	7	297.711
NSL-KDD [27]	148.517	41	1	147.790	2	71.361
Spambase	4.601	57	1	4.207	2	1.810

algorithms which we review briefly to understand the priciples: The initial stage consists of scanning two input sources from disk followed by *hash aggregations*. Finally, both aggregated inputs are fused using the *hash join* operator. Denote, a hash aggregation only requires one single scan of the given input to build the resulting hash table. The hash join relies on a build and probe phase where essentially each of the two incoming inputs is scanned only once. In comparison to other alternatives, these query plans perform without sorting, but require memory to build up the hash tables. Once a query runs out of memory, additional buckets are spilled to disk, which was not the case throughout the series of experiments. Even though both engines share similar algorithms, MSSQL is capable of running the queries in parallel while PSQL covers single core processing only. In general, we realized a very high CPU usage which is characteristic for the performance of our model. However we further observed that MSSQL does not scale well processing KDD99, because it is unable to distribute the workload evenly to all threads. We relate this issue to the lack of appropriate statistics in the given raw environment including its data distribution, where three equivalence classes represent 51% of all records. Therefore, we introduce a revised version called KDD99_M. In contrast, it holds an additional condition attribute splitting huge classes into chunks of 50K records. Note, this change does not influence the approximation, but results in a speed up of 76%. Further details of the runtime comparison are given in Figure 2. Summarized, we could achieve reasonable responses without major optimization steps. In particular, our model scales well appending additional cores in 9 out of 10 tests. Supported by this characteristic, MSSQL computes most queries within few seconds.

7 Future Work

The evaluation of the previous section shows how our RST model behaves in a native relational environment. However, further practical experiments are required, which we will address in the near future. In our domain of interest, i.e. network and data security, we will study classical as well as modern cyber attack scenarios in order to extract significant features of each single attack in both IPv4 and IPv6 environments. Our model is most suited for that subject, because it is designed to process huge amounts of data efficiently and can han-

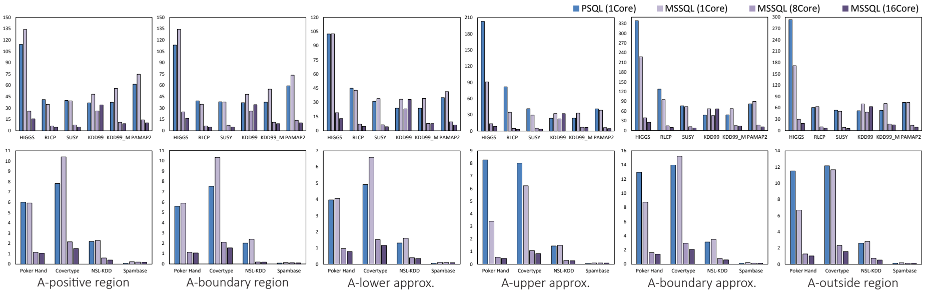


Fig. 2. Runtime comparison of the proposed rough set model in seconds

dle uncertainty which is required for proper intrusion detection. Additionally, we will use the outcome of our model to generate precise and characteristic attack signatures from incoming traffic and construct a rule-based classifier. Enabled by in-database capabilities, we can compute the resulting decision rules in parallel and integrate that approach into our existing data store. Hence, we can avoid huge data transports which is crucial for our near real time system.

8 Conclusion

In the past, the traditional Rough Set Theory has become a very popular framework to analyze and classify data based on equivalence relations. In this work we presented an approach to transport the concept approximation of that theory to the domain of relational databases in order to make use of well-established and efficient algorithms supported by these systems. Our model is defined on complete data tables and compatible with data inconsistencies. The evaluation on various prominent data sets showed promising results. The queries achieved low latency along with minor optimization and preprocessing effort. Therefore, we assume our model is suitable for a wide range of disciplines analyzing data within its relational sources. That given, we introduced a compact mining toolkit which is based on rough set methodology and enabled for in-database analytics. Immediately, it can be utilized to efficiently explore data sets, expose decision rules, identify significant features or data inconsistencies that are common challenges in the process of knowledge discovery in databases.

Acknowledgments. The authors deeply thank Maren and Martin who provided expertise and excellent support in the initial phase of this work.

References

1. M. Gawrys, J. Sienkiewicz: RSL - The Rough Set Library - Version 2.0. Technical report, Warsaw University of Technology (1994).
2. I. Düntsch, G. Gediga: The Rough Set Engine GROBIAN. In: Proc. of the 15th IMACS World Congress, pp. 613–618 (1997).
3. A. Ohrn, J. Komorowski: ROSETTA - A Rough Set Toolkit for Analysis of Data. In: Proc. of the 3rd Int. Joint Conf. on Information Sciences, pp. 403–407 (1997).

4. J.G. Bazan, M. Szczuka: The Rough Set Exploration System. TRS III, LNCS, vol. 3400, pp. 37–56 (2005).
5. M.C. Fernandez-Baizán, E. Menasalvas Ruiz, J.M. Peña Sánchez: Integrating RDMS and Data Mining Capabilities using Rough Sets. In: Proc. of the 6th Int. Conf. on IPMU, pp. 1439–1445 (1996).
6. A. Kumar: New Techniques for Data Reduction in a Database System for Knowledge Discovery Applications. JIIS, vol. 10(1), pp. 31–48 (1998).
7. X. Hu, T.Y. Lin, J. Han: A new Rough Set Model based on Database Systems. In: Proc. of the 9th Int. Conf. on RSFDGrC, LNCS, vol. 2639, pp. 114–121 (2003).
8. C.-C. Chan: Learning Rules from Very Large Databases using Rough Multisets. TRS I, LNCS, vol. 3100, pp. 59-77 (2004).
9. H. Sun, Z. Xiong, Y. Wang: Research on Integrating Ordbms and Rough Set Theory. In: Proc. of the 4th Int. Conf. on RSCTC, LNCS, vol. 3066, pp. 169-175 (2004).
10. T. Tileston: Have Your Cake & Eat It Too! Accelerate Data Mining Combining SAS & Teradata. In: Teradata Partners 2005 ”Experience the Possibilities” (2005).
11. Z. Pawlak: Rough Sets. Int. Journal of Computer and Information Science, vol. 11(5), pp. 341–356 (1982).
12. Z. Pawlak: Rough Sets - Theoretical Aspects of Reasoning about Data (1991).
13. Z. Pawlak: Information Systems - Theoretical Foundations. Inform. Systems, vol. 6(3), pp. 205–218 (1981).
14. F. Machuca, M. Millan: Enhancing Query Processing in Extended Relational Database Systems via Rough Set Theory to Exploit Data Mining Potentials. Knowledge Management in Fuzzy Databases, vol. 39, pp. 349–370 (2000).
15. H.S. Nguyen: Approximate Boolean Reasoning: Foundations and Applications in Data Mining. TRS V, LNCS, vol. 4100, pp. 334–506 (2006).
16. U. Seelam, C.-C. Chan: A Study of Data Reduction Using Multiset Decision Tables. In: Proc. of the Int. Conf. on GRC, IEEE, pp. 362–367 (2007).
17. S. Naouali, R. Missaoui: Flexible Query Answering in Data Cubes. In: Proc. of the 7th Int. Conf. of DaWaK, LNCS, vol. 3589, pp. 221–232 (2005).
18. T. Beaubouef, F.E. Petry: A Rough Set Model for Relational Databases. In: Proc. of the Int. Workshop on RSKD, pp. 100–107 (1993).
19. L.-L. Wei, W. Zhang: A Method for Rough Relational Database Transformed into Relational Database. In: Proc. of the Int. Conf. on SSME, IEEE, pp. 50–52 (2009).
20. D. Slezak, J. Wroblewski, V. Eastwood, P. Synak: Brighthouse: An Analytic Data Warehouse for Ad-hoc Queries. In: Proc. of the VLDB Endowment, vol. 1, pp. 1337–1345 (2008).
21. T.Y. Lin: An Overview of Rough Set Theory from the Point of View of Relational Databases. Bulletin of IRSS, vol. 1(1), pp. 30–34 (1997).
22. K. Bache and M. Lichman: UCI Machine Learning Repository. University of California, Irvine, http://archive.ics.uci.edu/ml (June, 2015).
23. J.A. Blackard, D.J. Dean: Comparative Accuracies of Neural Networks and Discriminant Analysis in Predicting Forest Cover Types from Cartographic Variables. In: Second Southern Forestry GIS Conf., pp. 189–199 (1998).
24. P. Baldi, P. Sadowski, D. Whiteson. Searching for Exotic Particles in High-energy Physics with Deep Learning. Nature Communications 5 (2014).
25. I. Schmidtmann, G. Hammer, M. Sariyar, A. Gerhold-Ay: Evaluation des Krebsregisters NRW Schwerpunkt Record Linkage. Technical report, IMBEI (2009).
26. A. Reiss, D. Stricker: Introducing a New Benchmarked Dataset for Activity Monitoring. In: Proc. of the 16th ISWC, IEEE, pp. 108–109 (2012).
27. NSL-KDD: Data Set for Network-based Intrusion Detection Systems. http://nsl.cs.unb.ca/NSL-KDD (June, 2015).

Resolving Unclassifiable Regions in Multilabel Classification by Fuzzy Support Vector Machines

Shigeo Abe

Kobe University
Rokkodai, Nada, Kobe, Japan
abe@kobe-u.ac.jp

In multilabel classification, a data sample is classified into one class or plural classes [1]. One of the widely used classification methods uses one-against-all classification, in which for an n-class problem, n decision functions are determined, with each decision function putting one class on the positive side and the remaining classes on the negative side. In classification, a data sample is classified into a single-label or multilabel class associated with positive decision functions. By this method, a data sample is unclassifiable if there is no positive decision function, and a data sample may be classified into a multilabel that is not included in the multilabels contained in the training set.

To solve this problem, in this paper, we propose one-against-all fuzzy support vector machines (FSVMs) for multilabel classification [2]. For each multilabel in the training data set, we define a new multilabel class. And for each single label or multilabel class, we define a fuzzy region using the decision functions determined by one-against-all classification. The degree of membership of a data sample to the fuzzy region is determined by the decision hyperplane that is nearest to the data sample. And the data sample is classified into the class with the highest degree of membership.

This classification strategy is simplified for an unclassifiable region. If no decision function is positive for a data sample, it is classified into a class with the maximum degree of membership. This is the same as the fuzzy SVM for single-class classification.

We compare the accuracies and subset accuracies of the proposed FSVMs with the conventional one-against-all, one-against-one, and the best accuracies in [1] using several benchmark data sets that are used in [1].

This work was supported by JSPS KAKENHI Grant Number 25420438.

References

1. G. Madjarov et al. An extensive experimental comparison of methods for multi-label learning. *Pattern Recognition*, 45(9):3084–3104, 2012.
2. S. Abe. Fuzzy support vector machines for multilabel classification. *Pattern Recognition*, 48(6):2110–2117, 2015.

Mining Sequential Patterns of Event Streams in a Smart Home Application

Marwan Hassani, Christian Beecks, Daniel Töws, and Thomas Seidl

Data Management and Data Exploration Group
RWTH Aachen University, Germany
{hassani,beecks,toews,seidl}@cs.rwth-aachen.de

Abstract. Recent advances in sensing techniques enabled the possibility to gain precise information about switched-on devices in smart home environments. One is particularly interested in exploring different patterns of electrical usage of indoor appliances and using them to predict activities. This in turns results with many useful applications like inferring effective energy saving procedures. The necessity to derive this knowledge in the real time and the huge size of generated data initiated the need for a precise stream sequential pattern mining approach. Most available approaches are less accurate due to their batch-based nature. We present a smart home application of the *PBuilder* algorithm which uses a batch-free approach to mine sequential patterns of a real dataset collected from appliances. Additionally, we present the *StrPMiner* which uses the PBuilder to find sequential patterns within multiple streams. We show through an extensive evaluation over a smart home real dataset the superiority of the StrPMiner algorithm over a state-of-the-art approach.

1 Introduction

Careful usage of indoor electrical devices is an important topic in the field of energy saving and sustainability. Understanding the usage patterns of appliances during a typical day is the key to induce savings of electrical energy. If a domain expert finds anomalies in the electricity usage of one house, which consumes a lot of energy, he can help the householder by suggesting lesser consuming patterns. Recent advances in sensing techniques enabled the possibility to gain precise information about different switched-on devices in a smart home environment. This information contains the time and the duration when a particular appliance was turned on. Gaining knowledge about correlation patterns between the activation of different devices is possible with an offline visualization of a small-sized data collected from a limited number of appliances (cf. Figure 1). This tends to be sophisticated when one requires an instant knowledge about the usage needed during the collection time. Additionally, the number of devices and

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

even the houses should usually be big enough to gain useful patterns. This signals the necessity to apply data mining methods to collect handy usage patterns.

A data stream produces an infinite and continuous flow of data. Regularities can often be found in those streams, which give information about the connection between the events in the data. To find this hidden information, sequential pattern mining algorithms can be used over the data stream. A suitable algorithm is able to reveal electric devices that are often used with or implied by each other. Sequential pattern mining is a special case of frequent item set mining, where patterns have to be frequent subsequences of the stream. Each pattern has to appear a certain number of times within a part of the stream (called *batch*) to count as a sequential pattern.

Additional challenges arise when looking at multiple streams at once, as patterns can be part of one or multiple streams. This is the case in a smart home environment, as each electric device provides a different data stream, feeding us with new information. For this a special treatment of data is needed, so that a useful connection among multiple electric devices can be found.

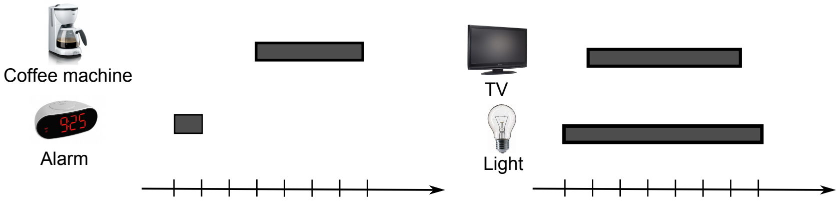


Fig. 1. If a person drinks a coffee every morning, the data would contain a connection between the alarm clock and the coffee machine. In particular the alarm would imply the coffee machine. In the second example the time frame, in which the TV is used, would be contained by the time frame, in which the light is used.

Multiple algorithms were proposed in the literature to mine sequential patterns from data streams. Most of them use a batch approach, like the *SS-BE* algorithm [11]. The batch approach is a simple and efficient solution to mine sequential patterns in a stream. However, it leaves a room for errors. Sequential patterns are, by definition, very sensitive to the order of items. This order can not be found when searched patterns are located between two consecutive batches. A batch-based algorithm will fail to detect such patterns. Moreover, single items might have a duration as in the case of the interval-based events in our smart home application (cf. Fig. 1). These items might also span multiple batches.

In this work, we present an application over a real smart home dataset using two algorithms [14] that avoid the above mentioned errors. The first algorithm is the Pattern Builder *PBuilder* which mines sequential patterns for given data using a batch-free approach. The second algorithm is the Streaming Pattern Miner *StrPMiner* which uses the *PBuilder* to find sequential patterns within multiple streams arriving from multiple indoor appliances and keeps track of their quality.

The remainder of this paper is organized as follows: Section 2 presents some related work. Section 3 looks at the preliminaries of sequential pattern mining. Additionally it will highlight the problem with the batch approach. In Section 4 two algorithms are presented. The algorithm *StrPMiner* is then tested against the *SS-BE* algorithm in Section 5, where we will also prove its superior accuracy. The paper is concluded with a summary and an outlook in Section 6.

2 Related Work

Optimizing sequential pattern mining is an important task in the streaming data mining field, which leads to a lot of different algorithms. A base algorithm for many approaches [11],[13],[15],[2], is the *PrefixSpan* algorithm [12]. The *PrefixSpan* algorithm was designed for a static data environment. Because of this it can use the apriori assumption [1], that every part of a frequent pattern also has to be frequent. In the *PrefixSpan* patterns are generated bottom up. Starting with a frequent item, each pattern will be checked for its frequency. If it is frequent, it will be used as a prefix for other frequent items to generate longer patterns. All algorithms using the *PrefixSpan* in a stream environment collect data in a batch instead of evaluating each item as soon as it arrives.

Since the streaming approach allows to only look at data once, algorithms have to make compromises in order to provide fast results. [11] proposes two algorithms with different pruning strategies, the *SS-BE* and *SS-BM* algorithms. These algorithms restrict memory usage but are able to find all true sequential patterns and allow an error bound on the false positives. The patterns are saved in a new designed tree structure, the T_0 tree. The tree will be frequently checked and pruned. Patterns that did not reappear frequently in the past will be deleted, so that only current frequent items are contained in the tree.

In a static data set, all information needed for the algorithm is provided from the beginning, while in the streaming approach new data arrives every second, thus, patterns that were not frequent in the beginning may become frequent later on. Yet, it is impossible to save every pattern and its information. The *FP-stream* [3] solves this issue by saving information in different time granularities. The newer the information, the more accurate it will be displayed. Another way to solve the memory problem is by using a sliding window model, in which only the most recent data is being looked at. The *MFI-TransSW* algorithm [10] optimizes this concept. The algorithm works in three steps: window initialization, window sliding and pattern generation. Previously described algorithms only provide solutions for one stream. In cases of multiple streams in parallel, the *MSS-BE* algorithm [8] is an idea to find sequential patterns in an multiple-stream environment, where pattern elements can be part of different streams.

The algorithms mentioned above only provide solutions for frequent pattern mining or find sequential patterns by using batches. The stream pattern miner (StrPMiner) algorithm which uses the PBuilder was first introduced in [14]. It uses a sliding window approach instead of the batch method while efficiently mining sequential patterns of the streams. The algorithm was successfully used in an application within the humanities domain, for analysis of translation data, where subjects are translating English texts into German. The two streams in that case were the eye gazes of the translators and their collected keystrokes during the translation session [4,5,14].

3 Preliminaries: Sequential Pattern Mining

We are given a set $\mathcal{S} = \{S^1, S^2, \ldots, S^{|\mathcal{S}|}\}$ of $|\mathcal{S}|$ different streams arriving from different observed parameters collected from the smart home. Each stream S^k is represented by streaming, time-stamped interval-based events that evolve over the time. Thus, the first n items of stream S^k are represented as $S^k = \{s_1^k, s_2^k, \ldots, s_n^k\}$ where s_i^k is an observed event that occurs at time t_i where $t_i < t_{(i+1)}$ for all $i = 1, \ldots, n$. Each event is additionally described by its label. A sequential pattern is a combination of multiple events that follow each other. These patterns can be used to find correlations in the data.

We are asked to obtain the different frequent patterns that appear within a *single* stream S^k and also within *multiple* streams from $\mathcal{S}$ (also called multimodal streams). The sequential pattern mining problem differs from the normal frequent item set mining in the fact that the order of items (events) matters. The problem of mining sequential patterns is defined as follows: Let $I = \{i_1, i_2, \ldots, i_{|I|}\}$ be a set of $|I|$ items, each item consists of a timestamp and a duration. A pattern is represented here by a sequence, which is an ordered list of items from I denoted by $\langle p_1, p_2, \ldots, p_k \rangle$. Thus, a sequence $p = \langle a_1, a_2, \ldots, a_q \rangle$ is a subsequence of a sequence $p' = \langle b_1, b_2, \ldots, b_r \rangle$ if there exists integers $i_1 < i_2 < \cdots < i_q$ such that $a_1 = b_{i_1}, a_2 = b_{i_2}, \ldots, a_q = b_{i_q}$.

This definition of sequential pattern mining is very feasible for the continuously emerging characteristics of stream data. A stream S^k in this context is an arbitrarily large list of sequences p_i. A sequence p in the data stream S^k *contains* another sequence p' from S^k if p' is a subsequence of p. The *count* of a sequence p, denoted as $count(p)$, is defined by the number of sequences that contain p in the stream S^k. If the frequency of a pattern (p) within a window w of the stream S^k is greater or equal to a user defined threshold min_supp, then the sequence p is a frequent sequence or a sequential pattern in that window of S^k.

Following the apriori principle [1], given two subsequences $p = \{p_1, p_2, ..., p_n\}$ and $p' = p \backslash \{p_n\}$, it holds that $supp(p') \geq supp(p)$ due to the anti-monotonicity property. Thus, if p is a sequential pattern, p' is also a sequential pattern.

To provide different views on the data, three different window concepts are used by the *StrPMiner*. The algorithm works with the Landmark Window, the Sliding Window and the Damped Window concept. In the Landmark Window, a point in time is defined as the *landmark*. All data is then collected starting from the *landmark*. This concept allows to look at big parts of the data. The Sliding Window concept uses a fixed window size and slides it over the data. Thus, only a snapshot of the data will be monitored at any given time. An advantage is that old patterns will be forgotten eventually, which leaves only current information. The Damped Window weights the objects to reflect their age. New items will be

more important than old ones. This allows a compromise between the Landmark Window and the Sliding Window concept. A good solution to find sequential patterns in a streaming environment is the batch approach. It allows to use the Apriori principle, since each batch provides a static data set. However it comes at a cost. Given a support threshold of 2, meaning a pattern has to appear two times within one batch to be counted as frequent, a batch size of 3 and following sequence: $(A, B, C, A, C, C, A, D, C, A...)$ with A, B, C, D being items of a stream. The online component would cut the data stream in following batches:
1. (A, B, C) 2. (A, C, C) 3. (A, D, C) 4. (A, ...) 5. ...

In this case, no pattern would be frequent. Looking at the whole data without cutting it into batches would reveal that the pattern C, A appears three times, which is over the support threshold of 2. This would lead to a frequent pattern. Additionally, all items except for C in the second batch, would be pruned away, although the item A and C appear in every batch. This leads to two reasons for errors through the batch approach: First: Patterns that appear between batches will not be found. Second: Items and patterns that do not appear often in one batch will be pruned, although they are frequent in the whole data set. The *StrPMiner* was designed to avoid the batch approach because of these two reasons which result into false statistics for sequential patterns.

4 The *StrPMiner* and the *PBuilder* Algorithms

Since the *PrefixSpan* algorithm only scales well when the candidates for sequential patterns can be pruned, the *StrPMiner* reverses the idea of the *PrefixSpan* and uses a new algorithm called the Pattern Builder (*PBuilder*). This allows the *StrPMiner* to work on each data item step by step as it comes in.

To provide a more focused view on the order of the items, the definition of sequential patterns was changed slightly. As stated previously, a sequential pattern is a frequent subsequence. We redefine subsequences, and sequential patterns, as only allowed to be a list of ordered items that directly follow each other. Thus, p is considered a subsequence of q if $p = (p_1, p_2, ..., p_n)$, $q = (q_1, q_2, ..., q_m)$ and there exist integers $i_1 < i_2 < ... < i_m$ such that $p_1 = q_{i_1}, p_2 = q_{i_2}, ..., p_n = q_{i_n}$ for $n < m$ *and* for all k, l with $l, k < m$ and $l = k + 1$.

The *StrPMiner* handles arriving data from multiple streams at once. For this, we assume that at each point in time only one item can arrive per stream. If multiple items from multiple streams arrive at the same time, they will be put into an ordered list and the algorithm handles each item after another. First an item will be compressed, as only the label and the timestamp are relevant for creating sequential patterns. Then the *StrPMiner* passes the item to the *PBuilder*. The *PBuilder* then uses this data to create sequential pattern candidates. After this, the *StrPMiner* saves the candidates in the T_0 tree structure and keeps track of those candidates and their corresponding statistics. Currently this is the count value, which allows to calculate the support and confidence value of a pattern. The tree will be updated with the new count values and if a pattern was not part of the tree a new node will be created. This approach allows full accuracy, and flexibility in the output, as the support threshold can be changed at every output request. This is not possible when using the *PrefixSpan*, since the threshold has to be previously set.

4.1 The *PBuilder*

The *PBuilder* creates only patterns that contain the newly arrived item. Since it is the last arrived item, all created patterns will end with this item. Given an item A as the newly arrived item, the *PBuilder* starts with this item as a pattern of length one. After this, the algorithm recursively adds older items as a prefix to the previously created postfix. To ensure that the *StrPMiner* only finds direct sequential patterns, the prefix is a direct predecessor of the postfix. As visible in the pseudocode, visible in Agorithm 1, the ItemList only contains the latest items ordered by their appearance. The newest item is the last item in the list. In the first iteration, the currentPattern parameter is empty. Line 7 will then recursively add a prefix to our current pattern. The resulting pattern will be inserted into the tree, as visible in Line 9. This will be repeated, until the complete ItemList was included.

For each created pattern, the *PBuilder* algorithm calls the update function of the T_0 tree. An example of the tree can be seen in Figure 2.

Algorithm 1: The *PBuilder* explained with pseudo code

```
1  PBuilder
   Data: ItemList, currentPattern
2  //ItemList contains the latest compressed items and is limited by
   maxPatternLength. The newly arrived item is at the last position
   Result: The new patterns that can be created with the new item
3  int index = ItemList.length;
4  //create patterns until maxPatternLength is reached
5  while $currentPattern.length \leq ItemList.length$ do
6      //add the next item to the pattern
7      currentPattern = ItemList.get(index-currentPattern.length) +
       currentPattern;
8      //update the tree with the new pattern
9      updateTree(currentPattern);
10 end
```

4.2 Maximum Pattern Length as a Solution for Exponential Growth

In contrast to a static database, where all information is available from the beginning, the streaming approach does not have any information on what future items and their frequency might look like. This means that any item and pattern that is currently not frequent in a stream, can become frequent at any later point in time. The support of every pattern changes with every new arriving item. To ensure that at every time the user requests an output all sequential patterns are part of the output, every possible pattern and its information have to be saved. This causes an exponential increase of the calculation time, as with every new arriving item more patterns can be created. Additionally, the memory space will eventually collapse, as the amount of data that has to be saved also increases exponentially.

To stop the exponential growth, the *StrPMiner* introduces a parameter called *maxPatternLength*, as an upper Bound for the pattern length. This variable restricts the *PBuilder* to only look at the last *maxPatternLength* items. A *maxPatternLength* of five, will cause patterns to maximally contain five items, as only those are given to the algorithm. Given this bound, the calculation time in each step only scales with the size of the *maxPatternLength* parameter. Additionally this parameter bounds the maximum growth of the required memory space. On the one hand, as the parameter will not change over the time, the calculation time for each new arriving item will be constant. On the other hand this upper bound filters patterns, before they have been created. Sequential patterns that have a length higher than the given bound, will not be found. With this in mind, a careful selection of the upper bound is important, as it provides a trade off between the calculation time and accuracy.

4.3 Different Window Models

As previously mentioned, the *StrPMiner* uses the T_0 tree introduced by [11]. For the algorithm slight adaptations were made, regarding the saved information. The *StrPMiner* saves the label of the item and the time stamps, at which it appeared, of the pattern in each node. The count of each pattern is then determined by the number of time stamps saved in the corresponding node. An example is shown in Figure 2.

The sliding window model helps to provide another view on the data, as it only contains knowledge of recent data and forgets old data. This helps in cases, where the data changes drastically over the duration of the stream. The landmark window would still show old patterns even though they did not reappear for a long time. In general, the whole algorithm works the same, as in the landmark window, except for an extra pruning step. For this the time stamp of the corresponding item and the patterns created with it have to be deleted from the T_0 tree, which is one path.

5 Experimental Results

Because of the problems that come with the batch approach, the *StrPMiner*, unlike the *SS-BE* algorithm, does not use the *PrefixSpan*. Instead it uses the *PBuilder*, which handles each newly arriving item immediately, without using the batch approach. In this section we compare the presented algorithm to the *SS-BE*, since it is a current state of the art algorithm that finds sequential patterns in a stream environment. Other algorithms we looked at did not fulfill both of these criteria.

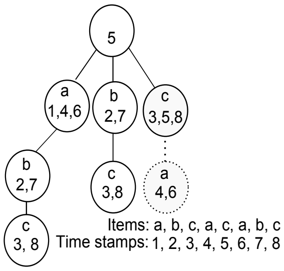

5
a
1,4,6
b
2,7
c
3,5,8
b
2,7
c
3,8
a
4,6
c
3, 8
Items: a, b, c, a, c, a, b, c
Time stamps: 1, 2, 3, 4, 5, 6, 7, 8

Fig. 2. An example of the T_0 tree. The dotted node represents the pattern (c,a).

For the experimental evaluation of both algorithms we used the REDD dataset [9]. This dataset contains information about the usage of electric devices in Smart Homes. For analyzing those information we preprocessed the data to an event stream. Each stream represents one electronic device, where the items contain the information about the on and off time of the objects. For example, if the oven is turned on at time t, the corresponding item at time t will be labeled *oven +* and *oven -* if it is turned off. Following this code, the patterns of the examples in Figure 1 would be *alarm +, alarm -, coffee +, coffee -* and *light +, tv +, tv -, light -*.

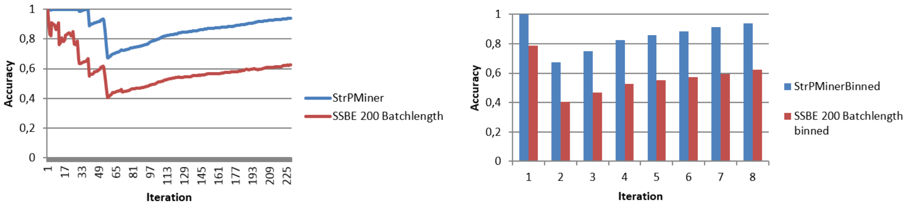
Since we are only interested of direct sequential patterns, we adapted the *PrefixSpan* in such a way that it will only create direct sequential patterns. The adaption will additionally effect the results output by the *SS-BE* algorithm, as it is dependent on the results produced by the *PrefixSpan*.

The support threshold, the only parameter used by both algorithms, was set to 1%.

For the *StrPMiner* we set the maximum pattern length at 200. As explained in 4.2, this parameter strongly influences the patterns that we find and our runtime. The runtime of the *StrPMiner* is slower than the runtime of the *SS-BE*, but with this parameter setting we still ensure real time results. Our assumption is, that, with this setting, the *PBuilder* will find every pattern that is shorter than 200. This result into full accuracy for those patterns. In this evaluation we only want to look at the strong accuracy of the *StrPMiner*, we will only use the Landmark Window here. The Sliding Window and the Damped Window show similar results.

The parameters we set for the *SS-BE* algorithm were the significance threshold ϵ with 0.0099 and the pruning period δ to 10. Those settings are close to those used by the authors [11]. This means, that after ten batches the algorithm will prune the t_0 tree. The batch length is either set to 200 or to 300. Those settings ensure that we will compare both algorithms to similar patterns and similar output.

In a first evaluation we compared both algorithms against a ground truth, which contains all patterns with a support of at least 1%. As the *SS-BE* algorithm uses the batch approach, an output can only be generated after batch



(a) Comparing each time stamp.

(b) The evaluated time stamps were binned and then compared.

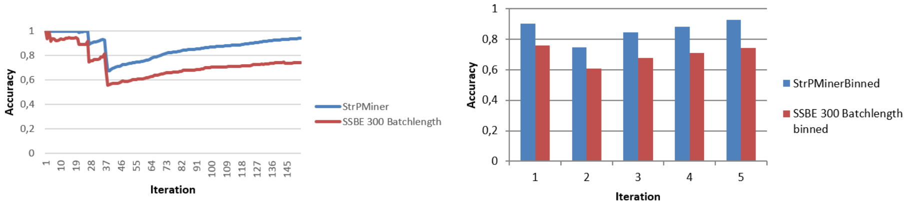
Fig. 3. A comparison of the *StrPMiner* to the *SS-BE* algorithm after evaluating one house. The y-axis displays the accuracy, while the x-axis shows the time. The batch length was set to 200.

length amount of items were evaluated. This means in our case, that only after each 200 or 300 items, an output is available. In contrast to this, the *StrPMiner* can produce a valid output after each item, as it will treat each item directly. In Figure 3 we compared the result of both algorithms to the output after each 200 items. Additionally another comparison is created, where we bin the single time steps. As visible in this figure, the *StrPMiner* has a significantly higher accuracy, which is 30% points higher at each single time step for the given data. Two other things are also visible in this figure. First, the accuracy of the *StrPMiner* stays 100% for the first few time steps, as long as there are no frequent patterns found with a higher length than 200. Second, there is a noticeable drop in the accuracy during the first third of the evaluation. A closer look into the data reveals, that during this time the amount of patterns, that have a higher length than 200, is rising. But, all of those patterns are single stream patterns, with a switching on and off event of one single device, happening in a few seconds. The binning is used to smooth out those abrupt changes and provide a focused view on the general direction of the results.

Although the accuracy of the *SS-BE* algorithm rises with a higher batch length, all three observations are still visible in Figure 4. We tested the algorithms against multiple houses, in which the accuracy of the algorithms changed slightly, but the general direction was the same, revealing the higher accuracy of the *StrPMiner*. In houses with less noisy data, we were even able to maintain full accuracy with the *StrPMiner*, as there were no frequent patterns with a high batch length.

In most of the evaluation the higher batch length setting shows to be more accurate, but still has a lower accuracy of nearly 20% points.

In a next step we wanted to prove our assumption. Only looking at the most important patterns, meaning the top 100 patterns with the highest support, reveals that the *PBuilder* has a full accuracy for all patterns with a length lower than the maximum pattern length. A comparison to the *SS-BE* algorithm is visible in Figure 5. This figure shows, how many of the hidden patterns in the data could be found. In this case, the *SS-BE* algorithm has a high accuracy of over 90%, but is still beaten by the full accuracy of the *StrPMiner*.

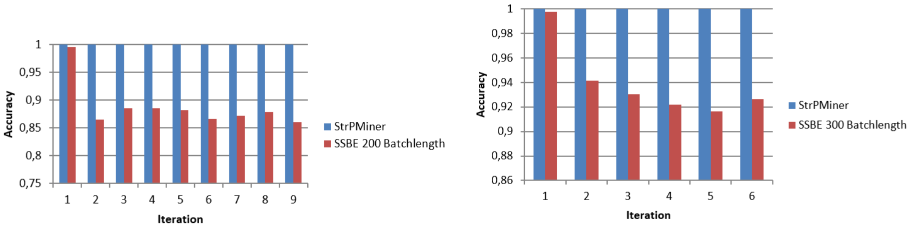


(a) (b)

Fig. 4. A similar comparison as in Figure 3, but with a batch length of 300 for the *SS-BE* algorithm.

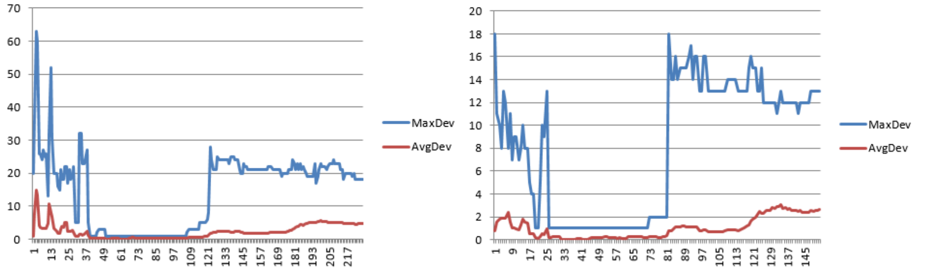
Taking a closer look at the order of the top 100 reveals, that, due to the full accuracy, the *StrPMiner* is able to show all important patterns in the correct order, sorted by their support value. The *SS-BE* algorithm is not able to keep the correct position of the patterns. Figure 6 shows the deviation of the patterns at each time step. The figure shows the mean deviation over all patterns, and the maximal deviation of one pattern.

Although these results show the higher accuracy for the *StrPMiner*, they only represent the average case, formed by looking at all patterns. The open question is, how can these results help in an application case, where we want to find and keep track of specific interesting patterns? The open assumption we want to test is, that both algorithms are able to find meaningful patterns. This means, patterns that show an existing connection between the items contained in it. To test this assumption, we created a correlation matrix for the devices in the data set. A snapshot of it is shown in Table 1, which gives information about how often the items were turned on or off *together*. A higher value means that the on and off time of those two items is close to each other. With this correlation matrix we may not gain information about the specifics of the connection of two items, but we can safely say, that there is a connection between those items.



(a) (b)

Fig. 5. For the most important patterns, the top 100, both algorithms show a higher accuracy. Notable is, that the *StrPMiner* provides full accuracy.



(a) The deviation for a batch length of 200 reaches up to 60.

(b) The deviation for a batch length of 300 reaches only to 20.

Fig. 6. This figure shows the deviation between the top 100 patterns created by the *SS-BE* algorithm, compared to the ground truth. A maximal deviation of 10 means, that a pattern a, that appeared at position x in the ground truth, will appear at position $x + 10$ or $x - 10$ in the results of the *SS-BE* algorithm.

	oven	oven	refrigerator	dishwasher	k_outlets	k_outlets	lighting
oven	1	0.828	0.046	0.387	0	0	0.006
oven	0.828	1	0.051	0.307	0	0	0.005
refrigerator	0.046	0.051	1	0.022	0	0	0.011
dishwasher	0.387	0.307	0.022	1	0	0	0
k_outlets	0	0	0	0	1	0	0
k_outlets	0	0	0	0	0	1	0
lighting	0.006	0.005	0.011	0	0	0	1

Table 1. A part of the correlation matrix between some appliances.

This is also reflected in the results of the *StrPMiner*, as patterns between two items with a high correlation, are the multimodal patterns with the highest support. Item combinations with a correlation of over 0.6 are part of the frequent patterns. These patterns, like *oven 3+, oven 4+* and *oven 3-, oven 4-* show that both items are often used with each other. Six of those multimodal patterns have a higher support than 1% in the ground truth and can be found with full accuracy in the results of the *StrPMiner*. In contrast to this, the *SS-BE* algorithm can find three of those with an error rate of over 5%. The other 3 items are not part of the results at all, as they were pruned out of lost between batches of *SS-BE*.

6 Conclusion and Future Work

In this paper we have presented a smart home application over a recent algorithm, the *PBuilder* [14], that is able to mine sequential patterns in data streams. The *StrPMiner* [14] uses the *PBuilder* for the pattern calculation in multiple streams. The results are saved in the T_0 tree. Three different window concepts allow to present the data in different perspectives, which helps users to analyze the data more effectively. Additionally the algorithm can create the output in a much more flexible way than other algorithms, that use the *PrefixSpan*. For each output request any support threshold can be given and the output can be created correctly. The usefulness of the algorithm is tested with the big smart home

REDD dataset. We compared the *StrPMiner* against the *SS-BE* algorithm. In our experimental evaluation we showed, that our algorithm has a significantly higher accuracy than the competitor. Additionally, we showed that the algorithm is capable of running over big real datasets.

In the future we plan to improve the time efficiency of our algorithm. Although our algorithm is able to calculate the results in real time, it is slower than the *SS-BE* algorithm. We found the bottleneck in the insertion step of the data into the T_0 tree. First changes could improve the runtime significantly. We would like additionally to test our approach in distributed, multi-source sensor streaming environments [7] and in anytime environments [6].

References

1. R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *VLDB*, pages 487–499, 1994.
2. Y.-C. Chen, C.-C. Chen, W.-C. Peng, and W.-C. Lee. Mining correlation patterns among appliances in smart home environment. In *PAKDD*, pages 222–233. 2014.
3. C. Giannella, J. Han, J. Pei, X. Yan, and P. S. Yu. Mining frequent patterns in data streams at multiple time granularities. *Next gen. DM*, 212:191–212, 2003.
4. M. Hassani. *Efficient Clustering of Big Data Streams.* PhD thesis, RWTH Aachen University, 2015.
5. M. Hassani, C. Beecks, D. Töws, T. Serbina, M. Haberstroh, P. Niemietz, S. Jeschke, S. Neumann, and T. Seidl. Sequential pattern mining of multimodal streams in the humanities. In *BTW*, pages 683–686, 2015.
6. M. Hassani, P. Kranen, and T. Seidl. Precise anytime clustering of noisy sensor data with logarithmic complexity. In *SensorKDD Workshop @KDD*, pages 52–60, 2011.
7. M. Hassani, E. Müller, P. Spaus, A. Faqolli, T. Palpanas, and T. Seidl. Self-organizing energy aware clustering of nodes in sensor networks using relevant attributes. In *SensorKDD Workshop @KDD*, pages 39–48, 2010.
8. M. Hassani and T. Seidl. Towards a mobile health context prediction: Sequential pattern mining in multiple streams. In *MDM*, pages 55–57. IEEE, 2011.
9. J. Z. Kolter and M. J. Johnson. Redd: A public data set for energy disaggregation research. In *SustKDD Workshop @KDD*, 2011.
10. H.-F. Li and S.-Y. Lee. Mining frequent itemsets over data streams using efficient window sliding techniques. *Expert Sys. w. App.*, 36(2):1466–1477, 2009.
11. L. F. Mendes, B. Ding, and J. Han. Stream sequential pattern mining with precise error bounds. In *ICDM.*, pages 941–946, 2008.
12. J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *ICDE*, pages 0215–0215, 2001.
13. A. F. Soliman, G. A. Ebrahim, and H. K. Mohammed. Speds: A framework for mining sequential patterns in evolving data streams. In *Communications, Computers and Signal Processing (PacRim), 2011*, pages 464–469. IEEE, 2011.
14. D. Töws, M. Hassani, C. Beecks, and T. Seidl. Optimizing sequential pattern mining within multiple streams. In *BTW*, pages 223–232, 2015.
15. S.-Y. Wu and Y.-L. Chen. Mining nonambiguous temporal patterns for interval-based events. *KDE*, pages 742–758, 2007.

Comparing Prediction Models for Active Learning in Recommender Systems

Rasoul Karimi, Christoph Freudenthaler, Alexandros Nanopoulos, Lars Schmidt-Thieme

Information Systems and Machine Learning Lab (ISMLL)
Samelsonplatz 1, University of Hildesheim, D-31141 Hildesheim, Germany
{karimi,freudenthaler,nanopoulos,schmidt-thieme}@ismll.uni-hildesheim.de

Abstract. Recommender systems help web users to address information overload. Their performance, however, depends on the amount of information that users provide about their preferences. Users are not willing to provide information for a large amount of items, thus the quality of recommendations is affected. Active learning for recommender systems has been proposed in the past, to acquire preference information from users. Early active learning methods for recommender systems used as underlying model either memory-based approaches or the aspect model. However, matrix factorization has been recently demonstrated (especially after the Netflix challenge) as being superior to memory-based approaches or the aspect model. Therefore, it is promising to develop active learning methods based on this prediction model. In this paper, we thoroughly compare matrix factorization with the aspect model to find out which one is more suitable for applying active learning in recommender systems. The results show that beside improving the accuracy of recommendations, the matrix factorization approach also results in drastically reduced user waiting times, i.e., the time that the users wait before being asked a new query. Therefore, it is an ideal choice for using active learning in real-world applications of recommender systems.

1 Introduction

Recommender systems guide users in a personalized way to interesting or useful objects in a large space of possible options. There are several techniques for recommendation and collaborative filtering is one them [1, 2]. Given a domain of items, users give ratings to these items. The recommender system can then compare the user's ratings to those of other users, find the most similar users based on some criterion of similarity, and recommend items that similar users have already liked.

Evidently, the performance of recommender systems depends on the number of ratings that the users provide. This problem is amplified even more in the case where we lack ratings due to a new user (cold-start problem). There are different solutions to deal with this problem. The first solution is to use the meta data of the new user. However, even a few ratings are more valuable than the meta data [21]. Therefore, the new user is requested to provide ratings to some items. But a well identified problem is that users are not willing to provide ratings for a large amount of items [4, 5]. Therefore, the queries presented to the new user have to be selected carefully. To address this situation active learning methods have been proposed to acquire those ratings from the new user that will help most in determining his/her interests [5, 4]. Another approach for the new user problem is to use implicit feedback. It means the recommender system uses implicit information from the user (browsing, viewing events) that can be used to quickly adjust his/her user model to his/her real taste, while interacting with the system [22]. In this paper, we focus on the active learning approach and do not deal with the other solutions.

Early active learning methods for recommender systems were developed based on Aspect Model (AM) [4, 5]. However, Matrix Factorization (MF) has been demonstrated (especially after the Netflix challenge) as being superior to other techniques. Therefore, it is promising to develop active learning methods based on this prediction model. In this paper we examine AM and MF for the new user problem in recommender systems. For this problem, in addition to the accuracy, training time of the prediction model is also important. It is because the preference elicitation of the new user is an interactive scenario and long time interruptions cause the new users to leave the conversation.

This paper is organized as follows: in section 2, related work is reviewed. In section 3, MF and AM are explained. In section 4, the training algorithms of MF and AM are compared. The experimental result is given in section 5. Finally the conclusion is stated in section 6.

2 Related Work

Active learning, in the context of the new-user problem, was introduced by Kohrs and Merialdo [9]. This work suggested a method based on nearest-neighbor collaborative filtering, which uses entropy and variance as the loss function to identify the queried items. Al Mamunur et al. [6] expanded this work, by considering the popularity of items and also personalizing the item selection for each individual user. Boutilier et al. [10] applied the metric of expected value of utility to find the most informative item to query, which is to find the item that leads to the most significant change in the highest expected ratings.

Jin and Si [4] developed a new active learning algorithm based on AM which is similar to applying active learning for parameter estimation in Bayesian networks [11]. This method uses the entropy of the model as the loss function. However, this work does not directly minimize the entropy loss function, because the current model may be far from the true model and relying only on the current model can become misleading. To overcome this problem, this work proposes to use a Bayesian network to take into account the reliability of the current model. This Bayesian approach is, however, complex and intractable for real applications (demands excessive execution time). Harpale and Yang [5] extended [4] by relaxing the assumption that a new user can provide a rating for any queried item. This approach personalizes active learning to the preferences of each new user as it queries only those items for which users are expected to provide a rating for. Karimi et. al [12] applied the most popular item selection to AM. The results show that it competes in accuracy with the Bayesian approach while its execution time is in the order of magnitude faster than the Bayesian method.

Karimi et. al [13] developed a non-myopic active learning which capitalizes explicitly on the update procedure of the MF model. Initially, this method queries items that if the new user's features are updated with the provided rating, it will change the features as much as possible. Its goal is to explore the latent space to get closer to the optimal features. Then, it exploits the learned features and slightly adjusts them. Karimi et. al. [14] by being inspired from existing optimal active learning for the regression task, exploits the characteristics of matrix factorization and develops a method which approximates the optimal solution for recommender systems. Karimi et. al. [15] improved the most popular item selection according to the characteristics of MF. It finds similar users to the new user in the latent space and then selects the item which is most popular among the similar users.

The idea of using decision trees for cold-start recommendation was proposed by Al Mamunur et. al [8]. Golbandi et. al [7] improved [8] by advocating a specialized version of decision trees to adapt the preference elicitation process to the new user's responses. Zhou et. al [20] modified [7] by associating matrix factorization to decision trees. Karimi et. al [16] proposed another approach to introduce matrix factorization in decision trees which is more scalable compare to [20]. Karimi et. al [17] improved the decision trees by splitting the nodes of the trees in a finer-grained fashion. Specifically, the nodes are split in a 6-way manner instead of 3-way split. Karimi et. al [18] proposed an innovative approach for active learning in recommender systems. The main idea is to consider existing users as (hypothetical) new users and solve an active learning problem for each of them. In the end, we aggregate all solved problems in order to learn how to solve the active learning problem for a real new user.

3 Background

In this section, a short introduction to AM and MF is provided.

3.1 Aspect Model

The Aspect Model is a probabilistic latent space model, which models user interests as a mixture of preference factors [24, 25]. The latent class variables

$f \in F := \{f_1, f_2, ..., f_k\}$ are associated with each user u and each item i. Users and items are independent from each other given the latent class variable f. The probability for each observation tuple (u, i, r) is calculated as follows:

$$p(r|i, u) = \sum_{f \in F} p(r|f, i)p(f|u) \tag{1}$$

where $p(f|u)$ is a multinomial distribution and stands for the likelihood for user u to be in the latent class f. $p(r|f, i)$ is the likelihood of assigning item i with rating r for class f. In order to achieve better performance, the training ratings of each user are normalized with zero mean and variance 1 [25]. The parameter $p(r|f, i)$ is a Gaussian distribution $N(\mu_{i,f}, \sigma_{i,f})$ with latent class mean $\mu_{i,f}$ and standard deviation $\sigma_{i,f}$.

3.2 Matrix Factorization

Matrix Factorization is the task of approximating the true, unobserved ratings-matrix R. The rows of R correspond to the users U and the columns to the items I. Thus the matrix has dimension $|U| \times |I|$. The predicted ratings $\hat{R}$ are the product of two feature matrices $W : |U| \times k$ and $H : |I| \times k$, where the u-th row w_u of W contains the k features that describe the u-th user and the i-th row h_i of H contains k corresponding features for the i-th item. The elements of h_i indicate the importance of factors in rating item i by users. Some factors might have higher effect and vice versa. For a given user the element of w_u measure the influence of the factors on user preferences. Different applications of MF differ in the constraints that are sometimes imposed on the factorization. The common form of MF is finding a low-norm approximation (regularized factorization) to a fully observed data matrix minimizing the sum-squared difference to it.

The predicted rating $\hat{R}$ of user u to item i is the inner product of the user u features and item i features $h_i^T w_u$. However, the full rating value is not just explained by this interaction and the user and item bias should also be taken into account. It is because part of the rating values is due to effects associated with either users or items,i.e biases, independent of any interactions.

By considering the user and item bias, the predicted rating is computed as follows [3]:

$$\hat{r}_{ui} = \mu + b_i + b_u + h_i^T w_u \tag{2}$$

where μ is the global average, b_i and b_u are item and user bias respectively. The major challenge is computing the mapping of each item and user to factor vectors $h_i, w_u \in R^k$. The mapping is done by minimizing the following squared error:

$$Opt(S, W, H) = \sum_{(u,i) \in S} (r_{ui} - \mu - b_u - b_i - h_i^T w_u)^2 + \lambda(\|h_i\|^2 + \|w_u\|^2 + b_u^2 + b_i^2) \tag{3}$$

where λ is the regularization factor, and S is the set of the (u, i) pairs for which r_{ui} is known, i.e the training set. The details of MF learning algorithm is described in [3].

When MF is applied to a specific data set, the predicted ratings should be in the range of the minimum rating and maximum rating of the dataset. However, sometimes this does not happen and we have to explicitly clip them. To solve this problem we use the sigmoidal function to automatically truncate the predicted rating to the range of minimum and maximum ratings. Therefore, the predicted ratings are computed as follows:

$$\hat{r}_{ui} = MinRating + \frac{(MaxRating - MinRating)}{1 + e^{-(\mu + b_i + b_u + h_i^T w_u)}} \tag{4}$$

3.3 Retraining Policy

When a new user enters the recommender system, the prediction model (AM or MF) should be updated to learn the new user latent features. As there are already a lot of users in the recommender system, training the model from scratch needs a lot of time. Therefore, we switch to online updating which means after a first training, further retraining is only done for new users.

For online updating, we use the method introduced in [23]. In this method after getting a new rating for the new user, the user's latent features are initialized to a random setting and then learned using all ratings of the new user. The complexity of retraining is the same as the training but the size of training data, S, is only the number of ratings used for online updating which is just the ratings provided by the new user.

When the online updating technique is applied in MF, the learning step should be reduced. This is because the number of training data (ratings provided by the new user) is small and updating the new user's latent features should be done more precisely and carefully. In our experiments the learning step in the training phase is 0.01 and is reduced to 0.001 when online updating is performed.

4 Comparing AM and MF

The training algorithm for MF has the time complexity of [23] :

$$O(L \times |S| \times k) \tag{5}$$

where L is the maximum number of iterations. The learning algorithm stops if the RMSE on the training data is smaller than ϵ.

The training algorithm for AM is shown in Algorithm 1. In this algorithm, the convergence criterion is the same as the convergence criterion in MF. According to this algorithm the time complexity of AM is $O(L \times |S| \times k)$ which is equal to Equation 5. Therefore MF and AM have the same time complexity. However, AM needs more computations because there are two essential differences between AM and MF.

Algorithm 1 Aspect Model Training Algorithm According to [25]

```
loop {repeat until convergence}
    for $r_{ui}$ in $S$ do
        for $f \leftarrow 1, ..., k$ do
            compute E-Step for each $f$
        end for
        for $f \leftarrow 1, ..., k$ do
            update $p(f|u)$, $\mu_{i,f}$ , and $\sigma_{i,f}$
        end for
    end for
    for $f \leftarrow 1, ..., k$ do
        normalize $p(f|u)$
    end for
    check the convergence
end loop
```

First, the learning algorithm of MF uses the gradient descent but AM is based on expectation maximization. While in the gradient descent the gradient is computed just by one training sample, in the expectation maximization the amount of change should be computed using all training data. This step is called E-step [24]. The time complexity of the E-step is $O(L \times |S| \times k)$. The second difference is that as AM is a probabilistic approach, the user features must be normalized so the summation of probabilities becomes 1. But MF is an algebraic approach, so it is not necessary to normalize the user features. The time complexity of the normalization is $O(L \times k)$. Finally though the maximum number of iterations L is the same for AM and MF (100 in our experiments), but the effective L in MF is lower than the effective L in AM, because MF converges faster than AM which consequently cuts down the training time.

5 Experimental Results

As the main challenge in applying active learning for recommender systems is that users are not willing to answer many queries in order to rate the queried items, we evaluate AM and MF with respect to their accuracy on the new users in terms of prediction error versus the number of queried items (simply denoted as number of queries). The mean absolute error (MAE) is used to evaluate the performance of each test user u :

$$MAE_u = \frac{1}{|M_u|} \sum_{i \in M_u} |r_{ui} - \hat{r}_{ui}| \tag{6}$$

where M_u is the set of test items of user u, r_{ui} is the true rating of user u for item i, and $\hat{r}_{ui}$ is the predicted rating. Since the test dataset includes multiple users, the reported MAE is the average over individual MAE for each test user.

5.1 Data Set

We use the MovieLens(100K)[1] dataset in our experiments. MovieLens contains 943 users and 1682 items. The dataset was randomly split into training and test sets. The training dataset consists of 343 users (the same number used in [5]) and the rest of the users are in the test dataset. Each test user is considered as a new user. The latent features of the new user are initially trained with three random ratings. 20 rated items of each test user are separated to compute the error. The test items are not new and already appeared in the training data. The remaining items are in the pool dataset, i.e the dataset that is used to select a query. For simplicity, we assume that the new user will always be able to rate the queried item. Of course, this is not a realistic assumption because there are items that the new user has not seen before, so it is not possible for him/her to provide the rating. As the focus of this paper is on the suitable prediction model for active learning in recommender system, we will leave this issue for future work. In our experiment, 10 queries are asked from each new user. Therefore, the pool dataset should contain at east 10 items which exist in the training data. Considering 10 queries and 20 test items, each test user has given ratings to at least 30 items. The number of latent dimensions k is 10 according to [5].

5.2 Results

In this section, we compare the accuracy of the active learning algorithm based on MF with the active learning algorithm based on AM. The objective is to show that MF is a better prediction model to be used for developing the active learning algorithm. For this reason, in order to have a fair comparison we focus only on the prediction model and simply apply random selection of the queried items for both MF and AM.

Learning the new user's features usually starts with 3 initial ratings [5, 4]. This can be done in two different ways. The first option is to add the ratings to the training user dataset and train AM or MF with all users together. The further retraining of the new user is done using the online updating technique. The second way is to train the prediction model (AM or MF) only with training users, and then train the new user with three initial ratings using the online updating technique.

For AM, both ways provide the same initial error, i.e before asking any query. But for MF, the error is lower when online updating is used from the beginning (i.e the second way). This evidence shows a new solution to improve the accuracy of MF. MF can not make accurate predictions for users with few ratings [26]. Therefore, after training all users and items, the latent features of such users can be retrained using the online updating technique. This is an open door for further research.

Now we move on to compare MF and AM for 10 queries. Fig. 1 depicts the resulting MAE as a function of the number of queried items. MF outperforms

[1] www.grouplens.org/system/files/ml-data0.zip

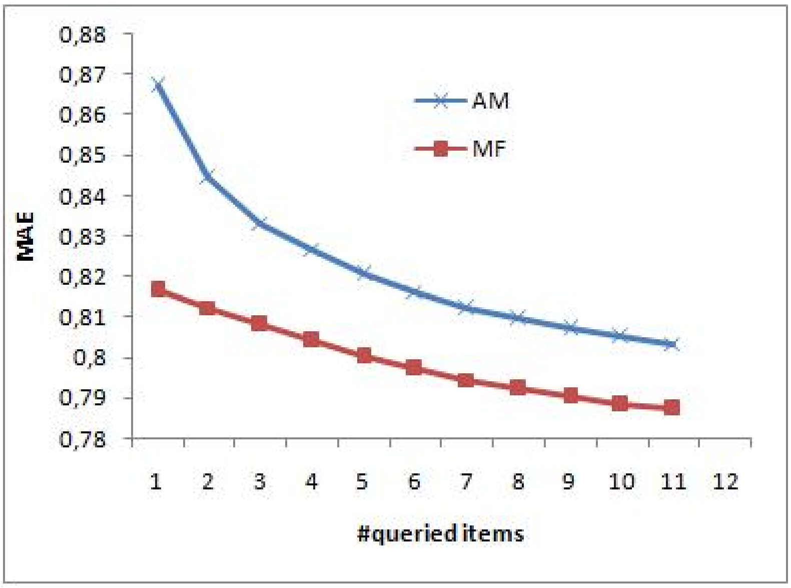


Fig. 1. Active Learning trends for 10 active-iterations

AM, indicating its superiority as the prediction model. In addition to the accuracy, the time required to retrain the new users latent features is also important. It is because the preference elicitation of the new user is an interactive scenario and long time interruptions make the new users leave the conversation. Table 1 compares the retraining time of new users latent features in AM and MF. Although both of them have the same complexity, but due to the reasons that have already been mentioned, MF is faster than AM.

Table 1. Retraining time of new users latent features in AM and MF

	Aspect Model	Matrix Factorization
MovieLens	44.5s	3.9s

6 Conclusion

In this paper, we proposed to develop active learning methods based on matrix factorization. We compared the training algorithm of matrix factorization with the aspect model and showed that matrix factorization is faster and its accuracy is also better.

As the future work, we plan to conduct online survey to validate the significance of our offline evaluation. To this end, it is crucial to design a software with a user-friendly user interface to encourage users to cooperate with the system [19].

References

1. G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.
2. J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl, "GroupLens: Applying collaborative filtering to usenet news," *Communications of the ACM*, vol. 40, no. 3, pp. 77–87, 1997.
3. Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, pp. 30–37, 2009.
4. R. Jin and L. Si, "A bayesian approach toward active learning for collaborative filtering," in *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, 2004.
5. A. S. Harpale and Y. Yang, "Personalized active learning for collaborative filtering," in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval.* ACM, 2008, pp. 91–98.
6. A. M. Rashid, I. Albert, D. Cosley, S. K. Lam, S. M. McNee, J. A. Konstan, and J. Riedl, "Getting to know you: Learning new user preferences in recommender systems," in *International Conference on Intelligent User Interfaces (IUI).* ACM Press, 2002, pp. 127–134.
7. N. Golbandi, Y. Koren, and R. Lempel, "Adaptive bootstrapping of recommender systems using decision trees." in *WSDM.* ACM, 2011, pp. 595–604.
8. A. M. Rashid, G. Karypis, and J. Riedl, "Learning preferences of new users in recommender systems: an information theoretic approach," *SIGKDD Explor. Newsl.*, vol. 10, no. 2, pp. 90–100, Dec. 2008.
9. A. Kohrs and B. Merialdo, "Improving collaborative filtering for new users by smart object selection," in *International Conference on Media Features (ICMF)*, 2001.
10. C. Boutilier, R. S. Zemel, and B. Marlin, "Active collaborative filtering," in *Conference on Uncertainty in Artificial Intelligence(UAI)*, 2003.
11. S. Tong and D. Koller, "Active learning for parameter estimation in bayesian networks," in *Advances in Neural Information Processing Systems(NIPS)*, 2000.
12. R. Karimi, C. Freudenthaler, A. Nanopoulos, and L. Schmidt-Thieme, "Active learning for aspect model in recommender systems," in *IEEE Symposium on Computational Intelligence and Data Mining (CIDM).* IEEE, 2011.
13. R. Karimi, C. Freudenthaler, A. Nanopoulos, and L. Schmidt-Thiemee, "Non-myopic active learning for recommender systems based on matrix factorization," in *IEEE Information Reuse and Integration (IRI).* IEEE, 2011.
14. R. Karimi, C. Freudenthaler, A. Nanopoulos, and L. Schmidt-Thieme, "Towards optimal active learning for matrix factorization in recommender systems," in *23th IEEE International Conference on Tools With Artificial Intelligence (ICTAI)*, 2011.
15. R. Karimi, C. Freudenthaler, A. Nanopoulos, and L. Schmidt-Thiemee, "Exploiting the characteristics of matrix factorization for active learning in recommender systems," in *RecSys*, 2012, pp. 317–320.
16. R. Karimi, M. Wistuba, A. Nanopoulos, and L. Schmidt-Thieme. Factorized decision trees for active learning in recommender systems. In *25th IEEE International Conference on Tools With Artificial Intelligence (ICTAI)*, 2013.

17. R. Karimi, A. Nanopoulos, and L. Schmidt-Thieme. Improved Questionnaire Trees for Active Learning in Recommender Systems. Proceedings of the 16th LWA Workshops: KDML, IR and FGWM, Aachen, Germany, September 8-10, 2014.
18. R. Karimi, A. Nanopoulos, and L. Schmidt-Thieme. A supervised active learning framework for recommender systems based on decision trees. User Model. User-Adapt. Interact. 25(1): 39-64 (2015).
19. R. Karimi, Fuzzy Model View Controller Pattern in International Conference on Advances in Intelligent Systems, Theory and Applications in cooperation with IEEE Computer Society, 2004.
20. K. Zhou, S.-H. Yang, and H. Zha, "Functional matrix factorizations for cold-start recommendation," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, ser. SIGIR '11. ACM, 2011, pp. 315–324.
21. Pilászy I, Tikk D (2009) Recommending new movies: even a few ratings are more valuable than metadata. In: RecSys, pp 93–100
22. Zhang L, Meng XW, Chen JL, Xiong SC, Duan K (2009) Alleviating cold-start problem by using implicit feedback. In: Proceedings of the 5th International Conference on Advanced Data Mining and Applications, Springer-Verlag, Berlin, Heidelberg, ADMA '09, pp 763–771
23. Rendle S, Schmidt-Thieme L (2008) Online-updating regularized kernel matrix factorization models for large-scale recommender systems. In: ACM Conference on Recommender Systems (RecSys), ACM, pp 251–258
24. Hofmann T, Puzicha J (1999) Latent class models for collaborative filtering. In: International Joint Conference on Artificial Intelligence, Morgan Kaufmann Publishers Inc., pp 688–693
25. Hofmann T (2003) Collaborative filtering via gaussian probabilistic latent semantic analysis. In: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrievall, ACM, pp 259–266
26. Salakhutdinov R, Mnih A (2008) Probabilistic matrix factorization. In: Advances in Neural Information Processing Systems (NIPS 2007), pp 134–141

A Novel Kernelized Classifier Based on the Combination of Partially Global and Local Characteristics

Riadh Ksantini[2] and Raouf Gharbi[1]

[1] Department of Computer Networks
Université Internationale de Tunis, Tunis, 2035 Tunisia
Tel.: +216 71 809 000
gharbiraouf@outlook.com
[2] University of Windsor, Windsor, ON N9B 3P4 Canada.
SUP'COM. Research Unit: Sécurité Numérique. Tunisia.
ksantini@uwindsor.ca

Abstract. The Kernel Support Vector Machine (KSVM) has achieved promising classification performance. However, since it is based only on local information (Support Vectors), it is sensitive to directions with large data spread. On the other hand, Kernel Nonparametric Discriminant Analysis (KNDA) is an improvement over the more general Kernel Fisher Discriminant Analysis (KFD), where the normality assumption from KFD is relaxed. Furthermore, KNDA incorporates the partially global information in the Kernel space, to detect the dominant normal directions to the decision surface, which represent the true data spread. However, KNDA relies on the choice of the κ-nearest neighbors ($\kappa - NN$'s) on the decision boundary. This paper introduces a novel Combined KSVM and KNDA (CKSVMNDA) model which controls the spread of the data, while maximizing a relative margin separating the data classes. This model is considered as an improvement to KSVM by incorporating the data spread information represented by the dominant normal directions to the decision boundary. This can also be viewed as an extension to the KNDA where the support vectors improve the choice of κ-nearest neighbors ($\kappa - NN$'s) on the decision boundary by incorporating local information. Since our model is an extension to both SVM and NDA, it can deal with heteroscedastic and non-normal data. It also avoids the small sample size problem. Interestingly, the proposed improvements only require a rigorous and simple combination of KNDA and KSVM objective functions, and preserve the computational efficiency of KSVM. Through the optimization of the CKSVMNDA objective function, surprising performance gains were achieved on real-world problems.

[1]Corresponding Author.

Keywords: Kernel Nonparametric Discriminant Analysis, Kernel Support Vector Machines, Partially Global Information, Local Information, Small Sample Size Problem.

1 Introduction

Supervised learning is the task of finding a function which relates inputs and targets. A training set $\mathcal{X}$ of input vectors $\{x_i\}_{i=1}^N$ is given, where $x_i \in \mathbb{R}^k (k \geq 1) \quad \forall i = 1, 2, ..., N$. The corresponding set $\mathcal{T}$ of tags are $\{t_i\}_{i=1}^N$, where $t_i \in 0, 1 \quad \forall i = 1, 2, ..., N$. The objective is to learn a model of dependency of the targets on the inputs. The ultimate goal is to be able to make accurate predictions of t for unseen values of x. Typically, we base our predictions upon some function $y(x)$ defined over the input/training space $\mathcal{X}$, and learning is the process of inferring the parameters of this function. A new representation of data is necessary to learn non-linear relations with a linear classifier. This is equivalent to applying a fixed non-linear mapping $\mathcal{F}$ of the data to a feature space, in which the linear classifier can be used. Hence, the objective function will be of the form:

$$y(x; w) = \sum_{i=1}^{N} f_i^x w_i + w_0 = \Phi^T(x_i)\mathbf{w} + w_0, \tag{1}$$

where $\Phi(x) = (f_1^x, f_2^x, \ldots, f_N^x) : \mathcal{X} \to \mathcal{F}$ describes a non-linear mapping from the input space to a feature space for the input variable x. Hence, non-linear classifiers have two stages: (i) a fixed non-linear mapping transforms the data into a feature space $\mathcal{F}$ and then (ii) a linear classifier is used to classify them in $\mathcal{F}$. Analysis of functions of the type (1) is facilitated since the adjustable weight vector $\mathbf{w}$ and the offset w_0 appear linearly, and the objective is to estimate optimum values of the weight coefficients. There are a large number of functions of type (1). Our concentration here is on some relevant state-of-the-art kernel-based models, such as, the Kernel Support Vector Machine (KSVM) and the Nonparametric Discriminant Analysis in kernel space, which we will call KNDA. KNDA extends the linear NDA based on the same principles that the Kernel Fisher Discriminant Analysis (KFD) is built upon. The advantage of KNDA over KFD is the relaxation of normality assumption. KNDA measures the between-class scatter matrix on a local basis in the neighborhood of the decision boundary in the higher dimensional feature space. This is based on the observation that the normal vectors on the decision boundary are the most informative for discrimination. In case of a two-class classification problem, these normal vectors are approximated by the $\kappa - NN$'s from the other class for one point. We can consider KND as a classifier based on the "near-global" characteristics of data. Although KNDA gets rid of the underlying assumptions of KFD and results in better classification performance, no additional importance is given to the boundary samples. In other words, the margin criterion (as calculated in KSVM) is not considered here. Moreover, it is not always an easy task to find a common and appropriate choice of $\kappa - NN$'s on the decision boundary for all data points to obtain the best linear discrimination.

Another category of kernel-based classifiers is the Kernel Support Vector Machine (KSVM). KSVM is based on the idea of maximizing the margin or degree of separation in the training data. There are many hyperplanes which can divide the data between two classes for classification. One reasonable choice for the optimal hyperplane is the one which represents the largest separation or margin between the two classes. KSVM tries to find the optimal hyperplane using support vectors. The support vectors are the training samples that approximate the optimal separating hyperplane and are the most difficult patterns to classify. In other words, they are consisted of those data points which are closest to the optimal hyperplane. As KSVM deals with a subset of data points (support vectors) which are close to the decision boundary, it can be said that the KSVM solution is based on the "local" variations of the training data.

It has been shown in the literature that maximum margin based classifiers like the KSVM typically perform better than discriminant (or average margin) based methods like the KNDA due to their robustness and local margin consideration. However, KSVM can perform poorly when the data varies in such a way that data points exist far from the classification boundary [12]. This can be the case especially when the data is of high dimension. This is because KSVM does not take into consideration the "near-global" properties of the class distribution (as in the case of KNDA). This limitation of KSVM can be avoided by incorporating variational information from the KNDA which will control the direction of the separating hyperplane of KSVM. In that way we will have a maximum margin based classifier which is not sensitive to skewed data distribution like KSVM.

Several methods exist in literature which have addressed these issues inherent in discriminant based and maximum margin based methods. The ellipsoidal kernel machine was proposed in [11], where a geometric modification is proposed for data normalization by considering hyperellipsoids instead of hyperspheres in the classical KSVM method. Similarly, in [5], radius/margin bound has been used to iteratively optimize the parameters of KSVM efficiently. In [15], a kernel-based method has been proposed which essentially calculates the KFD scatter matrices based on the support vectors provided by KSVM. While these methods were backed by experimental improvements, most of them are a combination of multiple locally optimal algorithms to separately solve the discriminant based problem and margin maximization rather than providing one algorithm with one unique globally optimum solution.

Although the method proposed in [12] is superior to the previously described methods in the sense that it is based on only one convex optimization problem, it does so by introducing new constraints to the optimization problem. New constraints means new Lagrangian variables, which in turns can degrade the computational time. The Gaussian Margin Machine proposed in [3] tries to find the least informative distribution that classifies training data correctly by maintaining a Gaussian distribution of weight vectors. The drawback with this method is the expensive objective function involving log determinants in the optimization problem.

Another approach to improve the classification performance of KSVM is to include additional training examples. This approach has been used in [1], where additional unlabeled samples are made available to the system in a semi-supervised learning system. The approach in [14] introduces a *neither* class, where additional samples are drawn from the same distribution for the classes under consideration. These additional samples are then used for improved margin consideration. However, we will stick to the simple binary classification model which does not rely on any additional assumption and, hence, is closer to a real-life pattern recognition problem in its truest form.

We propose a novel CKSVMNDA model which combines the KNDA and KSVM methods. In that way, a decision boundary is obtained which reflects both near-global characteristics (realized by KNDA) of the training data in feature space and its local properties (realized by the local margin concept of the KSVM). Being a kernel-based model, CKSVMNDA can deal with nonlinearly separable data efficiently. Rather than introducing new constraints like [12], our method modifies the objective function of KSVM by incorporating the scatter matrices provided by KNDA.

The proposed method improves upon our recently proposed models [6, 7] by preserving the same discriminative way while adding the following significant advantages:

- Unlike the method in [6], our proposed model is more theoretically founded and forms a convex optimization problem because the final matrix used to modify the objective function is positive-definite. As a result, the method generates one global optimum solution. Because of this global extremum, existing numerical methods can be used to solve this problem easily and efficiently.
- The methods in [6, 7] primarily focused on the linear version of SVM while our model derivation emphasizes on the kernel space. As stated before, the kernel space has the advantage of being able to learn non-linear relations by mapping to a higher-dimensional feature space.

We also show that our method is a variation of the KSVM optimization problem, so that even existing KSVM implementations can be used. The experimental results on real and artificial datasets show the superiority of our method both in terms of accuracy.

The rest of the paper is organized as follows: Section 2 provides formulations of the KSVM and KNDA. Section 3 contains derivation of the novel CKSVM-NDA model. Section 4 provides a comparative evaluation of the CKSVMNDA model to the KSVM and KNDA methods. This evaluation is carried out on a number of benchmark real datasets. Finally, Section 5 provides some conclusions.

2 KSVM and KNDA

Let $\mathcal{X}_1 = \{x_i\}_{i=1}^{N_1}$ and $\mathcal{X}_2 = \{x_i\}_{i=N_1+1}^{N_1+N_2}$ be two different classes constituting an input space of $N = N_1 + N_2$ samples or vectors in $\mathbb{R}^M$ where, class $\mathcal{X}_1$ contains N_1 samples and class $\mathcal{X}_2$ contains N_2 samples. Let the associated tags with these vectors be represented by $\mathcal{T} = \{t_i\}_{i=1}^{N}$, where $t_i \in \{0, 1\} \;\; \forall i = 1, 2, \ldots, N$. Since real-life data has inherent non-linearity, KSVM tries to map the data samples to a higher dimensional feature space $\mathcal{F}$, where linear classification might be achieved. Let the function Φ map the classes $\mathcal{X}_1$ and $\mathcal{X}_2$ to two higher dimensional feature classes $\mathcal{F}_1 = \{\Phi(x_i)\}_{i=1}^{N_1}$ and $\mathcal{F}_2 = \{\Phi(x_i)\}_{i=N_1+1}^{N}$, respectively.

However, in case when the dimension of $\mathcal{F}$ is very high, it is not possible to do mapping directly. In such a case, the *kernel trick* [13] is used. Instead of explicitly calculating the mapping, a kernel function $\mathcal{K}$ is used, which calculates the dot products of the higher dimensional data samples instead of the samples themselves. Mathematically it can be written as

$$\mathcal{K}(x_i, x_j) =< \Phi(x_i).\Phi(x_j) >, \forall i, j \in \{1, 2, \ldots, N\}.$$

Our target is to learn the weight vector $\mathbf{w}$ which minimizes (or maximizes) some objective function of the form of Equation (1).

2.1 The Kernel Support Vector Machine

As stated before, KSVM tries to map the samples to a higher dimensional feature space in the hope that the classification problem will be linear in that space. In the feature space, KSVM tries to find the optimal decision hyperplane. The optimal hyperplane is the one with the largest margin, or, in other words, the plane which has largest minimal distance from any of the samples. Maximizing the distance of samples to the optimal decision hyperplane is equivalent to minimizing the norm of $\mathbf{w}$. As a result, this becomes part of the objective function. However, it might be the case that the problem is non-linear even in the higher dimensional space. To solve this, the margin constraint is relaxed or *slacked*. Also, a penalty factor is introduced in the objective function to control the amount of slack. This penalty factor is of the form of a loss function, usually a hinge loss function. Incorporating all these, the KSVM optimization problem can be written as:

$$\min_{\mathbf{w} \neq 0, w_0} \quad \left\{ \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N} max(0, 1 - t_i(\Phi^T(x_i)\mathbf{w} + w_0)) \right\},$$

Here, $max(0, 1 - t_i(\Phi^T(x_i)\mathbf{w} + w_0))$ is the hinge loss function. For correctly classified training samples, this function does not incur any loss. For misclassification, the loss factor is controlled by C. Note that although KSVM is generally described as an optimization problem with constraints on the weights, we are presenting it slightly differently with the hinge-loss function so that it will be easier to derive the probabilistic interpretation of our proposed method later. This representation can easily be converted to the more familiar constrained optimization problem.

Since the weight vector $\mathbf{w}$ resides in the feature space, it cannot be calculated directly. Instead, the Lagrangian dual problem is solved [10]. The optimal weight vector for this problem is a linear combination of the data points and is of the form $\mathbf{w}^* = \sum_{i=1}^{N} t_i \alpha_i^* \Phi(x_i)$, where $\{\alpha_i\}_{i=1}^{N}$ are the Lagrangian variables. The decision function for any test sample x is obtained by:

$$g(x) = \sum_{i=1}^{N} t_i \alpha_i^* \mathcal{K}(x, x_i) + w_0^*, \tag{2}$$

where w_0^* is computed using the primal-dual relationship, and where only samples with non-zero Lagrange multipliers α_i contribute to the solution. The corresponding data samples are called Support Vectors (SVs). These points are the crucial samples for classification. Therefore, KSVM considers only those data points which are close to the decision hyperplane and are critical to find the decision boundary. In other words, KSVM only considers the local variations in data samples. The overall distributions of the training samples are not taken into consideration. Incorporating some kind of global distribution (e.g. results from classifiers like KNDA) can provide better classification.

2.2 The Kernel Nonparametric Discriminant Analysis

The NDA can be extended to the feature space $\mathcal{F}$. We call this the Kernel Nonparametric Discriminant Analysis (KNDA). Instead of calculating the simple mean vectors, the nearest neighbor mean vectors are calculated to formulate the between-class scatter matrix of the NDA. In our feature space, this vector can be defined as:

$$M_m^{\kappa}(\Phi(x_i)) = \frac{1}{\kappa} \sum_{j=1}^{\kappa} \Phi(x)_{NN}(j), \tag{3}$$

where, $\Phi(x_i)_{NN}(j)$ defines the j^{th} nearest neighbor from data point x_i of class m. κ is the free parameter which defines how many neighbors to consider. This parameter needs to be optimized for each dataset. Now, let us define two matrices $L_1(\Phi(x_i))$ and $L_2(\Phi(x_i))$. We will use the kernel trick to formulate these matrices. In that case, the matrices are calculated on a component by component basis, where, a component of $L_1(\Phi(x_i))$ is defined as:

$$\begin{aligned}&(L_1(\Phi(x_i)))_j = \mathcal{K}(x_j, x_i) - (M_2^{\kappa}(\Phi(x_i)))_j,\\ &\forall i \in \{1, 2, \ldots, N_1\}, \forall j \in \{1, 2, \ldots, N\},\end{aligned} \tag{4}$$

and a component of $L_2(\Phi(x_i))$ is defined as

$$\begin{aligned}&(L_2(\Phi(x_i)))_j = \mathcal{K}(x_j, x_i) - (M_1^{\kappa}(\Phi(x_i)))_j,\\ &\forall i \in \{N_1 + 1, N_1 + 2, \ldots, N_1 + N_2\}, \forall j \in \{1, 2, \ldots, N\}.\end{aligned} \tag{5}$$

With these formulations, the between-class scatter matrix in the feature space can be defined as:

$$\nabla = \frac{1}{(N_1 + N_2)} \sum_{i=1}^{N_1} \Psi_i L_1(\Phi(x_i)) L_1(\Phi(x_i))^T$$
$$+ \frac{1}{(N_1 + N_2)} \sum_{i=N_1+1}^{N_1+N_2} \Psi_i L_2(\Phi(x_i)) L_2(\Phi(x_i))^T. \tag{6}$$

Here, Ψ_i are the weighting functions to nullify the effects of samples that are far from the boundary. It is defined as follows [4]:

$$\Psi_i = \frac{min\{d(\Phi(x_i), \Phi(xNN_{1i}^{\kappa}))^{\gamma}, d(\Phi(x_i), \Phi(xNN_{2i}^{\kappa}))^{\gamma}\}}{d(\Phi(x_i), \Phi(xNN_{1i}^{\kappa}))^{\gamma} + d(\Phi(x_i), \Phi(xNN_{2i}^{\kappa}))^{\gamma}}, \tag{7}$$

where γ is a control parameter which can range from zero to infinity, and $d(\Phi(x_i), \Phi(xNN_{ji}^{\kappa}))$ is the Euclidean distance from x_i to its $\kappa - NN$'s from class $\mathcal{X}_j$ in the kernel space. γ controls how rapidly the value of weighting function falls to zero as we move away from the classification boundary.

The motivation behind KNDA is the observation that essentially the nearest neighbors represent the classification structure in the best way. For small values of κ, the matrices in Equation (4) and (5) represent the direction of the gradients of the respective class density functions in the feature space. If the weighting functions are not used, samples with large gradients that are far from the boundary may pollute the necessary information. Hence, these gradients with combination of the weighting functions form the between-class scatter matrix ∇, which preserves the classification structure.

The KNDA does not make any modifications to the within-class scatter matrix. As a result, the formula for within-class scatter matrix Δ is similar to the KFD, and can be written as follows:

$$\Delta = \mathbf{K}_1(I - 1_{N_1})\mathbf{K}_1^T + \mathbf{K}_2(I - 1_{N_2})\mathbf{K}_2^T, \tag{8}$$

where $\mathbf{K}_1$ is a $N \times N_1$ Kernel matrix for the class $\mathcal{X}_1$ and $\mathbf{K}_2$ is a $N \times N_2$ Kernel matrix for the class $\mathcal{X}_2$. I is the identity matrix and 1_{N_1} and 1_{N_2} are the matrices with all entries $\frac{1}{N_1}$ and $\frac{1}{N_2}$, respectively. With these definitions of ∇ and Δ, the KNDA method proceeds by computing the eigenvectors and eigenvalues of $\Delta^{-1}\nabla$. Since the higher dimensional feature space $\mathcal{F}$ is of dimension N, the matrix Δ is needed to be regularized before calculating the inverse. This is achieved by adding a small multiple β of the identity matrix I. Hence, the eigenvectors and eigenvalues of $(\Delta + \beta I)^{-1}\nabla$ are computed, and the eigenvector corresponding to the largest eigenvalue forms the optimal decision hyperplane. We can exploit the fact that the matrix ∇ is only of rank one (i.e., $\alpha^T \nabla \alpha = \frac{1}{(N_1+N_2)} \sum_{i=1}^{N_1} \Psi_i \big(\alpha^T L_1(\Phi(x_i)) L_1(\Phi(x_i))^T\big) + \frac{1}{(N_1+N_2)} \sum_{i=N_1+1}^{N_1+N_2} \Psi_i \big(\alpha^T L_2(\Phi(x_i))$ $L_2(\Phi(x_i))^T\big)\big)$. Thus, we can fix $\alpha^T \nabla \alpha$ to any non-zero value, for example 1 and minimize $\alpha^T \Delta \alpha$. This amounts to the following quadratic optimization problem:

$$\min_{\alpha \neq 0, \alpha_0} \quad \alpha^T \varDelta \alpha, \tag{9}$$

$$s.t. \quad \alpha^T \nabla \alpha = 1. \tag{10}$$

3 The CKSVMNDA Model

In this section, we present our proposed model CKSVMNDA which combines the data spread information represented by the normal vectors to the decision surface for the KNDA (partially global information), and the support vectors for the KSVM (local information). Thus, the CKSVMNDA overcomes the drawbacks of KSVM by controlling the spread of the data, which is represented by the KNDA dominant normal directions to the decision boundary, while maximizing a relative margin separating the data classes. Moreover, the choice of KNDA κ-nearest neighbors ($\kappa - NN$'s) on the decision boundary is improved by the KSVM support vectors. Therefore, the CKSVMNDA objective function is a simple and rigorous summation of the KSVM and KNDA objective functions:

$$\min_{\alpha \neq 0, \alpha_0} \quad \Big\{ \frac{1}{2} \alpha^T \big[2\lambda \big(\varDelta + \beta \nabla \big) + I \big] \alpha - \lambda \beta \tag{11}$$

$$+ C \sum_{i=1}^{N} max(0, 1 - t_i(\varPhi^T(x_i)\alpha + \alpha_0)) \Big\}. \tag{12}$$

In theory, CKSVMNDA should outperform both KSVM and KNDA if the control parameter λ can be optimally chosen. In practice, the values of λ and β will be tuned via the cross validation technique, where data is divided into a number of subsets. Then, one subset is used for testing while the others are used for training. All the subsets are used for testing in turns and the average is taken into consideration to reduce variability. This whole process is repeated with different values of λ and β. The latter are assigned the values with the best performance.

3.1 Solving the Optimization Problem

Since our optimization problem is similar to the KSVM optimization problem, we can solve it in a similar way, i.e., by using Lagrange multipliers. However, obtaining the CKSVMNDA solution this way requires an entirely new implementation to test this method. The following lemma gives us an easier alternative to implement this method:

Lemma 1. *The CKSVMNDA method formulation is equivalent to:*

$$\min_{\hat{\mathbf{w}} \neq 0, w_0} \quad \Big\{ \frac{1}{2} \hat{\mathbf{w}}^T \hat{\mathbf{w}} + C \sum_{i=1}^{N} max(0, 1 - t_i(\hat{\varPhi}^T(x_i)\hat{\mathbf{w}} + w_0)) \Big\}, \tag{13}$$

where

$$\hat{\mathbf{w}} = \Theta^{1/2}\mathbf{w}, \tag{14}$$

$$\hat{\Phi}(x_i) = \Theta^{-1/2}\Phi(x_i) \quad \forall i = 1, \ldots, N \tag{15}$$

and

$$\Theta = \eta\Delta(\nabla + \beta I)^{-1}\Delta + I. \tag{16}$$

Proof. Substituting Equations (14-16) into equation (13) we get the original CKSVMNDA problem (Equation (11)).

This lemma gives us a significant advantage from the implementation viewpoint. This essentially means that we can use the existing SVM implementations [8] provided we can calculate the terms $\Theta^{1/2}$ and $\Theta^{-1/2}$. The algorithm used to solve the optimization problem in this implementation is based on the interior-reflective Newton method described in [2].

4 Experimental Results

In this section we evaluate the proposed CKSVMNDA method against three other contemporary classifiers, namely, the KSVM, KNDA and the Kernel Fisher Discriminant (KFD). To strengthen the significance of our method, we provide results for both real-world datasets and a face recognition application.

For kernelization of the data, we use the Gaussian RBF Kernel $\mathcal{K}(x_i, x_j) = e^{-\|x_i - x_j\|^2/\sigma}$. This kernel is proven to be robust and flexible. Here, σ represents the positive "width" parameter. For KNDA and KFD, after finding the optimal eigenvector, Bayes classifier was used for conducting the final classification.

The involved parameters were optimized using exhaustive search to try all possible combinations. Although the parameter optimization is a lengthy process, this needs to be done only once for a new dataset, and, hence, does not contribute to the actual classification performance. If the optimization needs to be faster, efficient methods like coordinate descent technique can be used at the cost of a small degradation in accuracy values.

The number of parameters to tune for the CKSVMNDA method is 4, while it is 2 for KSVM and KNDA. It might seem that an accurate fit of the parameter values is necessary for CKSVMNDA to perform well, specially if we have a small training dataset. But as we will see from the results, CKSVMNDA performs better compared to other methods by tuning over a limited range of parameter values we have used (e.g. we use a set of only 20 κ values and 20 η values for parameter tuning to obtain the results of Table 1). Since this is a combination of KSVM and KNDA, the parameters compensate each other, and the fit doesn't necessarily have to be perfect. Also, for small training set, we tackle the problem of poor performance due to inaccuracies in matrix inversion by adding a regularization term before inverting.

4.1 Experiments on Real and Artificial Datasets

We have applied the classification algorithms on 11 real-world and artificial datasets.The datasets are obtained from the Benchmark Repository used in [9]. Namely, the datasets are: Flare-Sonar, Breast-Cancer, German, Heart, Banana, Diabetes, Ringnorm, Thyroid, Twonorm, Waveform and Splice. These datasets are obtained from the UCI, DELVE and STATLOG repositories. Some of these datasets are originally multi-class. In such cases, some of the classes were (randomly) merged to convert it into a two-class classification problem. 100 partitions are then generated for each dataset, where about 60% data is used for training and the rest for testing [9]. For our experimental results, we randomly picked 5 out of these 100 partitions (5 partitions each for training and 5 each for testing). Additionally, we repeated this random picking process 5 times to achieve the average result. This randomness was introduced to ensure that no method has a coincidental advantage over the others. For parameter tuning, 5-fold cross validation on the training dataset was performed for each model (i.e. 4 out of the 5 picked training partitions were used for training and the remaining one for validation at each stage of cross validation).

4.2 Interpretation of the Results

Accuracy

Table 1 contains the average accuracy values and the standard deviations obtained over all the runs. We see that the CKSVMNDA method outperforms the KSVM, KNDA and the KFD in almost all cases. Since the CKSVMNDA combines the global and near-global variations provided by the KSVM and the KNDA, respectively, it can classify the relatively difficult test samples. Also, being a variation of the KSVM and KNDA, this method is free from any underlying distribution assumption, and, hence, can provide better results. Concerning the parameters λ and β, in order to reduce the time of optimization, we had to restrict ourselves to only a few values. Still, as we can see, these limited values are good enough for almost all the datasets. This establishes the fact that our method can be used in practical applications. To measure the statistical significance of the results, we paired up the CKSVMNDA method with the other methods and performed paired t-tests on the accuracy values. The paired t-test determines whether or not two paired sets of measured values are significantly different. The last row of Table 1 provides the confidence intervals (in %) obtained from the performed t-tests. This confidence interval quantifies the probability of the paired distributions being the same. The higher the confidence interval, the lower is the probability that the underlying distributions are statistically indifferent. As we can see, all the confidence intervals are almost 100%, which proves that the CKSVMNDA method indeed provides statistically significant accuracy improvements.

If we compare the results between the KNDA and KFD, we see that in some cases, the KFD provides better classification results than the KNDA. This is due to the fact that the optimal nearest neighbor parameter for the KNDA (the

$\kappa - NN$'s) is not always easy to find. But since our method combines the KNDA with KSVM, the optimality of this parameter is not as crucial as it is in the KNDA.

Computational Complexity Analysis

The computational complexity of the KSVM scales with $\mathcal{O}(N^2)$ for one iteration. The KNDA and KFD scale with a computational complexity of $\mathcal{O}(N^3)$ (dominated by the inversion of the within-class scatter matrix). Each of the KNDA and KFD methods requires only one run as there is no iterative process involved.

In the CKSVMNDA, the complexity for the inversion of Θ scales with $\mathcal{O}(N^3)$. However, this inversion process can be considered to be part of pre-processing, as it is needed to be done only once before start of the training. Therefore, the computational complexity of our proposed CKSVMNDA can be considered similar to that of the KSVM, i.e., $\mathcal{O}(N^2)$ per iteration. This can also be seen from the obtained results (second last row of Table 1), where we see that the average computational time of our method is on par with that of KSVM.

Dataset	CKSVMNDA	KSVM	KND	KFD
Flare-Sonar	**67.7** (0.47)	66.9 (0.41)	*67.1* (0.65)	66 (0.40)
Breast-Cancer	*78.9* (1.96)	77.4 (2.1)	**79.3** (2.00)	77 (2.37)
German	**78.1** (0.40)	*77* (0.38)	76.3 (0.68)	75.7 (0.51)
Heart	**86.5** (2.21)	*85.4* (2.3)	81.7 (1.58)	82.9 (2.13)
Banana	**89.8** (0.25)	*89.6* (0.29)	*89.6* (0.22)	89.5 (0.20)
Diabetes	**78.6** (0.50)	*77.7* (0.69)	75.7 (0.90)	77.3 (1.03)
Ringnorm	**98.5** (0.04)	*98.4* (0.04)	98.3 (0.03)	97.4 (0.07)
Thyroid	**97.3** (0.6)	96.5 (1.02)	*97.1* (0.64)	96.8 (0.49)
Twonorm	**97.7** (0.04)	*97.6* (0.05)	96.5 (0.32)	96.9 (0.08)
Waveform	**90.7** (0.15)	*90.5* (0.15)	89.3 (0.19)	90 (0.12)
Splice	**88.9** (0.41)	*88.7* (0.38)	88.5 (0.41)	88.4 (0.36)
Avg. time	4.04	4.05	2.95	2.92
Confidence	-	99.8	97.6	99.9

Table 1. Average percentage classification accuracy and standard deviation (in parentheses) of each method for the 11 data sets (best method in **bold**, second best *emphasized*). The last two rows contain the average cpu time for each method (in *seconds*) and the t-test confidence interval, respectively.

5 Conclusion

In this paper, we have proposed a novel classification method named CKSVMNDA. The CKSVMNDA method incorporates the global variational information from the KSVM and the near-global information from the KNDA. Being a combination of these two methods, CKSVMNDA is a robust classifier, free from any underlying assumption regarding class distribution. Our method is also capable of tackling the small sample size problem. Being a convex optimization problem, our method provides a global optimum solution and can be solved efficiently by using numerical methods. Besides, we have shown that our method can be reduced to the classical KSVM model so that existing KSVM implementations can be used. The experimental results on some contemporary datasets verifies the superiority of our method, where we compare CKSVMNDA with the KSVM, KND and KFD. In future, we plan to build a multi-class classifier based on the principles of the CKSVMNDA method.

References

1. Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. On Manifold Regularization. In *Proceedings of the Artificial Intelligence and Statistics*, 2005.
2. T.F. Coleman and Y. Li. A reflective newton method for minimizing a quadratic function subject to bounds on some of the variables. *SIAM Journal on Optimization*, 6(4):1040–1058, 1996.
3. Koby Crammer, Mark Dredze, and Fernando Pereira. Exact Convex Confidence-Weighted Learning. *Advances in Neural Information Processing Systems 21*, 2009.
4. K. Fukunaga. *Introduction to Statistical Pattern Recognition, second ed.* Academic Press, 2000.
5. S.S. Keerthi. Efficient Tuning of SVM Hyperparameters Using Radius/Margin Bound and Iterative Algorithms. *IEEE Transactions on Neural Networks*, 13(5):1225–1229, Sep 2002.
6. N.M. Khan, R. Ksantini, I. Ahmad, and B. Boufama. A novel SVM+NDA model for classification with an application to face recognition. *Pattern Recognition*, 45(1):66–79, 2012.
7. R. Ksantini and B. Boufama. Combining partially global and local characteristics for improved classification. *Int. J. Machine Learning & Cybernetics*, 3(2):119–131, 2012.
8. MATLAB Bioinformatics Toolbox. The mathworks$^{\mathrm{TM}}$, 2011.
9. G. Ratsch, T. Onoda, and K.R. Muller. Soft Margins for Adaboost. *Machine Learning*, 42(3):287–320, 2000.
10. B. Scholkopf and A. Smola. *Learning With Kernels-Support Vector Machines, Regularization, Optimization and Beyond.* MA: MIT Press, Cambridge, 2001.
11. P.L. Shivaswamy and T. Jebara. Elliposoidal Kernel Machines. In *Proceedings of the Artificial Intelligence and Statistics*, 2007.
12. P.L. Shivaswamy and T. Jebara. Maximum relative margin and data-dependent regularization. *Journal of Machine Learning Research*, 11:747–788, 2010.
13. V.N. Vapnik. *Statistical Learning Theory.* John Wiley & Sons, New York, USA, 1998.
14. J. Weston, R. Collobert, F. H. Sinz, L. Bottou, and V. Vapnik. Inference with the universum. In *Proceedings of the International Conference on Machine Learning*, pages 1009–1016, 2006.
15. Baochang Zhang, Xilin Chen, Shiguang Shan, and Wen Gao. Nonlinear face recognition based on maximum average margin criterion. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 554 – 559, 2005.

New Graph regularized Sparse Coding Improving Automatic Image Annotation

Céline RABOUY[1,2], Sébastien PARIS[1,2] and Hervé GLOTIN[1,2,3]

[1]Aix-Marseille Université, CNRS, ENSAM, LSIS UMR 7296, 13397 Marseille, France
[2]Université de Toulon, CNRS, LSIS UMR 7296, 83957 La Garde, France
[3]Institut Universitaire de France, 75005 Paris, France
{celine.rabouy,sebastien.paris}@lsis.org
glotin@univ-tln.fr

Abstract. Typical image classification pipeline for shallow architecture can be summarized by the following three main steps: i) a projection in high dimensional space of local features, ii) sparse constraints for the encoding scheme and iii) a pooling operation to obtain a global representation invariant to common transformation. Sparse Coding (SC) framework is one particular example of this general approach. The main problem raised by it is the local feature encoding which is done independently, loosing correlation of the input space. In this work we propose to simultaneously encode sparse codes to tackle this problem with Joint Sparse Coding (JSC) inspired by Graph regularized Sparse Coding (GSC). We experiment SC, GSC and JSC on UIUCsports and scenes15 database. We will show that results obtained, for UIUCsports, with SC (87.27 ± 1.33), JSC (84.17 ± 1.57) and the State-of-the-Art (88.47 ± 2.32 [23]) are tackled by a simple fusion (95.37 ± 1.29). Several assumptions will be advanced to explain this phenomenon which can't be generalized.

Keywords: Scenes categorization, Sparse Coding, Graph regularized Sparse Coding, Dictionary Learning, Scale Invariant Feature Transform, Spatial Pyramid Matching, Joint Sparse Coding.

1 Introduction

In the field of computer vision and signal processing, significant progress has been made since the 2000s with more general methods such as Bag of Words (BoW) [19]. We have at our disposal a significant number of databases as, for example, UIUCsportss [11], scenes from 15 databases [8], where the goal is to label images into a finite number of classes. The first way could be to evaluate the metric distance between two images. Unfortunately, due to the high dimensionality of this input space, most of these distances are concentrated into a sub-manifold whatever the image class, making the discrimination by direct distances not robust. To overcome this problem, a solution has to be designed to find a general application $\Psi^j(.;\mu^j)$ with parameter μ^j which characterizes the class $\mathcal{C}^j$ satisfying:

$$\begin{cases} \text{dist}(\Psi^j(\mathbf{I}_1;\mu^j),\Psi^j(\mathbf{I}_2;\mu^j)) \to 0 & \text{if } \mathbf{I}_1 \in \mathcal{C}^j \text{ and } \mathbf{I}_2 \in \mathcal{C}^j \\ \text{dist}(\Psi^j(\mathbf{I}_1;\mu^j),\Psi^j(\mathbf{I}_2;\mu^j)) \to \infty & \text{if } \mathbf{I}_1 \in \mathcal{C}^j \text{ and } \mathbf{I}_2 \notin \mathcal{C}^j, \end{cases} \quad (1)$$

where $\mathbf{I}_1$ and $\mathbf{I}_2$ are two images. The choice of Ψ^j represents a trade-off between its representation capacity versus the μ^j optimization difficulty. In general, in order to estimate/optimize μ^j, we have to start from a local representation (patches) $\mathbf{x} \in \mathbb{R}^d$ to obtain the global representation $\Psi^j(.;\mu^j)$. From Ψ^j associated to BoW, Sparse Coding (SC) [21], up to ConvNet [3, 9] follow the three main procedures: i) high dimension local feature projection, ii) sparsity constraints into the representation model and iii) non-linearity operation and pooling to obtain a global invariant representation.

In this article, we will focus on a new formulation of encoding method, which corresponds more specifically to procedure ii), inspired by SC and more generally by Graph regularized Sparse Coding (GSC) [25]. This new formulation allows to encode simultaneously testing patches as with the GSC model which has good properties. Although we will only work on a single layer, we will show that a simple fusion will allow to improve considerably the classification accuracy and that our results will be close to CNN (convolutional neural nets) [6, 18] initialized on Image Net as shown in [3]. This article is divided into five parts. The first part focuses on SC models and its derivatives (GSC especially). The second part presents our modeling Joint Sparse Coding (JSC). The third part presents Graph regularized Sparse Coding (GSC) dictionary inspired by [13]. A fourth part presents results we obtained on UIUCsports and scenes15 databases and in the last part, we conclude on our contribution.

2 Related Works

In this part, we will focus on the encoding step using linear coding to reconstruct inputs. An approximation of any patches $\mathbf{x} \in \mathbb{R}^d$ can be given by $\mathbf{x}_i = \mathbf{D}\alpha_i$, where $\mathbf{D} \triangleq [\mathbf{d}_1,\ldots,\mathbf{d}_K] \in \mathbb{R}^{d\times K}$ is a given/trained dictionary where $\forall k = 1,\ldots,K,\ \|\mathbf{d}_k^T\mathbf{d}_k\|_2^2 = 1$ and $d_k^j \geq 0$. A patch is a vector extracted from an image. A dictionary is a matrix of "words" allowing the patch reconstruction. In many encoding methods, three common steps can be found: i) a projection into a higher dimension space with ($K >> d$) ii) sparse constraints and iii) a non-linear operation procedure. If α_i^* is obtained with Ordinary Least Square (OLS), the solution is full dense (all elements are non zero). One way to get around this problem is the use of the ℓ_1-norm constraint which corresponds to Lasso problem [21] or Basis Pursuit [4]:

$$\mathcal{L}_{SC}(\alpha_i|\mathbf{x}_i;\mathbf{D}) = \min_{\alpha_i\in\mathbb{R}^K} \frac{1}{2}\|\mathbf{x}_i - \mathbf{D}\alpha_i\|_2^2 + \lambda\|\alpha_i\|_1, \quad (2)$$

with λ the regularization parameter associated to the SC formulation. This parameter controls the sparsity level as is shown in [15]. Thus, the more λ is large, the more α_i^* (solution of eq.2) will be sparse.

Usually in SC framework, if we take two neighbor patches $\mathbf{x}_i$ and $\mathbf{x}_j$ (with a strong correlation between them), their respective sparse codes, α_i and α_j, can lose this strong correlation, especially indexes of non-zero inputs can completely mismatch. It means they are involving different atoms for their patches' reconstructions. An atom is an element of the vector patch. There exist some SC variations which have been introduced to tackle this behaviour. Principles of this improvement can be divided into two categories: one plays on adding of proximity constraint into the loss directly while the second adds some extra terms into the regularization term. To illustrate the first category, we can cite two approaches: Local Constrained linear Coding (LCC) [24] and the Local Sparse Coding (LSC) [20]. In the second category, we can mention GSC [25].

We will define the set of pre-computed sparse codes of $\mathbf{X}^{train} \triangleq \{\mathbf{x}_1^{train}, \ldots, \mathbf{x}_{N^{train}}^{train}\}$ by $\mathbf{A}^{train} \triangleq \{\alpha_1^{train}, \ldots, \alpha_{N^{train}}^{train}\}$ where N^{train} designates the number of local features sampled from the training set. Indeed, this adds a spatial constraint in the regularization term. Its equation is:

$$\mathcal{L}_{GSC}(\alpha_i|\mathbf{x}_i, \mathbf{A}^{train}; \mathbf{D}, \lambda, \beta) = \min_{\alpha_i \in \mathbb{R}^K} \|\mathbf{x}_i - \mathbf{D}\alpha_i\|_2^2 + \lambda\|\alpha_i\|_1 + \beta L_{ii}\alpha_i^T\alpha_i + 2\beta\alpha_i^T\mathbf{h}_i, \quad (3)$$

where $\mathbf{h}_i = \sum_{j \neq i}^{N^{train}} L_{ij}\alpha_j^{train}$, $\mathbf{L} = \{L_{ij}\}_{i,j=1,\ldots,N^{train}}$ is a Laplacian matrix and β a regularization parameter. The matrix $\mathbf{L}$ is defined by $\mathbf{L} = \mathbf{S} - \mathbf{W}$, where $\mathbf{W}$ is a weight matrix with and $W_{i,j} = \exp\{-\frac{\|\mathbf{x}_i - \mathbf{x}_j^{train}\|_2^2}{\sigma^2}\}$ if $\mathbf{x}_j^{train} \in V(\mathbf{x}_i)$ (where $V(\mathbf{x}_i)$ is the set of neighborhood of $\mathbf{x}_i$ excluding $\mathbf{x}_i$ itself), $W_{i,j} = 0$ else. The matrix $\mathbf{S}$ is diagonal and $S_{i,i} = \sum_{j=1}^{N^{train}} W_{i,j}$. We propose to improve SC by simultaneously encoding all the test local patches (for example associated with a test image). This new modeling will be inspired from the GSC.

3 Joint Sparse Coding - JSC

JSC principle is to jointly encode **all** local features $\mathbf{X}^{test} = \{\mathbf{x}_1^{test}, \ldots, \mathbf{x}_{N^{test}}^{test}\}$ **simultaneously** to overcome the decorrelation problem. We also enforce $\alpha_i^k \geq 0$ in the previous optimization problem. This additional constraint improves pooling performances, thus avoiding to pool simultaneously on positive and negative sparse code values and decreasing as a consequence the final size vector by a factor by two. The equation of our modeling is very similar to GSC:

$$\mathcal{L}_{JSC}(\alpha_i|\mathbf{x}_i, \mathbf{A}^{test}; \mathbf{D}, \lambda) = \min_{\alpha_i \in \mathbb{R}^K} \|\mathbf{x}_i - \mathbf{D}\alpha_i\|_2^2 + \lambda\|\alpha_i\|_1 + \beta L_{ii}\alpha_i^T\alpha_i + 2\beta\alpha_i^T\mathbf{h}_i, \ s.t.\ \alpha_i^k \geq 0, \quad (4)$$

where $\mathbf{h}_i = \sum_{j \neq i}^{N^{test}} L_{ij}\alpha_j^{test}$, $\mathbf{L} = \{L_{ij}\}_{i,j=1,\ldots,N^{test}}$ is a Laplacian matrix, β a regularization parameter. Here, $\mathbf{L} = \mathbf{S} - \mathbf{W}$, where $W_{i,j} = \exp\{-\frac{\|\mathbf{x}_i - \mathbf{x}_j^{test}\|_2^2}{\sigma^2}\}$ if $\mathbf{x}_j^{test} \in V(\mathbf{x}_i)$, $W_{i,j} = 0$ else and $S_{i,i} = \sum_{j=1}^{N^{test}} W_{i,j}$. Here, $\mathbf{A}^{test} \triangleq \{\alpha_1^{test}, \ldots, \alpha_{N^{test}}^{test}\}$ are computed and stacked initially. In practice $N^{test} << N^{train}$, so we need to store only a sparse $K \times N^{test}$ matrix.

Our Laplacian matrix ($N^{test} \times N^{test}$) is very sparse. If we don't need to compute the full matrix, one way is to only calculate the non-zero elements ($(v+1) \times N^{test}$) with the previous formulation. Each column of this $((v+1) \times N^{test})$ matrix is denoted by $\mathbf{L}_i$. To realize this, we use a fast NN-search technical (FLANN) [14] which speeds up the computation considerably. Thus, the solution of eq.4 is given by a modified Feature Sign Search (FSS) algorithm [10] by adding a) a positivity constraint on sparse codes and b) integrating the two right terms (in β) of eq.4 in the gradient formulation used during the FSS algorithm. JSC is given by the algorithm 1. To illustrate the correlation problem, viewed with SC, we compare the normalized correlation computed between two inputs vectors with the normalized correlation computed with their respective output vectors. In this example, 300 different pairs, extracted from UIUCsports local features, are chosen to realize this. The normalized correlation formulation between $\mathbf{x}$ and $\mathbf{y}$ is given by $\rho(\mathbf{x},\mathbf{y}) = \frac{\mathbf{x}^T\mathbf{y}}{\|\mathbf{x}\|_2\|\mathbf{y}\|_2} \in [0,1]$. We also introduce the scalar value $\overline{\nabla\rho}^2 = \frac{2}{300\times299}\sum_{i=1}^{300}\sum_{j=1}^{j<i}[\rho(\mathbf{x}_i,\mathbf{x}_j)-\rho(\alpha_i,\alpha_j)]^2$ which measures the average quadratic difference between normalized correlation of the input space and the output space. The lower $\overline{\nabla\rho}^2$ is the better. Table 1 summarizes our results including the sparsity percentage. The last line presents $\rho(\alpha_i,\alpha_j)$ correlation associated to output space, for a strong correlation $\rho(\mathbf{x}_i,\mathbf{x}_j) = 90\%$ in input space. We note that the correlation gain is accom-

Algorithm 1 Joint Sparse Coding

```
Inputs: $\mathbf{D}$, $\lambda$, $\beta$, $\mathbf{X}^{test}$, $\sigma$ and $v$
for $i = 1 : N^{test}$ do
    [$\mathbf{V}_i$, $\mathbf{dist}_i$] = $v$-nn search of $\mathbf{x}_i^{test}$ into $\mathbf{X}^{test}$
    $\mathbf{V}_i$ are indexes of $\mathbf{x}_i$ neighbors in $\mathbf{X}^{test}$
    Compute $\mathbf{L}_i$ from $\mathbf{dist}_i$ and $\sigma$
end for
$\mathbf{A}^{test} = \mathrm{lasso}(\mathbf{X}^{test};\mathbf{D},\lambda)$
for $i = 1 : N^{test}$ do
    $\alpha_i = \mathrm{JSC}(\mathbf{x}_i^{test},\mathbf{A}^{test},\mathbf{D},\mathbf{L}_i,\mathbf{V}_i,\lambda,\beta)$
end for
Output: $\mathbf{A}^{test}$
```

Method	SC (0.2)	GSC (0.4,0.2)	JSC (0.4,0.2)	GSC (0.2,0.2)	JSC (0.2,0.2)
Level Sparsity	**5.82%**	9.36%	15.05%	17.66%	22.75%
$\overline{\nabla\rho}^2$	126.75	116.59	81.83	108.77	**73.35**
$\rho = 90\%$	31%	75%	63%	**79%**	70%

Table 1. $\overline{\nabla\rho}^2$ and correlation $\rho = 90\%$, as an example of strong correlation, for SC, GSC and JSC for two couples (λ,β), on testing patches. The lower $\overline{\nabla\rho}^2$ is obtained for JSC $(0.2,0.2)$and the best result for correlation parameter ρ is for GSC $(0.2,0.2)$, however, the low sparsity level is obtained for SC.

panied by a sparsity level drop. Thus, λ is increasing sparsity while β is working in the opposite direction.

4 Dictionary Learning

The analytical solution to update a dictionary $\mathbf{D} \triangleq [\mathbf{d}_1,\dots,\mathbf{d}_K]$ off-line exists and it is formulated as $\mathbf{D} = (\mathbf{X}\mathbf{A}^T)(\mathbf{A}\mathbf{A}^T)^{-1}$, where $\mathbf{A} \triangleq \{\alpha_i\}, i = 1,\dots,N$ and $\mathbf{A} \in \mathbb{R}^{K\times N}$. The problems comes from the computation of $(\mathbf{A}\mathbf{A}^T)^{-1}$. It is a matrix of size $(K \times K)$ and the computational complexity of this matrix inversion is in $O(K^3)$. Moreover, we have to store the matrix $\mathbf{A}$ in central memory. Thus, we want efficient methods (in term of complexity and memory occupation) to train such dictionaries under basis constraints. One would minimize the regularized empirical risk $\mathcal{R}_n$:

$$\mathcal{R}_N(\mathbf{A},\mathbf{D}) \triangleq \frac{1}{N}\sum_{i=1}^{N} l(\mathbf{x}_i; f(\alpha_i,\mathbf{D})) + \Gamma(\mathbf{A}), \tag{5}$$

where $f(\alpha_i,\mathbf{D}) = \mathbf{D}\alpha_i$, $l(.)$ is typically a quadratic loss function and $\Gamma(.)$ represents the regularization term (for example SC and GSC regularization terms). Eq. 5 would be optimized iteratively by a (stochastic) gradient descent. Unfortunately, the problem is not jointly convex but only conditionally convex. Alternatively, we can minimize:

$$\mathcal{R}_N(\mathbf{A}|\hat{\mathbf{D}}) \triangleq \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}\|\mathbf{x}_i - \hat{\mathbf{D}}\alpha_i\|_2^2 + \Gamma(\alpha_i),\ \ s.t.\ \ \alpha_i^k \geq 1 \tag{6}$$

and

$$\mathcal{R}_N(\mathbf{D}|\hat{\mathbf{A}}) \triangleq \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}\|\mathbf{x}_i - \mathbf{D}\hat{\alpha}_i\|_2^2\ \ s.t. \|\mathbf{d}_k^T\mathbf{d}_k\|_2^2 = 1 \text{ and } d_k^j \geq 0. \tag{7}$$

In order to obtain a suboptimal solution of eq. 5., eq. 6 can be solved efficiently in parallel *via* SC/GSC procedures while eq. 7 can be solved by a constrained linear system [13].

5 Experiments

5.1 Metrics

In this section we present some results obtained with SC and GSC dictionaries when we use SC and JSC for the encoding part. We fix the dictionary size to $K = 1024$ and a positivity constraint on dictionary columns and sparse codes are applied. The regularization parameters are $\lambda = 0.2$ for SC, ($\lambda = 0.4$; $\beta = 0.2$) and ($\lambda = 0.2$; $\beta = 0.2$) for GSC and JSC for encoding part. Only the GSC ($\lambda = 0.2$, $\beta = 0.2$) dictionary will be used. We measure a classification rate given by a 1-vs-all approach thanks to a linear Support Vector Machine (SVM). Its regularization parameter is fixed to $C = 0.07$. This classification is made by an Average Overall Accuracy (AOA):

$$AOA = \frac{1}{M}\sum_{m=1}^{N}\left\{\frac{1}{N}\sum_{i=1}^{N}\delta(\hat{y}_{i,m} - y_{i,m})\right\}, \tag{8}$$

where N represents the number of available data, δ the loss function chosen (mean square error), M, the number of cross validation and $\hat{y}_{i,m}$ and $y_{i,m}$, the true and predicted label. We realize our experiments on UIUCsportss database [11] and scenes15 database [8]. UIUCsportss database contains 1579 images from 8 different classes. The number of images in each class varies from 137 to 250. We randomly select 70 images from each class for training and 60 for testing. scenes15 database contains 4485 images belonging to 15 different categories and the number of images per class varies between 200 to 400. 100 images are selected for training part and the others for testing part. In our experiments, $M = 10$, $N_{UIUCsports} = 60 \times 8 = 480$ and $N_{scenes15} = 4485 - 15 \times 100 = 2985$. We extract densely SIFT patches (24×24) [12] with a grey level and on one scale. The grid size is 80×80 for UIUCsportss database and 30×30 for scenes15 database. We apply a Spatial Pyramid Matching (SPM) [8] which is defined on L levels. For UIUCsportss, $L = 2$, thus pooling is performed on the entire image ((1×1) - first layer) and the second layer on (2×2) grid with stride of 25%. For scenes15, $L = 3$, thus we use (1×1), (2×2) and (4×4) sub-regions for SPM. We apply μ-pooling ($\mu = 2.5$) for the pooling step [1].

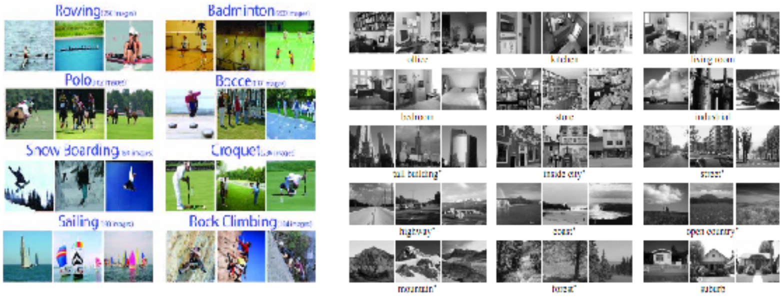


Fig. 1. UIUCsports dataset (left) - scenes15 (right)

5.2 Results on UIUCsports

Table 2 summarizes obtained results. We observe different behaviours. If we focus on encoding part variations (horizontal reading), we see that for all dictionaries choices, SC encoding is the best. Any gain is viewed for the others and a similar behavior is obtained if we read the table vertically. To go further more, in order to evaluate if SC and JSC models are complementary, we measure the accuracy of the arithmetic and geometric means of their estimates (AOA arithmetic and AOA geometric). AOA arithmetic is defined as the sum of probabilities of two selected models and AOA geometric as the

[1] As remind, μ-pooling is written as $f(\mathbf{v};\mathbf{w},\mu) = \sum_{m=1}^{c} w_m v_m^{\mu} = \mathbf{w}^T \mathbf{v}^{\mu} \;\; s.t. \|\mathbf{w}\|_2^2 = 1$ and $\mu \neq 0$, where $\mathbf{v}^{\mu} = \left\{\alpha_m^{\mu}\right\}, m = 1, \ldots, c$ and w_m encodes the contribution of the m-image location for specific visual words [7]

Encoding \ Dictionary	SC (0.2)	GSC (0.4, 0.2)	JSC (0.4, 0.2)	GSC (0.2, 0.2)	JSC (0.2, 0.2)
SC (0.2)	**87.27 ± 1.33**	80.75 ± 1.69	83.6 ± 1.66	80 ± 2.01	84.17 ± 1.57
GSC (0.2, 0.2)	**84.81 ± 1.87**	80.71 ± 2.05	81.6 ± 1.77	80.92 ± 2.15	84.17 ± 1.02

Table 2. Evolution of the Average Overall Accuracy for UIUCsports database. The best result is obtained with the couple SC dictionary and SC encoding

square root of the product of two selected models. Tables 3 and 4, associated to figures 2 and 3 respectively (only the arithmetic fusion is showed here, because geometric fusion is lower than the first), summarize results obtained with initial models and their associated fusion. Table 3 corresponds to a horizontal reading (encoding fusion) and table 4 to a vertical reading (dictionary fusion) for UIUCsports. We notice an important relative gain (until +8 points) with SC dictionary. This is less significant with GSC (0.2,0.2) dictionary where few relative gains are observed. For dictionary fusion, strong relative

Dictionary \ Encoding fusion		SC + GSC (0.4,0.2)	SC + JSC (0.4,0.2)	SC + GSC (0.2,0.2)	SC + JSC (0.2,0.2)
SC	AOA arithmetic	94.31 ± 1.28	94.77 ± 1.31	94.23 ± 1.3	**94.94 ± 1.05**
	AOA geometric	93.33 ± 1.23	93.94 ± 1.19	93.37 ± 1.22	94.19 ± 1.2
GSC (0.2,0.2)	AOA arithmetic	84.42 ± 1.5	85.08 ± 1.67	84.37 ± 1.51	85.12 ± 1.62
	AOA geometric	84.48 ± 1.52	84.9 ± 1.62	84.5 ± 1.65	84.98 ± 1.61

Table 3. Evolution of the arithmetic and geometric Accuracy for UIUCsportss database (encoding fusion). The best result is obtained with the couple SC dictionary associated with SC and JSC (0.2,0.2) encodings. An illustration is given in figure 2.

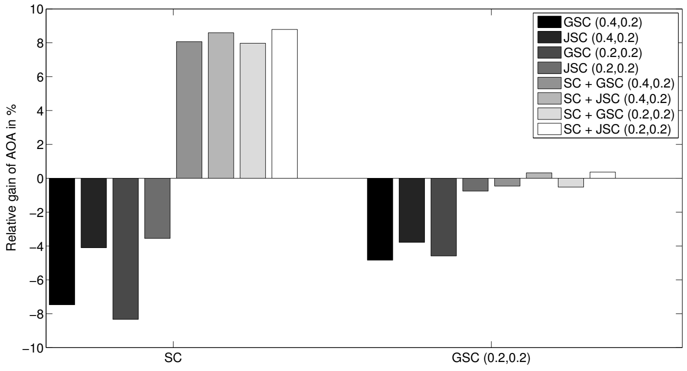


Fig. 2. Benefits and deficits obtained with GSC, JSC and arithmetic fusions encodings compared to SC encoding for the three different dictionaries for UIUCsports database.

Encoding / Dictionary fusion		SC	GSC (0.4,0.2)	JSC (0.4,0.2)	GSC (0.2,0.2)	JSC (0.2,0.2)
SC+GSC(0.2,0.2)	AOA arithmetic	**95.37 ± 1.29**	92.56 ± 1.11	83.33 ± 1.36	92.46 ± 1.15	84.25 ± 1.22
	AOA Geometric	94.62 ± 1.15	92.31 ± 1.42	83.89 ± 1.29	91.21 ± 1.58	84.31 ± 1.57

Table 4. Evolution of the arithmetic and geometric Accuracy for UIUCsportss database (dictionary fusion). The best result is obtained with the couple SC and GSC (0.2,0.2) dictionaries associated with SC encoding.An illustration is given in figure 3

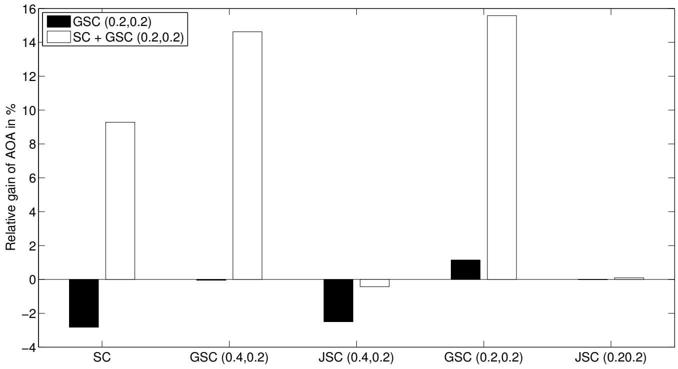


Fig. 3. Beneficits and deficits obtained with GSC and arithmetic fusions dictionaries compared to SC dictionary with five different encoding method choices for UIUCsports database.

gains are viewed for SC and the two GSC encoding models. There is no gain for the two JSC encoding models. The best result is for SC dictionary and encoding with SC and GSC (0.2,0.2) dictionary with SC encoding.

5.3 Results on scenes15

The table 5 summarizes our results: No gain is observed for this dataset. The best results are for SC dictionary and encoding. Fusion results which follow, are summarized in tables 6 and 7 which present fusion results obtained. Figures 4 and 5 illustrate the previous tables respectively. We notice that the behaviour is inverted for the two fu-

Dictionary / Encoding	SC (0.2)	GSC (0.4,0.2)	JSC (0.4,0.2)	GSC (0.2,0.2)	JSC (0.2,0.2)
SC (0.2)	**84.69 ± 0.6**	80.31 ± 0.6	80.82 ± 0.63	80.59 ± 0.64	81.47 ± 0.47
GSC (0.2,0.2)	**83.35 ± 0.59**	78.79 ± 0.66	78.4 ± 0.79	79.06 ± 0.62	80.81 ± 0.66

Table 5. Evolution of the Average Overall Accuracy for scenes15 database. The best result is obtained with the couple SC dictionary and SC encoding

Dictionary \ Encoding fusion		SC + GSC (0.4,0.2)	SC + JSC (0.4,0.2)	SC + GSC (0.2,0.2)	SC + JSC (0.2,0.2)
SC	AOA arithmetic	82.97 ± 0.69	83.46 ± 0.51	83.09 ± 0.62	83.60 ± 0.43
	AOA geometric	83.04 ± 0.69	83.48 ± 0.46	83.59 ± 0.59	**83.67 ± 0.42**
GSC (0.2,0.2)	AOA arithmetic	82.66 ± 0.66	82.33 ± 0.76	82.67 ± 0.52	82.79 ± 0.73
	AOA geometric	82.62 ± 0.71	82.44 ± 0.74	82.85 ± 0.62	82.84 ± 0.72

Table 6. Evolution of the arithmetic and geometric Accuracy for scenes15 database (encoding fusion). No results improve tha of SC. An illustration is given in figure 4.

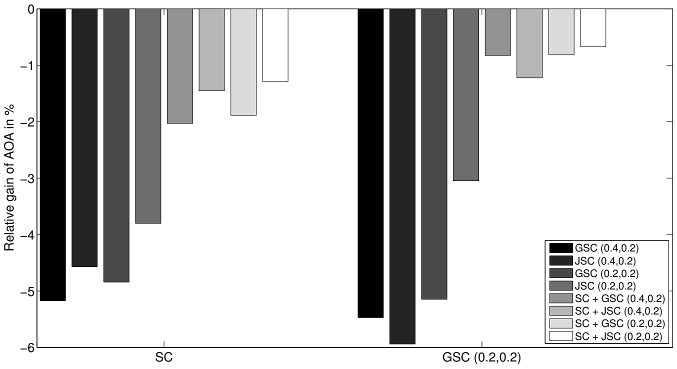


Fig. 4. Benefits and deficits obtained with GSC, JSC and arithmetic fusions encodings compared to SC encoding for the three different dictionaries for scenes15 database.

Dictionary \ Encoding fusion		SC	GSC (0.4,0.2)	JSC (0.4,0.2)	GSC (0.2,0.2)	JSC (0.2,0.2)
SC + GSC (0.2,0.2)	AOA arithmetic	**84.66 ± 0.64**	79.76 ± 0.62	81.4 ± 0.71	80.41 ± 0.67	82.35 ± 0.75
	AOA Geometric	84.62 ± 0.71	79.76 ± 0.63	81.38 ± 0.69	80.47 ± 0.57	82.2 ± 0.77

Table 7. Evolution of the arithmetic and geometric Accuracy for scenes15 database (dictionary fusion). The best result is obtained with the couple SC and GSC (0.2,0.2) dictionaries associated with SC encoding.An illustration is given in figure 5

sion cases. However, the deficits decrease with fusion and more specifically for GSC (0.2,0.2) dictionary. For the dictionaries fusion, it is between the two models that we obtain the most significant gain. The best result is for the couple (SC + GSC) dictionary associated with SC encoding.

5.4 Weighted fusion

To go further more, we plot the accuracy for a weighted arithmetic fusion. In a first time, the weights are the same for each classes and curves of figure 6 illustrate the weighted arithmetic fusion ($AOA_{arith} =_{SC} +(1-\mu)AOA_{GSC}$). We notice for UIUCsports, when we use adapted coefficients with fusion, no improvement is observed and the accuracy

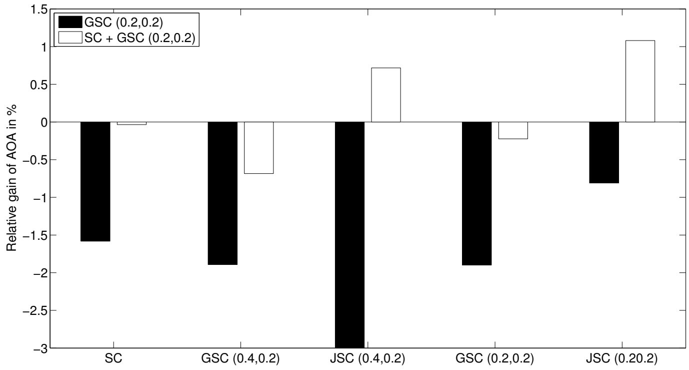


Fig. 5. Beneficits and deficits obtained with GSC and arithmetic fusions dictionaries compared to SC dictionary with five different encoding method choices for scenes15 database.

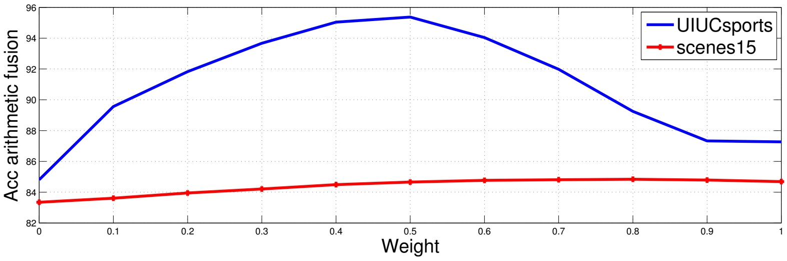


Fig. 6. Evolution of the accuracy with different coefficient. The first point corresponds to the chosen model for fusion and the last point is the SC model. Notice the best result for UIUCsports is obtained with a coefficient of 0.5, and for scenes15, it is 0.8 for SC and 0.2 for GSC (0.2,0.2) dictionary associated with SC. For these two examples, the fusion is between SC (dictionary and encoding) and GSC (0.2,0.2) dictionary with SC encoding.

decreases considerably for other couples. For scenes15, a very small improvement is seen but it does not allow us to conclude to the real benefit of the method. Another alternative would be to calculate others means as harmonic or energy means for examples. Also, the considerable gain obtained with UIUCsports database can be explained by putting forward two assumptions: the heterogeneity between images of training and testing sets and the correlation conservation between the input and output space. The study conducted so far shows that the second assumption is the one that goes in the right direction.

6 Conclusion

Although the results obtained with GSC and JSC alone are not living up to our expectations, we highlight the relevance of our proposal, thanks to the fusion procedure which

	Initial accuracy	Blinded fusion	Weighted fusion	State-of-the-Art
UIUCsports	87.27% ± 1.33	**95.37% ± 1.29**	**95.37% ± 1.29**	88.47 ± 2.32 [23]
scenes15	**84.69% ± 0.6**	**84.66% ± 0.64**	**84.88% ± 0.55**	81.04% ± 0.5 [8]

Table 8. Summarize of fusion results - details in Tables 2, 3, 4, 5, 6, 7.

greatly improves the State-of-the-Art for UIUCsports (88.47 ± 2.32) of [23] (our modeling: 95.37 ± 1.29). A complete study must be realized with different couples (λ, β) for dictionary and encoding parts to find the right setting for UIUCsports and scenes15 databases. Also, the nature of the images is to be considerate and a study of the heterogeneity level of images could be achieved [22] through the Shannon entropy measure. However, we think that our modeling can be improved by three ways. The first will be to get even better stabilized JSC results by adding an outer loop in the JSC algorithm. After multiple stages, we can expect some improvements. The second is a direct extension of the JSC by integrating some Laplacian regularization computed from a training set of local features. Here, sparse codes will be reconstructed by simultaneously minimize the deviation from both this training set and the image local features. The fusion could be improved by weighted average fusion using statistic from code image. Finally, it had been shown that adding some orthogonal constraints during the dictionary learning process can improves results [5, 17]. Here, too, a full study should be conducted with the two methods of sparse codes encoding.

Acknowledgement. We thank Direction Générale de l'Armement (DGA) for a financial support to this research. We thank Lucian ALECU for his comments.

References

1. C. Bauge, M. Lagrange, J. Andén, and S. Mallat. Representing environmental sounds using the separable scattering transform. In *ICASSP*, pages 8667–8671, 2013.
2. Y. Bengio. Learning deep architectures for ai. *Found. Trends Mach. Learn.*, 2(1):1–127, Jan. 2009.
3. K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *CoRR*, abs/1405.3531, 2014.
4. S. S. Chen, D. L. Donoho, Michael, and A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20:33–61, 1998.
5. A. Cherian. Nearest neighbors using compact sparse codes. In T. Jebara and E. P. Xing, editors, *Proceedings of the 31st International Conference on Machine Learning (ICML - 14)*, pages 1053–1061. JMLR Worshop and Conference Proceedings, 2014.
6. J. Deng, K. Li, M. Do, H. Su, and L. Fei-Fei. Construction and Analysis of a Large Scale Image Ontology. Vision Sciences Society, 2009.
7. J. Feng, B. Ni, Q. Tian, and S. Yan. Geometric ℓ_p-norm feature pooling for image classification. In *CVPR*, pages 2697–2704, 2011.
8. S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, CVPR '06, pages 2169–2178, Washington, DC, USA, 2006. IEEE Computer Society.

9. Y. LeCun, K. Kavukcuoglu, and C. Farabet. Convolutional networks and applications in vision. In *ISCAS*, pages 253–256. IEEE, 2010.
10. H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *In NIPS*, pages 801–808. NIPS, 2007.
11. L.-J. Li. What, where and who? classifying event by scene and object recognition. In *In IEEE International Conference on Computer Vision*, 2007.
12. D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ICCV '99, pages 1150–, Washington, DC, USA, 1999. IEEE Computer Society.
13. J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 689–696, New York, NY, USA, 2009. ACM.
14. M. Muja and D. G. Lowe. Scalable nearest neighbor algorithms for high dimensional data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36, 2014.
15. G. V. Pendse. A tutorial on the lasso and the "shooting algorithm". Technical report, P.A.I.N Group, Imaging and Analysis Group - McLean Hospital, Harvard Medical School, 8 February 2011.
16. F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, pages 143–156, Berlin, Heidelberg, 2010. Springer-Verlag.
17. I. Ramirez, F. Lecumberry, and G. Sapiro. Universal priors for sparse modeling. In *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2009 3rd IEEE International Workshop on*, pages 197–200, Dec 2009.
18. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge, 2014.
19. J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 1470–1477, Oct. 2003.
20. J. J. Thiagarajan, K. N. Ramamurthy, and A. Spanias. Local Sparse Coding for Image Classification and Retrieval. Technical report, 2012.
21. R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.
22. S. Tollari and H. Glotin. Lda versus mmd approximation on mislabeled images for keyword dependant selection of visual features and their heterogeneity. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume II, pages 413–416, may 2006.
23. X. Wang, B. Wang, X. Bai, W. Liu, and Z. Tu. Max-margin multiple-instance dictionary learning. In S. Dasgupta and D. Mcallester, editors, *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, volume 28, pages 846–854. JMLR Workshop and Conference Proceedings, May 2013.
24. B. Xie, M. Song, and D. Tao. Large-scale dictionary learning for local coordinate coding. In *Proceedings of the British Machine Vision Conference*, pages 36.1–36.9. BMVA Press, 2010. doi:10.5244/C.24.36.
25. M. Zheng, J. Bu, C. Chen, C. Wang, L. Zhang, G. Qiu, and D. Cai. Graph regularized sparse coding for image representation. *IEEE Transaction on Image Processing*, 20(5):1327–1336, 2011.

Anomaly and Event Detection for Unsupervised Athlete Performance Data

Jim O' Donoghue[1 2], Mark Roantree[2], Bryan Cullen[3], Niall Moyna[2], Conor O Sullivan[1], and Andrew McCarren[1]

[1] School of Computing
[2] Insight: Centre for Data Analytics
[3] School of Health and Human Performance,
Dublin City University, Glasnevin, Dublin 9, Ireland.

Abstract. There are many projects today where data is collected automatically to provide input for various data mining algorithms. A problem with freshly generated datasets is their unsupervised nature, leading to difficulty in fitting predictive algorithms without substantial manual effort. One of the first steps in dataset preparation and mining is anomaly detection, where clear anomalies and outliers as well as events or changes in the pattern of the data are identified as a precursor to subsequent steps in data mining. In the research presented here, we provide a multi-step anomaly detection process which utilises different combinations of algorithms for the most accurate identification of outliers and events.

1 Introduction

Anomaly detection is an important component in data science. In many situations, researchers are confronted with datasets which possibly contain a large number of features and more often than not, incorporates outliers and missing data. Implementing dimensionality reduction and incorporating cluster analysis techniques such as K-means are commonly used in performing unsupervised learning tasks in such data. In fact, principal components are the continuous solutions to the discrete cluster membership indicators for K-means clustering [4].

Anomalies are generally defined as unusual events which occur within a dataset, where a subset of these events are *outliers*. Outliers are occurrences that make either no physical sense, or appear so extreme they are considered probabilistically infeasible. The identification of outliers and anomalies is critical in avoiding poorly-fitting models for many machine learning algorithms [17].

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

1.1 Problem Background

The physiological demands of any sport are determined largely by the activity patterns of the game. Similar to other team sports, Gaelic football [1] involves repeated, short duration, high intensity bouts of anaerobic exercise interspersed with sustained light to moderate aerobic activity. The duration of these intervals of high intensity are largely unpredictable, due to the fact that they are imposed by the pattern of play, and can vary greatly from player to player and from one game to another. On average, senior players perform 96 bursts of high intensity activity lasting 6 seconds followed by an average recovery of 37 seconds. These players typically work at 80% of their maximum heart rate (HRmax) and cover an average distance of 8.5 km during competitive games. Superimposed on the physiological demands of match play are key technical activities such as winning possession of the ball, evading opponents and breaking tackles which involve high running velocities, agility, strength and power.

As part of the process for collecting data on each athlete, global positioning software (GPS) has become increasingly popular among sport scientists as a method of tracking movement patterns in many field based sports [13]. Modern GPS devices are portable, robust and lightweight making them particularly suited to field based sports. From a sports science perspective, the initial aim is to evaluate the characteristics and fitness levels of Gaelic football players and compare the physical and fitness characteristics relative to each playing position. The subsequent goal is to predict when these players are approaching or have reached optimal performance level. This requires the generation of a sufficiently rich dataset to build an initial model before it can be used in a real time environment. At each of 17 competitive games, 10 out of 15 players in the team were fitted with appropriate sensor devices to record heart rate, speed, distance covered, GPS location and accelerometer values, recording at multiple times per second. The resulting dataset contained in excess of 200 million values. Simple detection methods [14] can be useful for more obvious outliers, but encounter limitations in discerning more subtle anomalies. Due to the nature of contact sport, the devices incur a number of blows during each game, introducing many potential anomalies. The work presented here focuses on *anomaly detection* in unsupervised data.

Contribution. If one uses unsupervised clustering techniques such as K-means to determine an anomaly, then K (proposed number of clusters) can be calculated as part of the X-means cluster estimation technique [12]. However, such algorithms rely on the choice of good initial starting points [5] to find workable solutions. Our contribution is the development of an unsupervised outlier detection algorithm for time series data which employs both univariate and multivariate approaches for a more accurate detection rate and further our previously developed learning framework [11] to incorporate anomaly detection as well as classification. The univariate method is based on the approach taken in [15] but extends this work to manage time series data, while the multivariate approach builds upon the work of [16] and introduces a secondary decision statistic which detects when variables are unusually static. In dynamic environments such as GAA matches extremely static measurements are equally as anomalous as those which are extremely varying.

Paper Structure. The paper is structured as follows: in Section 2, we discuss related work in the area; in Section 3 we provide a description of our approach and detail how we identify and classify anomalies; we describe our experimental setup together with an evaluation of the results in Section 4; and finally, we present our conclusions in Section 5.

2 Related Research

In [8], the authors present a novel approach for anomaly detection incorporating both density and grid-based clustering algorithms. Their primary focus is high dimensional data and they test their algorithm on the KDD Cup 1999 network dataset [9]. The approach taken was to optimise the pMafia algorithm, using a Frequency-Pattern tree in an intermediate step in order to improve the detection rate. Similar to our approach, they provide an unsupervised anomaly detection algorithm. However, in their evaluation it was shown that the improvement in detection rate had a negative side effect in generating a higher number of false positives. By their own admission, the algorithm works best for datasets with certain characteristics i.e. data points sought will be statistically different from normal data. This means that if there is an entire window of anomalous data, this may affect the performance of the detection method.

In [3], the authors present an algorithm for anomaly detection in multivariate time series data. Their goal is to capture relationships across variables and by doing so, identify different types of anomalies that occur in the time series dataset. As we use a real-world dataset, our comments concern their evaluation with the several time series datasets from [2] and not the experiments with synthetic data. The evaluation used a sliding window of length 6 and clearly demonstrates that for time series, a subsequence of data points outperforms the basic data point approach, which is similar to our findings. Apart from the fact that our research is based fully on a real-world dataset using unsupervised learning, we also employ both univariate and multivariate algorithms to deliver a higher performance in anomaly and outlier detection.

The authors of [17] developed the Robust Support Vector Machine that demonstrates its ability to identify images (bullet holes) when outliers exist. This algorithm is an improvement on the standard support vector machine (SVM) algorithm as the incorporation of the averaging technique to an SVM makes the decision function less susceptible to outliers and thus, avoids overfitting. The process could be used to identify outliers however it requires a supervised training dataset which, as with our work, is not always available.

In [12], the authors propose an extension to K-means algorithm called X-means to identify outliers in Gaussian datasets without specifying the initial number of suspected clusters. The algorithm performed exceptionally well with regard to identifying the exact number of clusters and functioned commensurately against the K-means algorithm. However, the X-means algorithm is vulnerable to initial estimates and may attain a sub-optimal minima.

In [4], the authors demonstrate mathematically that principal components are the continuous solutions to the discrete cluster membership indicators for K-means clustering. This idea is extended in our work by using principal components as the basis for a decision based system to detect outliers in truly unsupervised data. In [16], the authors propose a novel Principal Components classifier in order to detect anomalies in the case of network intrusion identification on the KDD Cup 1999 network dataset, whose aim was to detect attacks on network access data. While they produced a false hit rate of only 1% and their PCC remained robust to false positives, all the other metrics degraded significantly in terms of quality. Our approach uses the PCC but extends it to detect highly static variables with a secondary chi-squared decision statistic. Furthermore, our training data is real-world, containing anomalous examples, whereas in [16] their classifier was trained on completely clean, non-anomalous data.

3 Outline Approach

Our anomaly detection algorithm has 4 primary components: Boundary Detection, Univariate Outlier Detection, Principal Component Transformation and Principal Component Classification. The role of the algorithm is to detect anomalies and classify these as outliers (data points which are far outside the expected norm) or events (samples which demonstrate a clear shift in the pattern of the data). Each of the algorithm's components detects anomalies within the dataset with the exception of Principal Component Transformation which transforms the data only. We now provide a brief overview of each component.

3.1 Boundary Detection

This represents a pre-processing stage where clearly erroneous data points are removed. This can only take place for those features where a domain expert has clearly specified boundaries, outside which data values make no sense. For example, if a player had a heart-rate below 40bpm or above 250bpm, the hardware has clearly malfunctioned. The process eliminates obvious errors so that later calculations are not affected.

3.2 Univariate Outlier Detection

This stage is based on Chauvnet's method, an approach for univariate outlier detection found in [15]. Euclidean distance [16] has shown to be of little value with this type of time series data as it detects far too many outliers and thus, excludes large amounts of data. Early experiments with Euclidean distance with sport scientists for manual evaluation confirmed this assumption. A first step marks a sample as an *anomaly_low* or *anomaly_high*, while a second classifies these anomalies as either: *outlier*, *event* or *untrue* (not an anomaly). Before detecting anomalous values, the algorithm first calculates summary statistics of mean, variance and standard deviation, denoted by $\bar{x}, \sigma^2$, and σ respectively, for each feature x_i in the dataset X. Unlike [15], where anomalies are detected with standard deviation alone, our univariate approach incorporates time differencing and compares the current time difference against *previous* time differences. If it is significantly different, we then compare with the *future* time difference to confirm the data point is an anomaly or outlier. This is achieved by iterating through each feature and examining every time point with a t-distribution coefficient for natural confidence intervals on the differenced data (which removes any non-stationary components of the data), a crucial factor for time series data as it is non-stationary.

The first steps of the algorithm presented get the dimensions of the dataset $|X| = (T \text{ x } n)$, where T is the number of time-points or samples and n is the number of features where $\forall x_{t,i} \in X$, $t \in (1, 2, \ldots, T-1, T)$ and $i \in (1, 2, \ldots, n-1, n)$. Subsequently a t-distribution coefficient β of 3 was chosen to give approximately a 99.9% confidence interval in order to detect anomalies with an α of 0.001 for each feature, where α is the probability of a false alarm. In concrete terms, this coefficient is multiplied by the standard deviation both positively and negatively ($\pm 3\sigma \mathit{\Delta} x_i$) for each feature in order to calculate upper and lower bounds for anomalies.

3.3 Principal Component Transformation

The aim of this step is to compute key characteristics and to transform the data for the final stage in the algorithm.

1. Recalculate the summary statistics from the previous stage. This is necessary due to removed outliers.
2. Impute the missing values.
3. Calculate the correlation matrix in order to provide input to the Principal Components Analysis.
4. Calculate eigenvectors and eigenvalues. The eigenvectors enable the creation of an orthogonal representation of the dataset, used to derive the principal components. Eigenvalues measure the energy contribution of each of the principal components, as well as providing input into the principal components classifier.
5. Standardise each feature to have unit variance. For each feature x, this involves subtracting the mean, $\bar{x}$ from x and dividing the result by the standard deviation, $\sigma^2 x$.
6. Compute the transformed dataset. Principal Component Analysis (PCA) provides an orthogonal representation of the data, describing it in terms of the axes of most variation for each component.

Before transforming the data into its principal components, it is differenced at a one second time lag and transformed into a 5 second moving average, centred on the value being transformed, essentially filtering noise from the data. Missing values are imputed with the R Amelia [6] package. Amelia is a multi-variate imputation mechanism which infers missing data in a single-cross section from times series and is the only R component in a Python application. The use of R was necessary as Amelia was not available in Python and was evaluated to best suit our imputation needs. Employing bootstrapping and Expectation Maximisation, it allows for imputation from the posterior distribution of the data.

After imputation, it is necessary to determine how much the features change together by calculating the correlation matrix. The dimensions of this matrix are (p x p) where p is the number of features in the dataset.

Algorithm 1 Data Transform

```
function DataTransform($X$)
    $X^{lagl} \leftarrow moving\_avrg(X)$    ▷ transform data into moving average at a lag of l
    $X^{imp} \leftarrow amelia(\Delta X^{lagl})$ ▷ impute missing data with Amelia on the differences
    $cov(X_{imp}, Y_{imp}) = \sum_{t=1}^{T} \frac{(x_{t,i}^{imp} - \bar{x_i}^{imp})(y_{t,i}^{imp} - \bar{y_i}^{imp})}{T-1}$    ▷ calculate covariance
    $corr(X^{imp}, Y^{imp}) = \frac{cov(X^{imp}, Y^{imp})}{\sigma X^{imp} Y^{imp}}$    ▷ calculate correlation
    $E = (e_1 \dots e_p)$ and $e_{vals} = \lambda_1 \dots \lambda_p$    ▷ calculate the eigenvalues and vectors
    for $x_{t,i}$ in $x_i$ where $t \in 0 \dots T$ do    ▷ standardise the data
        $Z_{t,i} = \frac{x_{i,t} - \bar{x}_i}{\sigma^2 x_i}$
    end for
    $P = (E^\top Z^\top)^\top$    ▷ calculate principal components
     return $X^{imp}, E, e_{vals},\ P$
end function
```

The eigenvectors E are then calculated on the non-standardised data using the correlation matrix (whose calculation effectively standardises the data) where each eigenvector $e_i \in e_1, e_2, \dots, e_p$. Eigenvalues $\lambda_1, \dots, \lambda_p$ are also calculated. Once the data is standardised as Z, the result is multiplied with the eigenvectors which gives the principal components of the original dimensions (T x n).

3.4 Principal Component Classification

The final stage has three main steps.

1. Calculate the number of major and minor components to use by calculating the cumulative percentage of the total eigenvalue energy for each.
2. Calculate the classification value or *test statistic* for each sample.
   (a) Sum the major components divided by their eigenvalues, giving you the chi-squared test-statistic.
   (b) Calculate the same value for minor components.
3. Classify anomalies using significance values calculated with the chi-squared distribution

(a) Use the test statistic generated from step 2, compute the decision statistics with the chi-squared cumulative distribution function and compare this to a chi-squared distribution with `num_components` degrees of freedom and all false alarm rates α providing the confidence interval. The chi-squared distribution was employed as we observed that the distribution of differenced, standardised variables at the univariate stage demonstrated a normal distribution.

(b) If confidence interval exceeds either of the chosen decision statistic thresholds (e.g. $> 95\%$ or $< 0.001\%$), for either the major or the minor components test statistics (who have the same false alarm α pairs), the data instance is classified as anomalous.

In algorithm 3, we first calculate the number of principal components involved and determine how much variation in both the major and minor components is to be included in the classifier. The percentage variation thresholds are set and subsequently the number of components to use are calculated with Algorithm 2 which takes the eigenvalues, type of components being summed (string of 'major' or 'minor') and the desired percentage eigenvalue energy (variation) as parameters. Eigenvalues are initially summed to calculate the total variance and then the parameter num_components and current_sum (running total) are initialised to zero before being calculated.

In lines 5 to 12 of algorithm 2, for each eigenvalue there is a check to see if the current percent variance sum is less than the desired variance. If this is the case, the number of components is incremented and added to variance sum is the current eigenvalue divided by the eigenvalue total sum, as this gives the percentage variance. It is worth noting that if it is the major components sum, we begin at $i = 1$ to start with the major components but with the minor components, we begin with the last eigenvalue $i = p - 1$ where p is the total number of eigenvalues.

Once the number of major components q and the number of minor components r are found, the chi-squared test statistics are then calculated for both component types by summing the value for the component p_i at time-point t divided by the appropriate eigenvalue λ_i up until the number of components is exhausted. In the case of the major components, the process begins at 1 and stops at component q whereas in the case of the minor components, it begins at the last component p and sum to $p - r$ as shown in lines 7 to 12.

In lines 13 and 14, the decision statistic is calculated by $1 -$ the chi-squared cumulative distribution function with q degrees of freedom for the major components and r for the minor components. Given that our data follows a multivariate normal distribution the overall false alarm rate is given by Equation 1.

$$\alpha_{total} = \alpha_{major} + \alpha_{minor} - \alpha_{major}\alpha_{minor} \tag{1}$$

We then chose the varying and static false alarm rates α_{large} and α_{static} (line 15). If a particular decision statistic is greater than $1 - \alpha_{static}$ or less than $1 - \alpha_{large}$ i.e. a certain significance threshold, the row is classified as anomalous

Algorithm 2 Calculate Number of Components

```
function GetNumComponents($e_{vals}$, $type$, $variance$)
    $\lambda_{total} = \lambda_1 + \lambda_2 + \ldots + \lambda_{p-1} + \lambda_p$
    $num\_components \leftarrow 0$
    $var\_sum \leftarrow 0$
    for all $\lambda_i$ where $i \in 1, 2, \ldots, p$ do
        if $type == 'major'$ && $var\_sum < variance$ then
            $num\_components \mathrel{+}= 1$
            $var\_sum \mathrel{+}= \frac{\lambda_i}{\lambda_{total}}$
        else if $type == 'minor'$ && $var\_sum < variance$ then
            $num\_components \mathrel{+}= 1$
            $var\_sum \mathrel{+}= \frac{\lambda_{p-i}}{\lambda_{total}}$
        end if
    end for
    return $num\_components$
end function
```

Algorithm 3 Principal Components Classifier

```
function PrincipalComponentsClassify (PCC)($P$, $E$, $e_{vals}$)
    $p \leftarrow |P|$
    $var_{maj} \leftarrow$ percentage variance for major classifier
    $var_{min} \leftarrow$ percentage variance for minor classifier
    $q =$GetNumComponents($e_{vals}, 'major', var_{maj}$)
    $r =$GetNumComponents($e_{vals}, 'minor', var_{min}$)
    for $p_i \in P$ where $i \in 1, 2, \ldots, s-1, s$ do                ▷ For each sample
        $test_t^{maj} \mathrel{+}= \frac{p_i}{\lambda_i}$
    end for
    for $p_j \in P$ where $j \in p, p-1, \ldots, p-d$ do
        $test_t^{min} \mathrel{+}= \frac{p_j}{\lambda_j}$
    end for
    $Decision_{maj} = 1 - P(\frac{q}{2}, \frac{test^{maj}}{2})$        ▷ 1 - chi-squared cumulative distribution
  function
    $Decision_{min} = 1 - P(\frac{r}{2}, \frac{test^{maj}}{2})$
    $c_{lrg} = 1 - \alpha_{large}$                    ▷ false alarm rates $\alpha_{large}^{maj} = \alpha_{large}^{min}$
    $c_{stat} = 1 - \alpha_{static}$                            ▷ $\alpha_{static}^{maj} = \alpha_{static}^{min}$
    for all $\delta_{major,t} \in Decision_{maj}$ and $\delta_{minor,t} \in Desicision_{min}$ do
        if $\delta_{major,t} < c_{large} \mid \delta_{minor,t} < c_{lrg} \mid \delta_{major,t} > c_{stat} \mid \delta_{minor,t} > c_{stat}$ then
            $X_t$ is anomalous
            $Anomalies_t \leftarrow True$
        end if
    end for
    return $Anomalies$
end function
```

as in line 18 of Algorithm 3. This implies a very large or very small degree of variation at this time-point. Our extension to the PCC captures where there is very little or no variation from sample to sample. Our features should be non-static and should be constantly changing, this minor variation parameter was a very important factor and was not incorporated by [16]. Finally, as our aim was to identify anomalous time-periods as opposed to particular time-points (as a time-point itself is not anomalous) and due to our rolling average transformation, we incorporated time-points within an 11 point centred window as anomalous.

4 Evaluation and Analysis

In this section, we briefly describe our dataset, approach to evaluation and provide a detailed analysis of experiments and results.

4.1 Experiment Setup

Experimental Set-up. Experiments were performed on a Dell Optiplex 790 running 64-bit Windows 7 Home Premium SP1 with an Intel Core i7-2600 quad-core 3.40 GHz CPU and 16.0GB of RAM. The code for the experiments was developed in Python using the Enthought Canopy (1.5.4.3105) distribution of 64-bit Python 2.7.6 and developed in PyCharm 4.5 IDE, making use of NumPy 1.8.1-1[18], Pandas 0.16.0[10] and SciPy 0.15.1[7] for mathematical and statistical operations and data manipulation. The imputations were performed with R and Amelia, package version 1.0 [6].

Dataset. For our evaluation, we used the results of one match with 81,165 instances in the dataset. The hardware devices generate at least 10 sets of measures per second, which were averaged giving one set of measures per second and providing just under 8,000 instances. As only heart rate and distance covered were used in this experiment and only 9 players generated data, each instance had 18 features per second, namely 9 sets of heart rates and distances. We selected the data beginning at the pre-match warm-up, until 4 minutes and 30 seconds after the end of the match which left a total 6,211 instances.

Evaluation Design. The design of the outlier detection algorithm allows for evaluation of different processes in combination. The univariate (P2) and multivariate steps (P3 & P4) were evaluated in isolation and with bounds detection (P1) added. We also evaluated after the univariate step (P1 & P2), without the univariate step (P1, P3, & P4) and with both (P1, P2, P3, & P4).

Before evaluating combinations of processes, we first tested various numbers of major components in isolation, at various false alarm rates α before performing the same evaluation for the minor components in the PCC. The secondary alpha measure added to P4 to detect unusually static time-points was kept at 0.01% for all experiments, keeping the measure sensitive. Window-size was also not varied and kept at 11, to account for the rolling average transformation.

Each anomaly detection algorithm was trained on the dataset in its entirety without partition. For testing, a random subset of the dataset was presented to the sports scientist involved in the original data collection for classification. He classified the subset as 69.89% anomalous with the remainder being non-anomalous. We then examined and cross-referenced the relevant subset of each result (from each configuration) and calculated an accuracy score for each. The evaluation metric use for all experiments was accuracy where $accuracy = \frac{TP+TN}{P+N}$.

4.2 Evaluation: Results and Analysis

Tuning the PCC (P3 & P4) Table 1 shows the results of our empirical search for the parameters to use in our PCC (P3 & P4). We first vary the percentage of total contribution to the classifier by the major components only ($r = 0$), in a range from 30% to 70% (columns 1 to 4) and then repeat the process with the minor components only ($q = 0$), in a range of 10% to 20%. We evaluate both at various false alarm rates α, in increments from 1% to 10%.

Table 1: Evaluation of Major only $r = 0$ and Minor only Components $q = 0$

α	MajC 30%	MajC 40%	MajC 50%	Maj 60%	MajC 70%	Min 10%	MinC 20%
1%	0.3656	0.3978	0.3979	0.3979	0.4409	0.4839	0.5807
2%	0.3979	0.3978	0.3871	0.4086	0.4409	0.5269	0.5807
4%	0.4409	0.4301	0.4301	0.4409	0.4409	0.5054	0.5914
6%	0.4624	0.4409	0.4409	0.4731	0.4731	0.5161	0.6237
8%	0.4839	0.4624	0.4731	0.4731	0.5054	0.5484	0.6237
10%	0.5054	0.4946	0.4946	0.4839	0.5054	0.5807	0.6022

The results in Table 1 furnishes us with interesting findings. First, the highest accuracy achieved with the major components (MajC) alone was just over 50% with an $\alpha = 10\%$. This was found at both the 30% and 70% contributions respectively. Increasing the number of major components actually decreased the accuracy for higher false alarm rates but slightly improved the values for lower false alarm rates. Our second finding is that when only the *minor* components were tested (MinC), a higher accuracy was immediately achieved. This is surprising as the major components explain the major variation in the dataset leading us to the conclusion that, at least for the dataset in question the components that contribute less to the dataset as a whole actually have greater capacity for modelling anomalies, suggesting that anomalous examples in this data are subtle and contribute to noise in the dataset as minor components generally contain a relatively high degree of the noise in a dataset.

After the analysis of Table 1 we chose the percentage contribution for the Major components to be 70% and those of the minor to be 20%. The ROC curve of this can be seen in Figure 1. Our first observation was the results were not optimal, at 66.67% accuracy this is just a 7% increase to using the minor components in isolation. When we again analysed the results of Table 1 we decided to test a configuration where an even greater emphasis was given to the minor components and less to the major components, as we realised that increasing the major components from 30% to 70% gave no great gains in accuracy. We chose to double the contribution from the minor components to 40%. The results again improved with an accuracy increase of just under 11% to 70.97% compared to using either of the major or minor components in isolation, and can also be seen in Figure 1.

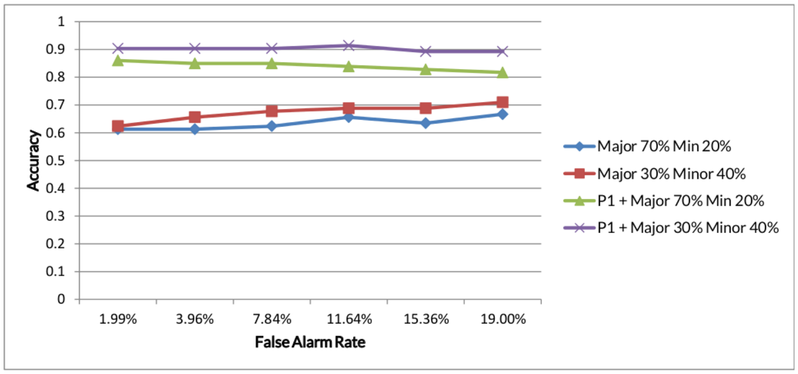


Fig. 1: P3/P4 and P1+P3/P4 Results at Various False Alarm Rates

Comparing Processes. Table 2 shows the comparison of the events found in P2 to the anomalies found with P3/P4, when taken as components in isolation, to those where P1 was coupled with P2 and with P3/P4. We chose to exclude the results from P1-P2-P3-P4 as only six clear outliers were found in P2 to be excluded from the P3/P4 processes and therefore did not have a great impact on the model. The results of the process in its entirety is also shown in Figure 1, demonstrating a clear gain in accuracy.

Measure	**P2 Chvnts. Univariate**	**P3/P4 PCC**	**P1+P2**	**P1+P3/P4**
Accuracy	0.5376	0.7097	0.8495	0.8925

Table 2: Comparing Processes

As we can see from 2, the PCC multivariate anomaly detector is far more accurate than the univariate outlier process, but once P1 is added, the results approach in accuracy. This is primarily due to zero values now being classed as anomalies from the added boundary detection step. A number of samples given to the sports scientist were found to contain zeros which he immediately classed as anomalies. Further evaluation could perhaps exclude samples containing zeros as these classifications do not accurately test the algorithm as this fundamental step is easy to compute and from the sports scientist's perspective, not difficult to identify via manual inspection. Given this caveat, once P1 was added we still achieved a final classification accuracy of 0.8925 once our algorithmic components were combined, showing the performance of the process as a whole was greater than any constituent process in isolation.

Some general findings include that anomalies (when the rolling window was not used), often occurred together in a sequential series. This gives credence to the hypothesis that in our time-series dataset, anomalies occurred in windows rather than at particular time-points. Furthermore, a subset of the events found from P2 and the anomalies found with P3/P4 overlapped at certain points, indicating strong events at these points. Finally, in an examination of the false positives, we found a number of the distance variables actually remained static, an unusual event for a GAA match and similarly for other sports events. Further evaluation will involve testing this hypothesis with the sports scientists, as we posit certain events even in the relatively small evaluation subset could have been missed upon manual inspection, due to fatigue or other factors incurred by examining vast swathes of data by eye. This secondary analysis could provide further motivation to this work, that is identifying anomalies in data that might have been missed by manual inspection.

5 Conclusions

Newly generated datasets often prove difficult for data mining as they can contain erroneous data-points and are often unsupervised (without classifications); datasets produced in areas such as sports science exemplify this. The first step in addressing these problems is anomaly detection, both to remove or adjust those values which are clear outliers, and to detect patterns which are signs of an event or change in the data. In this paper, we presented a novel anomaly detection algorithm which utilises both univariate and multivariate steps enabling us to determine which approach works best for a unsupervised time series dataset. Our results demonstrated the effectiveness of a combined approach when compared to even an improved univariate approach and that a multivariate approach outperforms it's univariate counterpart when used in isolation.

Acknowledgments

This work was partially funded by FP7 project Grant Agreement Number 304979 and also by Science Foundation Ireland under grant number SFI/12/RC/2289.

References

1. Rules of gaa football, 2015. Online; last accessed: 09/07/2015; www.gaa.ie/about-the-gaa/our-games/football/rules.

2. Yanping Chen, Eamonn Keogh, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, and Gustavo Batista. The ucr time series classification archive, July 2015. `www.cs.ucr.edu/~eamonn/time_series_data/`.
3. Haibin Cheng, Pang-Ning Tan, Christopher Potter, and Steven A Klooster. Detection and characterization of anomalies in multivariate time series. In *SDM*, pages 413–424. SIAM, 2009.
4. Chris Ding and Xiaofeng He. K-means clustering via principal component analysis. In *Proceedings of the twenty-first international conference on Machine learning*, page 29. ACM, 2004.
5. AM Fahim, AM Salem, FA Torkey, G Saake, and MA Ramadan. An efficient k-means with good initial starting points. *Georgian Electronic Scientific Journal: Computer Science and Telecommunications*, 2(19):47–57, 2009.
6. Matthew Blackwell James Honaker, Gary King. *Amelia II: A Program for Missing Data*, 2014. R package version 1.0 — For new features, see the 'Changelog' file (in the package source).
7. Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. Online; accessed 2015-06-29; Available on: `http://www.scipy.org/`.
8. Kingsly Leung and Christopher Leckie. Unsupervised anomaly detection in network intrusion detection using clusters. In *Proceedings of the Twenty-eighth Australasian conference on Computer Science-Volume 38*, pages 333–342. Australian Computer Society, Inc., 2005.
9. Moshe Lichman. 1999 kdd cup dataset, UCI machine learning repository, 2013. Dataset; Available on: `https://archive.ics.uci.edu/ml/datasets/KDD+Cup+1999+Data`.
10. Wes McKinney. Data structures for statistical computing in python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 51 – 56, 2010.
11. Jim ODonoghue and Mark Roantree. A framework for selecting deep learning hyper-parameters. In *Data Science*, pages 120–132. Springer, 2015.
12. Dan Pelleg, Andrew W Moore, et al. X-means: Extending k-means with efficient estimation of the number of clusters. In *ICML*, pages 727–734, 2000.
13. Mark Roantree, Donall McCann, and Niall Moyna. Integrating sensor streams in phealth networks. In *Parallel and Distributed Systems, 2008. ICPADS'08. 14th IEEE International Conference on*, pages 320–327. IEEE, 2008.
14. Mark Roantree, Jie Shi, Paolo Cappellari, Martin F. OConnor, Michael Whelan, and Niall Moyna. Data transformation and query management in personal health sensor networks. *Journal of Network and Computer Applications*, 35(4):1191 – 1202, 2012. Intelligent Algorithms for Data-Centric Sensor Networks.
15. Stephen M Ross. Peirce's criterion for the elimination of suspect experimental data. *Journal of Engineering Technology*, 20(2):38–41, 2003.
16. Mei-Ling Shyu, Shu-Ching Chen, Kanoksri Sarinnapakorn, and LiWu Chang. A novel anomaly detection scheme based on principal component classifier. Technical report, DTIC Document, 2003.
17. Qing Song, Wenjie Hu, and Wenfang Xie. Robust support vector machine with bullet hole image classification. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 32(4):440–448, Nov 2002.
18. Stefan Van Der Walt, S Chris Colbert, and Gael Varoquaux. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011.

FGWM: Workshop on Knowledge Management

Effectiveness of role plays on process-oriented behaviour in daily work practices: An analysis in the financial services sector

Michael Leyer[1], Ann-Kathrin Hirzel[2], Jürgen Moormann[2]

[1] University of Rostock, Rostock, Germany
michael.leyer@uni-rostock.de
[2] Frankfurt School of Finance & Management, Frankfurt, Germany
{a.hirzel;j.moormann}@fs.de

Abstract. We examine whether role plays have the potential to advance process-oriented behaviour (i.e. employees perform their activities while considering other activities and colleagues in the process) of employees in daily work practices. Process-oriented behaviour is difficult to achieve. To become process-oriented requires employees to have the ability (i.e. task and context-specific knowledge) and cognitive capabilities, as well as a willingness (i.e. intrinsic and extrinsic motivation) to change their daily work towards a cooperative and integrated procedure. In our paper we argue that role plays in which participants take over fictitious roles are a promising learning method. However, effects of role plays on subsequent behaviour in daily work practices are missing so far in the literature. Our results from 153 participants of a financial service institution reveal that the role play used has a significant impact on employees' process-oriented behaviour in terms of their cross-functional coordination, their process knowledge and their continuous process reflection, but not on employees' process awareness. Given that outcome, we argue that despite the application costs, role plays are beneficial for companies to train their employees in process orientation. Moreover, we show that there is no impact of the number of employees trained per department on individual process-oriented behaviour. Thus, initial pilot projects can be started and employees can be trained independent from their team. To increase the effect it should be repeated after a certain time, which we assume should not be more than one year, while keeping the cost of the role play training in mind.

Keywords: process knowledge, process-oriented behaviour, role play

Resubmission of Leyer, M./Hirzel, A./Moormann, J. (2015), Effectiveness of role plays on process-oriented behaviour in daily work practices: An analysis in the financial services sector, in: Becker, J./vom Brocke, J./de Marco, M. (Hrsg.), Proceedings of the European Conference on Information Systems 2015, 26.-29.05.2025, Münster, Deutschland, Paper 1870.

Relational Presentations Using Semantic Closeness
Spatial Narrative for Mathematical Content

Naomi Pentrel, Michael Kohlhase

Jacobs University Bremen

Abstract. Visualization of knowledge is important to foster learning. Especially so in Mathematics where students have to understand not just one topic at a time but also the related concepts. Taking the typical Mathematics lecture as an example, it is often the case that students come from different backgrounds. When a new topic is introduced it would therefore be ideal to have a simple way to find dependencies and present students with an easy way to catch up on topics they have not learned. To optimize the visualization of information and its interdependencies, a way to present information without losing context is therefore necessary. The following research takes an existing annotated corpus and presents its contents while allowing students to see dependencies between topics and encouraging them to explore related mathematical concepts. Thus students can interactively learn the concepts the current topic depends on by taking small detours through those topics, should they need to refresh their memories. This approach to presenting learning materials changes how we interact with course materials and it is ultimately applicable to almost all areas in which knowledge needs to be transferred.

Original Publication as:
Naomi Pentrel and Michael Kohlhase. *Relational Presentations Using Semantic Closeness Spatial Narrative for Mathematical Content.* In: Proceedings of the MathUI 2015 10th Workshop on Mathematical User Interfaces. Washington, USA. July 2015, published at http://www.cermat.org.

iClass – Applying Multiple Multi-Class Machine Learning Classifiers combined with Expert Knowledge to Roper Center Survey Data

Marmar Moussa[1] Marc Maynard[2]

[1]University of Connecticut, CT, USA
marmar.moussa@uconn.edu
[2]Roper Center for Public Opinion Research, CT, USA
mmaynard@ropercenter.org

Abstract. As one of the largest public opinion data archives in the world, Roper Center [1] collects datasets of polled survey questions as they get released from numerous media outlets and organizations with varying degrees of format ambiguity. The volume of data introduces search complexities over survey questions asked since the 1930s and poses challenges when analyzing search trends. Up to this point, Roper Center question-level retrieval applications used human metadata experts to assign topics to content. This has been insufficient to reach required levels of consistency in catalogued data, and provides an inadequate base for creating an advanced search experience for research clients.

The objective of this work is to combine the human expert teams' knowledge of the nature of the poll questions and the concepts and topics these questions express, with the ability of multi-label classifiers to learn this knowledge and apply it to an automated, fast and accurate classification mechanism. This approach cuts down the question analysis and tagging time significantly as well as provides enhanced consistency and scalability for topics' descriptions. At the same time, creating an ensemble of machine learning classifiers combined with expert knowledge is expected to enhance the search experience and provide much needed analytic capabilities to the survey question databases.

In our design, we use classification from several machine learning algorithms like SVM and Decision Trees, combined with expert knowledge in form of handcrafted rules, data analysis and result review. We consolidate this into a 'Multipath Classifier' with a 'Confidence' point system that decides on the relevance of topics assigned to poll questions with nearly perfect accuracy.

Keywords: ensembles; expert knowledge; knowledge base; machine learning; multi-label classifiers; supervised learning; survey datasets

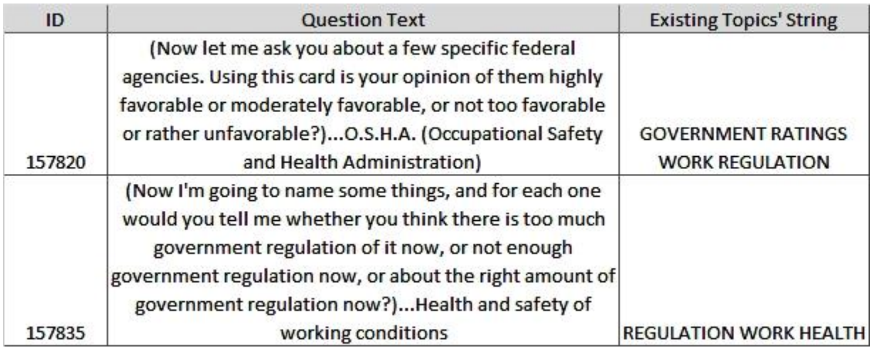
1 Introduction

In this paper, we present an overview of our work at the Roper Center in applying machine learning to the public opinion survey datasets in an attempt to classify the questions to their respective most relevant set of topics/classes. The application iClass is a collection of modules of autonomous classifiers and a knowledge expert 'Admin' module which allows us to combine both human knowledge and machine learning to the classification and review processes. This paper presents the nearly complete first phase of iClass. We describe the business context and motivation behind this development, a design overview and preliminary evaluation results. The last section describes the business payoff and trends for future phases.

1.1 Context and Motivation

The Roper Center collects datasets of survey questions from polls performed by think tanks, media outlets, and academic organizations. The data has been gathered since the 1930s with varying degrees of format ambiguity. The volume of legacy data introduces search complexities and poses challenges when analyzing search trends.

Homegrown backend systems serve up several data retrieval and analysis services to the Roper Center members. The primary two are 1) iPOLL, a question-level retrieval database containing over 650,000 polling questions and answers, and 2) Roper*Express*, a catalog of survey datasets conducted in the US and around the globe. Historically, datasets, iPOLL questions, and secondary material have been managed and cataloged by separate teams, which led to different descriptive practices. Dataset expert teams use free text key-word descriptors to assign topics to content. This means, even though there are clear topical and other kinds of connections among the content, lack of consistent description creates rifts, making these connections elusive (Fig.1). It results in costly string operations for even simple tasks, as well as costly retrospective updates to topics definitions and adding new topics. This approach also does not allow for any further data analytics capabilities.



ID	Question Text	Existing Topics' String
157820	(Now let me ask you about a few specific federal agencies. Using this card is your opinion of them highly favorable or moderately favorable, or not too favorable or rather unfavorable?)...O.S.H.A. (Occupational Safety and Health Administration)	GOVERNMENT RATINGS WORK REGULATION
157835	(Now I'm going to name some things, and for each one would you tell me whether you think there is too much government regulation of it now, or not enough government regulation now, or about the right amount of government regulation now?)...Health and safety of working conditions	REGULATION WORK HEALTH

Fig. 1. Example of inconsistent topics assigned to 'similar' content

Our objective is therefore to develop a scalable system for concept-based classification of questions that implements an intelligent automated approach for identifying conceptual links between content at point of acquisition/creation using machine learning classifiers while at the same time leverage existing expert knowledge.

1.2 Related Work

In statistics and machine learning, ensemble methods achieve performance by combining opinions of multiple learners [2]. There are different ways of combining base learners into ensembles [3]. We decided to design a combining method that is tailored to our specific goals like scalability and utilizing available expert knowledge. This is required to accommodate changes in topic definitions over time and the emergence of new topics from newly acquired studies. Our combining method is a mix of weighting, majority voting and performance weighting. In weighting methods a classifier has strength proportional to its assigned weight. In a voting scheme, the number of classifiers that decide on a specific label is counted and the label with the highest number of votes is considered. For performance weighting [4], the weight of each classifier is set proportional to its accuracy performance on a given validation set.

2 Design Overview

2.1 Data Analysis and Tools

Several housekeeping steps had to take place before we would be able to develop a reliable system with high accuracy. The first was performing a data cleanup. The initial classification tests revealed numerous discrepancies and inconsistencies between the actual concepts of questions and assigned topics as described in Section 1.1.

Also, the review revealed the need for a number of new topics and a three-level topic hierarchy. This meant defining categories at the parent level in a new topics hierarchy (Fig.2), as well as refining existing topic definitions to achieve consistency. The effort resulted in 119 topics for the current question bank, with over 20 new topics, identified as a result of initial classification test reviews and analysis. The topics were arranged into 6 main categories and 3 levels of hierarchy. We also needed to implement necessary workflow changes to include testing results review, a review of the ‘Before & After’ list of topics associated with each question. An ‘Admin’ role with the necessary expert knowledge reviews the result and ‘approves’ the topics assigned and selects some of the accepted question-topic pairs to be fed back into the training set.

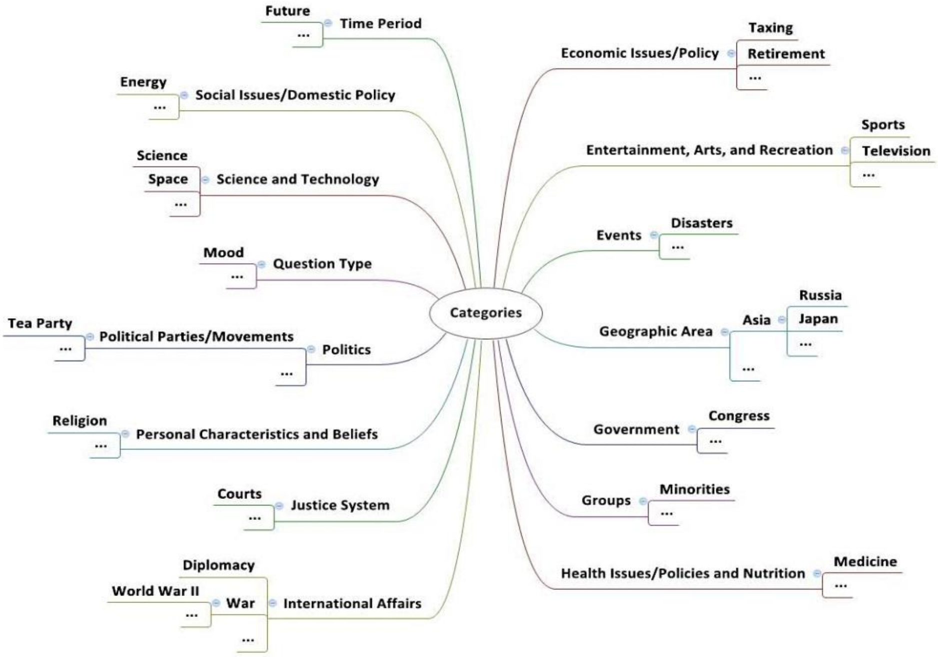


Fig. 2. New Categories Hierarchy

Roper Center metadata is stored in an Oracle 11g database, prompting an examination of machine learning algorithms supported by Oracle classifier functions. We conducted tests using the RTextTools package over datasets exported from the Oracle Database [5], also tests and evaluation using python scikit-learn package over exported data [6]. The main modules however used Oracle Pl/SQL for analysis as well as the training and classification for compatibility with the Roper Center's architecture.

2.2 Machine Learning Algorithms

We used two machine learning techniques for this phase of iClass, Support Vector Machine (SVM) and Decision Tree. SVM is known to perform well with significant accuracy, even with sparse data, also SVM classification attempts to separate target classes with the widest possible margin, and is very fast. Distinct versions of SVM use different kernel functions to handle different types of data sets. Linear and Gaussian (nonlinear) kernels are supported in SVM. We used linear kernels in this phase of iClass. SVM however does not produce human readable rules. In contrast, the Decision Tree (DT) algorithm produces human readable and extendable rules. Decision trees extract predictive information in the form of human-understandable rules. The rules are nearly if-then-else expressions; they explain the decisions that led to the prediction. DT has good missing value interpretation, is fast and performs with good accuracy [7].

3 Design Details

3.1 Classification Process Flow

The assembling of a comprehensive training set that represents all topics and their features was challenging yet critical for success. The data analysis and initial tests resulted in a selected set of expert-classified questions to use as the seed for the training set. The training set also included handcrafted question samples for under-represented topics. For new topic definitions, we used a set of SQL queries for a fine-grained selection of questions to be assigned the new topics.

After the training set is constructed, SVM and Decision Tree classifiers are trained to produce a set of rules for each topic. Each topic also gets an additional set of Admin/Expert-defined rules in the form of keywords to look for or exclude from the question text. These manually defined rules formed the third set of rules to process.

Three modules (DT, SVM and Rule-Based Classifiers) are created to use these sets of rules and 'vote' with different scores over the topics to be assigned. A fourth path for classification is formed by the direct SQL queries representing the more complex expert defined rules that are not included in the Rule-Based Classifier. For this path too, the implementation assigned confidence scores to the selected question-topic pairs. The four paths' (sources) results construct a vector for each question and topic pair, containing the source and the designated score/confidence.

(Fig.3) below provides a description of this process flow in iClass. Three values are then considered in combining the information from this ensemble of classifiers: 1) the (weighted) number of sources/votes that classified a topic to a specific question, 2) the threshold (possible one for each topic and source) that would consider this classification true positive or false positive, and 3) the confidence/score values.

A combined confidence/score is formed and then the classified question-topic pairs are reviewed by an expert to approve or reject. Approved results can then be fed back to the training set pool for a new round of training. This is needed as the dataset grows with incoming poll questions from newer studies acquired by the Roper Center.

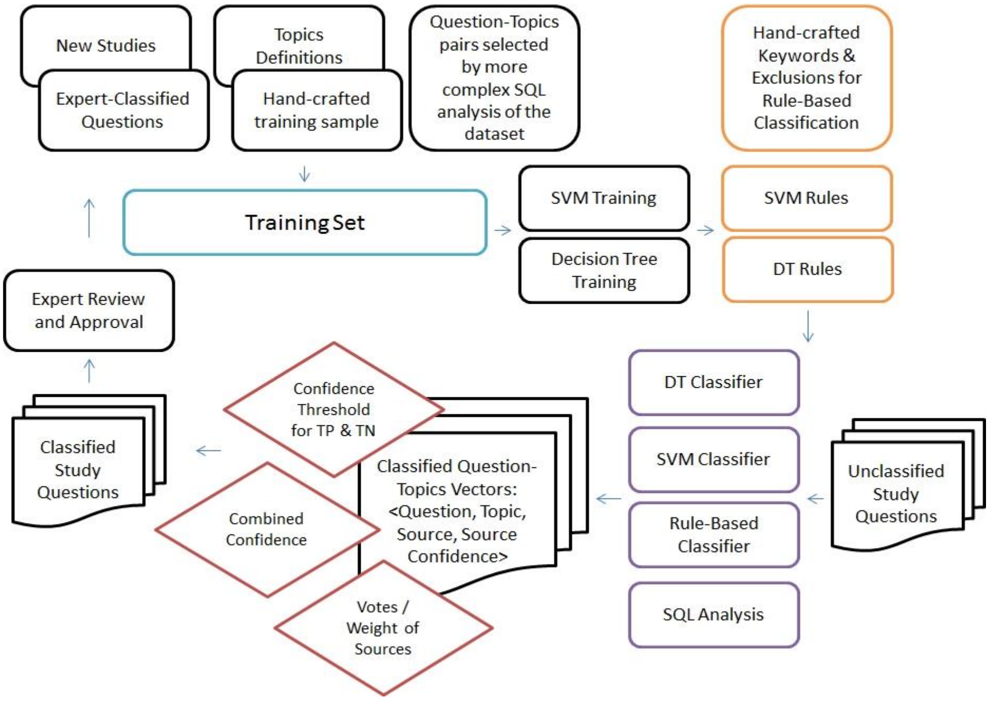

New Studies
Expert-Classified Questions
Topics Definitions
Hand-crafted training sample
Question-Topics pairs selected by more complex SQL analysis of the dataset
Hand-crafted Keywords & Exclusions for Rule-Based Classification
Training Set
SVM Training
Decision Tree Training
SVM Rules
DT Rules
Expert Review and Approval
DT Classifier
SVM Classifier
Rule-Based Classifier
SQL Analysis
Unclassified Study Questions
Classified Question-Topics Vectors: <Question, Topic, Source, Source Confidence>
Confidence Threshold for TP & TN
Combined Confidence
Votes / Weight of Sources
Classified Study Questions

Fig. 3. iClass Components and Process Flow details

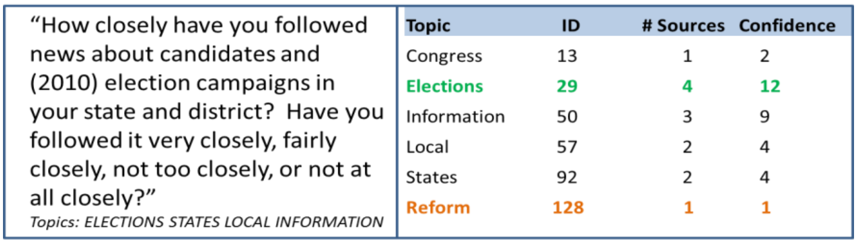
3.2 Confidence

1
2
3
4
5
6
7
8
9
10
11
12

Fig. 4. Confidence Points 1(low) to 12(high)

As described in previous sections, iClass current phase has four different sources of classification: SVM, DT, Rule-Based and Expert/Manual direct selection paths. To combine them, we applied a ‘Confidence Points System’. Confidence/relevance levels (Low, Medium, and High) from each classification algorithm/path aggregate to an (N*3) point system, where N is the number of classification paths. As we currently implement 4 paths, there are 12 Points of Confidence (Fig.4).

For sources 1 and 2, SVM and DT, the confidence is calculated via the Classifier functions as a value > 0 and <100, we convert this to a value 1→ 3 by using a pseudo-count, scaling, and rounding. For the Expert/Manual path, where the direct analysis process is implemented in the various SQL scripts, the confidence for each topic is configured directly based on the Admin’s analysis. For the Rule_Based Classifier, each topic is assigned a rule confidence level associated with the keywords and exclusion rules defined for that topic. A question-topic classified pair has therefore 1 → 12 possible confidence points: if 4 sources vote for a topic with the max confidence points (3) each, then the total confidence for this question-topic classification is 12. If on the other hand only one source votes for this assignment and with the lowest confidence possible (1), the total will be 1 point (Fig.5). The Admin sets different thresholds for different functionalities, for instance a threshold >3 to appear in search results, a threshold of ≤2 for admin review process to look at the weakest items.



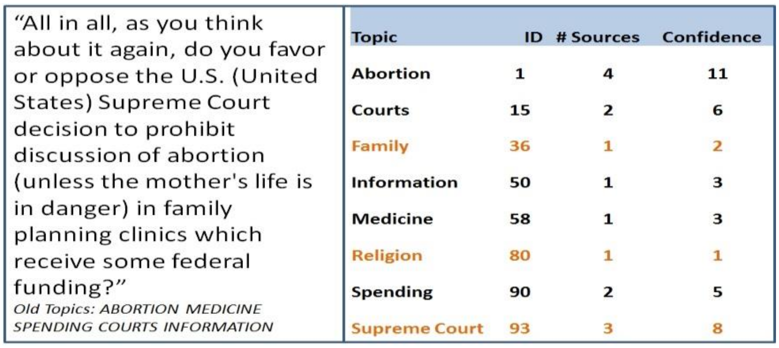
"How closely have you followed news about candidates and (2010) election campaigns in your state and district? Have you followed it very closely, fairly closely, not too closely, or not at all closely?"

Topics: ELECTIONS STATES LOCAL INFORMATION

Topic	ID	# Sources	Confidence
Congress	13	1	2
Elections	29	4	12
Information	50	3	9
Local	57	2	4
States	92	2	4
Reform	128	1	1

Fig. 5. Example of Question-topics assigned with highest and lowest confidence.

A tradeoff exists between adding more (maybe distantly related) topics which could cause a degree of confusion to the reviewer/user versus being extra cautious in assigning topics and risking that related questions might not appear in results of related but not main topic searches. (Fig.6) is an example of this tradeoff, and is also an example of how iClass identified more relevant topics than were assigned by a human cataloger. 'Family' and 'Religion' topics in this example, although both are topics long available in the system, were not initially assigned by manual classification during data entry. iClass assigned lower confidence to these topics compared to other more relevant topics, like 'Abortion' and 'Courts'. Topic 'Supreme Court' is a new topic. It is also very relevant and is correctly captured and assigned a high confidence level.



"All in all, as you think about it again, do you favor or oppose the U.S. (United States) Supreme Court decision to prohibit discussion of abortion (unless the mother's life is in danger) in family planning clinics which receive some federal funding?"

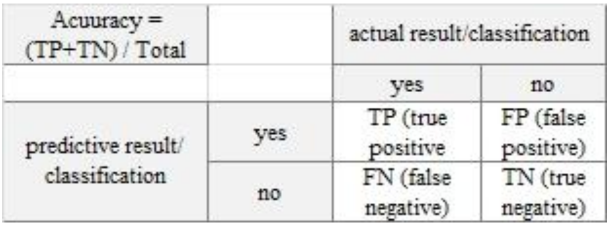
Old Topics: ABORTION MEDICINE SPENDING COURTS INFORMATION

Topic	ID	# Sources	Confidence
Abortion	1	4	11
Courts	15	2	6
Family	36	1	2
Information	50	1	3
Medicine	58	1	3
Religion	80	1	1
Spending	90	2	5
Supreme Court	93	3	8

Fig. 6. Example of the new classification results

4 Evaluation

The evaluation of classical multiclass classifiers is by nature challenging, as most of the metrics usually make the most sense when applied to binary classifiers. One way to explore the performance of a multiclass classifier is to construct the confusion matrix (Fig.7) and extend it to (NxN) matrix, where N is the number of classes (topics).



Acuuracy = (TP+TN) / Total		actual result/classification	
		yes	no
predictive result/ classification	yes	TP (true positive	FP (false positive)
	no	FN (false negative)	TN (true negative)

Fig. 7. Binary Confusion Matrix

Aside from having multi-classes in our system, we have a further complexity; a question can be assigned multiple topics with varying confidence/relevance levels. A threshold then determines whether or not questions with lower confidence points are counted towards FP or TP. The cutoff between TP and FP is therefore a little blurred.

Results based on SVM, DT & Rule-Based Classifier modules only:	
Hits: Avg. # of Questions with all topics TP (657,850 total Questions)	576,902
Average Accuracy ((TP+TN)/total)	**0.917**
False Negative/Miss Rate	0.026
"False Positive" Rate	0.030
# Newly identified Question-Topic pairs (Not present in training set)	695,792
# New correct topic assignments rate (added value)	0.455

Fig. 8. Evaluation Results

Our evaluation of only 3 paths, the SVM, DT and Rules_Based Classifier results showed accuracy of 91.7% and over 99% when enabling all 4 paths. This is expected as the 4th path involves more direct human knowledge over the classification decision.

5 Conclusion & Future Work

The development of the first phase of iClass, combining machine learning and expert knowledge, introduced performance and administrative benefits to the business process at the Roper Center. To name a few, the automated classification contributed to better consistency in topics definition and faster, streamlined topics assignment. The expert/Admin review process is dramatically shortened as it is focused on low confidence items. The process is now change-tolerant; when adding/updating topics, we can reflect the updates over the entire questions' bank retrospectively. In terms of performance enhancement, the elimination of costly string operations improves functionalities like search and navigation by topics. Although still a work in progress, iClass is scalable in terms of thresholds, confidence level configurations as well as adding entire extra classification paths to the system. Analytic capabilities are now part of the system, like efficient metadata statistics, especially about topics trends.

From the application perspective, several components of iClass need further work in next phase, a more user-friendly Admin module is planned, the system currently supports only one set of handwritten rules per topic definition, as well as only one admin user, which needs to be extended for business needs. The classification of the datasets only lays the groundwork for better data analytics, which is currently not fully leveraged. There is also the business need to extend the functionality of iClass to knowledgebase facets other than topics, such as the survey sample classes. In addition, there is still a great deal to be explored about learning techniques that best fit the business. As the classification process is prepared to accept more classification paths, part of the future work includes using other machine learning algorithms to create more classification paths, as well as study other ensemble classification methods for combining weights and votes, and compare the results of the different methods.

6 References

1. http://www.ropercenter.uconn.edu/
2. Polikar R(2006) Ensemble based systems in decision making. IEEE Circuits Syst Mag 6(3)
3. Rokach, L. (2010). "Ensemble-based classifiers". Artificial Intelligence Review Artificial Intelligence Review. February 2010, Volume 33, Issue 1-2, pp 1-39
4. Opitz D, Shavlik J (1996) Generating accurate and diverse members of a neural network ensemble. In: Touretzky DS, Mozer MC, Hasselmo ME (eds) Advances in neural information processing systems, vol 8. The MIT Press, Cambridge, pp 535–541
5. Timothy P. Jurka, Loren Collingwood, Amber E. Boydstun,Emiliano Grossman,Wouter van Atteveldt (2014) Automatic Text Classification via Supervised Learning
6. "Scikit-learn.": Machine Learning in Python — 0.17.dev0 Documentation.
7. Smola, Alex, and S.V.N. Vishwanathan. Introduction to Machine Learning. Cambridge: Cambridge UP, 2008.

A Pattern-based Approach to Transform Natural Text from Laws into Compliance Controls in the Food Industry

Andrea Zasada, Michael Fellmann

Rostock University, Rostock, Germany
azasada@web.de, michael.fellmann@uni-rostock.de

Abstract. In the food industry, regulations support companies to specify what needs to be done to minimize the risks of processing, trade and consumption of inferior food products. Complying with regulations protects companies from expensive and negative perceived product recalls, sanctions and financial penalties. A compliant manufacturing process requires a process design that conforms to legal requirements, quality and safety standards. Regulations are generally described in natural text so that relevant information has to be retrieved and formalized before it can be used for process description. In this contribution, we use a sample of laws and an initial set of generic control patterns to explore the scope of food regulations and the extent of formalization that can be reached by applying control patterns. All in all, we present a pattern-based approach to turn natural text from laws into formalized machine-readable constructs that may serve as basis for a compliant process design.

Keywords: Business Process Management, Control Pattern, Business Process Compliance, Regulations, Food Industry

1 Motivation and Introduction

The "act of being in alignment with guidelines, regulations and/or legislation" is defined as *compliance* [6]. This definition implies that compliance does not only comprise the adherence to laws but also standards, codes of practice and business partner contracts [9]. Compliance has been driven by reforms of the American banking and insurance sector since the 1990s, when more and more scandals of money laundering and insider trading have been revealed [10, 14]. The increasing reform pressure finally summits in the Sarbanes-Oxley Act (SOX) of 2002 which makes listed companies responsible for establishing and maintaining an internal control system [11].

Similar observations can be made for the food industry, where compliance is seen as a current issue but an old problem that has been subject of many regulative attempts [10, 14]. Most frequent compliance offences in the food industry relate to violations of disclosure information, tax and import regulations and to the processing and trade of spoiled food [4]. *Business process compliance* considers how a business operation or service should be carried out to comply with a normative system while executing a process [5]. In this regard, control patterns are important since they can be understood as high level domain-specific templates which can be applied to specify recurring process requirements like regulations [13]. A regulation is a declarative written statement defined as “a rule or order issued by an executive authority or regulatory agency of a government and having the force of law” [7].

The purpose of this paper is to reduce the complexity regulations making implicit information accessible and machine-readable through the use of control patterns. The challenge is to identify and convert relevant process information from natural text into formalized constructs that can implemented by process execution languages. The investigation’s focus lies on the degree of formalization (extent) and the thematic focus (scope) of a real-world domain (food industry), which is used as empirical basis for specifying control patterns. In behalf of that, the resulting research questions are:

RQ1: *What is the scope of regulations in the food industry?*
RQ2: *To what extent can regulations be formalized by control patterns?*

To answer these two research questions, we discuss related work and present a conceptual model for automating compliance checking in Section 2. In Section 3 we continue with the textual analysis of German food regulations. The regulations have been retrieved by querying the database of the Federal Ministry of Justice and Consumer Protection [2]. The title search for the keyword “food” led to 20 national regulations, which were analyzed to specify requirement, objective and risk for every single regulation. Control patterns that are extracted from regulations are classified with regard to the given process information. Concluding remarks and prospects on future work are given in Section 5.

2 Principles of Control Patterns

2.1 Related Work and Problem Specification

Considerable work on patterns has been provided by Dwyer, Avrunin and Corbett (1999), who developed a pattern system for finite-state verification based on a large sample of over 500 examples of property specifications [1]. Extensive work on compliance automation has also been conducted by Sadiq, Governatori (2015) [9], Namiri (2007) [8] as well as Turetken et al. (2012) [13] by exploiting formal techniques (e.g. MTL/LTL and FCL) in alignment to the de facto standard COSO for managing internal controls. COSO has been settled by the Committee of Sponsoring Organizations of the Treadway Commission (COSO) to comply with significant regulations like SOX [12]. We decided to build our conceptual model upon the four control patterns *Order*, *Occurrence*, *Resource* and *Time* suggested by Turetken et al. (2012) [13] because of the existence of a framework for the key elements of *business process compliance management* (BPCM) and its alignment to an established control framework like COSO. The key elements of BPCM refer to the operational activities of compliance management (e.g. risk assessment and response) and corresponding entities of the compliance repository (e.g. risk).

2.2 Conceptual Model for Capturing Compliance Controls

In order to capture compliance controls in the food industry we adopted the BPCM framework of Turetken et al. (2012) [13]. The focus of the framework has been shifted from operational compliance management activities to the formalization of natural text language through control patterns. *Control Patterns* form a separate layer in the continuum of abstraction ranging from *Regulations* to machine-readable *Process Execution Languages* (see Fig. 1). Each layer contains several process elements represented by different operands (compare Section 3.2). *Regulations* are the source of compliance requirements used to define the requirement, objective and risk of a control. The smallest entity of a *Regulation* is a rule. In this layer relevant rules are adopted, control objectives are set and possible risks are assed. The next layer is assigned to the scope of *Control Patterns.* Within this layer the templates for process controls are defined and classified. In the bottom layer we specified a number of criteria for selecting a compatible *Process Execution Language* to pave the way for automated compliance controls.

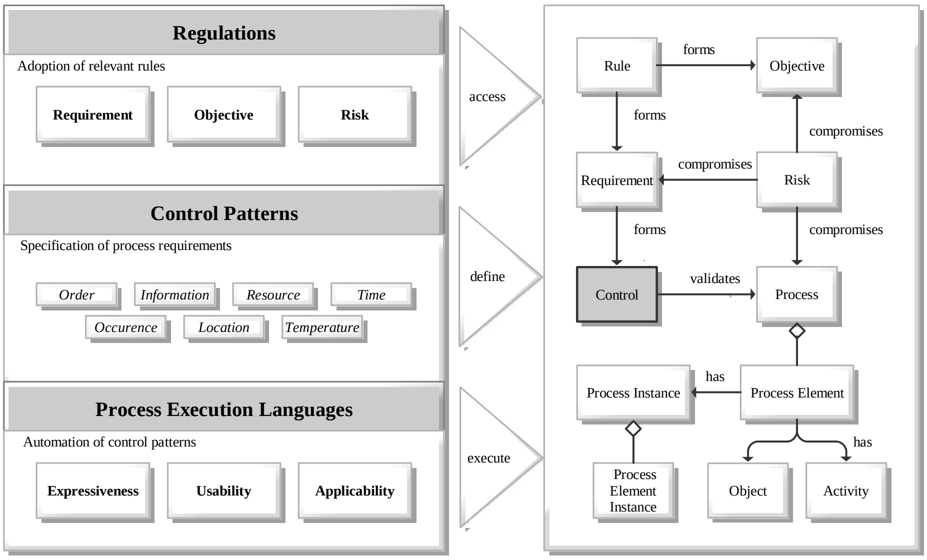


Fig. 1. Conceptual model for compliance checking[1]

As we conducted a facet classification on compliance checking approaches in previous work [3], we adopted one of its dimensions to assess the scope of regulations in the food industry. We chose the dimension *Scope* because we wanted to analyze the applicability of its elements in more detail. The dimension *Scope* is based on the compliance concerns identified by COMPAS, a study on Compliance-driven Models, Languages and Architectures for Services (COMPAS), which has been conducted by Tilburg University (2008) [12]. The study introduces two categories of compliance

[1] In alignment to Turetken et al. (2012).

concerns that have been aligned from business process modeling. The first category comprises the basic compliance concerns control flow, locative, information, resource and time. The second category describes more advanced compliance concerns (e.g. monitoring, privacy and quality aspects).

3 Applying Control Patterns to Capture Compliance Controls

3.1 Text Analysis of Regulations in the Food Industry

Regulations for the German food industry have been discovered by searching the database of the Federal Ministry of Justice and Consumer Protection, which claims to offer nearly the entire body of federal law [2]. A title search for the German equivalent for “food” returned 20 hits of national regulations, which were further analyzed to gain information on the requirement, objective and risk of each regulation. The analysis of subsequent paragraphs and sections of each regulation led us to a total of 108 single requirements with process characteristics. The requirements are used to extract important process information for specifying control patterns. While a requirement can be seen as an early stage of a control pattern, the objective is necessary to express the importance of each control and the risk to access the negative consequence of non-compliance. Table 1 shows an excerpt of the complete listing which is addressing the scope of compliance regulations (compare RQ1). The advantage of the chosen examples is that they cover nearly all facets of the dimension *Scope*, which is used in Section 3.2 to demonstrate the transformation from compliance requirements to control patterns. The retrieved types of regulations vary from the definition of:

- quality controls,
- hygiene and purity requirements,
- requirements regarding the processing of goods,
- preventing the spread of animal diseases,
- requirements regarding transport and storage,
- disclosure agreements to
- tax and export regulations.

	Regulation	**Requirement**	**Objective**	**Risk**
01	LMÜV *§ 5, Sec. 1*	(1) Fulfil occasionally imposed obligations to combat animal diseases. (2) Take precaution if infectious animal diseases occur.	Conduct quality controls if infectious animal diseases are reported.	Spread of infectious animal diseases.
02	LMEV *§ 9, Sec. 2*	Export goods within 30 days to a third country or store goods within 60 days in an approved or registered national storage unit.	Export goods within a certain time limit or store goods in an approved or registered national storage.	Violation of tax and import regulations.

03	LME *Appendix 4, Chapter I, No. 3*	Depending on the statutory sample size, a sensory testing and a legal assessment have to be conducted after opening the packaging.	Conduct quality control to check goods after opening the packaging.	Processing, trade and consumption of spoiled or contaminated goods.
04	TLMV *§ 2*	During deep freezing, goods have to be separated from specified inadmissible substances.	Prevent contact to forbidden substances.	Processing, trade and consumption of spoiled or contaminated goods.
05	ATP *§ 5*	Containers classified as thermal maritime by land without transloading the goods does not require an export permit.	Transport goods without permit if containers are classified as thermal maritime by land.	Violation of disclosure agreements.
06	LMHV *§ 20*	Transport and store chicken eggs 18 days after laying date at a temperature between 5 °C and 8 °C.	Transport special goods within a certain time limit at a given temperature range.	Processing, trade and consumption of spoiled or contaminated goods.

Table 1. Examples for regulations in the food industry

After completing the text analysis by following the example of Table 1, we were able to identify four different risk types that are representative for our sample of regulations in the food industry, namely the:

- processing, trade and consumption of spoiled or contaminated goods,
- spread of infectious animal diseases,
- violation of disclosure agreements and
- violation of tax and import regulations.

Subsequent risks are negative consequences like disposal costs, sanctions and financial penalties or even health hazards. However, these consequences depend on the risks above so they have not been considered as single risk types. Given these explicit information on requirement, objective and risk the next Section is dedicated to the control pattern layer that serves as intermediary to automate compliance controls with process execution languages (compare Section 2.2).

3.2 Specification of Control Patterns in the Food Industry

The formalization of legal text implies to find a reasonable abstraction level. This raises the question to what extent regulations can be formalized by simple constructs like control patterns (compare RQ2). Table 2 provides an overview on frequent control patterns in the food industry. Due to space limitations, only those patterns have been listed that have been applied to formalize compliance regulations. The frequency (FRQ) indicates how often a pattern has been used and to which category it belongs. The listing contains 21 unique control patterns that can be combined to express even more complex compliance requirements using operands and Boolean delimiters (see Table 3). Patterns can be defined using simple verb constructs and prepositions (e.g. O_i *CompliesWith* Q_i*)*. Operands are either used to specify general process elements (e.g. object O_i) or specific compliance concerns (e.g. quality control Q_i), which were introduced in Section 2.2. A complete description of operands is given in Table 2.

		Pattern	**FRQ**	**Description** Given *A, O, l, p, k* and *t* as operands representing process elements: *A* = activity, *O* = object, *l* = location, p = production facility, *k* = time, *t* = temperature and *Q, D* and *P* as operands representing compliance concerns: *Q* = quality, *D* = disclosure and P = security precautions, with *i, j* = 1, 2, 3, …*n*, $i \neq j$ and constant *m*.
Order	Basic	A_j *Precedes* A_i	1	A_i must be preceded by A_j.
		A_i *LeadsTo* A_j	1	A_i must be followed by A_j.
Res.	Basic	O_i *Exclusive* O_j	6	If O_i is present then O_j must be absent and vice versa.
		O_i *Exists*	3	O_i must exist in the process specification.
Location	Basic	*ProcessedWith* p_i	5	Used with order and occurrence patterns to denote a given O_i is processed with production facility $p_{i.}$
		StoredIn l_i	4	Used with order and occurrence patterns to denote a given O_i is stored in storage unit l_i.
		MovedFrom l_i *MovedTo* l_j	4	Used with order and occurrence to denote a given O_i is moved from storage unit l_i to another storage unit l_j.
		(O_i*, …; m) Multi-ProcessedWith* p_i	3	A set of objects *(*O_i *, …)* has to be processed with a certain number of *m* different production facilities p_i.
		(O_i*,…; m) Multi-StoredIn* l_i	1	A set of objects *(*O_i *, …)* has to be stored in a certain number of *m* different storage units l_i.
Information	Advanced	O_i *CompliesWith* Q_i	24	Object O_i complies with quality standards, hygiene and purity requirements by passing regular quality controls as well as extraordinary quality controls $Q_{i.}$ Subject of these controls are e.g. temperature, weight, date of expiry, ingredients, texture and consistence.
		O_i *CompliesWith* D_i	10	Object O_i complies with disclosure requirements D_i. Subject of these requirements are the consumer protection, tax and import regulations e.g. by correct and complete product declaration, complying with quality and security standards, transparent production processes and a traceable supply chain.
		A_i *CompliesWith* P_i	3	Activity A_i has to be performed with special security precautions P_i in order to protect users from e.g. infectious animal diseases.
		A_i *CompliesWith* Q_i	2	Activity A_i complies with quality standards, hygiene and purity requirements by applying regular quality controls as well as extraordinary quality controls Q_i (e.g. to prevent the spread of animal diseases).

Time	Basic	*Within k*	10	Used with order pattern to denote a given A_i to happen within *k* time units.
		Before k	2	Used with order patterns to denote a given A_i to happen before *k* time units.
	Adv.	A_i *ExistsMax/Min k*	2	A_i must hold at most/minimum k time units once it happens
		A_i *ExistsEvery k*	1	A_i must happen in every *k* time unit.
Temperature	Basic	*Within* t_j *and* t_i	7	Used with time patterns to denote a given O_i is tempered within temperature *t* (with $i > j$).
		Below t	7	Used with time patterns to denote a given O_i is tempered below temperature *t*.
		ExactlyAt t	1	Used with time patterns to denote a given O_i is tempered exactly at temperature *t*.
	Adv.	O_i *ExistsMax/Min t*	1	Object O_i has to be tempered at most/minimum at temperature *t*.

Table 2. Specification of frequent control patterns in the food industry[2]

To formalize the requirements given in Table 2, we distinguish a number of typical keywords for each pattern. For example, a control is often aligned to the assurance of quality standards, so that the word “control” is tied to an *Information* pattern. *Resource* patterns (Res.) are usually described by expressions that indicate how goods should be handled, which is indicated by word orders like “prevent contact”. *Location* patterns are clearly addressed if something is about to be “processed”, “moved” or “stored”. Depending on the context, keywords like “within” or “below” can also indicate if a pattern depends on *Time* and/or *Temperature* pattern. The most important indicators to classify control patterns with regard to our conceptual model for automating checking are:

- temporal order (e.g. precedes or leads),
- occurrence (e.g. exists, absent or universal),
- human resource (e.g. to segregate or merge activities),
- location in conjunction with the process status (e.g. processed, moved or stored),
- time limitation (e.g. interval, minimum or maximum) and
- temperature setting (e.g. within, below, above or exactly at).

Instead of the control flow proposed by COMPAS [12] we used the three patterns *Time*, *Order* and *Occurrence* recommended by Turetken et al. (2012) [13] and expanded the focus of the *Resource* pattern from the segregation of duties to the segregation of input goods. Besides, we added an information, location and temperature pattern. The *Information* pattern indicates which legal source, control objective and risk is addressed or whether the requirements of a quality control, security precaution or disclosure agreement is met. This ensures transparency and provides valuable con-

[2] According to Turetken et al. (2012). Newly added control patterns are indicated by a grey filled table row.

text information about the impact of different food regulations. The *Location* pattern considers how goods should be stored, moved or where they are processed with regard to time and temperature constraints. The *Temperature* pattern is necessary to capture compliance regulations regarding the storage and transport of perishable food. The final set of control patterns consists of seven categories: *Order, Occurrence, Resource, Location, Information, Time* and *Temperature*. Table 3 concludes with the formalization of compliance regulations that started with Table 1. It shows simple patterns as well as more complex patterns to demonstrate the applicability of the most frequent compliance patterns in the food industry.

	Control Patterns	**Order**	**Occurrence**	**Resource**	**Location**	**Information**	**Time**	**Temperature**
01	*Oi CompliesWith Qi AND Ai CompliesWith Pi AND Oi ProcessedWith pi*				■	■		
02	*(Oi MovedFrom li MovedTo lj Within k) OR (Oi StoredIn li Within k)*				■		■	
03	*Ai LeadsTo Aj AND Oi CompliesWith Qi*	■				■		
04	*Oi Exclusive Oj*			■				
05	*(Oi MovedFrom li MovedTo lj AND Oi Exists) AND Oi CompliesWith Di*			■	■	■		
06	*(MovedFrom li MovedTo li OR StoredIn li) Within tj and ti*				■			■

Table 3. Examples for control patterns in the food industry

4 Conclusion and Outlook

In this contribution we applied a pattern-based approach for specifying compliance controls in the food industry. Based on a sample of 20 legal text documents, provided by the German Federal Ministry of Justice and Consumer Protection law database, we derived 108 legal statements with process character. These were used to analyze the content of every regulation concerning requirement, objective and risk. To access the scope of food regulations we adopted a business process compliance framework and expanded it by refining the *Scope* of control patterns by *Resource, Location, Information* and *Temperature* patterns. Determining the frequency of compliance patterns we were able to present a list of relevant control patterns in the food industry. The use of control patterns has been illustrated by a choice of regulations which address the previously defined facets of the *Scope* dimension. This led to a deeper understanding of the involved process elements and compliance concerns, which will help to evaluate the benefits and boundaries of current process execution languages used for compliance checking. Future work will be guided by the research question, how control patterns can be used to automate compliance controls. Remaining challenges, regarding the syntax of control patterns, deal with the accuracy versus complexity of applied control patterns and a standardized use of patterns and connectors that enable the implementation of compliance patterns by common process execution languages. To improve the approach further, we will evaluate the usability for the average user with basic IT knowledge and the process modeler with high IT affinity as well.

References

1. Dwyer, M. B., Avrunin, G. S., Corbett, J. C.: Patterns in property specifications for finite-state verification. In: IEEE International Conference on Software Engineering, pp. 411–420. IEEE Press, New York (1999)
2. Federal Ministry of Justice and Consumer Protection, Juris (Bundesministerium für Justiz und Verbraucherschutz – BMJ): http://www.gesetze-im-internet.de
3. Fellmann, M., Zasada, A.: State-of-the-Art of Business Process Compliance Approaches: A Survey, Proceedings of the 22nd European Conference on Information Systems (ECIS), Tel Aviv (2014)
4. Foodwatch: http://www.foodwatch.org/en/what-we-do/campaigns/foodwatch-campaigns
5. Hashmi, M., Governatori, G., Wynn, M.T.: Normative requirements for business process compliance. Service Research and Innovation, pp. 100–116. Springer, Berlin (2014)
6. Merriam-Webster, An Encyclopædia Britannica Company: Compliance, http://www.merriam-webster.com/dictionary/compliance
7. Merriam-Webster, An Encyclopædia Britannica Company: Regulation, http://www.merriam-webster.com/dictionary/regulation
8. Namiri, K. and Stojanovic, N.: Pattern-based design and validation of business process compliance. In: Proceedings of 6th On The Move Conference (OTM), Tari, Z. (ed.), LNCS 4083, pp. 59–76. Springer, Berlin, (2007)
9. Sadiq, S. and Governatori, G.: Managing Regulatory Compliance in Business Processes. In: Handbook on Business Process Management 2: Strategic Alignment, Governance, People and Culture, International Handbooks on Information Systems, vom Brocke, J., Rosemann, M. (eds.), vol. 2, pp. 265–288. Springer, Berlin (2015)
10. Shears, P.: Food Fraud: A Current Issue but an Old Problem. British Food Journal, vol. 112, no. 2, pp. 198–213 (2010)
11. SOX: Sarbanes-Oxley Act of 30 July 2002, 15 USC 7201 note, Public Law 107-204, 107th Congress, 116 Statistics Act, Sec. 404, pp. 745–810 (2002)
12. Tilburg University: State-of-the-Art for Compliance Languages: Compliance-driven Models, Languages, and Architectures for Services, Specific Targeted Research Project. Information Society Technologies (COMPAS Project no. 215175, D2.1), Netherlands (2008)
13. Turetken, O., Elgammal, A., Van den Heuvel, W.J., Papazoglou, M.P.: Capturing compliance requirements: A pattern-based approach. IEEE, vol. 29, no. 3, pp. 28–36. IEEE Press, New York (2012)
14. Weber, O., Diaz, M., Schwegler, R.: Corporate social responsibility of the financial sector – Strengths, weaknesses and the impact on sustainable development. Sustainable Development, vol. 22, no. 5, pp. 321–335 (2014)

Subgroup Discovery as a Method for Generating Test Ontologies

Daniel Knoell[1], Constantin Rieder[1], Martin Atzmueller[2], and Klaus Peter Scherer[1]

[1] Karlsruhe Institute of Technology
D-76344, Eggenstein-Leopoldshafen, Germany
firstname.lastname@kit.edu

[2] University of Kassel, Research Center for Information System Design
Knowledge and Data Engineering Group
Wilhelmshöher Allee 73, 34121 Kassel, Germany
atzmueller@cs.uni-kassel.de

Abstract. For the validation of ontologies, test cases can be applied. In order to ease the acquisition of such test cases, automatic methods can be used. This paper introduces such an approach to unit testing of ontologies using subgroup discovery methods. The tests are given by small ontologies, which will be compared to an existing ontology. The generation of the test ontologies uses subgroup discovery methods to derive concepts and relations from real data which refers to the ontology. For each concept of the ontology, we apply subgroup discovery in order to identify subgroup patterns consisting of concepts that are (un-)correlated with the target concept. Then, subgroups with very high (or low) confidence will be selected for the testing ontology. We exemplify the approach using real-world data from the ophthalmology domain, and present first results and experiences of a case study demonstrating the test case generation process.

1 Introduction

Not only ontology learning [9], but also the evaluation [15] of the learned ontologies is getting more and more important. In particular, this relates to the automatic evaluation of ontologies [17]. In this area, there exist different approaches like OntoClean[8] and AEON (Automatic Evaluation of Ontologies)[13]: OntoClean, for example aims for validating an ontology with the taxonomic relationships while AEON continues this approach with an automatically tagging of the OntoClean meta-properties.

One prominent method for the validation of knowledge systems considers the utilization of test cases, e. g., [14, 7]. Below, we sketch a method for ontology validation using automatically generated testing ontologies. For obtaining the appropriate test axioms automatically, we apply subgroup discovery – a versatile method for data analytics. Subgroup discovery [16, 3] aims at identifying interesting subsets of a dataset with

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

respect to a certain property of interest. Each subset that can be characterized by a certain descriptive *pattern* is then called a *subgroup*. For the construction of test cases, we generate patterns that are correlated with the target concept, or those that exclude the target concept.

Intuitively, we identify candidates for derivable axioms using patterns that have a high share of a certain target concept, i. e., the concepts making up the description of the pattern show a (strong) correlation with the target concept. Likewise, we obtain candidates for non-derivable axioms by constructing relations for which we are sure that they are not observed in the data. That is, for the validation of an ontology O, we apply two test ontologies as in [14]: T^+ and T^-, that specificy all axioms that should be derivable from the given ontology O, and the axioms that should not be derivable given O.

Below, we first introduce some background before we sketch our novel approach for the generation of automatic test cases. After that, we provide a case study using real-world data from the ophthalmology domain. The data is extracted from anonymized surgery sheets which are filled in by the eye specialist itself. These sheets include structured data like which eye was medicated or which medicine was used. We demonstrate the test case generation process, and discuss first experiences and implications.

2 Background

In this section, we briefly summarize necessary notation and basic concepts on subgroup discovery and the proposed approach for ontology validation using test ontologies.

2.1 Patterns and Subgroups

Formally, a *database* $DB = (I, A)$ is given by a set of individuals I and a set of attributes A. A *selector* or *basic pattern* $sel_{a_i = v_j}$ is a Boolean function $I \to \{0, 1\}$ that is true if the value of attribute $a_i \in A$ is equal to v_j for the respective individual. The set of all basic patterns is denoted by S. For a numeric attribute a_{num} selectors $sel_{a_{num} \in [min_j; max_j]}$ can be defined analogously for each interval $[min_j; max_j]$ in the domain of a_{num}. The Boolean function is then set to true if the value of attribute a_{num} is within the respective range.

Definition 1. *A* subgroup description *or (complex)* pattern *sd is given by a set of basic patterns $sd = \{sel_1, \ldots, sel_l\}$, where $sel_i \in S$, which is interpreted as a conjunction, i.e., $sd(I) = sel_1 \wedge \ldots \wedge sel_l$, with $length(sd) = l$.*

Without loss of generality, we focus on a conjunctive pattern language using nominal attribute–value pairs as defined above in this paper; internal disjunctions can also be generated by appropriate attribute–value construction methods, if necessary.

Definition 2. *A* subgroup (extension)

$$sg_{sd} := ext(sd) := \{i \in I | sd(i) = true\}$$

is the set of all individuals which are covered by the pattern sd.

As search space for subgroup discovery the set of all possible patterns 2^S is used, that is, all combinations of the basic patterns contained in S. For the practical implementation, we utilize the VIKAMINE platform [5] that provides efficient algorithms e. g., [10, 3] and according interestingness measure described below.

2.2 Interestingness of a Pattern

The interestingness of a pattern is determined by a quality function, that is selected according to the analysis task.

Definition 3. *A* quality function $q\colon 2^S \rightarrow \mathbb{R}$ *maps every pattern in the search space to a real number that reflects the interestingness of a pattern (or the extension of the pattern, respectively).*

While a large number of quality functions has been proposed in literature, many quality functions for a single target concept, e. g., in the binary or numerical case, trade-off the size $n = |ext(sd)|$ of a subgroup and the deviation $t_{sd} - t_0$, where t_{sd} is the average value of a given target concept in the subgroup identified by the pattern sd and t_0 the average value of the target concept in the general population. In the binary case, the averages relate to the *share* of the target concept. Then, this is equivalent to the concept of *confidence* commonly used in the field of association rule mining, e. g., [1]. In our context, we focus on binary target concepts only. Then, the subgroup patterns introduced above are equivalent to *class association rules* [11].

For subgroup discovery, typical quality functions are of the form

$$q_a(sd) = n^a \cdot (t_{sd} - t_0),\ a \in [0; 1]\,.$$

For binary target concepts, this includes, for example, the *weighted relative accuracy* for the size parameter $a = 1$ or a simplified binomial function, for $a = 0.5$. An extension to a target concept defined by a set of variables can be defined similarly, by extending common statistical tests.

The result of a subgroup discovery task is then the set of k (subgroup) patterns $sd_1, \ldots, sd_k$, where $sd_i \in 2^S$ with the highest interestingness according to the quality function. For our experiments in test ontology generation, we applied the simplified binomial function $q_{0.5}$ described above that conventiently balances the size of the subgroup with a high target share of the target concept.

2.3 Formalization of Test Ontologies

For the unit testing of ontologies, we adopt the approach proposed by Vrandecic and Gangemi [14]: We consider two testing ontologies. The ontology T^+ contains the axioms that should be derivable, and T^- contains the axioms that essentially should not be derivable. Thus, according to [14], for ontology O we have

$$O \models A_i^+ \, \forall A_i^+ \in T+\,,$$

for all axioms $A_i^+ \in T+, i = 1, \ldots, n$, and for all axioms $A_i^- \in T-, i = 1, \ldots, m$:

$$O \models A_i^- \, \forall A_i^- \in T-\ .$$

In this framework, the specific axioms themselves can be considered as atomic test cases for the ontology that need to be formalized accordingly. In the next section, we describe our approach for generating these axioms in an automatic fashion.

3 Method: Generating Test Ontologies

In the following, we first provide an overview on the proposed approach, before we discuss the test ontology generation in detail and present an algorithm for this task.

3.1 Overview

The basic idea of test ontology generation using subgroup discovery is the following: We start with all relevant test concepts contained in ontology O and map these to our test instances contained in the dataset. Then, for each target concept $t \in S$, we discover subgroups according to the total selector set S using a specific quality function (we used $q_{0.5}$ in our experiments). After that, we map these subgroup patterns to axioms that are integrated in the testing ontologies $T+$ and $T-$, respectively. For that step, we create complex classes for each selected subgroup pattern.

3.2 Generating Test Ontologies

For generating test ontologies from the subgroup discovery results, we map *(un-)correlated* patterns to the axioms of $T+$ and $T-$, respectively. Intuitively, when generating the according axioms using the discovered subgroup patterns, there must be a value which decides which subgroup is used and which not. There are several strategies for selecting the respective subgroup patterns used for creating the axioms. Below, we consider two variants: A confidence-based approach, and a test-coverage-based one.

1. A simple variant is a confidence-based approach. According to the share of the target concept in the subgroup, i. e., its *confidence*, we use the respective confidence of a subgroup to decide whether it should be used for the test ontology or not. If the confidence is higher than a specific threshold, it is used as for the T^+ ontology. If the confidence is below a specific other threshold, it is used for the T^- ontology. The threshold for T^- has to be very low or zero, because otherwise the test will also fail for the same dataset. Therefore it is also possible to define that not all of the test cases in the T^- ontology have to pass. It would be suitable if for example eight out of ten test cases pass to let the whole test pass.
2. An alternative approach is test-coverage-based and will be described using an example. For a specific target value two subgroups were detected. These subgroups only contain three different selectors. Subgroup one, for example, contains selector A and B and subgroup two contains selector A and C. In our example we add the two subgroups to the test ontology. In reality we would use a quality function to rate them and only add for example the five best subgroups to the test ontology. Back to our example with the two subgroups: If an ontology contains the target value and is connected with two out of three selectors it would be a coverage of $\frac{2}{3}$. With this approach we also need a threshold which decides if the test will pass or not. If this threshold would be $\frac{1}{3}$, our test in the example would have passed.

In our algorithm presented below, we use the described confidence-based approach, because it is easier to implement in this initial stage. In the future, we also plan a test-coverage-based approach to compare the results.

3.3 Algorithm

For the automatic generation of the described test ontologies we developed the algorithm shown in Listing 1.1. It uses subgroup discovery to derive interesting correlations and transforms them to small test ontologies consisting of complex classes.

Listing 1.1: Algorithm in pseudo code

```
for each attribute{
  subgroupset = findSubgroups(attribute)
  candidatesTplus = [], candidatesTminus = []
  for each subgroup in subgroupset{
    if subgroup.confidence >= thresholdPlus
      candidatesTplus.add(subgroup)
    if subgroup.confidence <= thresholdMinus
      candidatesTminus.add(subgroup)
  }
  makeComplexClasses(candidatesTplus, candidatesTminus)
}
```

In the first step, subgroup discovery is performed for each attribute of the dataset. Therefore, the current attribute $t \in S$ is used as target value. For each target variable a set of subgroups is calculated. Each set contains all subgroups with the appropriate target variable.

Each subgroup in the set that exceeds a certain confidence threshold is added as a test candidate to the positive test candidate set (candidatesTplus). If the confidence is under a specific different threshold. In this case the subgroup is added to the negative candidate set (candidatesTminus). Thus each attribute has its own two test candidate sets, which contain all test cases for the specific attribute. A test candidate consists of any number of selectors that are associated by the AND function. This combination of selectors is characteristic for the target variable related to a high confidence and therefore will be added to the positive test candidate set. If we consider the exemplary “vehicle”-domain, for example, then the target variable “vehicle type = tricycle” has the significant describing combination of the selectors “number of wheels = 3” and “engine = false”.

For the generation, we use the confidence as a threshold. From the test candidate set a complex class is created for each attribute. This set is composed of the individual test candidates concatenated by the OR function. Thus generated complex classes will be compared or tested against an existing ontology in the next step.

4 Case Study

This section demonstrates the proposed approach using real-world data and presents first results. It covers the ontology structure and the working on the generation of T^+ ontologies.

4.1 Overview

In order to verify our approach we used anonymised operative reports from the cataract surgery domain to set up a dataset for testing. We have extracted the structured data from the data sheets. All records were reduced, encoded, typed and converted to a suitable format. Finally, after the data wrangling the latest version of the test dataset for this work contained more than 500 instances described by 80 attributes. As output format “ARFF”(Attribute-Relation File Format) was used, which is an input format for data mining tools like WEKA or VIKAMINE. For subgroup discovery, the VIKAMINE platform was used, because WEKA does not support subgroup discovery out of the box. The VIKAMINE system offers a rich-client environment for subgroup discovery and analytics [5]. We iterated over every attribute and performed a subgroup discovery with VIKAMINE for each attribute as target value. With the results of the subgroup discovery we create test ontologies for each attribute using the confidence based approach. These test ontologies are written in a KnowWE(Knowledge Wiki Environment) Wiki. KnowWE has export functions for the ontology which supports various popular formats like Turtle, RDF/XML or RDF/JSON, cf. [2].

4.2 Structure of the basic ontology

The output ontology follows a specific order. A cataract ontology[12] was designed and the structure of the attribute is displayed in Figure 1. Each real attribute is a subclass of the “Attribute”-class and each class has its own wiki page.

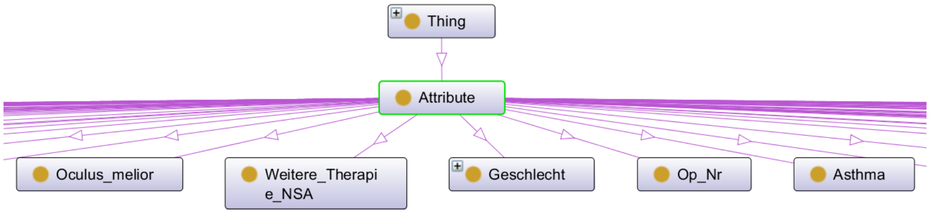
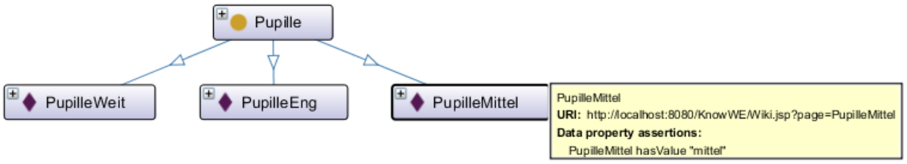


Fig. 1: Overview of the attribute classes of the ontology

The attribute value is stored in the instances of the specific attribute. A visualisation of the the instances of the attribute “Pupille” (German for pupil) is shown in Figure 2. This attribute has three possible values: “weit”, “eng” and “mittel”. For each value there is a corresponding instance of the attribute with the matching data property. For the data property the relation “hasValue” is used. As an example the instance “PupilleMittel” of type “Pupille” has a data property “hasValue” with the value “mittel”.



PupilleMittel
URI: http://localhost:8080/KnowWE/Wiki.jsp?page=PupilleMittel
Data property assertions:
PupilleMittel hasValue "mittel"

Fig. 2: Pupil class with the corresponding individuals

4.3 Automatic generation of the test ontologies

Below, we describe first exemplary results of our proof-of-concept implementation of the proposed approach. In the basic setting for subgroup discovery we applied the BSD algorithm [10] with the quality function $q_{0.5}$ and a minimum quality limit of 0.8.

After performing subgroup discovery and constraining the subgroup description length to one concept for each attribute of our dataset we pick up exemplary the target value “Patient_unruhig = true” which means that the patient is nervous and in a state of agitation during the surgery. One of the subgroups from the derived amount of subgroups which fulfill the quality limit in this case is “Sedativum = true”. That is no surprise because one of possible logical consequences for the operating surgeon could be to sedate the patient. The resulting subgroup from the corresponding dataset confirms this practice. The identified pattern can now be used to generate a test case for the cataract ontology because we know there is a correlation with a certain quality limit between the above mentioned items. Incrementing the description length to two concepts yields further patterns, for example, “Diabetes = true” is described among other descriptions by the combination of concepts “Infusion_Lidocain = false” and “Gerinnungshemmer = true”. In this way further test cases can be created to extend our test ontology by mapping the identified patterns to a complex class.

Our first results and experiences in the case study demonstrate, that the proposed approach is working well. It is a very transparent process for test ontology generation since the subgroup patterns can be directly inspected, cf. [6], and transparently mapped to complex classes in the ontology. Currently, we are still refining the results with respect to the testing ontologies $T+$ and $T-$ and more complex test-coverage strategies.

5 Conclusions

In this paper, we have presented a novel approach for the automatic generation of test ontologies. Using subgroup discovery, we identify patterns that are mapped to axioms in our testing ontology. In particular, these are formalized using complex classes which can then be directly implemented in unit testing manner. We demonstrated the basic concepts in a case study using a real-world dataset. Currently, our proof-of-concept implementation focuses on a confidence-based strategy for test ontology generation. We aim to investigate the approach further for generating more complex testing ontologies ($T+$ and $T-$) and to apply a test-coverage-based approach in order to have fine-grained control when a testing ontology passes or fails. Furthermore, the presented basic approach can then also be extended using semi-automatic strategies, e. g., cf. [4].

Acknowledgements

The research project is a cooperation of the Karlsruhe Institute of Technology, Germany (KIT) and the denkbares GmbH. It is funded as a ZIM-KOOP by the German Federal Ministry of Economics and Technology (BMWI). The authors also thank the project executing organization AiF in Berlin, which is responsible for the allocation of the budget to the research center and the commercial company.

References

1. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules. In: Bocca, J.B., Jarke, M., Zaniolo, C. (eds.) Proc. VLDB. pp. 487–499. Morgan Kaufmann (1994)
2. Allemang, D., Hendler, J.A.: Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL. Morgan Kaufmann, Waltham Mass., 2. ed edn. (2011)
3. Atzmueller, M.: Subgroup Discovery – Advanced Review. WIREs: Data Mining and Knowledge Discovery 5(1), 35–49 (2015)
4. Atzmueller, M., Baumeister, J., Hemsing, A., Richter, E.J., Puppe, F.: Subgroup Mining for Interactive Knowledge Refinement. In: Proc. Conf. on Artificial Intelligence in Medicine (AIME). pp. 453–462. LNAI 3581, Springer, Heidelberg (2005)
5. Atzmueller, M., Lemmerich, F.: VIKAMINE - Open-Source Subgroup Discovery, Pattern Mining, and Analytics. In: Proc. ECML/PKDD. LNCS, vol. 7524, pp. 842–845. Springer, Heidelberg (2012)
6. Atzmueller, M., Puppe, F.: A Case-Based Approach for Characterization and Analysis of Subgroup Patterns. Journal of Applied Intelligence 28(3), 210–221 (2008)
7. Baumeister, J.: Advanced Empirical Testing. Knowl Based Syst 24(1), 83–94 (2011)
8. Guarino, N., Welty, C.: An Overview of OntoClean. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies, pp. 151–171. Springer Berlin Heidelberg (2004)
9. Lehmann, J., Völker, J.: Perspectives on Ontology Learning, Studies on the Semantic Web, vol. Volume 018. IOS Press / AKA (2014)
10. Lemmerich, F., Rohlfs, M., Atzmueller, M.: Fast Discovery of Relevant Subgroup Patterns. In: Proc. FLAIRS. pp. 428–433. AAAI Press, Palo Alto, CA, USA (2010)
11. Liu, B., Hsu, W., Ma, Y.: Integrating Classification and Association Rule Mining. In: Proc. KDD. pp. 80–86. AAAI Press (August 1998)
12. Scherer, K.P., Rieder, C., Henninger, C., Germann, M., Baumeister, J., Reutelshöfer, J.: Modeling and Visualization of Cataract Ontologies. In: Proc. ADVCOMP (2014)
13. Völker, J., Vrandečić, D., Sure, Y.: Automatic Evaluation of Ontologies (AEON). In: Gil, Y., Motta, E., Benjamins, V., Musen, M. (eds.) The Semantic Web – ISWC 2005, Lecture Notes in Computer Science, vol. 3729, pp. 716–731. Springer Berlin Heidelberg (2005)
14. Vrandecic, D., Gangemi, A.: Unit Tests for Ontologies. In: Proc. 1st Intl. Workshop on Ontology Content and Evaluation in Enterprise. LNCS, Springer, Montpellier, France (2006)
15. Vrandečić, D.: Ontology Evaluation. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies, pp. 293–313. Springer Berlin Heidelberg (2009)
16. Wrobel, S.: An Algorithm for Multi-Relational Discovery of Subgroups. In: Proc. PKDD-97. pp. 78–87. Springer, Heidelberg, Germany (1997)
17. Zablith, F., Antoniou, G., d'Aquin, M., Flouris, G., Kondylakis, H., Motta, E., Plexousakis, D., Sabou, M.: Ontology Evolution: A Process-Centric Survey. KER 30, 45–75 (1 2015)

POQL: A New Query Language for Process-Oriented Case-Based Reasoning

Gilbert Müller and Ralph Bergmann

Business Information Systems II
University of Trier
54286 Trier, Germany
[muellerg][bergmann]@uni-trier.de,
http://www.wi2.uni-trier.de

Abstract. Sharing and reuse of best-practice process models is an important knowledge management approach for business process modelling. Process-oriented Case-Based Reasoning (PO-CBR) supports this by retrieving and adapting processes or workflows based on models stored in the repository, which requires an expressive query language. Hence, we present a novel query language for workflows that enables to express generalized query terms and negation. Further, it allows a ranking of the repository workflows.

Keywords: Process-oriented Case-based Reasoning, Business Process Querying, Workflows

1 Introduction

Nowadays, business processes have to be organized highly flexible due to increasing globalization and competitive pressure. Thus, business processes and workflows implementing them have to be promptly defined or adapted to new circumstances. For this purpose, sharing and reuse of existing best-practice process models is an important approach that introduces knowledge management concepts into business process modelling [11]. Thus, important knowledge management tools are searchable repositories of process models that enable the retrieval of reusable processes and may in addition propose ways of reusing them. Process-oriented Case-based Reasoning (POCBR) [18] is a research area that deals with applying case-based reasoning (CBR) to experiential knowledge represented in process models and workflows and thus provides the foundations for building knowledge management tools supporting process modelling. Current research in POCBR addresses the similarity-based retrieval and adaptation of process and workflow models. However, the formulation of queries for the purpose of modelling and adaptation support has not been discussed extensively.

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

Current POCBR research usually assumes that a query just consists of a partial workflow/process currently being designed. Then retrieval searches for similar workflows that mostly match the current query. Thus, the found workflows can be considered a source of the auto-completion of the current partial workflow, which can be supported by case-based adaptation methods. However, for appropriate retrieval and adaptation, a query language is needed that is able to capture as best as possible all current requirements on the workflow/process to be created. In this paper, we address this issue by proposing a new, more expressive approach for the formulation of queries in POCBR and we sketch a way of tweaking existing retrieval methods to deal with this language.

2 Foundations

We build this research upon our previous work in POCBR which addresses the similarity-based retrieval and adaptation of semantic workflows [7,19,20]. The methods developed so far are implemented in the prototypical software system CAKE [6] and analyzed in various application domains. In this paper, we illustrate our approach in the domain of cooking recipes. A cooking recipe is represented as a workflow describing the instructions for cooking a particular dish. We now briefly outline previous relevant work as the main foundation of this paper.

Broadly speaking, workflows consist of a set of activities (also called tasks) combined with control-flow structures like sequences, parallel (AND) or alternative (XOR) branches, as well as repeated execution (LOOPs). Tasks and control-flow structures form the control-flow. In addition, tasks exchange certain data items, which can also be of physical matter, depending on the workflow domain. Tasks, data items, and relationships between them form the data flow. In our work we extend this traditional view of workflows by adding semantic annotations to (potentially) all workflow items as a means to support case-based reasoning. In formal terms, a semantic workflow is defined as a directed graph $W = (N, E, S, T)$ where N is a set of nodes and $E \subseteq N \times N$ is a set of edges. Nodes and edges have types assigned by the function T, which partitions the nodes N of a workflow into a single workflow node N^W and several data nodes N^D, task nodes N^T, and control-flow nodes N^C. Likewise we distinguish data-flow edges E^D, control-flow edges E^C and part of edges N^P. Further, nodes have a semantic description from a semantic meta data language Σ, which is assigned by the function $S : N \to \Sigma$.

2.1 Semantic Workflows

Figure 1 shows a simple fragment of a workflow graph from the cooking domain. Here, the tasks represent the cooking steps and the data items refer to the ingredients being processed by the cooking steps. The main source of knowledge in POCBR is a repository of semantic workflows (in CBR terminology called the case base) available for reuse. In order to obtain a reusable workflow similarity

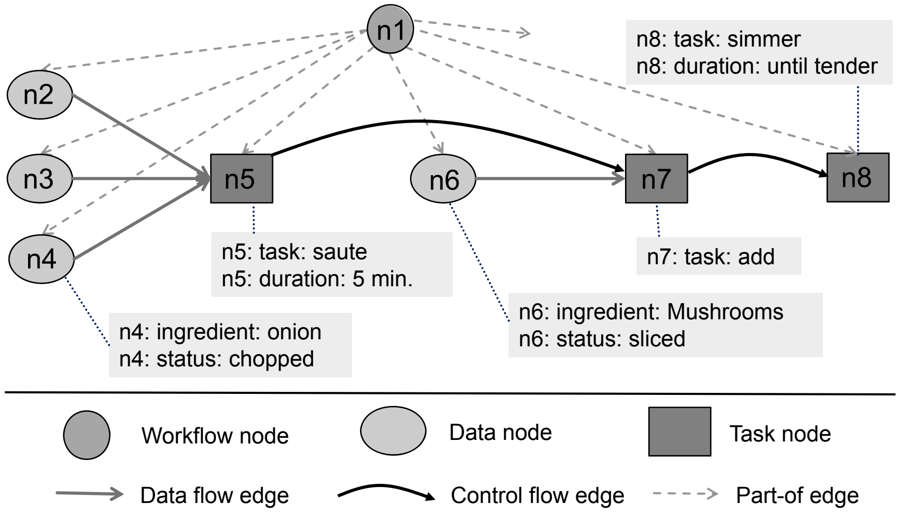


Fig. 1. An example workflow graph

search or process model querying [11,21] can be applied. According to Dijkman et al. [11] "the main difference between querying and similarity search is that querying searches for exact matches of a query to a part of a process model, while similarity searches for inexact matches of the query to a complete process model". We focus on similarity search as it is able to provide results even if exact matches are not available, which is very likely in many application scenarios. Various approaches for similarity search of workflows have been proposed in the literature, such as graph edit metrics, graph/subgraph isomorphism, most common subgraph approaches [3,7,10,14,15]. In our research [7], we developed an approach that follows the tradition of CBR and uses explicitly modelled local similarity measures for task and data items, based on task and data ontologies, which are also used for the semantic annotation of the workflows. The overall similarity $sim(QW, CW) \in [0, 1]$ between a query workflow QW and a case workflow CW from the repository is defined as an optimization problem aiming at finding the best possible type-preserving, partial, injective mapping of the nodes and edges of QW to those of CW. The optimization target is the average similarity of the mapped nodes and edges. This similarity measure assesses how well the query workflow is covered by the case workflow. In particular, the similarity is 1 if the query workflow is exactly included in the case workflow as a subgraph.

3 Query Language Requirements

Previous work on POCBR (including our own) is limited by the type of queries that can be considered. As described in the previous section, a query is a single (partial) workflow that describes task and data items and structural relationships of the desired workflow. This can be roughly considered a conjunctive query, as the ideal workflow contains the whole query workflow, i.e., all components it contains. Thus, disjunction and negation cannot be expressed. However, the user may also want to express undesired workflow elements and structures. For example, some tasks or data elements must not occur in the workflow or a certain sequence of activities is undesired. Also, disjunction/generalization is sometimes required, e.g. by providing more general conditions, such as specifying a class of tasks (from the ontology) or by specifying that a certain task must occur some time (but not necessarily directly) before another one. More expressive queries are not only desirable for retrieving more suitable workflows but are also essential to guide automatic adaptation methods from POCBR as they can provide hints concerning which workflow elements need to be added, deleted, or moved to a different position. Besides these usage scenarios, literature also lists a wide range of additional purposes for queries [17,1], e.g., dependency analysis between workflow elements [9] and decision making support [12]. However, in the following we focus our investigations on retrieval and adaptation support. Based on this, we derive the following main requirements for a new query language, which have also been partially mentioned in the literature [16,2]:

- **Expressiveness:** The query language must be expressive enough to be able to represent the relevant requirements of the user. Thus, the query language should not only be able to represent what the user desires, it should be also able to handle undesired workflow elements, which requires a kind of negation. Additionally, generalization of structure and items is required.
- **Intuitiveness:** The query language should be easy to understand. Thus, new notations should be only introduced if required and it should be based on the already known concepts. Additionally, a visual query language is preferable as workflows can become very complex and thus also its queries.
- **Ranking:** For the specified query language it must be able to identify all matching workflows. Moreover, as fully matching workflows are very unlikely in many application scenarios, it must be possible to rank the workflows w.r.t. suitability for the query, if they don't match perfectly.

Thus, the similarity-based retrieval must be extended towards a retrieval considering suitability w.r.t. a more expressive query.

4 POQL: A New Query Language for Workflows

POQL is a workflow query language developed according to the previously mentioned requirements. It extends the purely conjunctive query approach by introducing negation and generalized query workflows. Formally, a POQL query Q is defined as follows: $Q = DW \wedge \neg RW_1 \dots RW_n$. Here, DW is the desired workflow representing properties that the searched workflow should fulfill, RW_i are restriction workflows, each of which represents one undesired situation that should be avoided. The desired workflow and the restriction workflows are so-called generalized query workflows. They are in principle workflows in the previously introduced sense, but they are extended to represent generalizations by two means. As a consequence, generalized query workflows are not executable workflows anymore, but they are just a means to express a query in a way similar to a workflow. Thereby, the intuitiveness requirement can be fulfilled as building a query is very similar to building a workflow.

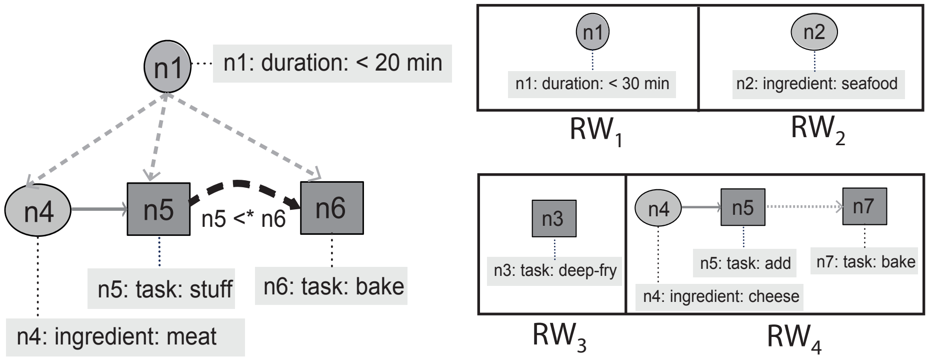
Generalized Task and Data Labels: While workflows from the repository usually contain ontology instances for tasks and data items specifying the flow of activities in executable terms, a generalized workflow [20] is allowed to contain concept labels of the data- and task ontology, thus specifying classes of them. For example, a recipe workflow may specify that meat is desirable without specifying in detail which meat should be used. For the representation of generalized task and data labels, the meta-data language used for semantic annotation must also include the ontology concepts.

Transitive Control-Flow and Dataflow Connectedness: While workflows exactly specify the control- and dataflow of a workflow, a generalized query workflow may just specify that a certain tasks must be executed some time before another task or that one task produces data that at some later point is used by another tasks. In formal terms these relations are defined as follows.

1. Let $t_1 < t_2$ define that there exists a control-flow edge between $t_1 \in N^T$ and $t_2 \in N^T$ defining that t_1 is executed before t_2. We define that two tasks $t_1, t_2 \in N^T$ are transitively control-flow connected $t_1 <^* t_2$ iff $t_1 < t_2 \vee \exists t \in N^T : t_1 < t < *t_2$. We represent $t_1 < *t_2$ in a generalized query graph by introducing a new type of edge leading from t_1 to t_2.
2. Let $t_1 \ltimes t_2$ denote that the tasks $t_1, t_2 \in N^T$ are data-flow connected, i.e., it holds $t_1 \ltimes t_2$ iff $t_1 < t_2 \vee \exists d \in N^D : ((t_1, d) \in E^D \wedge (d, t_2) \in E^D)$. Based on this we define that two tasks $t_1, t_2 \in N^T$ are transitively data-flow connected $t_1 \ltimes^* t_2, if t_1 \ltimes t_2 \vee \exists t \in N^T : t_1 \ltimes t \ltimes^* t_2$. To represent transitive data-flow connectedness, a new type of edge is introduced. For example, this edge can be used to express in the query that after a specific preparation step using an ingredient another preparation step follows that uses the same ingredient.

Figure 2(a) shows an example of a generalized query workflow. This query specifies that a workflow is desired that requires less than 20 minutes of preparation time, which contains some kind of meat (generalized data label) , and the preparation steps stuff and bake. Furthermore, it is specified that the tasks for stuffing the meat must occur before the baking tasks (transitive control-flow connected).

The restriction workflows can be constructed in a similar manner as illustrated in figure 2(b). The four restrictions shown specify that the searched workflow should require less than 30 minutes of preparation time (RW_1), that it should not contain any seafood (RW_2) or deep-fry preparation steps (RW_3), as a frying machine is maybe not available. Furthermore, it is required that no



(a) Query Workflow Example

(b) Restriction Workflow Examples

Fig. 2. POQL Query Example

cheese is added to a dish component which is later baked (RW_4), thus casserole recipes are undesired.

Please note that in principle it would be possible to allow more than one desired workflow in a query. However, this is not necessary as it would not extend the expressiveness of the language, as several desired workflows could be easily merged into a single desired workflow representing the same query semantics.

The processing of a POQL query requires ranking all workflow from the repository and presenting the best ranked results. For a query which does not include any restriction workflows, our approach for workflow similarity can be extended in a straight forward manner. Generalized task and data labels are addressed by applying taxonomic similarity measures, which are well-established in CBR [5].

To consider transitive control-flow and dataflow connectedness, the workflow graph of all workflows in the repository is extended to represent all relations $t_1 < *t_2$ and $t_1 \ltimes^* t_2$ which must be pre-computed during the initialization of the repository. Given this, these relations can be matched in the same manner as all other edges of the graph. However, this approach obviously increases the complexity of the representation and thus the complexity of the similarity assessment, which is an issue of our future research.

To handle restriction workflows in POQL requires dealing with negation and conjunction. We propose an approach adopted from fuzzy logic [8] by treating the similarity value computed by comparing a case workflow with a generalized query workflow as a fuzzy membership value. Then we can apply fuzzy negations and a t-norm to compute the conjunction. In particular, we compute the rank of a workflow as follows:

$$rank(Q,CW) = min\{sim(DW,CW), 1-sim(RW_1,CW), \ldots, 1-sim(RW_n,CW)\} \tag{1}$$

The resulting $rank(Q, CW) \in [0, 1]$ reflects how well the workflow matches the query, while the following conditions hold:

1. $Rank = 1$ if the case workflow exactly matches the desired workflow and does not contain any subworkflow which is somehow similar to any of the restriction workflows.
2. $Rank = 0$ if the case workflow contains a subworkflow which exactly matches one of the restriction workflows.
3. $Rank \in]0, 1[$ if the case workflow partially matches the desired workflow and if all restriction workflows are at most matching partially (not exactly).

5 Conclusions

We presented the novel query language POQL for POCBR which is highly intuitive and enables not just the retrieval but also the adaptation of workflows. A POQL query can be easily constructed using a graphical workflow editor by introducing additional link types among tasks as well as ontological concepts for semantic annotation. We presented an approach that allows an ordering of the best workflows from the repository by a ranking approach.

In process model querying, BPMN-Q [1,21] is a related approach applied to BPMN business processes. The approach is able to identify processes that match the partial modelled workflow. However, the approach is so far neither able to consider the dataflow nor undesired data or tasks. Furthermore, there is no ranking between the processes found. Awad et al. [2] extend BPMN-Q by regarding semantics between workflow elements. The related approaches presented by Beeri et al. [4] or by Markovic et al. [16,17] are not able to support negations or to rank the results by similarity which both is required for the modelling and adaptation support of workflows. Recently, PQL [13] has been presented, which also addresses the querying and changing of process models. However, in contrast to our work where required changes are implicitly derived from the query, PQL required to define those changes explicitly in a SQL-like statement.

Future work will extend the query language to support other usage scenarios (see section 3), i.e., scenarios in which only fragments of workflows are searched rather than complete workflows. Additionally, an evaluation of the presented POQL will be undertaken.

Acknowledgements. This work was funded by the German Research Foundation (DFG), project number BE 1373/3-1.

References

1. Awad, A.: BPMN-Q: A language to query business processes. In: EMISA. vol. 119, pp. 115–128 (2007)

2. Awad, A., Polyvyanyy, A., Weske, M.: Semantic querying of business process models. In: Enterprise Distributed Object Computing Conference, 2008. EDOC'08. 12th International IEEE. pp. 85–94. IEEE (2008)
3. Bae, J., Liu, L., Caverlee, J., Rouse, W.B.: Process mining, discovery, and integration using distance measures. In: Web Services, 2006. ICWS'06. International Conference on. pp. 479–488. IEEE (2006)
4. Beeri, C., Eyal, A., Kamenkovich, S., Milo, T.: Querying business processes. In: Proceedings of the 32nd international conference on Very large data bases. pp. 343–354. VLDB Endowment (2006)
5. Bergmann, R.: Experience management: foundations, development methodology, and internet-based applications. Springer-Verlag (2002)
6. Bergmann, R., Gessinger, S., Görg, S., Müller, G.: The collaborative agile knowledge engine cake. In: Proceedings of the 18th International Conference on Supporting Group Work. pp. 281–284. GROUP '14, ACM, New York, NY, USA (2014)
7. Bergmann, R., Gil, Y.: Similarity assessment and efficient retrieval of semantic workflows. Inf. Syst. 40, 115–127 (Mar 2014)
8. Burkhard, H.D., Richter, M.M.: On the notion of similarity in case based reasoning and fuzzy theory. In: Soft computing in case based reasoning, pp. 29–45. Springer (2001)
9. Dai, W., of Waterloo. School of Computer Science, U.: A Query-based Approach to Workflow Process Dependency Analysis. University of Waterloo (2005)
10. Dijkman, R., Dumas, M., García-Bañuelos, L.: Graph matching algorithms for business process model similarity search. In: Business process management, pp. 48–63. Springer (2009)
11. Dijkman, R.M., La Rosa, M., Reijers, H.A.: Managing large collections of business process models-current techniques and challenges. Computers in Industry 63(2), 91–97 (2012)
12. Hepp, M., Leymann, F., Domingue, J., Wahler, A., Fensel, D.: Semantic business process management: A vision towards using semantic web services for business process management. In: e-Business Engineering, 2005. ICEBE 2005. IEEE International Conference on. pp. 535–540. IEEE (2005)
13. Kammerer, K., Kolb, J., Reichert, M.: PQL - A Descriptive Language for Querying, Abstracting and Changing Process Models. In: BPMDS'15. pp. 135–150. No. 214 in LNBIP, Springer (June 2015)
14. Kapetanakis, S., Petridis, M., Knight, B., Ma, J., Bacon, L.: A case based reasoning approach for the monitoring of business workflows. In: Case-Based Reasoning. Research and Development, pp. 390–405. Springer (2010)
15. Ma, Y., Zhang, X., Lu, K.: A graph distance based metric for data oriented workflow retrieval with variable time constraints. Expert Systems with Applications 41(4, Part 1), 1377 – 1388 (2014)
16. Markovic, I.: Advanced querying and reasoning on business process models. In: Abramowicz, W., Fensel, D. (eds.) Business Information Systems, Lecture Notes in Business Information Processing, vol. 7, pp. 189–200. Springer Berlin Heidelberg (2008)
17. Markovic, I., Costa Pereira, A., de Francisco, D., Muoz, H.: Querying in business process modeling. In: Di Nitto, E., Ripeanu, M. (eds.) Service-Oriented Computing - ICSOC 2007 Workshops, Lecture Notes in Computer Science, vol. 4907, pp. 234–245. Springer Berlin Heidelberg (2009)
18. Minor, M., Montani, S., Recio-Garca, J.A.: Process-oriented case-based reasoning. Information Systems 40(0), 103 – 105 (2014)

19. Müller, G., Bergmann, R.: Workflow streams: A means for compositional adaptation in process-oriented cbr. In: Case-Based Reasoning Research and Development, pp. 315–329. Springer (2014)
20. Müller, G., Bergmann, R.: Generalization of Workflows in Process-Oriented Case-Based Reasoning. In: 28th International FLAIRS Conference. AAAI, Hollywood (Florida), USA (2015)
21. Sakr, S., Awad, A., Kunze, M.: Querying process models repositories by aggregated graph search. In: Business Process Management Workshops. pp. 573–585. Springer (2013)

Dependencies between knowledge for the Case Factory maintenance approach

Pascal Reuss and Klaus-Dieter Althoff
pascal.reuss@dfki.de
klaus-dieter.althoff@dfki.de

Intelligent Information Systems Lab, University of Hildesheim
Competence Center Case Based Reasoning, German Center for Artificial Intelligence, Kaiserslautern

Abstract. In many knowledge-based systems the used knowledge is distributed among several knowledge sources. These knowledge sources may have dependencies between each other, which should be considered when maintaining these sources. An integrated maintenance approach for multiple Case-Based Reasoning (CBR) systems has to consider dependencies between the individual knowledge containers within one CBR system and the dependencies between the knowledge containers of different CBR systems. This paper describes the dependencies between knowledge containers in CBR systems from the perspective of the Case Factory approach and how possible maintenance actions could be derived from these dependencies.

1 Introduction

Today knowledge based systems handling a huge amount of knowledge to provide solutions to given problems. This knowledge is often distributed over several internal or external knowledge sources. These knowledge sources may be independent from each other, but they also may have dependencies between each other. In many systems the knowledge is distributed among sub-domains. For example a travel medicine application may have knowledge divided into knowledge about regions, hospitals and medication. Between the regions and the hospitals existing dependencies, because a hospital is linked to a specific region. If the spelling of the region is changed or the region is deleted, the corresponding hospital can not be found any more and there will be inconsistent knowledge. In the following we assume that all knowledge sources in a knowledge based system are CBR systems. When maintaining an application with several different CBR systems as knowledge sources, it is important to consider the dependencies between the knowledge inside these CBR systems. These dependencies could be between knowledge containers inside a single CBR system and between knowledge containers of different CBR systems. Current maintenance approaches for CBR systems focus on one single CBR system or a single knowledge container and considering

Copyright © 2015 by the papers authors. Copying permitted only for private and academic purposes. In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

only dependencies inside a single CBR system. The extended Case Factory approach [10] considers the dependencies between knowledge containers. In this paper we describe the dependencies that could exist between knowledge containers from a Case Factory perspective and how these dependencies could be processed to derive possible maintenance actions. In Section 2 we give an overview of related work to this topis, while Section 3 describes briefly the Case Factory approach and the dependencies between knowledge containers in more detail. In addition, we describe the modeling of dependencies with the help of a Maintenance Map and an algorithm to identify and process dependencies and derive possible maintenance actions. Section 4 gives a short conclusion and an outlook to future work.

1.1 SEASALT architecture

The SEASALT (Shared Experience using an Agent-based System Architecture Layout) architecture is a domain-independent architecture for extracting, analyzing, sharing, and providing experiences [2]. The architecture is based on the Collaborative Multi-Expert-System approach [1] and combines several software engineering and artificial intelligence technologies to identify relevant information, process the experience and provide them via an interface. The SEASALT architecture consists of five components: the knowledge sources, the knowledge formalization, the knowledge provision, the knowledge representation, and the individualized knowledge. The knowledge sources component is responsible for extracting knowledge from external knowledge sources like databases or web pages and especially Web 2.0 platforms. The knowledge formalization component is responsible for formalizing the extracted knowledge from the Collector Agents into a modular, structural representation. The knowledge provision component contains the so called Knowledge Line. The basic idea is a modularization of knowledge analogous to the modularization of software in product lines. The modularization is done among the individual topics that are represented within the knowledge domain. The Topic Agents can be any kind of information system or service. If a Topic Agent has a CBR system as knowledge source, the SEASALT architecture provides a Case Factory for the individual case maintenance [2]. The knowledge representation component contains the underlying knowledge models of the different agents and knowledge sources. The synchronization and matching of the individualized knowledge models improves the knowledge maintenance and the interoperability between the components. The individualized knowledge component contains the web-based user interfaces to enter a query and present the solution to the user.

2 Related work

The DILLEBIS methodology from Markus Nick [8] focuses on identifying necessary maintenance actions using user feedback. He considers dependencies between knowledge sources only implicitly. A dependency can be assumed, if a user advises to change more than one knowledge source in his feedback. A knowledge engineer has to confirm a dependency manually. In our approach we define the dependencies explicitly and process them automatically to give the knowledge engineer a list of possible maintenance actions. The SIAM methodology from Thomas Roth-Berghofer [11] focuses on maintenance for CBR systems and extends the CBR cycle with to additional steps for evaluation and maintenance of a single CBR system. Dependencies are considered only implicit in this methodology, too. The evaluation of the knowledge containers can show dependencies, if more than one knowledge container requires maintenance in a specific situation. But the confirmation of dependencies had to be done manually before a maintenance action can be performed. There are many maintenance approaches for CBR systems like [5], [6],[12], and [13] that presents strategies to maintain the case base or the similarity measures. But all of these approaches are only considering one knowledge container or a single CBR system, while we will consider dependencies between all knowledge containers of a single CBR system and dependencies between knowledge containers of different CBR systems. Leake and his co-authors worked with different multiple knowledge sources for CBR systems and the combination of maintenance actions to preserve the competency and efficiency of a CBR system [7]. Their approach is focused on a single CBR system, but the idea is also applicable for multiple CBR systems and may be combined with our approach.

3 Dependencies between knowledge containers in CBR systems

In a multi-agent system like *docQuery*, the knowledge is distributed over several knowledge sources. Each knowledge source is a software agent with an underlying CBR system, representing the knowledge of a sub-domain of the travel medicine domain. For example one CBR system contains knowledge about regions, anothr CBR system contains knowledge about medication. In the *docQuery* system exist seven different CBR systems for knowledge about regions, hospitals, medication, infectious diseases, chronicle diseases, activities and conditions (climate, security, etc) [9]. Between these CBR systems dependencies can be found, either because two CBR systems share the same vocabulary or cases are linked to each other. For example, the CBR systems for regions and infectious diseases have partially the same vocabulary and there are links between case from the region case base and the infectious disease case base. These dependencies have to be considered, when thinking about maintaining these CBR systems. The extended Case Factory approach [10] for maintaining CBR systems is able to consider these dependencies. A Case Factory is part of the *knowledge provision* component of the SEASALT architecture and is responsible for maintaining a single CBR system. Several software agents are monitoring the knowledge containers and propose possible maintenance actions, if defined conditions are met. Based on monitoring results and defined dependencies additional possible maintenance actions may be derived. A Case Factory can process the dependencies inside a single CBR system. Following our approach, each of the seven CBR systems has its own Case Factory to monitor and maintain the knowledge. To process dependencies between CBR systems, a so-called Case Factory Organization (CFO) is used. This high-level layer manages all Case Factories, the dependencies between knowledge containers of different CBR systems, and coordinates the maintenance process. There can be more than one CFO to manage the maintenance on different levels. A CFO can be used to split a system with multiple CBR system into several organizational units. For example in the *docQuery* application it would be possible to have 4 CFOs. One CFO contains the region and hospital CBR systems, the second CFO the infectious diseases, chronicle diseases and medication CBR systems and the third CF contains the activities and conditions CBR systems. Each of this CFOs manage the dependencies between the corresponding CBR systems. The fourth CFO manages the dependencies between CBR system of different CFOs and can also be used to manage the overall maintenance process to identify maintenance actions that have to be processed in combination with other maintenance actions to address problems as stated in [7]. In the following section, the dependencies between knowledge containers from our Case Factory perspective are described in more detail.

3.1 Intra- and inter-system dependencies

A dependency exists between different knowledge containers. We define a dependency d as

$$d = (kc_{sysS}, kc_{sysT}, t)$$

$$\begin{aligned} &\text{where } kc \in \{voc, sim, cb, ada\} \\ &\text{and } sysS, sysT \in \{1 \ldots n\} \\ &\text{and } t \in \{u, b\} \end{aligned}$$

A dependency can be described as a triple of two knowledge containers (*kc*) and the direction (*t*) of the dependency. The knowledge containers are the vocabulary (*voc*), the similarity measures (*sim*), the case base (*cb*), and the adaptation knowledge (*ada*). We assume there are 1 to n CBR systems. The indexes *sysS* and *sysT* identify the CBR systems a knowledge container belongs to, where *sysS* is the source of a dependency and *sysT* the target. The last element of the triple determines the direction of a dependency, either uni-directional (*u*) or bi-directional (*b*). A uni-directional dependency is only processed from the source knowledge container to the target knowledge container, while for a bi-directional dependency both directions have to be considered when deriving possible maintenance actions. From our Case Factory perspective two different categories of dependencies, intra-system and inter-system dependencies. Intra-system dependencies exist between different knowledge containers of the same CBR system, while inter-system dependencies exist between knowledge containers of different CBR systems. Distinguishing between intra- and inter-system dependencies is important for processing the dependencies. An intra-system dependencies can be processed by the corresponding Case Factory itself. If no dependencies points to another CBR system, there is no need to propagate the dependencie to the CFO.

An intra-sytem dependency is defined as follows:

$$d_{intra} = (kc_{sysS}, kc_{sysT}, t)$$

$$\begin{aligned} &\text{where } kc \in \{voc, sim, cb, ada\} \text{ and } kc_{sysS} \neq kc_{sysT} \\ &\text{and } sysS, sysT \in \{1 \ldots n\} \text{ and } sysS = sysT \\ &\text{and } t \in \{u, b\} \end{aligned}$$

while an inter-system dependency is defined as follows:

$$d_{inter} = (kc_{sysS}, kc_{sysT}, t)$$

$$\begin{aligned}\text{where } kc \in \{voc, sim, cb, ada\}\\ \text{and } sysS, sysT \in \{1 \dots n\} \text{ and } sysS \neq sysT\\ \text{and } t \in \{u, b\}\end{aligned}$$

There are three intra-system dependencies that could be called trivial dependencies. These trivial dependencies exist between the vocabulary and the other three knowledge containers and are uni-directional. The trivial dependencies are uni-directional, because the vocabulary sets the surrounding conditions of the other knowledge containers: changing the name of an attribute or its value range or creating a new concept for a taxonomy has to be done in the vocabulary and has then an effect on the other knowledge containers. Therefore the dependencies is only pointing from the vocabulary to the other knowledge containers and not backwards, too. These dependencies describe the fact that a change in the vocabulary has a direct impact on the other knowledge containers in the same CBR system. These trivial dependencies are defined per default for every CBR system and are defined as follows:

$$\begin{aligned}d_{triv} &= (voc_{sysS}, sim_{sysT}, u) && \text{where } sysS, sysT \in \{1 \dots n\} \text{ and } sysS = sysT\\ d_{triv} &= (voc_{sysS}, cb_{sysT}, u) && \text{where } sysS, sysT \in \{1 \dots n\} \text{ and } sysS = sysT\\ d_{triv} &= (voc_{sysS}, ada_{sysT}, u) && \text{where } sysS, sysT \in \{1 \dots n\} \text{ and } sysS = sysT\end{aligned}$$

3.2 Dependency modeling in a Maintenance Map

The dependencies between knowledge containers have to be defined by a knowledge engineer. The construct to store the modeled dependencies is a so-called Maintenance Map. The Maintenance Map is based on the Knowledge Map from Davenport and Prusak [4] and was adapted to multi-agent systems by Bach et al. [3]. A Maintenance Map can be represented as a bi-directional graph. The vertices represent knowledge sources, for example a CBR system, and the edges the dependencies between these knowledge sources. There are also loop edges from a vertex to itself to represent the trivial dependencies and it is possible to have multiple edges between two vertices to represent dependencies between multiple knowledge containers of CBR systems. In addition, the edges could be weighted to describe the importance of a dependency. The following figure 1 shows the Maintenance Map for the *docquery* application as a graph. There are dependencies between the vocabularies and the case bases for each CBR system and the number on the edges represent the importance of the dependencies.

Inside the Maintenance Map, the dependencies are modeled in RDF language to simplify the interchange of the Maintenance Map between MAS with multiple CBR systems. In the following we will describe an example based on the docQuery multi-agent system to show the modeling of dependencies:

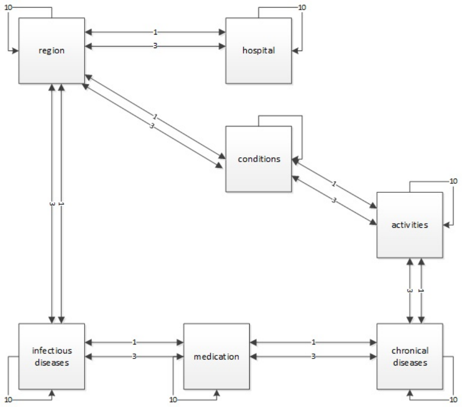


Fig. 1. Maintenance Map for the docquery application as graph

Listing 1.1. Exerpt from a Maintenance Map of the docQuery application

```
<rdf:Description rdf:about=”’dependency1”’>
        <dep:kcsource>vocabulary </dep:kcsource>
        <dep:kctarget>vocabulary </dep:kctarget>
        <dep:cbrsource>DQ_region </dep:cbrsource>
        <dep:cbrtarget>DQ_hospital </dep:cbrtarget>
        <dep:type>bidirectional </dep:type>
        <dep:weight>1</dep:weight>
</rdf:Description>
```

For every dependency the required attributes are modeled in RDF language. The knowledge containers are set with the attributes *kcsource* and *kctarget*, while the CBR systems are set with *cbrsource* and *cbrtarget*. The attribute *type* determines whether a dependency is uni-diretional or bi-directional and the *weight attribute* defines the importance. In this example, the first dependency is an inter-system dependency between the CBR system for region information and the CBR system for hospital information. We have a dependency between the vocabularies of both CBR systems, because several attributes of the different case structures use the same vocabulary. The attribute values for the name of the region in the region CBR system and the region part of the hospitals address are the same. A change of a region's name in the first CBR system has to lead to a change of the same region's name in the hospital CBR system. This way inconsistencies in the knowledge should be avoided. The second dependency is an intra-system and trivial dependency. It exists between the vocabulary and the case base of the region CBR system. Changing the vocabulary may lead to a change of attribute values in one or more cases. This dependency is uni-directional, because an attribute value in a case can only be set after it is defined in the vocabulary. In addition, the Maintenance Map could contain information about preferred maintenance actions for knowledge containers based on the dependencies and required combinations of maintenance actions to preserve the problem solving competence. Information about evaluation strategies for the CBR systems and knowledge containers can be stored, too.

3.3 Deriving maintenance actions from dependencies

After defining dependencies for multiple CBR systems in a multi-agent system, these dependencies are used to derive possible maintenance actions to keep the knowledge in all CBR systems consistent. Each Case Factory derives possible maintenance actions for the assigned CBR system based on intra-system dependencies and the Case Factory Organization derives possible maintenance actions based on inter-system dependencies. In the following we present an algorithm on an abstract level to derive possible maintenance actions based on given dependencies. A maintenance action for this algorithm is defined as a change on a knowledge container *changeKC*. *changeKC($d.kc_{sysS}$)* is a function that changes the knowledge container given as a parameter.

Listing 1.2. Algorithm to derive maintenance actions

```
Input:
D Set of given dependencies (intra- or intersystem)
M Set of initial maintenance actions
Output:
$M_p$ Set of proposed maintenance actions

$M_p$ = M
while (M not empty)
 for (m in M) {
        for (d in D) {
                if ($d.kc_{sysS}$ == $m.kc_{sysS}$
                OR ($d.kc_{sysT}$ == $m.kc_{sysS}$ AND d.t == b)) {
                        if (!$M_p$.contains(changeKC($d.kc_{sysT}$)) {
                                $M_p$.add(changeKC($d.kc_{sysT}$))
                                M.add(changeKC($d.kc_{sysT}$))
                        }
                }
                M.remove(m)
        }
 }
return $M_p$
```

The algorithm requires a set of defined dependencies D and a set of initial maintenance actions M as input. If M is empty, the algorithm terminates, because no starting point for the algorithm would be given. The output of the algorithm is a set of possible maintenance actions that could be proposed to the knowledge engineer. At first, the initial set of maintenance actions will be added to M_p, because these maintenance actions should be proposed, too. The condition for the while loop is that no new maintenance actions could be derived, so no more dependencies have to be considered. The inner loops process all defined dependencies and the initial and derived maintenance actions. If a new maintenance action is derived, it is added to M and M_p. A new maintenance action added to M leads to another cycle of the inner loop to determine if further dependencies fire for the new maintenance action. And the new maintenance actions is added to M_p to be proposed to the knowledge engineer. Two conditions are responsible for deriving new maintenance actions: If the source knowledge container of a maintenance action is the same as the source knowledge container of a dependency OR if the dependency is bi-directional and the source knowledge container of the maintenance action is the same as the target knowledge container of the dependency. If one condition is met, a new maintenance action is derived and added to the sets. A maintenance action can only be in a set once. After processing all dependencies for a maintenance action, this maintenance action is removed from M. This is necessary to have an empty list after processing all maintenance actions and dependencies.

4 Summary and Outlook

In this paper we describe the dependencies between knowledge containers of CBR systems from a Case Factory perspective to use them to derive possible maintenance actions. We describe the categories and elements of a dependency and show how defined dependencies could be modeled with the help of a Maintenance Map. In addition, we present an algorithm to use these dependencies to derive possible maintenance actions. The next steps in our work are to define and model all dependencies in our docQuery multi-agent sytem and detail, implement and test the algorithm to derive maintenance actions. Therefore, the possible maintenance actions and their combinations have to be defined. Based on the result of our evaluation, we will revise our algorithm and dependency modeling.

References

1. Althoff, K.D.: Collaborative multi-expert-systems. In: Proceedings of the 16th UK Workshop on Case-Based Reasoning (UKCBR-2012), located at SGAI International Conference on Artificial Intelligence, December 13, Cambride, United Kingdom. pp. 1–1 (2012)
2. Bach, K.: Knowledge Acquisition for Case-Based Reasoning Systems. Ph.D. thesis, University of Hildesheim (2013), dr. Hut Verlag Mnchen
3. Bach, K., Reichle, M., Reichle-Schmehl, A., Althoff, K.D.: Implementing a coordination agent for modularised case bases. In: Proceedings of the 13th UK Workschop on Case-Based Reasoning. pp. 1–12 (2008)
4. Davenport, T.H., Prusak, L.: Working Knowledge: How Organizations Manage What they Know. Havard Business School Press (2000)
5. Ferrario, M.A., Smyth, B.: Distributing case-based maintenance: The collaborative maintenance approach. Computational Intelligence 17(2), 315–330 (2001)
6. Iglezakis, I., Roth-Berghofer, T.: A survey regarding the central role of the case base for maintenance in case-based reasoning. In: ECAI Workshop Notes. pp. 22–28 (2000)
7. Leake, D., Kinley, A., Wilson, D.: Learning to integrate multiple knowledge sources for case-based reasoning. In: Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence. pp. 246–251. Morgan Kaufmann (1997)
8. Nick, M.: Experience Maintenance Loop through Closed-Loop Feedback. Ph.D. thesis, TU Kaiserslautern (2005)
9. Reuss, P.: Concept and implementation of a Knowledge Line - retrieval strategies for modularized, homogeneous topic agents within a multi-agent-system (in German). Master's thesis, University of Hildesheim (2012)
10. Reuss, P., Althoff, K.D., Henkel, W., Pfeiffer, M.: Case-based agents within the omaha project. In: Case-based Agents. ICCBR Workshop on Case-based Agents (ICCBR-CBR-14) (2014)
11. Roth-Berghofer, T.: Knowledge maintenance of case-based reasoning systems. The SIAM methodology. Akademische Verlagsgesellschaft Aka GmbH (2003)
12. Smyth, B., Keane, M.: Remembering to forget: A competence-preserving case deletion policy for case-based reasoning systems. In: Proceedings of the 13th International Joint Conference on Artificial Intelligence. pp. 377–382 (1995)
13. Stahl, A.: Learning feature weights from case order feedback. In: Case-Based Reasoning Research and Development: Proceedings of the Fourth International Conference on Case-Based Reasoning (2001)

On the Semantification of 5-Star Technical Documentation

Sebastian Furth[1] and Joachim Baumeister[1,2]

[1]denkbares GmbH, Friedrich-Bergius-Ring 15, 97076 Würzburg, Germany
[2]University of Würzburg, Institute of Computer Science,
Am Hubland, 97074 Würzburg, Germany
{sebastian.furth,joachim.baumeister}@denkbares.com

Abstract. Technical documentation is a special purpose content describing machines and plants with high complexity. The documentation covers operation, maintenance and repair of the technical artifacts. The high complexity of the machines yields a voluminous documentation, where it increasingly becomes difficult to find the relevant information for a given problem. The paper discusses the use of semantic technologies to organize the documentation on a syntactic and semantic level. Also, a scheme for the assessment of the maturity of existing documentation is proposed, that simplifies the application of semantic technologies.

Keywords: Semantic Publishing, Ontology Engineering, Information Extraction

1 Introduction

The complexity of machines has grown dramatically in the past years. As a consequence, the technical documentation became a fundamental source for service technicians in their daily work. Service technicians need fast and focused access methods to handle the massive volumes of technical documents. For this reason semantic search emerged as the new system paradigm for the presentation of technical documentation. However, the existing corpora are usually not semantically prepared. The best existing solutions may give access to dedicated sections, while the information relevant for the service technician remains concealed. In this paper we present a novel ontological representation for technical documents that combines structural and rhetorical elements to enable direct access to *Core Documentation Entities.* We additionally introduce a maturity schema that allows the assessment of existing technical documentation with respect to these Core Documentation Entities.

The remainder of this paper is structured as follows. In Section 2 we first give a general introduction to technical documentation and present a novel maturity scheme for the assessment of their quality. The maturity scheme relies on semantic technologies, hence we present ontologies for the ontological description of technical documents in Section 3. Section 4 shows the practical applicability of the presented ontologies. We conclude with a summary and a statement of future research directions.

2 5-Star Technical Documentation

In this section we introduce the domain of technical documentation as a special type of textual and multimedia resources. We motivate that the semantification enables reuse and integration of the resources for various applications.

2.1 Uses of Technical Documentation

Builders of machinery and plants provide technical documentation to support the service technician to ensure the save operation and maintenance of their products. Typically, the documentation is created to efficiently support the following tasks:

1. Operation of the machine
2. Maintenance of the machine
3. Localization of specific components
4. Diagnosis of problems
5. Repair of a localized damage

Historically, the documentation is partitioned into a number of books supporting the particular tasks by technical descriptions:

User Manual describes the operation of the machine, i.e., how to activate and perform the machine functions.

Repair Manual shows the replacement and maintenance of specific components of the machine.

Technical Functions and Diagnosis describe the logical connections and relations of components within the machine, in order to support the diagnosis of observed faults. Typical examples are electrical and hydraulic wiring diagrams.

Spare Parts provide a detailed view of parts located in particular components. Service technicians locate parts by using this documentation, but also to order new parts in exchange for faulty parts.

In the past, the documentation was printed on paper. With the increasing complexity of the machines many vendors switched to electronic versions of the books in recent years (PDF and HTML). For instance, the documentation for a full-featured harvesting machine or other special purpose vehicles comprise about 10,000 pages. With the electronic availability the metaphor of a single 'book' is not necessary anymore.

In recent years, semantic technologies emerged to (re-)organize the structuring of documents in corporate environments [3]. Hence, advanced methods are emerging for searching for relevant chapters and navigating between information units. Applications within the infrastructure of the technical documentation were improved, such as automated term extraction and general information extraction tasks. More importantly, interesting end-user applications become possible such as *semantic search* [7, 11] and *semantic assistants* [15].

However, existing documentation data does not necessarily fulfill all requirements for semantic applications. The quality state of existing documentation data can vary massively, ranging from scanned image PDF documents to products of XML content management systems. In practice, it is helpful to provide a classification schema to assess the maturity level of the existing documentation. This schema also gives advices for improving the current state of documentation.

2.2 Towards the quality of Technical Documentation

We introduce a maturity schema for the assessment of technical documentation data. The schema lists a number of quality criteria building on each other. For each criteria we give one star; that way the maturity of documentation data can range from one star to five stars. This schema is inspired by the idea of evaluating the quality of data in the linked open data cloud [1, 9], and was adapted to the needs of technical documentation. The aims of the schemes, however, are identical: First, users should obtain an intuitive impression about the maturity of their data; second, users should get motivated to increase the stars of their data by adding more semantics. The schema for *5-Star Technical Documentation* data is depicted in Figure 1.

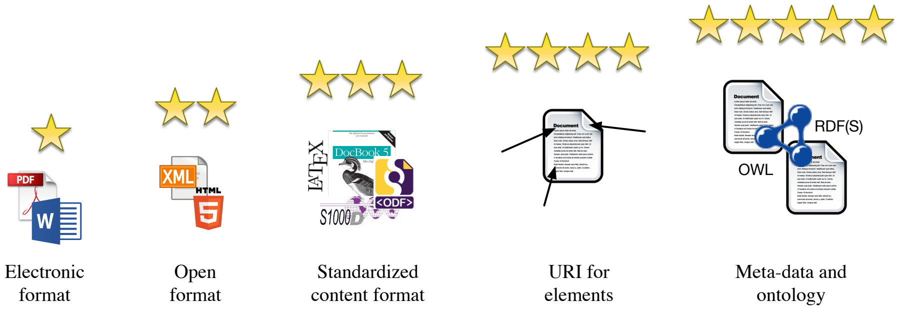


Fig. 1. The levels of the 5-stars maturity schema for technical documentation.

The first star is given, when the documentation is accessible in an electronic format, for instance, as PDF or MS-Word. The documentation gets two stars, when it is accessible in a structured and non-proprietary format, e.g., XML, SGML, or Markdown. Three stars are received for documentation that is accessible in a standardized format, e.g., DocBook XML or ASD S1000D[1]. Documentation with four stars provide URIs for all relevant elements of the content. That way, the book itself, the particular chapters, and paragraphs can be clearly named and thus can be linked by external applications. Five stars documentation adds semantics to the relevant elements by attaching meta-data to the elements that refers to concepts of an ontology. Using an ontology enables the automated interlinkage of document elements by using the same concepts of the ontology. Also external ontologies with similar semantics can by aligned to the used ontology. In the following, we discuss the use of open documentation standards and ontologies in order to receive the 5-stars level.

3 Ontologies for Technical Documentation

For the semantic representation of technical documentation we pick up the established idea from the semantic publishing community of the definition of OWL [8] or RDFS [14] vocabularies that describe certain aspects of the publishing domain. Such aspects typically comprise structural components (e.g. paragraphs, sections, sentences) and rhetorical elements (e.g. discourse elements / sections like "Motivation", "Problem Statement" or "Discussion"). Complementary ontologies often provide annotation vocabulary that allows the definition of additional meta data. In the following we first describe suitable vocabularies for the representation of structural and rhetorical aspects of a technical document. Building upon these vocabularies we introduce a novel ontology that exploits structural and rhetorical aspects to facilitate direct access to core documentation entities like component overviews or repair procedures. At this point the technical documentation already gets four out of five stars. The addition of annotation vocabularies completes the section with the achievement of 5-star technical documentation.

3.1 Structural Components

Considering only the pure structural composition of a document, the required vocabulary is rather independent of the underlying problem domain. The Document Ontology schema of the SALT ontology [6] or the pattern ontology [4] are popular examples for the description of (scientific) publications. However, for publications in the technical domain DocBook [12] is a de facto standard maintained by the Organization for the Advancement of Structured Information Standards (OASIS)[2]. Following the maturity schema introduced in Section 2.2 documents written according to this standard receive the 3-stars level. Thus we encourage the usage of a DocBook-like ontology for the structural description of technical documentation. Şah and Wade [13] proposed an ontology that covers a

[1] http://www.s1000d.org/

[2] https://www.oasis-open.org/

reasonable subset of the DocBook standard. Table 1 briefly introduces the most important elements of this ontology, e.g. `docbook:Book`, `docbook:Article`, `docbook:Chapter` or block elements like `docbook:Paragraph`, `docbook:Procedure` or `docbook:Figure`.

Element	Type	Description
`docbook:Book`	`Class`	Represents the top level element that has a number of sub-components like articles or chapters.
`docbook:Article` / `docbook:Chapter`	`Class`	Articles and chapters contain (sequences of) block elements.
`docbook:BlockElement`	`Class`	Block elements are typically used as atomic information units. Common examples that are available as subclasses are `docbook:Paragraph`, `docbook:Table`, `docbook:List`, `docbook:Procedure` or `docbook:Figure`
`dc:hasPart`	`Property`	Property from the Dublin Core ontology that connects instances of the DocBook classes

Table 1. Important elements of the DocBook ontology [13].

3.2 Rhetorical Components

In contrast to the structural organisation of a document the rhetorical ontology concentrates on modeling the rhetorical structures and elements of the document. A correspondence of structural components does not necessarily exist in the rhetorical organisation of the document. However, core rhetorical structures like safety instructions can often be linked explicity to particular structures like chapters, sections or paragraphs. For the representation of scientific articles the Rhetorical Ontology schema of the SALT ontology [6] or the Discourse Elements Ontology [2] provide appropriate vocabulary. Thus, rhetorical aspects like the motivation, background, methods etc. can be modeled as instances of respective classes. While the underlying idea also facilitates the rhetorical modeling of technical documentation the concrete classes do not fit the technical domain. For instance law requires technical documentation to follow a certain rhetorical organisation, e.g. safety notes need to preceed actual operation instructions. Thus it would be benefitial to semantically represent safety notes. Table 2 gives a non-exhaustive overview of common rhetorical elements in technical documents.

3.3 Core Documentation Entities = Structure + Rhetoric

The maturity schema introduced in Section 2.2 requires that relevant elements are identifiable by URIs. Representing the structural and rhetorical aspects of technical documentation is a considerable step in this direction. However, the most important aspects of technical documents are interweaved in these two

Element	Description
`rtc:Index`	Indices like table of contents, subject catalogs, list of abbreviations etc.
`rtc:GeneralInformation`	General aspects of the document or the machine in focus.
`rtc:SafetyInstruction`	Safety notes to be obtained while working with the machine.
`rtc:Description`	Information about specific components or functions.
`rtc:Operation`	Information about the usage of the machine, specific components or functions.
`rtc:Repair`	Repair procedures; important subclasses are `rtc:Assembly` and `rtc:Disassembly`
`rtc:Maintenance`	Information about maintenance works, schedules etc.
`rtc:Adjustment`	Information about necessary adjustments in specific situations.
`rtc:FaultIsolation`	Detailed troubleshooting information.
`rtc:Parts`	Spare part information.

Table 2. Common rhetorical components in technical documentation.

structures. The entropy of these aspectes is typically sufficient to satisfy an immediate information need. In the following we give excerpts of a novel ontology, that combines structural and rhetorical aspects in order to make these *Core Documentation Entities* easily accessible. A typical example for such an information need is a (dis-)assembly procedure. The corresponding information can be obtained by combining the rhetorical structure `rtc:Assembly` with the structual element `docbook:Procedure`:

$$\texttt{cde:AssemblyProcedure} \sqsubseteq \texttt{rtc:Assembly} \sqcap \texttt{docbook:Procedure}$$

Another example are component overviews that can typically be found in a section describing the machine or in the spare part information. Component overviews typically consist of an exploded-view drawing and an associated list of labels, product numbers etc.:

$$\begin{aligned}&\texttt{cde:ComponentOverview} \sqsubseteq \\ &\quad (\texttt{rtc:Description} \sqcup \texttt{rtc:Parts}) \sqcap \\ &\quad \exists\,(\texttt{dc:hasPart.docbook:Figure} \sqcap \texttt{dc:hasPart.docbook:List})\end{aligned}$$

3.4 Linked Documentation Data

The structural and rhetorical representation of technical documents and the subsequent identification of core documentation entities receives a publication four stars in the presented maturity schema. The maturity schema requires that documents have meta-data from an ontolgy attached to receive the fifth star. We recommend the usage of the `dc:subject` property from the Dublin Core [10] ontology for the annotation of structural, rhetorical or core documentation entities with concepts from (enterprise) ontologies. For instance, consider a document that has been annotated with concepts describing relevant components or functions of a machine. Then a complete repair instruction (assembly + disassembly) for a concrete component (`ex:componentA`) can be identified as follows:

$$\begin{aligned}&\texttt{ex:RepairComponentA} \sqsubseteq\\ &\quad(\texttt{rtc:AssemblyProcedure} \sqcup \texttt{rtc:DisassemblyProcedure}) \sqcap\\ &\quad\forall\,(\texttt{dc:subject.ex:componentA})\end{aligned}$$

4 Extended Example

The following Turtle excerpt, is an example of how the ontologies described in Section 3 may be used to represent a technical document. The example gives an ontological description of a repair manual that contains detailed information (`docbook:Step`) about the assembly and disassembly of a concrete component.

```
:repair-manual a docbook:Book ;
  dc:hasPart :index , :general , :safety , :repair .

:repair a docbook:Chapter , rtc:Repair ;
  dc:hasPart :repair-a , :repair-b .

:repair-b a docbook:Chapter ;
  dc:hasPart :disassembly-b, :assembly-b ;
  dc:subject :component-b .

:disassembly-b a docbook:Chapter, rtc:Disassembly ;
  dc:hasPart :safety-note; :some-text; :some-procedure .

:some-procedure a docbook:Procedure ;
  dc:hasPart
    [ a docbook:Step ;
      dc:description ”Insert stem into the fork.” ],
    [ a docbook:Step ;
      dc:description ”Point stem towards the front.” ] .
...
```

Listing 1. Example ontology representing a repair manual.

5 Summary and Future Work

This paper introduced a maturity schema that allows the assesment of existing technical documents according to certain quality criterias. The schema is inspired by the 5-star Linked Open Data idea but consideres important aspects of the Technical Documentation and Publishing domain. The maturity schema requires the usage of documentation standards and ontologies. Thus we proposed the representation of technical publications in a DocBook-like ontology. This representation is accompanied by a novel ontology that covers the rhetorical aspects of a technical document. Combining both ontologies in complex OWL [8] classes reveals core documentation entities. These high entropy elements can immediatly satisfy an information need. Hence, effective access to these elements yields huge time savings. The completion of rhetorical elements for technical documentation as well as the definition of supplementary core documentation entities will be subject of future work. We additionally plan to implement methods for the automatic conversion of 1-star legacy data to 4-star ontological data. These methods shall also be combined with our existing semantification approaches [5].

References

1. Berners-Lee, T.: Linked data (http://www.w3.org/designissues/linkeddata.html)
2. Constantin, A., Peroni, S., Pettifer, S., Shotton, D., Vitali, F.: The Document Components Ontology (DoCO). Semantic Web Preprint(Preprint) (2015)
3. Coskun, G., Streibel, O., Paschke, A., Schäfermeier, R., Heese, R., Luczak-Rösch, M., Oldakowski, R.: Towards a corporate semantic web. In: International Conference on Semantic Systems (I-SEMANTICS '09). pp. 602–610. Graz, Austria (2009)
4. Di Iorio, A., Peroni, S., Poggi, F., Vitali, F.: Dealing with structural patterns of xml documents. Journal of the Association for Information Science and Technology 65(9), 1884–1900 (2014)
5. Furth, S., Baumeister, J.: Towards the semantification of technical documents. In: FGIR'13: Proceedings of German Workshop of Information Retrieval (at LWA'2013) (2013)
6. Groza, T., Handschuh, S., Möller, K., Decker, S.: SALT-Semantically Annotated LaTeX for Scientific Publications. In: The Semantic Web: Research and Applications, pp. 518–532. Springer (2007)
7. Guha, R., McCool, R., Miller, E.: Semantic search. In: Twelfth International World Wide Web Conference (WWW 2003) (2003)
8. Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P.F., Rudolph, S. (eds.): OWL 2 Web Ontology Language: Primer. W3C Recommendation (27 October 2009), available at http://www.w3.org/TR/owl2-primer/
9. Janowicz, K., Hitzler, P., Adams, B., Kolas, D., II, C.V.: Five stars of linked data vocabulary use. Semantic Web 5(3) (2014)
10. Kokkelink, S., Schwänzl, R.: Expressing qualified dublin core in RDF/XML (2001)
11. Mäkelä, E.: Survey of semantic search research. In: Proceedings of the seminar on knowledge management on the semantic web (2005)
12. Norman, W., Hamilton, R.L.: DocBook 5: The Definitive Guide. O'Reilly Media, Inc. (2010)
13. Şah, M., Wade, V.: Automatic metadata extraction from multilingual enterprise content. In: Proceedings of the 19th ACM international conference on Information and knowledge management. pp. 1665–1668. ACM (2010)
14. W3C: RDF Schema 1.1 – W3C Recommendation. http://www.w3.org/TR/rdf-schema (February 2014)
15. Witte, R., Gitzinger, T.: Semantic Assistants – User-Centric Natural Language Processing Services for Desktop Clients. In: 3rd Asian Semantic Web Conference (ASWC 2008). LNCS, vol. 5367, pp. 360–374. Springer, Bangkok, Thailand (February 2–5 2009), http://rene-witte.net/semantic-assistants-aswc08

A framework for using social media channels in knowledge exchange with customers

Susanne Durst[1], Michael Leyer[2*]

[1] University of Skövde, Skövde, Sweden
susanne.durst@his.se
[2] University of Rostock, Rostock, Germany
michael.leyer@uni-rostock.de

Abstract. Social media channels become more and more important for service providers in contacting customers. Given the variety of offers it is important to understand the contribution of social media channels to knowledge exchange with customers. We analyse the requirements of customer contact in service provision and develop a framework how different social media channels can be used for knowledge exchange. In particular, we show from the perspective of service providers how these organisations may apply different social media channels in different stages of service processes.

Keywords: Knowledge exchange, social media channels, framework

1 Introduction

Knowledge exchange between customers and providers occurs through interaction channels [1]. An interaction channel is described as medium or customer contact point through which a customer contact takes place. Such interaction channels can be traditional (e.g. store, personal contact or mail) or modern (e.g. email, websites or social media). Irrespective from the specific channels, multi-channel management is relevant before, during and after service delivery as there are contact points between provider and customer including suppliers [1].

Among these channels, social media channels are becoming more and more important as there has been a dramatic increase in using social media platforms for a variety of communication purposes [2]. The advancements in the area of social media applications have opened up a feeling of self-determination and co-determination of customers [3]. Given the fact that there is a variety of social media channels available [4] and activities are less controllable [5], which can lead to detrimental consequences

[*] Both authors contributed equally to this work and should be considered co-first authors.

(“online firestorms”) [6], the research question is how knowledge exchange in a multichannel setting of different social media channels should take place. While for firms it is necessary to analyse how customers react to service offering in terms of adoption or satisfaction, the question how knowledge that may be critical for service innovation or other areas of business development is exchanged across several channels is not well understood so far [1]. The aim is to develop how knowledge exchange between customers and service providers may take place through social media channels.

2 Theoretical background

2.1 Service delivery

Irrespective from the context, a service delivery process consists of a start event followed by activities and an end event [7]. The start event is triggered by customers who make the decision to buy the service and the end event occurs when the service delivery has finished. The activities between are either performed by the provider (and eventually a supplier) or by the customer [8]. Service delivery processes can occur more than once for a specific customer and are performed for different customers. Thus, service delivery is characterised by customer integration which is heterogeneous, i.e. how this integration takes place can differ between customers and each time a service is executed for a customer [9].

Additionally, there can be processes before and after the service delivery that are connected to the service delivery process [10]. Processes before are typically information search, negotiation and contract conclusion [10] while within usage or enjoyment of the result afterwards [11] customer complaints might occur for example. The phases can be described as follows:

1. Information search: The first step towards a service delivery is to search for information. Customers have a certain need and look for information how to fulfil it.
2. Negotiation: If a customer is demanding a service or product then potentially negotiations can take place. This covers e.g. the price or characteristics of delivery.
3. Contract conclusion: If there is mutual agreement on the service offer, a contract between buyer and seller is the result.
4. Integration during delivery: Customers are integrated during the service delivery process. Knowledge is exchanged in terms of specific customer characteristics and details regarding the delivery.
5. Usage, enjoyment: Once a service is delivered/ produced it will be used by customers. In this phase knowledge exchange can occur through complaints by customers which have to be handled by service providers.

2.2 Knowledge exchange

Knowledge exchange incorporates the exchange of knowledge between customers, suppliers and provider as well as within supplier and provider companies involved [8]. Such knowledge can be directly related to a specific service delivery, general knowledge about services offered or regarding customer experience. The communication and thus the exchange of knowledge with the customer happens through a number of different channels.

Thanks to the ICT advancements, it is easier than ever to find and share knowledge, detached from time and space, and most individuals do it; also in their leisure time. Some technologies can be considered as better drivers or facilitators of these activities than other ones [12], social media have shown their enormous potential. At the same time, boundaries between work related knowledge sharing and private knowledge sharing are increasingly blurred, since the different social media sites can easily be assessed and used through computers and mobile devices. Furthermore, an increasing number of organisations apply social media with different groups of stakeholders and thus expand their scope of knowledge sharing [13].

2.3 Social media channels

The communication and thus the exchange of knowledge with the customer can occur through different channels [1]. While traditional interaction channels such as branches, telephone or the own website are highly under control of service providers, this does not hold true for social media channels. Here, service providers have a user status of a third-party program in a similar way as their customers. Customers have more possibilities to interact and make their knowledge public the latter being rarely the case for traditional channels. To better cope with the variety of social media channels [4, pp. 8-12] grouped them into four zones:

1. Zone 1 Social community: Social communities refer to channels that focus on relationships and the gathering of people that share the same interest or identification. Examples are social networking sites (SNS), forums, and wikis.
2. Zone 2 Social Publishing: Social publishing sites support the dissemination of content to a target group. Examples are blogs and media sharing sites.
3. Zone 3 Social Entertainment: These channels aim at offering opportunities for play and enjoyment. Examples are social games and entertainment communities.
4. Zone 4 Social Commerce: Social commerce is about the usage of social media for online buying and selling of physical goods and services. The channels in this zone also cover review sites, deal sites, social shopping markets, and social storefronts.

3 Social media channel usage in service processes

In line with the research question, we develop a conceptual framework that is displayed in Table 1. The framework links knowledge exchange in the phases of service processes (Section 2.1) with the four social media zones. More precisely, our aim is to show how different forms of knowledge exchange (i.e. information search, negotiation, contract conclusion, integration during delivery and usage/enjoyment) can be realised by applying social media channels.

	Zone 1	Zone 2	Zone 3	Zone 4
Information search	X	X		X
Negotiation	X			X
Contract conclusion	X		X	X
Integration during delivery	X		X	X
Usage/Enjoyment	X	X	X	X

Table 1. Usage of social media channels for knowledge exchange in different phases of service processes

Table 1 clarifies that depending on the form of knowledge exchange taking place during service processes different social media zones are affected:

- Information search: Using social media channels, information in this phase can be obtained from other individuals active in social communities. Here, customers seek for information such as how to solve a possible problem, which service offers other people may know and which experiences others have been made with the services offered. Thus, individuals can look for information or offer information/experience to others. A provider can step in this phase and also provide information. Another way of knowledge exchange in this phase is via social publishing, i.e. an individual or company provides a special blog on a specific solution. Additionally, information can be distributed through websites which aim at social commerce. Social entertainment channels are not relevant here, as these are typically not used for information search by customers.
- Negotiation: Next to direct negotiations between a customer and a company, it can happen online as well, e.g. in the case of online auctions. Within such social commercial websites, information exchanges within bidding processes take place between a company and many bidders.
  Furthermore, offers can be negotiated through social communities, in that case using them as communication channel. Such a type of communication is not feasible for social publishing as feedback is not possible, and does not represent the aim of social entertainment channels.
- Contract conclusion: As this phase also requires communication between the parties, execution via social publishing channels is not suitable. Yet, in addition to social communities and social commerce channels, offers can also be accepted in social entertainment environments. Examples are social online games (e.g. World of Warcraft) in which digital items can be bought.
- Integration during delivery: As in the prior phase, social publishing channels are not suitable. If a personal contact between customers and service providers is not required during service delivery (e.g. hair cutting), knowledge can also be exchanged through the other social media channels. Customers can provide specific characteristics on their own or these are already stored in applications within the social channels. Additionally, information provided on social networks or observed behaviour can be extracted.
- Usage/enjoyment: After a service is delivered and experienced by the customers, they can share the experiences within every social media channel. Such knowledge dissemination can include the description of features, benefits, weaknesses, opinions or complaints. Service providers can be present with their own accounts and actively

seek and accept inputs from customers (i.e. social commerce channels). However, customers may also complaint about or praise products with their social peers or visible to everyone (social publishing). Here, providers need to continuously monitor relevant social channels to pick up reactions to their services.

It should be emphasised that the separate description of channels does not implicate that knowledge exchanges regarding a specific service take place within one channel only. In fact, several social channels can be used stepwise, at the same time and in combination with other channels such as email, phone or a branch.

4 Influence on knowledge exchange

The usage of social media channels in the different phases of knowledge exchange will also have implications for the parties involved. Next, we highlight a number of implications, taking the perspective of a service provider.

The integration of customers in the service delivery process provides service providers the opportunity of getting access to outside knowledge and thereby expanding or updating their own knowledge base. Ideally, the combination of customers' knowledge and service provider's knowledge results in new or improved services. In order to do so, virtual reality labs may be considered as promising tools of service creation. Service providers and customers (e.g. lead users) can collaborate in real time and across geographical boundaries. They can pick up questions from and concerns of customers by answering via the same social media channel in a fast and detailed way.

Customers are searching for information on products and services using different social media channels, but are also using these channels to share information on their preferences and market trends [14]. In addition, they will also share their positive and negative experience with the service. Thus, customers act as boundary spanners between the firms and the market.

Provision of information and knowledge to potential customers about the attributes of products can take place where the customer is. Thus, a switch between channels is not necessary which reduces the risk of customers switching to other providers.

Information and (explicit) knowledge is not only shared through social media channels but can also easily be stored and then processed for later usage. This allows service providers to continuously develop their knowledge base. However, knowledge from different social channels is mainly stored with the provider of the social channel. A service provider has to export such knowledge to its own systems and combine knowledge on customers to enable internal reporting. Next to the direct communication with a customer, this can also be e.g. information on how well the last Facebook-campaign performed or how the number of followers develop. Furthermore, other customers can add their experience and knowhow to knowledge exchanges between customer and service providers that take place openly to the public. Such knowledge can also be accessed, stored, used to improve a service offer or to develop new ones.

5 Conclusions and Outlook

The challenge for service providers will lie in their capability of combining knowledge per customer over different channels. This is difficult as information is typically stored with other providers offering platforms for social media channels. Moreover, analytics often rely on the provider offerings and are not standardised across them. The present study provides practitioners' insights and ideas how best use could be made of different social media channels for information and knowledge exchange.

From a theoretical point of view, this study provides novel insights into the study of interactions between providers and customers as it draws particular attention to the contribution of social media channels in these interactions. These insights thus expand our body of knowledge regarding multi-channel management activities.

Future work should especially shed light on the following topics to strengthen our understanding of knowledge exchange within social media channels:

- Are there specific types of customers demanding specific combinations of social media channels which can be described in a standard knowledge exchange?
- Are there significant differences or similarities between industries regarding knowledge exchange via social media channels?
- How can service providers make use of knowledge which cannot be transferred to the own systems in order to develop a conjoint knowledge datasheet per customer?
- How should algorithms be designed to analyse customer knowledge if it is drawn from different social media channels with various formats?
- Does a company have to be present in at least four channels one for each zone to gather all possible feedback from potential and existing customers?

References

1. Neslin, S.A., Grewal, D., Leghorn, R., Shankar, V., Teerling, M.L., Thomas, J.S., Verhoef, P.C.: Challenges and opportunities in multichannel customer management. Journal of Service Research 9, 95-112 (2006)
2. Whiting, A., Williams, D.: Why people use social media. A uses and gratifications approach. Qualitative Market Research. An International Journal 16, 362-369 (2013)
3. Weiber, R., Wolf, T.: Disruptive Empowerment. Auswirkungen von Kundeninteraktionen auf den Social-Media-Erfolg. Marketing Review St. Gallen 4, 42-47 (2012)
4. Tuten, T.L., Solomon, M.R.: Social Media Marketing. Sage, Thousand Oaks (2015)
5. Mangold, W.G., Faulds, D.J.: Social media. The new hybrid element of the promotion mix. Business Horizons 52, 357-365 (2009)
6. Pfeffer, J., Zorbach, T., Carley, K.M.: Understanding online firestorms. Negative word-of-mouth dynamics in social media networks. Journal of Marketing Communications 20, 117-128 (2014)
7. Davenport, T.H., Short, J.E.: The New Industrial Engineering. Information Technology and Business Process Redesign. Sloan Management Review 31, 11-27 (1990)
8. Wynstra, F., Spring, M., Schoenherr, T.: Service triads. A research agenda for buyer-supplier-customer triads in business services. Journal of Operations Management 35, 1-20 (2015)
9. Leyer, M., Moormann, J.: A method for matching customer integration with operational control of service processes. Management Research Review 35, 1046-1069 (2012)

10. Williamson, O.E.: The economic institutions of capitalism. Firms, markets, relational contracting. Free Press, New York (1985)
11. Vargo, S.L., Lusch, R.F.: Evolving to a new dominant logic for marketing. Journal of Marketing 68, 1-17 (2004)
12. Sicilia, M.-A., Lytras, M.D.: The semantic learning organization. The Learning Organization 12, 402-410 (2005)
13. Kaplan, A.M.: If you love something, let it go mobile. Mobile marketing and mobile social media 4x4. Business Horizons 55, 129-139 (2012)
14. Cheung, F.Y.M., To, W.M.: Do task- and relation-oriented customers co-create a better quality of service? An empirical study of customer-dominant logic. Management Decision 53, 179-197 (2015)

Wissensmanagement in volatilen und temporären Organisationen.

Andreas Korger
Angesagt GmbH, Dettelbachergasse 2, 97070 Würzburg
a.korger@angesagt-gmbh.de

Abstract: In volatilen und temporären Organisationen erschweren sich viele Aufgaben des Wissensmanagements. Mit volatil ist gemeint, dass Akteure einer Organisation oft wechseln, in ihren Eigenschaften divers sind, räumlich verteilt sind oder organisationstypische Strukturen wie fixe Stellen oder Weisungs- und Informationspflichten kaum vorhanden sind. Temporär meint, dass die Organisation in ihrer Aktivität zeitlich beschränkt ist. Dieses Papier ist ein Vorschlag für eine Forschungsarbeit. Ziel ist es, alle Beteiligten solcher Organisationen zu integrieren und damit den Zugang zu Wissensmanagement zu ermöglichen.

Keywords: Wissensmanagement, Wissensmanagementsysteme, Erfahrungsmanagement, Prozessmanagement, Organisation

1 Problemstellung und Motivation

In einer „normalen“ Organisation sind Strukturen relativ klar erkennbar und verlässlich. Akteure gehören der Organisation in der Regel einige Jahre an, haben Erfahrung gesammelt und besetzen eine Stelle mit konkret definierten Aufgaben. Akteure unterliegen einer Hierarchie oder vergleichbaren Struktur, die Weisungs- und Informationssystematik zwischen den Akteuren regelt. Es gibt formalisierte Prozesse, nach denen sich die Akteure richten können und die Abläufe in der Organisation steuern. Die Organisation besitzt ein gewisses, von den Akteuren abstrahiertes Wissen, welches formalisiert vorhanden ist. Auch haben solche Organisationen meist einen Bestand von mehreren Jahrzehnten und können dementsprechend längerfristig planen und handeln. Solche Organisationen erfüllen auch die Voraussetzungen bewährte Methoden des Wissensmanagements effizient einsetzen zu können [1, S. 300]. Diese Voraussetzungen liegen in Organisation, Mensch und Technik begründet. Beispiele sind das Vorhandensein einer technischen und organisatorischen Infrastruktur, ein Minimum an Prozessorientierung oder motivationale Unterstützung [1, S. 310].

Es gibt aber auch Organisationen, bei denen die Strukturen sehr veränderlich sind. Das Wissen ist weitgehend in den Akteuren „gespeichert“ und liegt meist als Erfahrungswissen vor. Wenn Wissen gespeichert ist, dann in der Regel unstrukturiert. Das

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

Wissen der Akteure liegt in verschiedener Form vor. Man hat gewisse Prinzipien (Regeln), man erinnert sich an ähnliche Situationen (Fälle) oder hat erlerntes Wissen (Ontologien). Der Informationsfluss zwischen den Akteuren ist unterschiedlich stark ausgeprägt und kaum geregelt. Im Vordergrund soll hier weniger die rechtliche und betriebswirtschaftliche Form der Organisation stehen. Der Begriff meint eher den ursprünglicheren Sinn von gemeinsamer zielgerichteter Tätigkeit unter Einfluss gewisser Regeln. Diese Charakteristik hat zur Folge, dass Voraussetzungen, auf Basis derer konventionelle WM-Methoden entwickelt wurden, ganz oder teilweise fehlen. Es stellt sich die Frage, wie man nun Wissensmanagement an eine solche Umgebung anpassen kann. Ist das im Einzelfall überhaupt sinnvoll, weil ggf. die Kosten den Nutzen übersteigen. Forschungsfragen sind in diesem Zusammenhang wie man eine temporäre Organisation, ein zugehöriges Kosten- und Nutzenkonzept sowie ein Kommunikationskonzept geeignet modellieren kann. Wie können bestehende Methoden vereinfacht werden? Wie können Kosten des WM reduziert werden? Das Ziel der Forschung ist es, die Nutzung von Wissensmanagement insbesondere Erfahrungsmanagement einem größeren Kreis zugänglich zu machen.

1.1 Charakteristik volatiler Organisationen

Kennzeichen einer volatilen Organisation sind die räumliche Verteilung, der häufige Wechsel und die ggf. hohe Anzahl zugehöriger Akteure. Außerdem sind die Akteure sehr unterschiedlich, was ihre Eigenschaften betrifft. Dies hat gravierende Folgen. Es bleibt kaum Zeit, das Wissen der Akteure dem Organisationswissen hinzuzufügen, den Akteuren etwas beizubringen oder sie an die Ablaufstrukturen der Organisation anzupassen (soweit überhaupt vorhanden). Die Akteure wechseln häufig und damit auch deren Qualifikation sowie die Besetzung von Stellen in der Organisation. Es besteht Unsicherheit darüber, welches Wissen neue Akteure mitbringen und Wissen der alten Akteure geht regelmäßig verloren. Akteure zeigen kaum Bestreben freiwillig Organisationsinteressen und damit auch Wissensziele zu verfolgen, da ihr Verbleib ja ohnehin nur von kurzer Dauer ist. Die Flüchtigkeit des Wissens ist ein zentrales Problem des Wissensmanagements [1, S. 7] und wird hier noch verstärkt. Außerdem wird die Festlegung und Verfolgung einer Organisationsstrategie schwierig. Gleichwohl besteht natürlich trotzdem das kollektive Interesse am Erhalt und Erfolg der Organisation, man will ja an ihr teilhaben.

1.2 Charakteristik temporärer Organisationen

Wissen geht von Periode zu Periode verloren, wenn es nicht rechtzeitig formalisiert und damit konserviert wird, da sich die Akteure nicht mehr erinnern können. Fehler werden so immer wiederholt, insgesamt ist der Lernprozess für Akteure deutlich schwieriger und langsamer im Vergleich zu einer dauerhaft ausgeführten Tätigkeit. Für alle Tätigkeiten steht ein maximales Zeitfenster zur Verfügung, während dessen diese abgeschlossen sein müssen. Zeitknappheit ist eine der höchsten Barrieren für Wissensmanagement [1, S. 310]. Im Gegensatz zu einer dauerhaften Organisation ist es nicht ohne weiteres Möglich mehr Ressourcen für eine Aufgabe bereitzustellen.

Außerdem stellt sich die Frage, wie und für wen Wissen bewahrt werden kann oder soll, wenn die Organisation verschwindet. Das gilt z.B. für Organisationen und Branchen, deren Tätigkeit eine Art allgemeines Kulturgut darstellt.

1.3 Skizze eines generischen und systemischen Models

Die Art der Modellierung ist inspiriert durch den Ansatz zur systemischen Organisationstheorie von Fritz B. Simon sowie Ideen der Lehre der Synergetik, die die Selbstorganisation komplexer Systeme beschreibt [7][8]. Teil von Simons Theorie ist, dass Organisationen in erster Linie auf ihre Selbsterhaltung bedacht sind. Dies ermöglicht eine zunächst ziellose (einziges Ziel=Selbsterhaltung) und ggf. führungslose Modellierung der Organisation. Wenn keine aktive Führung vorliegt, so muss eine Form der Selbstorganisation aktiv sein. Es ist wahrscheinlich sinnvoll, nicht gegen die Selbstorganisation zu arbeiten, sie zu nutzen, zu verstärken ggf. aber auch zu verhindern. Weitere Anregungen lassen sich in der betriebswirtschaftlichen Organisationslehre finden. So gibt es hier bereits Modelle für alternative Organisationsformen wie die virtuelle Organisation oder die Netzwerkorganisation, die ebenfalls einen temporären Charakter haben können.

Im Folgenden eine grobe Skizze, wie man eine temporäre Organisation modellieren könnte. Das Modell erhebt keinerlei Anspruch auf Korrektheit oder Vollständigkeit, sondern versucht erste Ideen zu einer Struktur zusammenzufassen.

Sei $A_0=\{a_1,\dots,a_n\}$ eine Menge von Akteuren, sei $E_0=\{e_1,\dots,e_n\}$ eine Menge von Ereignissen, sei $T_0=\{t_1,\dots,t_n\}$ eine Menge von Aufgaben, sei $O_0=\{o_1,\dots,o_n\}$ eine Menge von Handlungen, sei $R_0=\{r_1,\dots,r_n\}$ eine Menge von Regeln, sei $C_0=\{c_1,\dots,c_n\}$ eine Menge von Kommunikationsvorgängen. Sei die Zusammenfassung $ORG_0=\{A_0,E_0,T_0,O_0,R_0,C_0\}$ dieser Mengen eine temporäre Organisation für das Intervall 0, die „Startkonfiguration“ vor dem ersten Ablauf der Organisation. Sei $ORG_1=\{A_1,E_1,T_1,O_1,R_1,C_1\}$ die Organisation nach dem ersten Durchlauf und ORG_n nach dem n-ten Durchlauf. ORG_{n-1} ist ab n=1 jeweils die Startkonfiguration für das Intervall n. Änderungen die nach Abschluss von n auftreten, werden erst in n+1 realisiert. Zusätzlich besteht Unsicherheit, nicht alle Elemente der Mengen, sowie deren Eigenschaften müssen bekannt sein. Es besteht keine vollständige Information. Sei $V_i(ORG_j)$ eine Teilmenge von ORG_j, die beschreibt, wie der Akteur a_i die Organisation zum Zeitpunkt j wahrnimmt. $V_i(ORG_{j-1})$ zum Zeitpunkt j könnte die Sicht auf die Vergangenheit modellieren, $V_i(ORG_{j+1})$ Erwartungen an die Zukunft. Sei $PR_0=\{pr_1,\dots,pr_n\}$ ein Menge von Prozessen.

2. Vorgeschlagener Lösungsweg

Prozesse bzw. Handlungspläne sind ein wichtiger Aspekt in der Planung temporärer und volatiler Organisationen. Die richtige inhaltliche und zeitliche Abfolge von Handlungsschritten ist erfolgskritisch [2, S. 230]. In jeder Periode ist eine Menge von Aufgaben (T_j) zu erledigen. Die Aufgaben stehen in Relation zueinander und werden zu Prozessen (PR_j) zusammengesetzt. Jeder Akteur bekommt einen individuellen Handlungsplan. Diese sollen im Einklang mit einem idealtypischen Ablaufplan der temporären Organisation stehen. Die Pläne werden in BPMN (Business Process Modelling and Notation) [3] dargestellt. Dies hat den Vorteil, dass die Prozessbeschreibung automatisiert und standardisiert behandelt werden kann. Nachteil ist, dass BPMN alleine für die Darstellung von z.B. Erfahrungswissen nicht gut geeignet ist, da Aufbauorganisation, Daten, Strategie oder Geschäftsregeln nicht abgebildet werden können [3, S. 20]. Das ARIS-Konzept wäre eine Option, die Lücken von BPMN zu schließen. Fraglich ist und zu prüfen bleibt, ob die Architektur für den hier benötigten Zweck nicht zu komplex ist. Gesucht wird ein semi-automatisches Vorgehen, das aus unstrukturiertem Wissen, Fällen, Regeln und Ontologien ein Prozessmodell entwickelt und an die Umwelt anpasst.

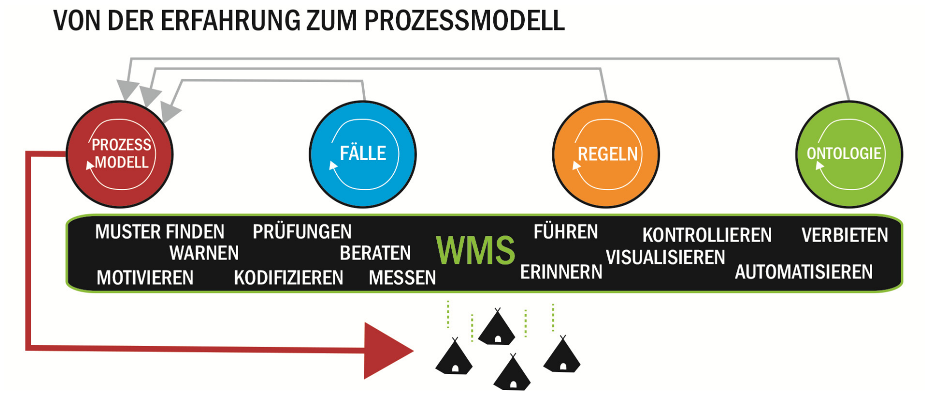


Abb. 1. Von der Erfahrung zum Prozessmodell

Die Entwicklung und Anpassung des Prozessmodells richtet sich nach Zielen. Sinnvoll erscheint es, sich hier an klassischen Organisationszielen der Betriebswirtschaftslehre zu orientieren und diese dann individuell zu gewichten. Die in Kapitel 1 beschriebenen Schwierigkeiten beachtend, ist es das grundlegende Ziel, den Bestand der Organisation nicht zu gefährden. Das bedeutet beispielsweise, dass Unfälle, illegales Handeln der Akteure oder Insolvenz vermieden werden. Weiter ist zu beachten, dass die Organisation eigene Ziele hat, jeder Akteur aber auch von individuellen Zielen getrieben ist und zweckrational handelt. Die Umwelt bewertet die Organisation ebenfalls vor dem Hintergrund allgemeinerer Ziele, Zielkonflikte sind deshalb kaum vermeidbar, ließen sich aber ggf. minimieren. Diese unterschiedliche Wahrnehmung und Bewertung der Organisation durch die Akteure a_i wird über die Sichten $V_i(ORG_j)$ realisiert.

Das Prozessmodell wird von den Akteuren durchlaufen, welche als Agenten modelliert werden. Die in Kapitel 1 beschriebenen Charakteristika von volatilen und temporären Organisationen werden in den Eigenschaften des Agentenmodells abgebildet. So könnte ein Agent z.B. die Eigenschaften: Art, Ort, Volatilität, Risikobereitschaft, Kommunikationskonto, monetäres Budget, Altruismus und Egoismus und Treue haben. Auf Grund der Verschiedenartigkeit der Akteure muss die Kommunikation über verschiedene Kanäle möglich sein. Abbildung 2. zeigt, wie eine Architektur aussehen könnte, die es ermöglicht, sich an verschiedenste Benutzer und Szenarien zu adaptieren. Templates ermöglichen je nach Art des Kommunikationskanals die geeignete Darstellung und Vermittlung der zu erledigenden Aufgaben. Der Kontext beeinflusst z.B. welche Aufgaben aktuell an Akteure vermittelt werden sollen. Durch den Zugriff auf externe Dienste und Daten können möglicherweise Lücken geschlossen werden. So ließe sich ein externer Übersetzungsdienst nutzen, um Templates an verschiedene Sprachen der Akteure anzupassen.

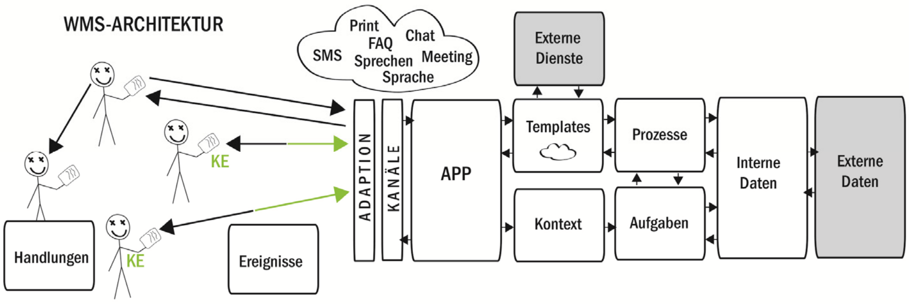


Abb. 2. WMS-Architektur

Durch einen Aufnahmetest wäre es möglich, Eigenschaften der Akteure einzugrenzen. So kann für neue Akteure festgelegt werden, welche Kommunikationskanäle bevorzugt werden und wie hoch das Maß der individuellen Fähigkeiten ist. Sind die Fähigkeiten eines Akteurs hoch, so kann z.B. ein komplexeres Prozessmodell kommuniziert oder dem Akteur die Aufgabe eines Knowledge Engineers (KE) zugeordnet werden. Analog zu den Agenteneigenschaften kann man auch Prozesselementen Eigenschaften zuordnen die volatilem und temporärem Charakter der Organisation geschuldet sind: Prozesselementeigenschaften: wann etabliert, wann geändert, wie oft geändert, Volatilität, organisationskritisch, Priorität, etc. Die Eigenschaften der Prozesselemente können dann unter Einfluss von Erfahrungswissen und Agentenverhalten angepasst werden. Für die Bewertung der Zielerfüllung lassen sich durch das Verhalten der Agenten in Abgleich mit dem Prozessmodell Metriken entwickeln, die aggregiert wiedergeben, wie nah die Gemeinschaft am idealtypischen Prozessablauf ist. Geeignete Visualisierungen geben komplexe Hintergründe an die Agenten weiter. Hinweise, auf deren Basis sich geeignete Prozessmodelländerungen ableiten lassen, werden aus den Disziplinen Handlungspsychologie [4], der Massenpsychologie [5] oder Theorien wie dem „Nudging-Prinzip" [6] entnommen. Geeignete Quellen sind sicher auch Theorien aus der neuen Institutionenökonomik wie die Principal-Agent-Theorie, die Theorie der Verfügungsrechte oder die Transaktionskostentheorie.

3. Algorithmusskizze

Im Folgenden eine erste Skizze, wie ein Wissensmanagementsystem mit den Akteuren interagieren könnte.

Initialisierung
Bei der ersten Zusammenkunft der Organisation müssen Ausgangsparameter festgelegt werden.

- *Wissensidentifikation*
    - *Handlungen und Ereignisse identifizieren*
    - *Grundlage für zeitliche Reihenfolge identifizieren*
    - *Grundlage für inhaltliche Reihenfolge identifizieren*
    - *Agenten modellieren: Startbelegung für alle Parameter ermitteln z.B. durch Befragen der Akteure*
    - *Überschneidende Handlungen (Interaktionen) identifizieren*
- *Organisationsziele / Wissensziele*
    - *Zusammenhang zwischen Zielen und Handlungen bewerten*
    - *Einfachste Strategie: Alle Akteure befragen, und die Handlungen in Reihenfolge bringen und bewerten lassen*
    - *Andere Strategien: z.B. 3 wichtigste Handlungen, bei welchen 3 Handlungen gibt es möglicherweise Probleme, vor welchen 3 Handlungen haben Sie Angst, welche Handlungen fehlen, Erfahrene Akteure werden höher gewichtet, nur die 3 erfahrensten Akteure bewerten (Knowledge Engineers), ...*

Iteration 1
erstmaliger Zusammentritt der Organisation

- *Wissenserwerb*
    - *Umweltparameter abfragen (Wetter, Feiertage, Ferien, Parallelveranstaltungen, etc.)*
    - *Externe Experten beauftragen*
- *Wissensentwicklung*
    - *Prozesskette unter Befragung der Akteure an Umweltparameter anpassen*
- ***Durchlauf der Prozesskette (Wissensverteilung, Wissensnutzung)***
- *Wissensbewahrung / Wissensbewertung*
    - *Anonyme gegenseitige Bewertung der Akteure*
    - *Erfassung, wie tatsächlich gehandelt wurde (unvollständige Information, nicht zu 100% möglich, wie viel % wurden erfasst)*
    - *Akteure fragen, wie zufrieden sie mit dem Organisationsablauf sind*
    - *Handlungswünsche für nächste Periode*

Iteration n
Mögliche Ereignisse: neuer Akteur, neue Regel, neue Erfahrung, neuer Fehler, …

- *Wissensentwicklung: Prüfen ob neue Ereignisse Prozesskette beeinflussen = Vergleich mit Regeln, Wissensbasis, Erfahrungswissen (Befragen der Akteure), Vergleich mit „älteren" Veranstaltungen ähnlichen Fall finden, etc.*
- *Wissenserwerb: Umweltparameter abfragen und Prozesskette anpassen*
- ***Durchlauf der Prozesskette (Wissensverteilung, Wissensnutzung)***
    - *Wie viel Prozent der Handlungen habe ich erfüllt*
    - *Wo stand ich in der letzten Periode zu diesem Zeitpunkt*
    - *Wo stehen andere Akteure aktuell und in der letzten Periode*
- *Wissensbewahrung / Wissensbewertung: Bewertungs- und Anpassungsvorgang für Iteration n+1*

Mögliche weitere Datenquellen: Presseartikel, Soziale Medien, Wetter, Geodaten, Bewegungsdaten der Akteure, Medizinische Daten der Akteure wie Puls, etc.

Fragestellungen sind z.B.

- Wie können Handlungen synchronisiert werden?
- Wie kann verhindert werden, dass Handlungen gleichzeitig von vielen Akteuren durchgeführt werden (Anlieferung > Verkehrsstau, Überfüllung von Plätzen, etc.)?
- Welche Kommunikationswege stehen mit welchen Vor- und Nachteilen zur Verfügung?
- Wie kann das Prozessmodell verfeinert werden?
- Wie lassen sich Zusammenhänge geeignet in BPMN modellieren?
- Wie kann man verschiedene „Erfahrungshintergründe" (Akteur, Organisation, Umwelt, …) modellieren?
- Wie lassen sich Informations- und Weisungsstrukturen abbilden?
- Wie kann die Zahl der Kommunikationskanäle reduziert, bzw. optimiert werden?

4. Beispielhafte Anwendung und Ausblick

Basis für eine Implementierung des Modells ist die langjährige Erfahrung bei der Organisation und Durchführung eines Festes mit ca. 100.000 Besuchern. Feste unterliegen einem starken demografischen Wandel, haben meist eingeschränkte Organisationsstrukturen und finden nur temporär statt. Die Akteure sind hinsichtlich ihrer Eigenschaften divers, räumlich verteilt, wechseln oft und die Zahl der Organisationsteilnehmer kann sehr groß werden. Klar ist, man will gemeinsam ein Fest veranstalten. Weitere Ziele liegen zunächst nicht vor; man möchte wirtschaftlich erfolgreich sein. Es existiert ein Regelwerk bestehend aus Gesetzen, Sicherheitsvorschriften und Vorgaben der Verwaltung. Rechtlich ist die „Organisation" als Verein konstruiert. In der Realität gibt es aber keine oder nur sehr unverbindliche Aufgabenverteilung.

Problemlösung erfolgt meist unter hohem Kommunikationsaufwand und Einbindung vieler Akteure, man will nicht selbst entscheiden. Es gibt eine Art Führungsgremium (Vorstandschaft des Vereins), jedoch mit begrenzter Weisungsbefugnis. Das Ziel der Organisationsführung ist es, einen Prozessablauf für das Fest vorzugeben, nach dem sich alle teilnehmenden Akteure richten können und diesen Prozessablauf jedes Jahr aufgrund der Erfahrungen zu verbessern. Eingehend auf Abbildung 2, würden die Mitglieder der Vorstandschaft als Knowledge Engineers tätig sein. Rollen an die sich das WMS adaptieren muss sind z.B. Mitarbeiter der Feuerwehr, der Stadtverwaltung, des kommunalen Ordnungsdienstes, der Presse, Festwirte aber auch Besucher des Festes. Jeder möchte ganz spezielle Informationen und auf unterschiedlichen Informationskanälen. Feedback von Gästen muss genauso für die nächste Periode berücksichtigt werden, wie Beschwerden von Anwohnern oder neue technische Anforderungen.

Weitere Anwendungen wären Veranstaltungen mit unabhängigen Teilnehmern bei hohem Anspruch an die Teilnehmer wie eine Regatta (z.B. Kieler Woche mit 5.000 teilnehmenden Booten und Schiffen) oder ein großer Stadtlauf (Frankfurter Iron Man). Als nächster Schritt erscheint es sinnvoll, sich auf die Entwicklung des generischen Modells zu konzentrieren. Danach kann der Algorithmus und die WMS-Architektur an das Modell angepasst und verfeinert werden. Stehen Modell, Architektur und Algorithmus kann die Systematik auf reale Welt der beispielhaften Anwendung übertragen werden.

Referenzen

1. Lehner, F.: Wissensmanagement. Hanser-Verlag, München, 2014
2. Paul, S.; Ebner, M.; Klode, K.; Sakschewski, T.: Sicherheitskonzepte für Veranstaltungen. DIN, Beuth Verlag GmbH, Berlin, 2014
3. Freund, J.; Rücker, B.: Praxishandbuch BPMN 2.0. Hanser, 2012
4. Kaiser, H. J.; Werbik, H., Handlungspsychologie. Eine Einführung, UTB GmbH, 2012
5. Keith Still, G.: Introduction to Crowd Science. CRC Press Taylor & Francis Group, 2013
6. Thaler, R.; Sunstein, C.: nudge – Improving decisions about health, wealth and happiness. Penguin Books, 2008
7. Simon, F.: Einführung in die systemische Organisationstheorie. Carl-Auer, 2015
8. Haken, H.: Die Selbstorganisation komplexer Systeme – Ergebnisse aus der Werkstatt der Chaostheorie. Picus, 2013

Towards cross-layer monitoring of cloud workflows

Eric Kübler and Mirjam Minor

Wirtschaftsinformatik, Goethe University, Robert-Mayer-Str.10, Frankfurt am Main, Germany,
{ekuebler, minor}@informatik.uni-frankfurt.de

Abstract. Prospective cloud management requires sophisticated monitoring capabilities. In this paper, we introduce a novel monitoring framework for cloud-based workflow systems called cWorkload. cWorkload integrates monitoring information from different layers of the cloud architecture. The paper puts its focus on the two-layer monitoring regarding the workflow layer and the PaaS layer. We present the layered monitoring architecture, an implementation of the two-layer cross-monitoring part, and an experimental evaluation with sample workflow data. Further, we discuss related work on cloud monitoring divided into one-layer, multi-layer, and cross-layer approaches. Our plans for future work on extending the implementation by further layers towards a cross-layer, prospective monitoring for prospective cloud management are described. The original version of this re-submission has been published at CLOSER 2015 [Kübler and Minor, 2015].

Keywords: Cloud Management, Cloud Monitoring, Workflow Management, Case-Based Reasoning

References

[Kübler and Minor, 2015] Kübler, E. and Minor, M. (2015). Towards cross-layer monitoring of cloud workflows. In *Proceedings of the 5th International Conference on Cloud Computing and Services Science (Accepted for publication)*, pages 389 – 396, Lisbon, Portugal. SciTePress.

Towards Context-aware Technical Service

Alexander Legler[1] and Joachim Baumeister[1,2]

[1]denkbares GmbH, Friedrich-Bergius-Ring 15, 97076 Würzburg, Germany
[2]University of Würzburg, Institute of Computer Science,
Am Hubland, 97076 Würzburg, Germany
{alexander.legler,joachim.baumeister}@denkbares.com

Abstract. Context-aware systems have long found application in everyday use cases, assisting users with their daily lives. Technical service covers any tasks concerning the maintenance, diagnosis, and repair of industrial machinery. It is a more specific domain that would also benefit from the introduction of context-aware systems. This domain requires the filtering and consumption of a vast amount of information resources. Employing semantic technologies enables engineers to more precisely find information as compared to full-text search. However, it still requires a search query to be actively formulated to the system. This paper applies the principles of context-aware systems to information systems for the technical service, where the technician is guided though the service process influenced by various sensors defining their current context. An implementation based on an established ontology for context-aware systems is presented that integrates with semantically enriched documentation.

Keywords: Context-aware Computing, Ontology Engineering, Decision Support

1 Introduction

In the technical domain, the trouble-shooting and maintenance of advanced machinery is a complex task. Technical documentation describes the service-related tasks for these machines and typically comprises some thousand pages of information for a single machine. Consequently finding relevant information bits for a specific fault is difficult and time-consuming.

In the last years, many semantic information systems were introduced in the technical domain to support technicians during service tasks [3]. Semantic information systems add ontological knowledge to the information bits included in standard information systems. In advance to full-text search, such semantic systems introduce semantic search [4], where the retrieval of information is based on semantic queries. Due to the unambiguous query statement, the research time for finding the relevant information is reduced dramatically. Nevertheless, the amount of information is overwhelming in many cases.

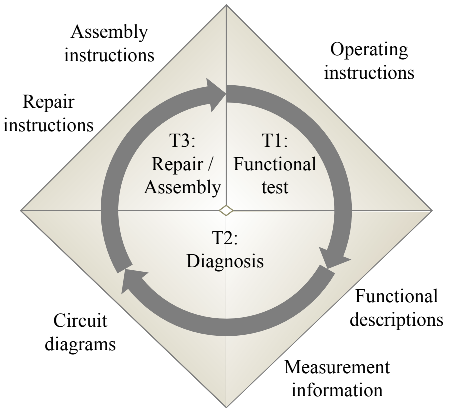


Fig. 1. The typical tasks of a service technician are shown in the center and the relevant information bits in the surrounding space.

In this paper, we propose the extension of semantic information systems by context-aware techniques, that provide the users with relevant information for their current tasks. For example, a service technician in a trouble-shooting task is unlikely to be interested in maintenance information when formulating a query. More generally, by knowing and tracing the context of a working engineer and their corresponding use of technical documentation, the system will be able to provide more relevant information. In Figure 1 the typical trouble-shooting workflow of a service technician is depicted. Essentially, the workflow is partitioned into three sub-tasks: 1. The functional test assures that the failure is actually present, 2. The diagnosis aims to find the cause of the failure, and 3. The repair and assembly fixes the failure. In every task, the service technician has different information needs, i.e. is interested in different types of documentation. The most common documentation types are depicted along the edge of the figure. For example, during the task *functional test* the service technician needs the operating instructions in order to know how the failing function is operated properly. Context-awareness tries to guess the actual task of the technician and to recommend the best-fitting information for the current situation.

The paper is organized as follows: Context-awareness is based on the interpretation of sensors. Thus, we first describe specific sensors with respect to technical service scenarios. For an implementation the context-awareness needs to be represented within the semantic system. Therefore, we introduce an ontological representation and show its application in a selected case study. The paper is concluded with a summary and planned future work.

2 Context-aware Systems in Technical Service

Modeling a specialized application domain such as technical service in a context-based system requires the use of various sources of information. In a context-based system, this information is provided by *sensors* that can be physical, but in this case mostly are *virtual* and *logical* (following the definition in [1]), i.e. providing data from software systems and combining data from other sensors. This is the case as data from a physical sensor, predominantly the location, only affects few environmental parameters and not the machine's overall condition on which the focus lies. Figure 2 shows the main entities (machine, engineer, service case, and location) and their properties that we propose to be described by sensor information.

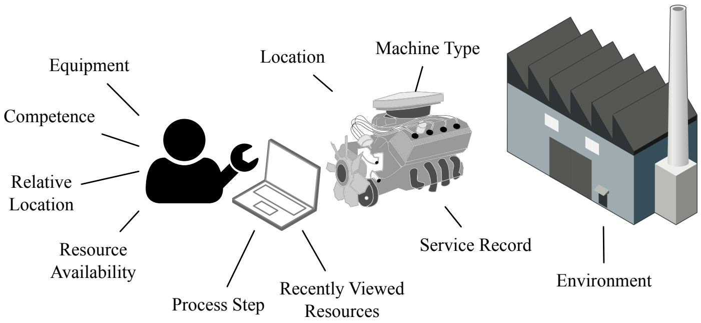


Fig. 2. Additional sensors providing information about an engineer's context while servicing a machine.

The following list describes potentially useful sensors for technical service:

Machine type The most essential sensor captures the exact model and equipment status of the machine that is being worked on. Functions are often provided by different components within a model range, each of which having their own maintenance procedures and documentation. Misidentifying the specific component setup can increase service turnover times or even damage equipment.
There are several feasible implementations: a physical sensor beacon on the machine can transmit its type or a virtual sensor can provide it as manually set information.

Machine service record As the equipment status as provided by the first sensor can change over time, it is useful to provide the service record and part changes as well. This information is also relevant when faults in machines reappear after a period of time. Previous repair attempts can be factored into the support process to directly suggest a remedy or exclude it if it can not fix a fault permanently.

Machine-relative location One of the most common sensor types in everyday use of context-based systems is the location sensor, using AGPS to determine a person's position with an accuracy of up to a few meters.

Location information is also valuable in the target domain, but required on a finer scale. Its use becomes evident when dealing with machines that exceed a certain installed size as it for instance is the case with offset printing machines. Knowing the module at which the engineer currently is located at enables a context-based information system to narrow down the relevant documentation to that specific part.

Engineer equipment Another factor to consider is the equipment available to the engineer. This includes both the available tools in the engineer's toolkit as well as the devices they have available to consume documentation: augmented or virtual reality displays and expert systems may not be available on all device types or require special gear.

Using this information enables a timely detection of faults that are not remediable with the currently available material and improves clarity in the software system by hiding information that can not be displayed.

Information and resource availability The applicability of documentation items is further determined by the resources at the engineer's disposal. Most importantly, it needs to be determined if the engineer has Internet access to reach further materials on a company network. For problems requiring in-depth analysis and triage, the ability to contact off-site support staff may be required as well.

Engineer competence level Given the ever increasing complexity of modern appliances, training engineers is expensive, both financially and in terms of time consumption. The context can factor in the engineer's competence level to offer additional guidance for lesser experienced engineers while not disturbing the workflow of seasoned mechanics with basic knowledge. Additionally, the system can sense when a procedure is potentially unsafe if performed by untrained staff.

Step in the service process This logical sensor captures the step the engineer is currently working on to influence the choice of documentation provided. As service processes are usually provided by the manufacturer and to be followed in a specific order, the position in the overall process can be determined.

Consideration should be given to the level of detail used for modeling the process. The inclusion of atomic steps like removing a screw would incur unnecessary modeling complexity.

History of viewed documents Together with the current process step, the previously used information within this task is an valuable sensor. The already consumed information spans the knowledge and status of the technician and can also be used to deduce the next steps in the service process.

Environment at the repair site Much like the *engineer equipment* sensor provides information about the resources made available by the engineer, this sensor describes the environment at the work site. This information is important as the environment can be vastly different when working in a specialized workshop or on-site at the customer's premises. The latter location will most likely not have specialized equipment for advanced repair scenarios.

3 Ontological Representation of Context Awareness

There are various instances of existing ontological context models, such as the ontologies COBRA-ONT [2] and CONON [9]. In the context of this work, we use CONON as the base ontology due to its simplicity that facilitates ontology reuse and the fact that no other specific domains are included that are of no use for technical service. CONON provides a minimal upper ontology that can be easily extended by domain-specific ontologies, as shown in excerpts in Figure 3. Its root class `ContextEntity` is extended by four general concepts: `CompEntity` (computational entity), `Location`, `Person`, and `Activity`. The list of pre-defined computational entities (not shown) includes `Service`, `Application`, `Device`, `Network`, and `Agent`. Locations are further distinguished between indoor and outdoor places, and activities can either be scheduled, or deduced from the other context sensors (not shown).

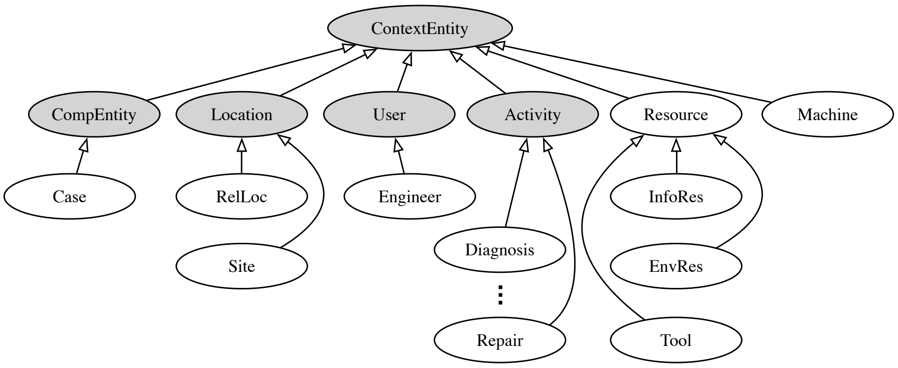


Fig. 3. The CONON upper-ontology (grey, in excerpts) and extensions for the technical service domain.

In the outlined technical service scenario, most classes are intuitively reusable: We extend `Person` with an `Engineer` class representing the technician working on a machine. An instance of this class will have several properties for identification (using SKOS' `skos:prefLabel` [8] or FOAF's `foaf:name`) as well as indications of their training status (`competenceLevel`) and information about provided resources, i.e. the tools they currently carry (`providesResource`).

To be able to further model the technical service domain, we also employ a few other entity classes, directly sub-classing `ContextEntity`. First, a `Resource` is defined as a resource that is available to or required by the engineer. Such a resource can be any kind of information (`InformationResource`), a `Tool`, or an `EnvironmentalResource`. Examples for information resources could be documentation (like sections of a manual, schematics, or wiring diagrams), expert systems, or the possibility to contact other support staff for further consultation. While tools are items contained in the engineer's toolkit, environmental resources are to be provided at the service location (service lift, expensive diagnostic utilities). The `Location` class provided by CONON is used to model the machine's location as well as the engineer's relative location. Its `Site` sub-class is instantiated for each work site to set `providesResource` properties to denote available resources. The other sub-class, `RelativeLocation`, is to be used to capture the current machine-relative location of the engineer. Finally, a `Case` class which is added as a computational entity represents a service case linking engineer, location, and machine. It also contains information on the current state in the process, modeled as instances of CONON's `Activity` class. We define a set of activities representing the service process: `FunctionalTest`, `Maintenance`, `Diagnosis`, `Repair`, etc.

Given the `tso` namespace (technical service ontology) and `ns` for the target application ontology, an exemplary minimal scenario could be as follows:

```
ns:SmallToolkit a tso:Tool .
ns:ServiceLift a tso:EnvironmentalResource .
ns:Machine_1 a tso:Machine .

ns:Engineer_1 a tso:Engineer ;
  tso:competenceLevel 4 ; tso:providesResource tso:SmallToolkit .

ns:Workshop_1 a tso:Site ; tso:providesResource ns:ServiceLift .

ns:Case_1 a tso:Case ;
  tso:locatedAt ns:Workshop_1 ; tso:servicedBy ns:Engineer_1 ;
  tso:machine ns:Machine_1 ; tso:currentStep tso:Diagnosis .
```

An Engineer (`Engineer_1`) has access to the `SmallToolkit` resource. Their competence level in this case is modeled as an integer and at level 4. The service case (`Case_1`) takes place at `Workshop_1` which provides a `ServiceLift` to work on `Machine_1`.

4 Implementation Example

To test the modeled ontology, we employ a small and understandable technical domain, in this case bicycles. We use the namespace prefix of `bts` (short for bike technical service) for framing the concepts of this domain. The example encompasses several bicycle models and contains information about repair steps and resources fulfilling the engineer's information needs while performing them.

We develop a demonstration application for the use by the service technician in the field. It has access to the ontology in order to read and write the context state and access the contained information resources. After setting initial parameters for the case, the application knows what process the engineer is about to begin, as an example diagnosis of faulty headlights on a bike. Initially, the `currentStep` property of the `Case` instance is `FunctionalTest` (c.f. Figure 4).

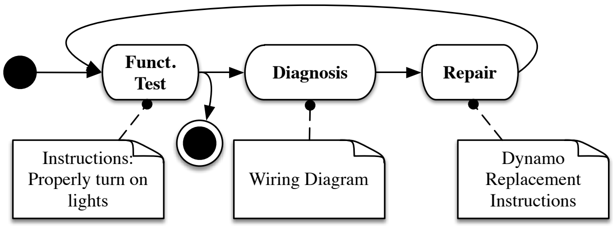


Fig. 4. A simple service process with associated information resources.

For every step of the process, the application queries the ontology of the semantic information system for relevant information, depending on the current context state. This task is performed by the semantic search engine as outlined in the introductory chapter. Context information additionally influences the results, for instance the engineer's competence level should be taken into consideration to ensure they can proficiently perform the actions suggested by the retrieved information resources. This functionality can be implemented using a SPARQL query that yields only resources matching `Engineer_1`'s level of competence for instance by using its `FILTER` functionality.

After reviewing the usage documentation, the current step changes, and so does the context. In the rest of the process, relevant resources for the next steps (diagnosis and repair) are retrieved, until a second functional test results in a working lighting system.

5 Conclusion and Further Work

In this paper, we motivated the introduction of context-based systems into the technical service domain. We proposed a basic ontology that provides a framework for modeling service steps and entities involved in the process of servicing industrial appliances. It can be easily integrated into semantic information systems that use ontologies to represent their semantically enriched documentation as well. In combination, such an application can be used to provide precise information to technicians without the need to manually invoke search operations.

Related work can be found both in different domains as well as system types: The application of context-aware information systems was proposed for instance in the medical domain [5]. The authors also present a specialized approach, like this paper does to allow for the integration of semantic search and domain-specific information sources. Reuss et al. [7] also discuss the application of case-based agents as a form of automated diagnosis support. While this is a different system type, we could envision the combined usage of such a tool and the system we propose in this paper as they both profit from context-awareness to reach the same goal.

The focus of our approach lies in ontology reuse and alignment. Based on an existing lean upper-ontology, we in turn implement a flexible domain-specific layer. Future work will keep this aspect in mind: Modeling the remaining sensor types such as machine history can be done by aligning the established PROV ontology [6]. Further research will be done on the user interface and applicable sensor types, including a survey of other specialized domain sensors that could be adopted. We expect to provide case studies with more complex appliances in the field of agricultural machines and explore reasoning strategies for the resulting, more complex models.

Acknowledgements

The work described in this paper is supported by the Bundesministerium für Wirtschaft und Energie (BMWi) under the grant ZIM KF2959903BZ4 "Mobile, Sprach- und MR-gestützte Dokumentation und Diagnose im Technischen Service (Service Troi)".

References

1. Baldauf, M., Dustdar, S., Rosenberg, F.: A survey on context-aware systems. IJAHUC 2(4), 263–277 (2007)
2. Chen, H., Finin, T., Joshi, A.: An ontology for context-aware pervasive computing environments. Knowledge Eng. Review 18(3), 197–207 (2003)
3. Elst, L., Abecker, A.: Ontologies for knowledge management. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies. pp. 435–454. Springer, Heidelberg (2004)
4. Guha, R., McCool, R., Miller, E.: Semantic search. In: Twelfth International World Wide Web Conference (WWW 2003) (2003)
5. Jahnke, J.H., Bychkov, Y., Dahlem, D., Kawasme, L.: Context-aware information services for health care. In: Proceedings Modeling and Retrieval of Context. Ulm, Germany (2004)
6. Moreau, L., Groth, P.: Provenance: An Introduction to PROV. Synthesis Lectures on the Semantic Web: Theory and Technology, Morgan and Claypool (2013)
7. Reuss, P., Hundt, A., Althoff, K.D., Henkel, W., Pfeiffer, M.: Case-based agents within the omaha project. In: Vattam, S., Aha, D.W. (eds.) Case-based Agents. ICCBR (2014)
8. W3C: SKOS Simple Knowledge Organization System reference: `http://www.w3.org/TR/skos-reference` (August 2009)
9. Wang, X., Zhang, D., Gu, T., Pung, H.K.: Ontology based context modeling and reasoning using OWL. In: 2nd IEEE Conference on Pervasive Computing and Communications Workshops (PerCom 2004 Workshops), 14-17 March 2004, Orlando, FL, USA. pp. 18–22. IEEE Computer Society (2004)

Importing the OEIS library into OMDoc

Enxhell Luzhnica, Mihnea Iancu, Michael Kohlhase

Computer Science, Jacobs University, Bremen, Germany
`initial.last@jacobs-university.de`

Abstract. The On-line Encyclopedia of Integer Sequences (OEIS) is the largest database of its kind and an important resource for mathematicians. The database is well-structured and rich in mathematical content but is informal in nature so knowledge management services are not directly applicable.
In this paper we provide a partial parser for the OEIS that leverages the fact that, in practice, the syntax used in its formulas is fairly regular. Then, we import the result into OMDoc to make the OEIS accessible to OMDoc-based knowledge management applications. We exemplify this with a formula search application based on the MathWebSearch system.

1 Introduction

Integer sequences are important mathematical objects that appear in many areas of mathematics and science and are studied in their own right. The On-line Encyclopedia of Integer Sequences (OEIS) [6] is a publicly accessible, searchable database documenting such sequences and collecting knowledge about them. The effort was started in 1964 by N. J. A. Sloane and led to a book [10] describing 2372 sequences which was later extended to over 5000 in [11]. The online version [8] started in 1994 and currently contains over 250000 documents from thousands of contributors with 15000 new entries being added each year [9]. Documents contain varied information about each sequence such as the beginning of the sequence, its name or description, formulas describing it, or computer programs in various languages for generating it.

The OEIS library is an important resource for mathematicians. It helps to identify and reference sequences encountered in their work and there are currently over 4000 books and articles that reference it. Sequences can be looked up using a text-based search functionality that OEIS provides, most notably by giving the name (e.g. "Fibonacci") or starting values (e.g. "$1, 2, 3, 5, 8, 13, 21$"). However, given that the source documents describing the sequences are mostly informal text, more semantic methods of knowledge management and information retrieval are limited.

In this paper we tackle this problem by building a (partial) parser for the source documents and importing the OEIS library into the OMDoc/MMT format which is designed for better machine support and interoperability. This opens up the OEIS library to

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

OMDoc-based knowledge management applications, which we exemplify by a semantic search application based on the MathWebSearch [4] system that permits searching for text and formulas.

This paper is organized as follows: in Section 2 we describe our import of the OEIS library into OMDoc. In Section 3 we show an initial application of our import by providing formula search for the OEIS library. Then, in Section 4 we discuss future work and conclude.

Acknowledgements This work has been supported by the German Research Foundation (DFG) under grant KO 2428/13-1. The authors gratefully acknowledge the foresight of the OEIS foundation to license the OEIS content under a Creative Commons license that allows derivative work like this one and the practical help of Jörg Arndt in obtaining the OEIS corpus.

2 Translating OEIS to OMDoc

The OEIS database is stored as a collection of text documents (one for each sequence) written in the *internal format* of OEIS which defines the document-level structure of the sources. Therefore, parsing the *document structure* is straightforward. However, at the formula level the format is not standardized which makes parsing them non-trivial. Still, in practice, the syntax used in the formula snippets is somewhat regular and we built a *formula parser* that succeeds on most OEIS formulas.

2.1 Preliminaries

OMDoc [7] is a content markup format and data model for mathematical documents. It models mathematical content using three levels.

Object Level: uses OpenMath and MathML as established standards for the markup of *formulae*.

Statement Level: supplies original markup for explicitly representing the various kinds of mathematical statements including *symbol declarations* and *definitions* (which introduce a new symbol names), *assertions* (which can represent theorems, lemmas or corollaries), and *examples*.

Theory Level: offers original markup that allows for clustering sets of statements into *theories* as well as specifying relations between them (inclusions, morphisms).

The Mmt [14] language can be seen as a restricted version of OMDoc but with a fully specified semantics. Additionally, for Mmt there is an Mmt system [13] which is a Scala-based [2] open source implementation of the Mmt language. The key features of the Mmt system for this paper are that it provides an infrastructure for writing importers from any compatible format into Mmt as well as exporters from Mmt, most notably into (MathML-enriched) HTML for both local and web-based presentation.

For the sake of simplicity, we often do not differentiate between Mmt and OMDoc languages in the following and refer to [7] and, respectively, [14] for details on each language. Throughout this paper we will use OMDoc/MMT to refer to both OMDoc and Mmt.

2.2 The OEIS document format

The *internal format* [12] is line-based in the sense that each line starts with a marker that represents the kind of content found in that line. We briefly introduce the relevant kinds below but refer to [12] for details.

The *identification* line gives the unique ID of the sequence declared in that document and the *name* line gives the name, a brief description or the definition of the sequence. There are also *start values* lines which give the beginning of the sequence. *Formula* lines give formulas that define or hold for the sequence described in the current document. The formulas are in plain text ASCII syntax that is similar to LaTeX math markup and can contain text as part of the formula or as comments. There are many other dedicated line types including those for implementations (in various programming languages), references, examples, or comments.

Running Example 1 (Fibonacci numbers). The article for Fibonacci numbers [5] was one of the first entries in the OEIS and is one of the most comprehensive. We will use it as a running example throughout the paper, although we will heavily trim the document for simplicity by presenting here only a few sanitized lines. Listing 1 shows the document with identification, values, name and reference lines, followed by three formula lines and the author line.

```
%I A000045 M0692 N0256
%S A000045 0,1,1,2,3,5,8,13,21,34,55,89,144,233,377,610,987
%N A000045 Fibonacci numbers: F(n) = F(n-1) + F(n-2) with F
    (0) = 0 and F(1) = 1.
%D A000045 V. E. Hoggatt, Jr., Fibonacci and Lucas Numbers.
    Houghton, Boston, MA, 1969.
%F A000045 F(n) = ((1+sqrt(5))^n-(1-sqrt(5))^n)/(2^n*sqrt(5))
%F A000045 G.f.: Sum_{n>=0} x^n * Product_{k=1..n} (k + x)/(1
     + k*x). - _Paul D. Hanna_, Oct 26 2013
%F A000045 This is a divisibility sequence; that is, if n
    divides m, then a(n) divides a(m)
%A A000045 _N. J. A. Sloane_, Apr 30 1991
```

2.3 Parsing the OEIS Formula Format

We built a partial parser for OEIS formulas by identifying and analyzing well-behaved formulas to produce a workable grammar. We leverage the fact that, although there is no standardized format for OEIS formulas, many of them use a sufficiently regular syntax.

OEIS is known for the human-readable mathematical terms, so a variety of syntactic rules are encountered when forming these mathematical terms. We use the following classification for notations, inspired by [1] and further motivated by our analysis of the OEIS formulas:

1. **infix operators** are used to combine two terms to one complex term, e.g the + symbol in `m+n`.
2. **suffix operators** are added after a term to form another term, e.g. the `!` symbol in `n!`.

3. **prefix operators** (with or without bracketed application) are added in front of a term to form another term, e.g. `sin` in `sin(x)` or `sin x`, respectively.
4. **infix relation symbols** are used to construct a formula out of two terms, e.g. the `<` symbol in `x<2`.
5. **binding operators** that bind a context to a body to construct a term, e.g. the $\forall$ symbol in $\forall$`x. x^2 > 0`

The classification presented above guides our grammar and, in principle, covers virtually all important notations used in OEIS formulas. However, in practice, we encountered several important challenges which we discuss individually below.

Open Set of Primitives Since the formulas are not standardized, not only is the syntax flexible, but so is the set of primitive operators that are used. For instance, the formulas in Listing 1 (on lines 5-6) use square root, power, as well as the sum (Σ) and product (Π) binders. The challenges arise because of the many different notations used for such primitives. For instance, in line 6 of Listing 1 the range for sum and product is given in two different ways. Similar problems appear with limits and integrals as well as numerous atypical infix and suffix operators. In order to parse these correctly, we investigate the documents and the grammar failures manually and incrementally extend the grammar.

Ambiguity As it is often the case with informal, presentation-oriented formulas, there can be ambiguity in the parsing process when there exist several reasonable interpretations. Since the OEIS syntax is not fixed, this is quite common, so we do additional disambiguation during parsing to resolve most of the ambiguities. Here we discuss a few of the many ambiguities that arise.

The multiplication sign is usually implicit so, instead of `a*(x+y)`, we encounter `a(x +y)` which could represent either a function application or a multiplication depending on whether `a` is a function or an individual (constant or variable). There is no general way to solve this, so we rely on several heuristics. First, we check if the symbol in question is used somewhere else in the same formula with an unambiguous meaning. Specifically, we default to function application unless the same symbol is used as an individual somewhere else in the formula. This already disambiguates most such cases in OEIS but we use several additional heuristics. For instance, having `name(`arg`)` will result in marking `name` as function since it is unlikely to be a multiplication between two individuals. Similarly, having `name(`arg_1`,...,`arg_n`)` results in marking `name` as a function.

The natural way of using the power operator also leads to ambiguities. For example, `T^2(y)` is used for $(T(y))^2$, however `T^y(x^2+2)` is ambiguous. We solve this using similar heuristics as for the implicit multiplication.

For unbracketed function application as in `sin x`, we rely on the heuristic that this form of function application is used only in well known functions. Therefore, we code these notations for well known functions in the grammar itself. This form of function application can also mean multiplication, for instance `Pi x`. One can already see that parsing and disambiguating the mathematical expressions in this context has a lot of aspects. Additional cases of ambiguities are handled in similar ways and we omit the details for brevity.

Delineating formulas OEIS formula lines freely mix text and formulas so it is required to correctly distinguish between text and formula parts within the lines in order to accurately parse each line. For instance, line 6 in Listing 1 starts with the text `G.f.:` (meaning "Generating function:") and continues with the formula. The line then has the author and date, separated from the formula by a dash (`-`) which could also be interpreted as a minus and, therefore, a continuation of the formula. In the extraction of the formulas we use the help of a dictionary. The text in the OEIS documents has words that are not found in the dictionaries since it contains many technical terms so we first run a pre-parsing procedure which enriches the dictionary. The final grammar tries to parse words until it fails and then tries to parse formulas; this process repeats.

2.4 Importing into OMDoc/MMT

For each OEIS document we create a corresponding OMDoc/MMT document that contains a single theory. Then, OEIS lines roughly correspond to OMDoc/MMT declarations inside that theory. We use the OEIS sequence ID as the name of the OMDoc theory. Then, the identification line produces an OMDoc symbol declaration representing the sequence (as a function from integers to values). The start values and example lines are both represented as OMDoc/MMT examples. Specifically, the starting values are considered as examples of sequence elements. Formula lines are represented as OMDoc assertions (about the sequence symbol). Finally, name, reference and author lines are represented as metadata using the Dublin Core standard.

Running Example 2 (Fibonacci Numbers). The corresponding OMDoc/MMT document for the Fibonacci numbers article from Example 1 is shown in Listing 2. We omit most formulas and some XML boilerplate for conciseness and simplicity.

```
<omdoc xmlns:dc="http://purl.org/dc/elements/1.1/">
  <theory id="A000045">
  <metadata>
    <dc:creator>N. J. A. Sloane</dc:creator>
    <dc:title>Fibonacci numbers</dc:title>
  </metadata>
  <symbol name="seq"/>
  <assertion>
    <!-- OpenMath for $\forall n.seq(n) = \frac{(1+\sqrt{5})^n-(1-\sqrt{5})^n}{2^n\sqrt{5}}$ -->
    <OMBIND>
      <OMS cd="arith" name="forall"/>
      <OMBVAR> <OMV name="n"/> </OMBVAR>
      <OMA>
        <OMS cd="arith" name="equal"/>
        <OMA><OMS name="seq"/><OMV name="n"></OMA>
        ⋮
      </OMA>
    </OMBIND>
  </assertion>
  ⋮
```

```
    </theory>
  </omdoc>
```

2.5 Implementation and Evaluation

The importer is implemented in Scala as an extension for the MMT system and consists of about 2000 lines of code. It is available at `https://svn.kwarc.info/repos/MMT/src/mmt-oeis/`. The implementation is mostly straightforward, other than the formula parser which we discuss separately below.

There are 257654 documents in OEIS totaling over 280MB of data. The OMDoc/MMT import expands it to around 9GB, partly due to the verbosity of XML and partly due to producing the semantic representation of formulas. The total running time is around 1h40m using an Intel Core i5, 16GB of RAM and a SATA hard drive.

Formula parsing The formula parser is implemented using the Packrat Parser [3] for which Scala provides a standard implementation. Packrat parsers allow us to write left recursive grammars while guaranteeing a linear time worst case which is important for scaling to the OEIS.

There are 223866 formula lines in OEIS and the formula parser succeeds on 201384 (or 90%) of them. Out of that, 196515 (or 97.6%) contain mathematical expressions. Based on a manual inspection of selected formulas we determined that most parser fails occur because of logical connectives since those are not yet supported. Other failures include wrong formula delineation because of unusual mix of formulas and text.

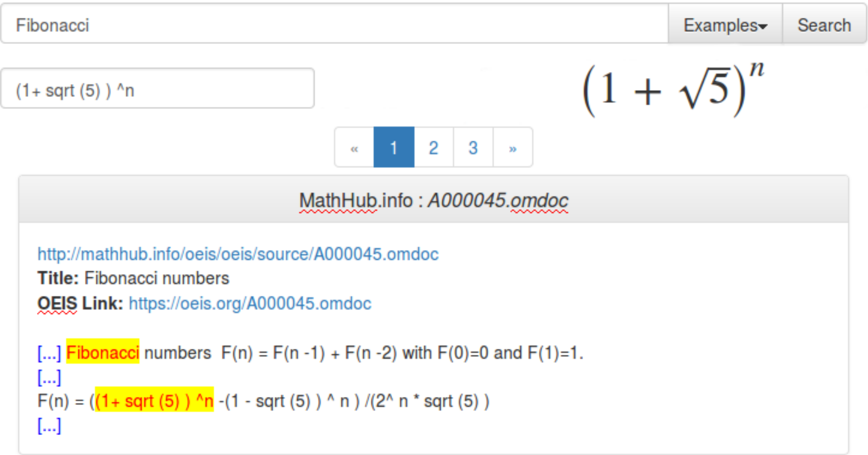
The statistics above refer just to the successful parses, but we cannot automatically evaluate if the result returned by the parser is actually the expected one. For this, we did a manual evaluation of the parsing result for 40 randomly selected OEIS documents and evaluated 85% of succesfully parsed formulas as semantically correct. The main contributor of incorrect formula parses was badly delineated formulas, which causes text to be wrongly parsed as part of a formula.

3 Application: Search

MathWebSearch (MWS)[4] is an open-source, open-format, content-oriented search engine for mathematical expressions. We refer to [4] for details.

To realize the search instance in MWS we need to provide two things:

1. A *harvest* of MathML-enriched HTML files that the search system can resolve queries against. The content-MathML from the files will be used to resolve the formula part of the query while the rest of the HTML will be used for the text part. The harvest additionally requires a configuration file that defines the location in the HTML files of MWS-relevant metadata such as the title, author or URL of the original article. This, together with the HTML itself is used when presenting the query results.
2. A *formula converter* that converts a text-based formula format into MathML. This will be used so that we can input formulas for searching in a text format (in our case OEIS-inspired ASCII math syntax) rather than writing MathML directly.



MathHub.info : *A000045.omdoc*

http://mathhub.info/oeis/oeis/source/A000045.omdoc
Title: Fibonacci numbers
OEIS Link: https://oeis.org/A000045.omdoc

[...] Fibonacci numbers F(n) = F(n -1) + F(n -2) with F(0)=0 and F(1)=1.
[...]
F(n) = ((1+ sqrt (5)) ^n -(1 - sqrt (5)) ^ n) /(2^ n * sqrt (5))
[...]

Fig. 1: Text and Formula Search for OEIS

To produce the harvest of the OEIS library for MWS we export the HTML from the content imported into MMT. We reuse the MMT presentation framework and only enhance it with OEIS-specific technicalities such as sequence name or OEIS link. For the formula converter we use the same parser used for OEIS formulas and described above, except extended with one grammar rule for MWS *query variables*. We then forward the resulting formula in MMT to produce the presentation (MATHML) and return it to the MWS frontend. The web-server infrastructure, needed to communicate with MWS, is provided by MMT and we just extend it. Figure 1 shows (a part of) the current interface answering a query about Fibonacci numbers. The search system is available at `http://ash.eecs.jacobs-university.de:9999/`.

4 Conclusion and Future Work

We presented a partial parser for the On-line Encyclopedia of Integer Sequences that covers the majority of formulas and an import of the parsed OEIS into OMDOC. We exemplified the added value by providing a formula-search service for the OEIS based on the MATHWEBSEARCH system. Our importer does not currently handle all line types in OEIS, most notably the program code lines. We also only analyze formulas that appear inside formula lines, but in OEIS they may appear elsewhere (for instance instead of the sequence name or inside comment lines). In the future, we plan to extend the structure parser to cover these cases as well as improve the formula parser to handle some of the failures discussed in Section 2. Moreover, since we have the defining formulas in their content representations for a significant number of sequences, we plan to analyze them to try and find additional relations between sequences as well as generate new ones.

References

[1] Marcos Cramer, Peter Koepke, and Bernhard Schröder. “Parsing and Disambiguation of Symbolic Mathematics in the Naproche System”. In: *Intelligent Computer Mathematics - 18th Symposium, Calculemus 2011, and 10th International Conference, MKM 2011, Bertinoro, Italy, July 18-23, 2011. Proceedings*. Ed. by James H. Davenport et al. Vol. 6824. Lecture Notes in Computer Science. Springer, 2011, pp. 180–195. DOI: 10.1007/978-3-642-22673-1_13. URL: http://dx.doi.org/10.1007/978-3-642-22673-1_13.

[2] École polytechnique fédérale de Lausanne. *The Scala Programming Language*. URL: http://www.scala-lang.org (visited on 10/22/2009).

[3] Bryan Ford. “Packrat Parsing:: Simple, Powerful, Lazy, Linear Time, Functional Pearl”. In: *Proceedings of the Seventh ACM SIGPLAN International Conference on Functional Programming*. ICFP ’02. Pittsburgh, PA, USA: ACM, 2002, pp. 36–47. DOI: 10.1145/581478.581483. URL: http://doi.acm.org/10.1145/581478.581483.

[4] Radu Hambasan, Michael Kohlhase, and Corneliu Prodescu. “MathWebSearch at NTCIR-11”. In: *NTCIR 11 Conference*. Ed. by Noriko Kando and Hideo Joho andKazuaki Kishida. Tokyo, Japan: NII, Tokyo, 2014, pp. 114–119. URL: http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings11/pdf/NTCIR/Math-2/05-NTCIR11-MATH-HambasanR.pdf.

[5] The OEIS Foundation Inc. *Fibonacci Numbers, The On-Line Encyclopedia of Integer Sequences*. http://oeis.org/A000045. 2015.

[6] The OEIS Foundation Inc. *The On-Line Encyclopedia of Integer Sequences*. http://oeis.org/. 2015.

[7] Michael Kohlhase. OMDoc – *An open markup format for mathematical documents [Version 1.2]*. LNAI 4180. Springer Verlag, Aug. 2006. URL: http://omdoc.org/pubs/omdoc1.2.pdf.

[8] N. J. A. Sloane. *An On-Line Version of the Encyclopedia of Integer Sequences*. http://www3.combinatorics.org/Volume_1/PDF/v1i1f1.pdf. 1994.

[9] N. J. A. Sloane. *The On-Line Encyclopedia of Integer Sequences*. http://neilsloane.com/doc/eger.pdf. 2012.

[10] N.J. A. Sloane. *A Handbook of Integer Sequences*. Academic Press, 1973.

[11] N.J. A. Sloane and Simon Plouffe. *The Encyclopedia of Integer Sequences*. Academic Press, 1995.

[12] *OEIS Help*. http://oeis.org/eishelp1.html.

[13] Florian Rabe. “The MMT API: A Generic MKM System”. In: *Intelligent Computer Mathematics*. Conferences on Intelligent Computer Mathematics. (Bath, UK, July 8–12, 2013). Ed. by Jacques Carette et al. Lecture Notes in Computer Science 7961. Springer, 2013, pp. 339–343. DOI: 10.1007/978-3-642-39320-4.

[14] Florian Rabe and Michael Kohlhase. “A Scalable Module System”. In: *Information & Computation* 0.230 (2013), pp. 1–54. URL: http://kwarc.info/frabe/Research/mmt.pdf.

Entwicklung einer Balanced-Scorecard zur Nutzenbewertung eines Lehr-Lern-Portals für die wissenschaftliche Weiterbildung

Dirk Stamer, Kurt Sandkuhl, Ulrike Borchardt, Felix Timm

Universität Rostock
Lehrstuhl für Wirtschaftsinformatik
Albert-Einstein-Str. 22
18059 Rostock
{dirk.stamer, kurt.sandkuhl, ulrike.borchardt, felix.timm}@uni-rostock.de

Abstract: Lehr-Lern-Portale können eingesetzt werden, um Lernende durch das Bereitstellen von relevanten Informationen an einem zentralen Ort sinnvoll zu unterstützen. Es stellt sich jedoch schnell die Frage, wie der intuitiv erwartete Mehrwert bezüglich des Nutzens eines solchen Portals objektiv gemessen werden kann? Ansätze zur Qualitätsbewertung von Software fokussieren sich häufig auf einzelne Teilaspekte wie den wahrgenommenen Nutzen. Diese Arbeit präsentiert als ganzheitlichen Ansatz zur Nutzenbewertung eines Lehr-Lern-Portals eine angepasste Balanced-Scorecard am Beispiel des Portals „myKosmos". Diese Arbeit entstand im Rahmen des Projekts KOSMOS, das darauf abzielt, ein Konzept für das Lebenslange Lernen an Hochschulen zu entwickeln, um traditionellen und nicht-traditionellen Zielgruppen individuell angepasste Studienmöglichkeiten auf universitärem Niveau anzubieten.

1 Einleitung

Die Universität Rostock hat sich zum Ziel gesetzt, ein Konzept für das Lebenslange Lernen zu implementieren, in dessen Rahmen traditionellen und nicht-traditionellen Zielgruppen maßgeschneiderte Studienmöglichkeiten auf universitärem Niveau angeboten werden. Neue Studienformate ermöglichen die Aufnahme eines Studiums in allen Lebensphasen. Sie bieten Anschlussmöglichkeiten an Ausbildung und Berufstätigkeit. Im Rahmen des Projekts „KOSMOS[1]" soll in den Fakultäten – die eigenen Grenzen der Fachdisziplin überschreitend – Bildung für neue Zielgruppen maßgeschneidert und nachfrageorientiert angeboten werden können.

[1] http://www.kosmos.uni-rostock.de/

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

Die Umsetzung der oben genannten Ziele erfordert nicht nur neue Studienmodelle und Studienformate, sondern muss auch die technischen und organisatorischen Voraussetzungen und Hilfsmittel berücksichtigen, die für die Lernenden und Lehrenden zur Verfügung stehen. Im Rahmen von „KOSMOS“ konzentrieren sich Arbeiten auf diese mediale Infrastruktur, da neue Zielgruppen, Studienformate und Lernkulturen auch neue Anforderungen an die unterstützenden IT-Systeme (z.B. sogenannte Learning Management Systeme oder auch Lernsysteme) und die relevanten Inhalte bedeuten.

Dabei wurde das Portal „myKosmos“ für den Einsatz an der Universität Rostock konzipiert und realisiert, das in verschiedenen Studienformaten und für unterschiedliche Zielgruppen eingesetzt werden kann. Ein Portal bietet dem Anwender einen zentralen Zugriff über eine einheitliche Benutzungsoberfläche auf integrierte Datenquellen und Anwendungen an. Informationstechnische Portale bündeln im Allgemeinen den Zugang zu unterschiedlichen Anwendungen und Informationsquellen unter einer Oberfläche, die auf den aktuellen Benutzer ausgerichtet ist und vor ihm verbirgt, dass verschiedene Anwendungen dahinter liegen [7].

Ziel ist es dabei, die Lernenden mit ihren unterschiedlichen Vorkenntnissen im Lernprozess individualisierter zu begleiten und weitere elektronische Unterstützungsmöglichkeiten anzubieten. Die Ausgestaltung der technischen Realisierung sollte unter zwei Gesichtspunkten geschehen: zum einen liegt der Fokus auf der bedarfsgerechten individuellen Informationsversorgung des Lernenden während der unterschiedlichen Lernphasen und zum anderen auf der individuellen Anpassbarkeit der Lernumgebung durch den Lernenden. Hier werden positive Effekte auf die Reduktion des Phänomens der Informationsüberflutung erwartet [12]. Eine große Bedeutung hat dabei, dass sowohl digital weniger erfahrene Menschen als auch „digital natives“ der jüngeren Generationen mit ihren unterschiedlichen Bedürfnissen unterstützt werden müssen. Die unterschiedliche Informations- und Medienkompetenz, die auch bei den traditionellen Studierenden zu beobachten ist, wurde bei der Konzeption berücksichtigt.

Diese Arbeit präsentiert zum einen das Vorgehen zur Entwicklung einer Balanced-Scorecard zur Bewertung der Effekte aus der Nutzung des oben beschriebenen Portals und zum anderen die entwickelte Balanced-Scorecard selbst.

Die weitere Arbeit gliedert sich wie folgt: Kapitel 2 stellt gängige Verfahren zur Bewertung von Software dar. Kapitel 3 beschreibt das methodische Vorgehen zur Entwicklung der Balanced-Scorecard. Abschnitt 4 erläutert die Ergebnisse der Balanced-Scorecard exemplarisch zur Messung des Nutzens für ein Lehr-Lernportal. Abschnitt 5 fasst die Ergebnisse der Arbeit zusammen und gibt einen Ausblick auf fortführende Arbeiten in dem Gebiet.

2 Verfahren zur Bewertung von Software

In diesem Abschnitt werden gängige Verfahren zur Nutzen- und Qualitätsbewertung präsentiert und hinsichtlich ihrer Eignung zur Bewertung eines Lehr-Lern-Portals diskutiert und bewertet.

Eine der Herausforderungen auf dem Gebiet der Nutzen- und Qualitätsbewertung von IT-Anwendungen und -Artefakten ist, dass zwar eine Vielzahl von Verfahren und Metriken vorgeschlagen worden sind, aber keine Einigkeit darüber besteht, welche Verfahren für eine ganzheitliche Betrachtung erforderlich bzw. für welchen Einsatzzweck welche Verfahren relevant sind. Nutzen und Qualität sind eng mit einander verbunden, da Nutzen als ein Aspekt von Qualität verstanden werden kann. Garvin [6] unterscheidet beispielsweise u.a. die Produkt- und Nutzer-bezogene Perspektive. Die Nutzer-bezogene Perspektive geht davon aus, dass sich Qualität während der Verwendung des Produkts zeigt, d.h. „im Auge des Betrachters" liegt [1]. Die Produkt-bezogene Perspektive sieht Qualität als präzise und messbare Variable, d.h. Unterschiede in der Qualität spiegeln sich in unterschiedlichen Werten bestimmter Attribute des Produkts wider[8]. Der Nutzen eines Produkts kann dabei eine dieser messbaren Variablen sein. Trotz dieser Nähe werden in vielen Bewertungsansätzen Nutzen und Qualität getrennt voneinander betrachtet.

Die bisher vorliegenden Ansätze lassen sich hinsichtlich ihres Schwerpunkts gliedern in Ansätze zur wirtschaftlichen Nutzenbewertung, Ansätze zur Bewertung der technischen Qualität und Ansätze für die Betrachtung der sozio-technischen Qualität. Zur wirtschaftlichen Nutzenbewertung sind wiederum verschiedene Kategorien zu unterscheiden, von denen hier jeweils ein typischer Vertreter genannt werden soll:

- Prozess-orientierte Ansätze, wie die IT Business Value Metrik von Mooney et.al. Bei diesen Ansätzen wird die Prozessverbesserung gemessen, wobei die zentralen Kriterien Durchlaufzeit, Ressourcenverbrauch und Fehleranzahl im Prozess sind [14],

- Ansätze mit Fokus auf den wahrgenommenen Nutzen, wie das IS Success Model von DeLone und McLean. Sie haben einen Katalog von Kriterien entwickelt, der u.a. die Qualität des Systems und die Qualität der bereitgestellten Informationen umfasst. Die Nutzer müssen aus ihrer subjektiven Sicht bewerten, wie sie diese Kriterien einschätzen [5],

- Projekt-orientierte Ansätze, wie Information Economics von Parker und Benson konzentrieren sich auf die Bewertung einzelner IT-Projekte. Zentrale Idee ist vor Projektstart eine Einschätzung zu geben, ob das Projekt wirtschaftlich sinnvoll ist. Der Ansatz hat Ähnlichkeit mit der klassischen Nutzwertanalyse [15],

- Scorecard-basierte Ansätze, wie BTRIPLEE-Framework, streben die Einbeziehung unterschiedlicher Perspektiven an, um somit ein besseres Gesamtbild zu zeigen. Bei BTRIPLEE sind dies beispielsweise finanzielle und prozessorientierte Aspekte [16].

Für die technische Perspektive sind vor allem Fragen der Nutzbarkeit (Usability) von Anwendungen und Bedienoberflächen relevant. Die Usability eines Software-Produkts wird durch sogenannte Evaluationsverfahren ermittelt. Zu unterscheiden sind hauptsächlich analytische und empirische Verfahren. Ein Vertreter der analytischen

Evaluationsverfahren ist das „Cognitive Walkthrough“. Hierbei versetzt sich ein Usability-Experte in die Rolle eines hypothetischen Benutzers und ermittelt anhand definierter Schritte die Gebrauchstauglichkeit des Produkts. Als Nachteil dieser Methode kann angesehen werden, dass nicht der zukünftige Nutzer, sondern ein unabhängiger Experte die Untersuchung vornimmt. Der bekannteste Vertreter von empirischen Evaluationsverfahren ist der „Usability-Test“. Hierbei führen die potentiellen Benutzer die Untersuchung unter Anleitung durch. Der zukünftige Benutzer führt definierte Arbeitsabläufe am System durch und ist angehalten alle seine Gedanken möglichst spontan laut auszusprechen („Methode des Lauten Denkens“), dabei wird der Proband intensiv beobachtet – dies kann technologisch durch Kameras, Messungen der Augenbewegungen o.ä. unterstützt werden. Das Ziel dieser Untersuchung ist es, eine Umgebung zu schaffen, die der späteren Arbeitsumgebung ähnlich ist. Dies hat trotz des hohen Aufwands einer derartigen Untersuchung zur Folge, dass schon mit wenigen Probanden ein sehr hoher Anteil der Fehler in einem Software-System gefunden werden können.

Aus sozio-technischer Sicht werden Arbeitspraktiken und prozess-bezogene Kriterien für relevant erachtet. Dazu gehören beispielsweise die Art der durchgeführten Aktivitäten (z.B. koordinieren, integrieren, beschreiben, anwenden, vereinfachen, kommunizieren zwischen Rollen bei der Nutzung der Anwendung) oder die Nutzung von Artefakten (z.B. Dokumentationen, Hilfsmittel, Werkzeuge). Ansätze zur Messung dieser Kriterien stammen häufig aus der empirischen Sozialforschung und umfassen beispielsweise ethnographische Studien mit offenen Tiefeninterviews, partizipatorische Beobachtungen und Dokumentenanalyse [2-4, 9-11, 13].

Bezüglich der Anwendung des Portals ist nicht nur die individuelle sondern auch die organisatorische Perspektive von Bedeutung. E-Learning Portale sollten idealerweise nicht nur für die Lehrenden und Lernenden relevant sein, sondern auch für ein Unternehmen oder eine Organisation einen Wert darstellen, weshalb eine Bewertung aus unterschiedlichen Perspektiven unabdingbar ist.

3 Methodisches Vorgehen zur Entwicklung der Balanced-Scorecard

Der vorherige Abschnitt zeigt deutlich eine breite Palette von Möglichkeiten, wie Nutzen- und Qualitätsbewertungen von Software durchgeführt werden können. Alle vorgestellten Ansätze könnten möglicherweise für eine maßgeschneiderte Evaluation angewendet werden. Eine genauere Betrachtung zeigt jedoch Unterschiede zwischen den Ansätzen in Bezug auf ihre Eignung.

Die Validierung hat wirtschaftliche Faktoren und ebenfalls wirtschaftliche Alleinstellungsmerkmale wie zum Beispiel eine erhöhte Flexibilität zu berücksichtigen. Diese wirtschaftlichen Alleinstellungsmerkmale sind messbare Kriterien, die in den Systemen des Controllings in vielen Unternehmen berücksichtigt werden. Ansätze, die eine Bewertung über den wahrgenommenen Nutzen durchführen wie der Ansatz nach DeLone und McLean, sind nicht in der Lage diese Aspekte zufriedenstellend abzudecken. DeLone und McLean bieten auf der anderen Seite eine Vielzahl von möglichen zu untersuchenden Aspekten, die als Inspiration bei der Festlegung von Kriterien unterstützend verwendet werden können.

Prozessorientierte Ansätze sind von Natur aus eher spezifisch für einzelne Unternehmen zugeschnitten, dies berücksichtigt ein tiefes Verständnis der Geschäftsprozesse, der möglichen Auswirkungen auf das Geschäft und der möglichen Auswirkungen der IT. Dies macht die Ansätze sehr aufwendig und daher teilweise unwirtschaftlich.

Der Balanced-Scorecard-Ansatz erfüllt hingegen alle bereits beschriebenen Anforderungen:

- Messung der wirtschaftlichen Alleinstellungsmerkmale können in einer Scorecard unter Zuhilfenahme von relevanten Indikatoren erfasst werden,

- Scorecards sind ein wichtiger Bestandteil von Managementsystemen, die die Überwachung der Leistung als Hauptelement enthalten,

- die Gesamtziele können auf die gleiche Weise, wie wirtschaftliche Alleinstellungsmerkmale erfasst werden und

- die Entwicklung und Umsetzung einer Scorecard sind angemessen in Bezug auf die zur Verfügung stehenden Mittel im Teilarbeitspaket.

Der Balanced-Scorecard-Ansatz kann daher als geeignetes Mittel zur Messung des Nutzens eines Portals angesehen werden. Im Folgenden wird ein genereller Entwicklungsablauf einer Balanced-Scorecard beschrieben. Abbildung 1 verdeutlicht die Schrittfolge bei der Entwicklung grafisch.

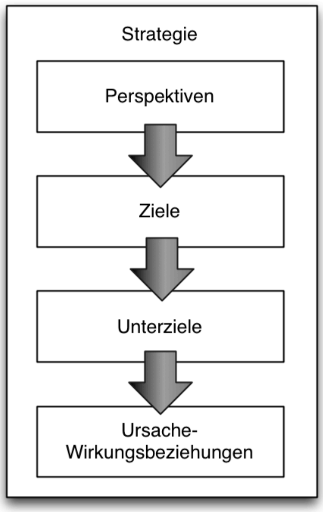


Abbildung 1: Schrittfolge zur Entwicklung der Balanced-Scorecard

Im ersten Schritt der Entwicklung wird beurteilt, ob die vom Balanced-Scorecard-Ansatz vorgeschlagen Perspektiven (d.h. Finanzen, Kunden, Prozesse, Lernen und Wachstum) anwendbar sind und für den vorliegenden Anwendungsfall in Frage kommen. Ein Ausgangspunkt für die Identifizierung der relevanten Perspektiven stellt die Geschäftsstrategie oder das Leitbild dar. Im konkreten Fall wurden hier das Leitbild der Universität bzw. die Ziele des Projekts KOSMOS berücksichtigt. Das Ergebnis dieses Schrittes ist eine erste Vereinbarung über die Perspektiven und wird in der Scorecard erfasst.

Für jede Perspektive müssen strategische Ziele oder auch kritische Erfolgsfaktoren definiert und vorzugsweise quantifiziert werden. Die Quantifizierung hilft hier Unbestimmtheit in den strategischen Zielen zu reduzieren.

Den Ausgangspunkt für die Balanced-Scorecard-Entwicklung bildete die Durchführung von Workshops zur Definition der initialen Ziele je bereits definierter Perspektive. Diese Workshops produzierten eine erste Version der Scorecard, die den Ausgangspunkt für Verbesserungen und weitere Entwicklung darstellte. Die Workshops wurden im Rahmen eines Masterkurses mit Studierenden und Dozenten durchgeführt.

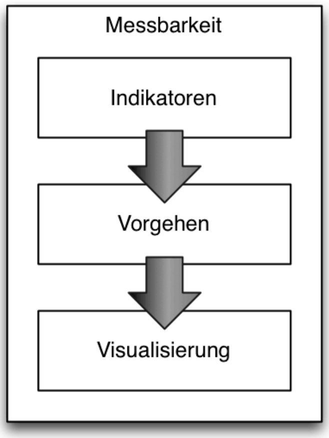


Abbildung 2: Messbarkeit

Die definierten strategischen Ziele werden in einem nächsten Schritt in Teilziele zerlegt. Leitfrage bei der Definition der Teilziele war: "Was haben wir zu tun, um unsere strategischen Ziele zu erreichen?". Ziel sollte es sein, nicht mehr als fünf bis sieben Teilziele pro Ziel zu definieren, um die Übersichtlichkeit – als Ziel einer Scorecard nicht zu konterkarieren.

Der letzte Schritt der strategischen Aspekte ist die Identifizierung von Ursache-Wirkung-Beziehungen. Es kann strategische Ziele geben, die nicht zur gleichen Zeit erreicht werden können, weil sie sich gegenseitig negativ beeinflussen. Es ist wichtig, diese Konflikte oder Ursache-Wirkungs-Beziehungen zwischen Zielen zu verstehen. Während der ersten Scorecard-Workshops sollten Ursache-Wirkungs-Beziehungen hingegen nicht berücksichtigt werden, um eine Reflektion und Diskussion der Ziele und Unterziele nicht zu belasten.

Nachdem die strategischen Aspekte abgedeckt wurden, wird der Schwerpunkt wird auf die Frage nach der Messbarkeit verlagert. Das Vorgehen bezüglich der Messbarkeit wird in Abbildung 2 verdeutlicht.

Für jedes Teilziel in den verschiedenen Perspektiven muss definiert werden wie dieses Teilziel in Bezug auf das übergeordnete Ziel gemessen werden kann. Zu diesem Zweck müssen Indikatoren oder auch „Key Performance Indikators“ (KPI) definiert werden. Bei der Definition der Indikatoren muss berücksichtigt werden, dass es eine praktikable Möglichkeit geben muss, um den Indikator erfassen zu können. In diesem Zusammenhang erfolgt eine Untersuchung über vorhandene Systeme oder Indikatoren (z.B. aus dem Qualitätsmanagement) und einer Möglichkeit diese Informationen wiederzuverwenden.

Für jeden Indikator wird festgelegt wie eine Erfassung oder Messung durchgeführt werden kann. Die Machbarkeit der Umsetzung des Messansatzes sollte sorgfältig geprüft werden. Ein Messverfahren umfasst hierbei typischerweise:

- die Möglichkeit des Messens eines Indikators,
- den Zeitpunkt und das Intervall für die Messung,
- die verantwortliche Rolle oder Person, die die Messung vornimmt und
- die Definition wie die Messergebnisse dokumentieren werden sollen.

Darüber hinaus muss ein Bezugswert definiert werden. Dieser könnte vorzugsweise auf vorhandenen alten Daten basieren, d.h. Datensätze oder Dokumenten aus der Vergangenheit.

Der letzte Schritt in diesem Zusammenhang ist die Visualisierung der Entwicklung der Indikatoren im Verlauf der Zeit, um das Ergebnis der Untersuchung anschaulich darzustellen und entsprechende Maßnahmen abzuleiten.

Das beschriebene Vorgehen zur Entwicklung der Balanced-Scorecard wurde in drei Iterationen mit unterschiedlichen Personengruppen durchgeführt und führte zu einer weiteren Detaillierung der Ergebnisse.

4 Balanced-Scorecard zur Bewertung eines Lehr-Lern-Portals

Essentiell bei der Entwicklung einer Balanced-Scorecard ist die Entscheidung über die zugrundeliegenden Perspektiven. Der ursprüngliche Ansatz beschrieben von Kaplan und Norton sieht die Perspektiven Finanzen, Kunden, Prozesse, Lernen und Wachstum vor. Da die vorliegende Balanced-Scorecard für ein Lehr-Lern-Portal für die Nutzung an einer Universität entwickelt wird, bietet es sich an die Perspektiven anzupassen. Es werden für diesen Kontext die Perspektiven: Student, Dozenten, Organisation und Zukunftsfähigkeit vorgeschlagen.

Die Perspektive „Student“ zielt auf die Hauptnutzergruppe des Portals hin ab und ähnelt daher bei Kaplan und Norton der Perspektive „Kunde“. Das Hauptziel der Perspektive ist, dass die Studierenden die Nutzung des Portals als Unterstützung bei ihren Studien empfinden sollen. In einer heterogenen Anwendungslandschaft wie an einer Universität ist ein Portal geeignet, die Nutzung der Systeme durch einen „Single Point of Entry“ zu vereinfachen. Die individuelle Informationsbereitstellung unterstützt ein kontextabhängiges Arbeiten mit dem Portal. Die Unterstützung des selbstständigen Lernens und der Kollaboration mit anderen Nutzern sind weitere wichtige Ziele dieser Perspektive.

Neben Studierenden sind allerdings auch Lehrkräfte der Universität als Nutzer zu berücksichtigen. Dies geschieht in der Perspektive „Dozent“, die bei Kaplan und Norton der Perspektive „Mitarbeiter“ ähnelt. Die Dozenten der Universität sollen das Portal „myKosmos“ als Unterstützung bei der Organisation und Durchführung ihrer Lehrtätigkeit empfinden und nutzen. Hier liegt der Fokus sowohl auf der einfachen Verteilung von Lehrmaterialien als auch auf einer engen Kommunikation mit den Studierenden, um diese bei ihren selbstständigen Studien zu unterstützen.

Die Perspektive „Organisation“ stellt die Sicht der verwaltenden Organe einer Hochschule auf das Portal dar. Hier sind langfristige Ziele wie die Senkung der Quoten von Studienabbrechern, Steigerung der Absolventenzahl oder der Abschlussnoten zu nennen. Die Individualisierung des Lehrangebots trägt dazu bei, neue Zielgruppen und Lernformen in der wissenschaftlichen Weiterbildung zu unterstützen.

Perspektive „Zukunftsfähigkeit“ stellt eine eher technisch orientierte Sicht auf die Balanced-Scorecard dar. Hier sind fragen nach der Gebrauchstauglichkeit und Fragen nach der Wartung, Betreibung und Erweiterung des Portals subsummiert.

Aufgrund von Platzgründen wird im Folgenden nur teilweise die Perspektive des Studierenden als Hauptzielgruppe in der Abbildung 3 erläutert und auf eine Darstellung der anderen Perspektiven verzichtet.

Das Ziel der Schaffung eines „Single Point of Entry“ zu allen studienrelevanten Systemen der Universität zielt daraufhin ab, dass dadurch ein steter Wechsel der zu benutzenden Systeme vermieden wird. Alle studienrelevanten Informationen werden den Studierenden an einem zentralen Ort bereitgestellt.

Wie auch die heterogenen Studiengänge sind die Lernprozesse eines jeden einzelnen individuell und müssen in einer individuellen bedarfsgerechten Informationsbereitstellung berücksichtigt werden. Hierbei werden Empfehlungen durch ein Empfehlungssystem berücksichtigt, das Studierenden z.B. Dateien empfiehlt, die für andere Studierende in einem ähnlichen Kontext von Bedeutung waren.

Ziel	**Indikator**	**Möglichkeit der Messung**	**Bezugswert**
Schaffung eines Single Point of Entry zu allen relevanten Systemen der Universität	Anzahl integrierter studienrelevanter Systeme	Ja, durch Identifikation studienrelevanter Systeme möglich.	Prozentuale Angabe integrierter Systeme zu allen relevanten Systemen.
Individuelle Informations-bereitstellung verbessern	Anteil genutzter Empfehlungen	Ja, durch technische Erfassung der Nutzung der angebotenen Empfehlungen im Portal.	Es wird angenommen, dass ab einem Prozentsatz von 75% die Empfehlungen akzeptiert werden.
Selbstständiges Lernen unterstützen	Häufigkeit der Nutzung von Lernangeboten	Ja, durch technische Erfassung der Nutzung der bereitgestellten Lernangebote.	Es wird angenommen, dass ab einem Prozentsatz von 75% die Lernangebote akzeptiert werden.
Kollaboration innerhalb von Gruppen unterstützen	Häufigkeit der Nutzung von Kollaborations-werkzeugen	Ja, durch technische Erfassung der Nutzung der angebotenen Kollaborations-werkzeuge.	Es wird angenommen, dass ab einem Prozentsatz von 75% die Kollaborations-werkezuge akzeptiert werden.
Nutzer-zufriedenheit steigern	Subjektiver Zufriedenheitsindex als z.B. Schulnote	Ja, durch Befragungen, Interviews oder Umfragen.	Zufriedenheitsindex vor der Nutzung des Portals ermitteln.

Abbildung 3: Perspektive Student

Insbesondere im Kontext einer universitären Weiterbildung ist das selbstständige Lernen notwendig. Ein Portal muss daher in der Lage sein, Studierende hierbei zu unterstützen. Hier erfolgt eine Messung der Nutzung der unterschiedlichen Angebote wie Online-Kurse, Webinare oder auch die einfache Nutzung von durch Dozenten bereitgestellten Materialien.

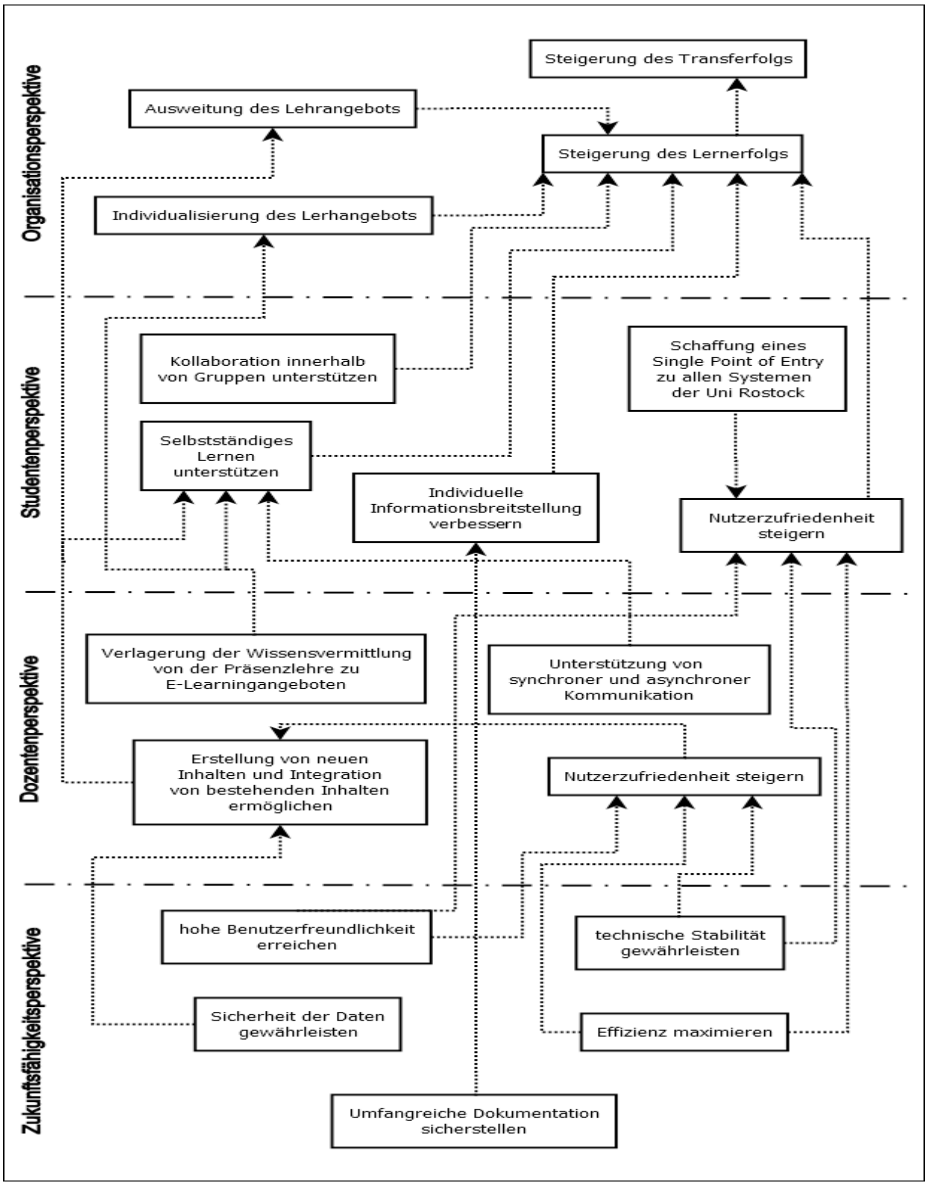

Organisationsperspektive
Studentenperspektive
Dozentenperspektive
Zukunftsfähigkeitsperspektive
Steigerung des Transferfolgs
Ausweitung des Lehrangebots
Steigerung des Lernerfolgs
Individualisierung des Lerhangebots
Kollaboration innerhalb von Gruppen unterstützen
Schaffung eines Single Point of Entry zu allen Systemen der Uni Rostock
Selbstständiges Lernen unterstützen
Individuelle Informationsbreitstellung verbessern
Nutzerzufriedenheit steigern
Verlagerung der Wissensvermittlung von der Präsenzlehre zu E-Learningangeboten
Unterstützung von synchroner und asynchroner Kommunikation
Erstellung von neuen Inhalten und Integration von bestehenden Inhalten ermöglichen
Nutzerzufriedenheit steigern
hohe Benutzerfreundlichkeit erreichen
technische Stabilität gewährleisten
Sicherheit der Daten gewährleisten
Effizienz maximieren
Umfangreiche Dokumentation sicherstellen

Abbildung 4: Ursache-Wirkungs-Beziehungen

Neben dem selbstständigen Lernen wird Gruppenarbeit während eines Studiums häufig eingesetzt. Dies geschieht innerhalb des Portals z.B. durch Nachrichten-, Chat- oder Telefoniefunktionalitäten.

Die Steigerung der Nutzerzufriedenheit stellt einen Indikator für den Erfolg oder Misserfolg einer Bildungsmaßnahme dar und spielgelt sich in der Akzeptanz des Portals wider. Hierbei sollten vor der Anwendung der Balanced-Scorecard Referenzwerte durch Umfragen bei Studierenden, die das Portal nicht nutzen, erhoben werden.

Die Ursache-Wirkungs-Beziehungen zwischen den beschriebenen Zielen der Perspektive „Student“ und der anderen Perspektiven stellt die Abbildung 4 dar.

5 Zusammenfassung und Ausblick

Lehr-Lern-Portale bieten Studierenden eine Hilfestellung im Rahmen ihres Studiums insbesondere durch eine individuelle und bedarfsgerechte Informationsbereitstellung. In dieser Arbeit wurden die Möglichkeiten zur Bewertung des Nutzens eines solchen Portals diskutiert. Der Ansatz der Balanced-Scorecard wurde als ganzheitlicher Ansatz, der nicht nur einzelne Teilbereiche der Qualität von Software berücksichtigen kann, ausgewählt.

Die vorgestellte Balanced-Scorecard bietet die Möglichkeit den Nutzen eines Lehr-Lern-Portals zu bewerten. Dazu wurde der Ansatz von Kaplan und Norton in dieser Arbeit adaptiert und angepasst an die Besonderheiten eines Portals und einer Universität. Berücksichtigung fanden hierbei nicht nur die Sicht der Studierenden, sondern auch die Perspektiven der Dozenten, der Organisation und der Zukunftsfähigkeit.

Die Ergebnisse dieser Arbeit sind zum einen die Vorstellung einer Balanced-Scorecard zur Nutzenmessung eines Portals und zum anderen die Vorstellung des Prozesses zur Erstellung einer Scorecard.

Für eine Überwachung des Erfolgs des Portals über einen längeren Zeitraum ist die kontinuierliche Durchführung der Anwendung der beschriebenen Balanced-Scorecard notwendig. Daher werden Arbeiten hinsichtlich einer stärkeren Automatisierung der Erfassung der Kennzahlen durchgeführt, um die Nutzung der Scorecard weiter zu vereinfachen und effizienter zu gestalten.

Die Entwicklungen am Portal „myKosmos“ werden bezüglich der Anbindung weiterer studienrelevanter Systeme und der Erweiterung des inhaltlichen Angebots fortgeführt. Daher werden weitere Iterationen bei der Entwicklung der Balanced-Scorecard notwendig werden, um eine Aktualität zu gewährleisten.

Ebenfalls sind weitere Arbeiten hinsichtlich der Validierung der Scorecard selbst notwendig. Erste Validierungsschritte wurden mit kleinen Gruppen von Studierenden und Dozenten bereits durchgeführt. Es ist jedoch geplant, das Portal zeitnah im Rahmen einer Veranstaltung einzusetzen und die Nutzung mit technischen Hilfsmitteln kontinuierlich zu überwachen.

Danksagung: Die im Rahmen dieses Beitrags dargestellten Arbeiten wurden am Mitteln des BMBF und des ESF Programms der EU durch das Projekt KOSMOS finanziert.

Literaturverzeichnis

1. Bevan, N.: Measuring usability as quality of use. Software Quality Journal. 4, 115–150 (1995).
2. Blomberg, J. et al.: Ethnographic field methods and their relation to design. D. Schuler, & A. Namioka (Eds.), Participatory design: Principles and practices (pp. 123-156). (1993).
3. Crawford, L.: Personal ethnography. Communications Monographs. 63, 2, 158–170 (2009).
4. Cunningham, S.J., Jones, M.: Autoethnography: a tool for practice and education. Presented at the CHINZ "05: Proceedings of the 6th ACM SIGCHI New Zealand chapter"s international conference on Computer-human interaction: making CHI natural, New York, New York, USA July (2005).
5. DeLone, W.H., McLean, E.R.: Information Systems Success: The Quest for the Dependent Variable. Information Systems Research. 3, 1, 60–95 (1992).
6. Garvin, D.: What Does "Product Quality" Really Mean? MIT Sloan Management Review. (1984).
7. Gurzki, T., Hinderer, H.: Eine Referenzarchitektur für Software zur Realisierung von Unternehmensportalen. Wissensmanagement. (2003).
8. Hallak, J.C., Schott, P.K.: Estimating Cross-Country Differences in Product Quality. (2008).
9. Hughes, J.A. et al.: Faltering from ethnography to design. Presented at the CSCW '92: Proceedings of the 1992 ACM conference on Computer-supported cooperative work, New York, New York, USA December (1992).
10. Jordan, B.: Chapter 3 Ethnographic workplace studies and CSCW. The Design of Computer Supported Cooperative Work and Groupware Systems. pp. 17–42 Elsevier (1996).
11. Marti, P.: Structured task analysis in complex domains. Ergonomics. 41, 11, 1664–1677 (1998).
12. Melinat, P. et al.: Information Overload: A Systematic Literature Review. Presented at the 13th International Conference on Perspectives in Business Informatics Research, Lund, Sweden January 24 (2014).
13. Millen, D.R.: Rapid ethnography: time deepening strategies for HCI field research. Presented at the DIS '00: Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques, New York, New York, USA August (2000).
14. Mooney, J.G. et al.: A process oriented framework for assessing the business value of information technology (Reprinted from Proceedings of the sixteenth annual International Conference on Information Systems, pg 17-27, 1995). Data Base for Advances in Information Systems. 27, 2, 68–81 (1996).
15. Parker, M.: Information Economics - an Introduction. Datamation. 33, 23, 86–& (1987).
16. Van Der Zee, H., Zee, H.T.: Measuring the value of information technology. Measuring the value of information technology. (2002).

Konzeption und Realisierung eines Portals für nicht-traditionelle Studienformate einer Universität

Kurt Sandkuhl, Dirk Stamer, Ulrike Borchardt, Felix Timm

Universität Rostock, Lehrstuhl Wirtschaftsinformatik, 18051 Rostock

{kurt.sandkuhl, dirk.stamer, ulrike.borchardt, felix.timm}@uni-rostock.de

Abstract: Die in diesem Aufsatz dargestellten Arbeiten entstanden im Projekt KOSMOS, das sich zum Ziel gesetzt, ein Konzept für das Lebenslange Lernen zu implementieren, in dessen Rahmen traditionellen und nicht-traditionellen Zielgruppen maßgeschneiderte Studienmöglichkeiten auf universitärem Niveau angeboten werden. Neue Studienformate ermöglichen die Aufnahme eines Studiums in allen Lebensphasen. Der Aufsatz konzentriert sich im Umfeld von KOSMOS auf die Frage, wie eine geeignete IT-Unterstützung für neue Zielgruppen und Studienformate aussehen muss. Die zentrale Idee ist ein kontext-orientiertes informationstechnisches Portal für das e-Learning. Das Ergebnis ist das „MeinKOSMOS" Portal mit einer individualisierten und bedarfsgerechten Informationsversorgung für die Lernenden. Der Beitrag beschreibt das Konzept für das Portal und Erfahrungen aus dessen Realisierung. Ein wichtiger Bestandteil ist dabei ein Leitfaden zur Bewertung, ob MeinKOSMOS für ein Studienformat geeignet ist bzw. wie das Portal dafür anzupassen ist.

1 Einführung

Pflegepersonal, Landschaftsarchitekten, Mediziner und Psychologen in einem gemeinsamen Studiengang? Kindergärtner, Lehrer und Sonderpädagogen mit gemeinsamem Studieninteresse? – Diese Situationen sind für traditionelle Studiengänge an Universitäten eher ungewöhnlich, gehören aber zum Alltag der neu entwickelten Studienformate im Projekt KOSMOS (s. Abschnitt 2.1). KOSMOS hat die Öffnung der Universität für nicht-traditionelle Zielgruppen zum Ziel, wobei nicht nur neue Studienformate, wie „Gartentherapie" und „Hochbegabtenpädagogik", Gegenstand der Arbeiten sind, sondern auch eine geeignete informationstechnische Unterstützung des Lehr- und Lernprozesses. Dieser Beitrag konzentriert sich im Umfeld von KOSMOS auf die Frage, wie eine geeignete IT-Unterstützung für neue Zielgruppen und

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

Studienformate aussehen muss. Die zentrale Idee ist ein kontext-orientiertes informationstechnisches Portal für das e-Learning. Das Ergebnis ist das „MeinKOSMOS“ Portal mit einer individualisierten und bedarfsgerechten Informationsversorgung für die Lernenden. Der Beitrag beschreibt das Konzept für das Portal und Erfahrungen aus dessen Realisierung. Ein wichtiger Bestandteil ist dabei ein Leitfaden zur Bewertung, ob MeinKOSMOS für ein Studienformat geeignet ist bzw. wie das Portal dafür anzupassen ist.

Ein zentraler Aspekt im Rahmen der Portalentwicklung war dabei die Kontext-Orientierung, d.h. die Erprobung und Bewertung kontext-basierter Lernsysteme und Lehrinhalte für die universitäre Weiterbildung. Viele Untersuchungen aus dem Wissensmanagement und der Informationslogistik weisen darauf hin, dass das Verstehen und die Unterstützung des Nutzerkontextes einen signifikanten Einfluss auf die Akzeptanz und die vom Nutzer empfundene Qualität von IT-Systemen und Inhalten haben. Der Nutzerkontext umfasst in diesem Zusammenhang nicht nur die aktuelle Rolle oder Aufgabe eines Nutzers, sondern auch dessen Ausbildungshintergrund, Erfahrungen und persönliche Präferenzen. Die damit verbundenen Forschungsfragen sind:

- Wie muss die Konzeptualisierung des Nutzerkontextes gestaltet werden, um eine Kontexterkennung und Kontextanpassung in Lernsystemen und von Lerninhalten vornehmen zu können?
- Wie lässt sich die Einführung von kontext-basierten Lernsystemen und Lehrinhalten so gestalten, dass dies als Leitfaden und Entscheidungshilfe für dessen Einsatz verwendet werden kann?

Aus methodischer Sicht war bei den oben genannten Forschungsfragen der Ansatz der konstruktionsorientierten Forschung (Design Science [22]) von hoher Bedeutung. Ein großer Teil der Forschung in KOSMOS war darauf gerichtet, mit dem Lehr- und Lernportal „MeinKOSMOS“ eine innovative Lösung und ein korrespondierendes Handlungssystem im Bereich e-Learning zu entwerfen und zu erproben. Darin spiegelt sich ein konstruktionsorientierter Forschungsansatz (vgl. [21]) wider, wie er besonders in der Wirtschaftsinformatik weit verbreitet ist. IT-Systeme werden dabei nicht nur bzgl. ihrer technischen Eigenschaften betrachtet, sondern dezidiert als Mittel zur Erreichung organisatorischer Zielsetzungen verstanden. Dies zeigte sich darin, dass die Forschung auf die Entwicklung einer innovativen Lehr- und Lernumgebung sowie die Gestaltung entsprechender organisatorischer Kontexte ausgerichtet war: der Software-Prototyp des Portals „MeinKOSMOS“ wurde für neue Formen der individuellen Nutzung des Portals durch Lernende sowie im Kontext der Abläufe des Lernens in neuen Studienformaten entworfen.

Während konstruktionsorientierte Forschung den Gesamtansatz der Forschung darstellt, wurde bei der Durchführung der Forschungsarbeiten eine Vielzahl von Einzelmethoden eingesetzt, die sich in drei miteinander integrierte Forschungszyklen einordnen lassen:

- Der Relevanzzyklus verankert die Entwurfsentscheidungen und Merkmale des konstruierten IT-Systems in Anforderungen des betreffenden Handlungskontextes bzw. evaluiert die entwickelten Forschungsresultate in diesem Kontext. Dies erfolgte im Wesentlichen auf Grundlage von Interviews und Fallstudien (siehe, z.B. [23])

- Der Rigorositätszyklus setzt die Forschungsresultate in Verbindung zum Stand des Wissens im jeweiligen Fachgebiet, was über Literaturanalysen in KOSMOS und über Publikationen auf Tagungen erfolgte (siehe [24], [25], [26] und [27])

- Der Entwurfs-/Evaluationszyklus dient der eigentlichen Konstruktion des innovativen Artefakts auf Grundlage der Ergebnisse aus dem Relevanzzyklus und unter Berücksichtigung der Resultate des Rigorositätszyklus. Hier wechselten sich hier Entwurf- und Validierungsschritte ab, was zu mehreren Portalversionen führte.

Die weitere Gliederung des Papiers ist wie folgt: Kapitel 2 enthält Hintergrundinformationen zum Projektkontext, zu Lehr- und Lernsystemen und zu Portalen. Kapitel 3 stellt anschließend das Grundkonzept des kontext-basierten Portals „MeinKOSMOS“ vor, was die dort enthaltenen grundlegenden Funktionalitäten einschließt. Das Kapitel 4 benennt Erfahrungen hinsichtlich der genutzten Technologien und beschreibt den entwickelten Leitfaden zum Portaleinsatz. Kapitel 5 fasst die wesentlichen Ergebnisse zusammen und gibt einen Ausblick auf weiterführende Entwicklungen.

2 Hintergrund

Dieser Abschnitt fasst Hintergrundinformation zu den in diesem Papier präsentierten Arbeiten zusammen. Dazu gehört das Projekt, in dem die Ergebnisse entstanden sind (2.1), das Gebiet der Lehr- und Lernsysteme (2.2) und IT-gestützte Portale (2.3)

2.1 Projektkontext: KOSMOS

Die in diesem Aufsatz dargestellten Arbeiten entstanden im Projekt „Konstruktion und Organisation eines Studiums in offenen Systemen (KOSMOS[2])“, das mit Mitteln des BMBF und der EU an der Universität Rostock gefördert wurde. Die Universität Rostock hat sich zum Ziel gesetzt, ein Konzept für das Lebenslange Lernen (LLL) zu implementieren, in dessen Rahmen traditionellen und nicht-traditionellen Zielgruppen maßgeschneiderte Studienmöglichkeiten auf universitärem Niveau angeboten werden. Neue Studienformate ermöglichen die Aufnahme eines Studiums in allen Lebensphasen. Sie bieten Anschlussmöglichkeiten an Ausbildung und Berufstätigkeit. Die Integration des Lebenslangen Lernens ist ohne Reorganisation der Institution Universität nicht zu leisten. Dementsprechend ist die Organisationsentwicklung ein Teil des Projekts und mit dem Ziel verbunden, inhaltliche, strukturelle und organisatorische Rahmenbedingungen für Lebenslanges Lernen zu implementieren.

Die Umsetzung der oben genannten Ziele erfordert nicht nur neue Studienmodelle und Studienformate, sondern muss auch die technischen und organisatorischen

[2] http://www.kosmos.uni-rostock.de/

Voraussetzungen und Hilfsmittel berücksichtigen, die für die Lernenden und Lehrenden zur Verfügung stehen. Im Rahmen von KOSMOS konzentriert sich daher ein Arbeitspaket speziell auf die „mediale Infrastruktur“, da neue Zielgruppen, Studienformate und Lernkulturen auch neue Anforderungen an die unterstützenden IT-Systeme (z.B. sogenannte Learning Management Systeme oder auch Lernsysteme) und die relevanten Inhalte bedeuten können. Dieses Papier präsentiert Ergebnisse dieses Arbeitspakets.

2.2 Lehr- und Lernsysteme

Die heute verfügbaren technischen, didaktischen und organisatorischen Möglichkeiten erlauben die Bereitstellung von Lernsystemen für lebensbegleitendes Lernen mit örtlich und zeitlich flexiblen Dimensionen. Hierbei kann bereits auf eine Vielzahl von existierenden Diensten sowie etablierten und offenen Standards in einer modernen Umgebung für mediengestütztes Lernen zurückgegriffen werden. Die Integration dieser Dienste unter einer einheitlichen Oberfläche, wie Lehr-Lern-Management-Systeme [5] (z. B. Ilias, Stud.IP, Moodle, Sakai, OLAT, Clix etc.), stellt die Infrastruktur für darauf aufsetzende E-Learning-Angebote dar. Neuartige Lernszenarien [4], in denen sich Präsenzphasen mit Onlinephasen abwechseln (z. B. Aufzeichnungssysteme wie Lecturnity, Opencast Matterhorn u. ä. oder Kommunikations- und Kooperationswerkzeuge wie Adobe Connect, Open Meeting u.ä.), bis zu einer komplett über das Netz angebotenen Lehrveranstaltung, an der Lehrende und Lernende interaktiv teilnehmen, durchdringen mehr und mehr den Lehr- und Lernalltag an Hochschulen. Hohe Zuverlässigkeit und Performance der technischen Infrastruktur unterstützen ferner die Benutzerfreundlichkeit der online-basierten Lehre und begünstigen ihre Durchdringung im universitären Alltag. Trotzdem sind Systeme, die alle Prozessschritte integrieren und auf den Kontext des jeweiligen Lernenden oder Lehrenden anpassen, nicht vorhanden.

Die Suche nach qualitativ neuen Formen eines wissenschaftlichen Weiterbildungsbetriebs unter Nutzung des genuinen Potentials digitaler Techniken ist seit einigen Jahren Gegenstand der Forschung. Eines dieser Anzeichen war bereits 2003 die Publikation des so genannten Atkins-Report der US-amerikanischen National Science Foundation (NSF) [6] mit der für den USA-Kontext formulierten Vision einer Cyber-Infrastructure als Grundlage für einen unter neuen technischen Bedingungen in weiten Bereichen neu formierten Wissenschaftsbetrieb postulieren. Eine Reihe von Neuentwicklungen und unterschiedlichen Folgeinitiativen (z. B. e-science in Großbritannien oder d-grid in Deutschland) wurden durch eine massive Investition in wissenschaftsrelevante Middleware und Infrastruktur für eine konsequente Digitalisierung wissenschaftlicher Kernprozesse erreicht. Diese Stimulierung ist auch für den Bereich Weiterbildung erforderlich.

2.3 Portale

Informationstechnische Portale bündeln im Allgemeinen den Zugang zu unterschiedlichen Anwendungen und Informationsquellen unter einer Oberfläche [17], die auf den aktuellen Benutzer ausgerichtet ist und vor ihr/ihm verbirgt, dass verschiedene Anwendungen dahinter liegen [16]. Portale sind eine in der Unternehmens-IT und speziell dem Wissensmanagement eingesetzte Technologie [3]. Maßgeblich für die Nutzung vom Portalen im universitären Kontext ist der Gedanke eines Portals als „Single Point of Entry“ [1] für alle Anwendungen, die in der Universität in der Weiterbildung online genutzt werden bzw. diese unterstützen können.

Eine Marktanalyse von Portalen und Portalplattformen [9] zeigte eine weite Verbreitung von Liferay[3]. Kennzeichnend für Liferay ist die Arbeit auf Basis des Model-View-Controller Prinzips, welches es erlaubt, die Darstellung der Inhalte von der verarbeitenden Logik zu trennen. So ist es möglich, die verarbeitende Logik in Softwaremodulen zu erstellen und sie dem Nutzer aggregiert ohne Brüche mit bestehenden Funktionalitäten darzustellen. Erweiterungen können in Liferay auf zwei Arten eingebunden werden. Zum einen gibt es die einfache Einbindung ohne weitere Anpassung als zusätzliche Seiten. Die zweite Variante, die zwar technisch aufwendiger ist, aber die angepasste Darstellung an den Nutzer erst ermöglicht, ist die Arbeit über Portlets. Diese ermöglichen u.a. den Datenaustausch mit anderen Anwendungen.

3 Kontext-basiertes Portal für das lebenslange Lernen

3.1 Grundkonzept

Vor dem Hintergrund des KOSMOS Projekts (s. 2.1), das neue Zielgruppen und Studienformate im Rahmen der universitären Ausbildung erschließen soll, wurde hinsichtlich der einzusetzende Lehr- und Lernsysteme eine stärkere Individualisierung auf den einzelnen Lernenden und seinen/ihren aktuellen Kontext untersucht. Der „Kontext“ fasst in diesem Zusammenhang alle Informationen zusammen, die die Situation des Lernenden beschreiben und dadurch bei der Individualisierung berücksichtigt werden sollten. Der Fokus auf eine stärkere Individualisierung erfolgt vor folgendem Hintergrund:

- Lernen ist ein Prozess, den jede Person selbst in ihre individuellen Abläufe und Hintergründe verankern muss. Das Lernen geschieht in unterschiedlichen Geschwindigkeiten, mit unterschiedlichen Assoziationen und mit unterschiedlichen Vorerfahrungen.

- Die spezifischen Lehr-Lern-Verfahren in einzelnen universitären Disziplinen (Ingenieur-, Sozial-, Geistes-, Naturwissenschaften etc.) unterscheiden sich in hohem Maße. Daher kann von einer Kultur der Disziplinen gesprochen werden, die bei Lehr-Lern-Prozessen der Weiterbildung eine Berücksichtigung finden müssen.

- Die Lernenden in den neuen Studienformaten im KOSMOS Projekt unterscheiden sich in ihrem Alter, ihrem Hintergrundwissen, ihren Lernzielen, ihrer Zeitverfügbarkeit, ihrem Geschlecht usw.. Diese fordern eine viel stärkere

[3] Siehe http://www.liferay.com

Individualisierung des Lernens, des Lehrangebots und der Lehr-Lern-Organisation im Vergleich zum universitären Alltag.

Untersuchungen aus dem Wissensmanagement und der Informationslogistik [18] weisen darauf hin, dass das Verstehen und die Unterstützung des Nutzerkontextes einen signifikanten Einfluss auf die Akzeptanz und die vom Nutzer empfundene Qualität von Lernsystemen und Lehrinhalten haben. Der Nutzerkontext umfasst in diesem Zusammenhang nicht nur die aktuelle Rolle oder Aufgabe eines Nutzers, sondern auch Ausbildungshintergrund, Erfahrungen und persönliche Präferenzen. Ziel ist es dabei, die Lernenden mit ihren unterschiedlichen Vorkenntnissen im Lernprozess individualisierter zu begleiten und weitere elektronische Unterstützungsmöglichkeiten anzubieten.

3.2 Aktuelle Realisierung

Die Ausgestaltung der technischen Realisierung umfasst aktuell zwei wesentliche Aspekte: zum einen liegt der Fokus auf der bedarfsgerechten individuellen Informationsversorgung des Lernenden während der unterschiedlichen Lernphasen und zum anderen auf der individuellen Anpassbarkeit der Lernumgebung durch den Lernenden.

Individuelle Informationsversorgung

Für die Teilnehmer essentiell ist der Zugang zu den Materialien aus unterschiedlichen Plattformen, die während des Lernprozesses genutzt werden sollen unter einem einzigen Zugang, um den Zugriffsprozess zu verkürzen. Der erste Schritt war hier die Nutzung der Single Sign On Funktionalität des Liferay Portals über den LDAP Dienst der Universität, da dies die Barriere der Mehrfachanmeldung an verschiedenen Diensten umgeht und eine automatische Einbindung verschiedener Inhalte ermöglicht. In einem zweiten Schritt wurde eine Softwarekomponente geschaffen, die individuelle Nutzerprofile der Lernenden ermöglicht und Inhalte und Funktionalität auf die Profile anpassen kann. Das Nutzerprofil umfasste zunächst nur den Studiengang/ Zertifikatskurs, die aktuell besuchten Lehrveranstaltungen, eine evtl. Zugehörigkeit zu Arbeitsgruppen und individuelle Vorlieben des Lernenden. Entsprechend des Profils werden dann Inhalte bei der Suche bzw. in den Lehr- und Lernsystemen gefiltert bzw. priorisiert. Durch Änderungen im Nutzerprofil kann das Portal hinsichtlich Inhalte und Oberfläche auf den Nutzerbedarf angepasst werden.

Auf Grundlage des Nutzerprofils erfolgte im dritten Schritt einerseits die Einbindung der aktuell in der Universität genutzten Lehr- und Lernsysteme, Stud.IP und ILIAS, sowie die Integration eines Suchagenten für die Meta-Suche. Das Grundkonzept bei der Meta-Suche ist es, die wichtigsten Informationsquellen für jedes Studienformat zu definieren und Suchanfragen der Lernenden auf genau diese Informationsquellen zu fokussieren, ohne dass der Lernende diese Vorauswahl selbst vornehmen muss. Informationsquellen sind dabei zum einen die für das Studienformat einschlägigen Literaturdatenbanken oder Suchsysteme der Universitätsbibliothek und zum anderen Informationssysteme und –dienste im Internet. Ziel der Einbindung einer solchen Suchmaschine ist es den Teilnehmern gebündelt das Suchen in fachrelevanten Kanälen zu ermöglichen, auch wenn ihnen diese vorher nicht bekannt sind. Hierbei kann wiederum der wissenschaftliche Fokus der Weiterbildung an der Universität unterstützt werden bei gleichzeitiger Rücksichtnahme auf die Gewohnheiten und Nutzungsmuster der Teilnehmer. Als Suchagent wurde „Wegtam“[4] vom gleichnamigen Unternehmen ausgewählt. Die Integration von Wegtam in das Portal umfasst die Vereinbarung eines einfachen Protokolls zur Übergabe von Profilinformation und Suchanfrage an den Suchagenten und die Rückgabe von Suchergebnissen an das Portal.

Unterstützung kollaborativen Lernens

Neben den reinen Selbstlernphasen soll das Portal auch die Gruppenarbeit [12] stärker in den Fokus rücken, da diese auch von den Teilnehmern als besonders wertvoll an einer Weiterbildung erachtet wird [7] [13]. Dazu soll es möglich sein, durch das Profil die Zugehörigkeit zu Gruppen zu erkennen (Arbeitsgruppen, Studienprogramm) und für diese Gruppen entsprechende Arbeitsbereiche und Funktionalitäten zur Verfügung zu stellen. Die Gruppenunterstützung soll während der wissenschaftlichen Weiterbildung zur Netzwerkbildung zwischen den Teilnehmern beitragen. Die Unterstützung der Gruppenarbeit umfasst Kommunikationsmöglichkeiten innerhalb der Gruppe mit asynchronen und synchronen Mechanismen. Aktuell sind sowohl Forum als auch Chatraum im Portal vorhanden, eine Anpassung auf die Gruppenfunktionalität ist beim Forum im Hinblick auf die Strangsichtbarkeit möglich. Ein synchroner Aspekt in der Gruppenarbeit ist die Förderung der Awareness für weitere Gruppenmitglieder. Dies bezieht sich primär auf die Möglichkeit zu sehen, dass andere Gruppenmitglieder online sind und welchen Tätigkeiten sie ggf. nachgehen. Als konkrete Unterstützung neben dem Chatraum wurde Skype in das Portal integriert. Neben den Kommunikationsmöglichkeiten ist die Verwendung möglicher Kollaborationswerkzeuge zur Erstellung von gemeinsamen Arbeitsdokumenten relevant. Hier wurde entschieden, eine Anbindung von GoogleDrive bzw. GoogleDocs zu realisieren.

Kontakt während der Selbstlernphasen - Tutoring

Neben den Kontakten unter den Studierenden soll auch eine stärkere mediale Unterstützung im durch die Universität geleiteten Lernprozess stattfinden. Dazu gehört der Einsatz aller zur Verfügung stehenden Kommunikationskanäle als Kontaktmöglichkeit zu fachlichen Ansprechpartnern, d.h. die Nutzung von Chats, Foren und Skype. Das Forum hat dabei den Vorteil, dass Fachfragen durch den Tutor nur einmal beantwortet und für alle Lernenden sichtbar dokumentiert werden müssen. Des Weiteren kann so der Kontakt zu Lehrenden gehalten werden, bei z.B. Rückfragen zu Einsendeaufgaben, die ein fester Bestandteil vieler wissenschaftlicher Weiterbildungsangebote sind. Die Realisierung ist hier in Form von Sprechstunden erfolgt, die über das Portal den Nutzer bekanntgegeben werden und für die das Portal die Kommunikationskanäle bereitstellt.

Studienformat-spezifische Lerninhalte

Verschiedene Studienformate benötigen verschiedene Inhalte, entsprechend ist auch die mediale Unterstützung unterschiedlich zu gestalten. Das Portal erlaubt die Integration spezifischer Inhalte zum Studienformat über die Integration neuer Portlets oder das Verlinken von Anwendungen oder Inhalten. Dies gilt zum Beispiel im Zusammenhang mit der Nutzung von ILIAS. Hier ist die Möglichkeit der Einbindung von z.B. Java-

[4] http://www.wegtam.biz/

basierten Einheiten oder Simulationsprogrammen begrenzt, was wiederum durch das Portal ermöglicht werden kann.

4 Technische Erfahrungen und Leitfaden zum Portaleinsatz

Die Realisierung des Portals wurde zwar im Mai 2014 in einer ersten Version abgeschlossen, musste aber wegen eines Upgrades der Liferay-Plattform in eine 2. Version überführt werden, die im Oktober 2014 fertiggestellt wurde. Da die Stud.IP Integration wegen fehlender Zuarbeiten externer Partner im Oktober 2014 noch nicht abgeschlossen war, wurde die Entwicklung einer 3. Version begonnen, die Im Januar 2015 beendet wurde. Dieser Abschnitt stellt einen Teil der „lessons learned" dar, die im Laufe der Realisierung der drei Versionen gesammelt wurden. Dies beinhaltet zum einen Erfahrungen mit den eingesetzten Technologien (4.1). Zum anderen wurde ein Leitfaden zur Nutzung und Anpassung des Portals für Studienformate entwickelt (4.2), der aus der Erfahrung entstanden ist, dass ein definierter Prozess zur Vorbereitung des Portaleinsatzes in neuen Studienformaten hilfreich ist.

4.1 Technische Erfahrungen

Die Nutzung von Liferay als Portalplattform erwies sich insgesamt als sinnvolle Entscheidung. Liferay verfügt über eine große Community von Entwicklern, die viele Portlets über Open Source Foren zugänglich machen. Hinsichtlich der Leistungsfähigkeit der Plattform und des Supports ist unsere Erfahrung, dass die kostenpflichtige Version von Liferay deutliche Vorteile besitzt. Beim Versionswechsel von Liferay waren einige der entwickelten Portlets nicht mehr funktionsfähig, was Anpassungsaufwand verursachte. Dies sollte bei der Planung zukünftiger Portal-Projekte berücksichtigt werden.

Die Einbindung von GoogleDrive und GoogleDocs in das Portal erwies sich als zwar funktional sinnvoll aber technisch aufwändig und problematisch. Konzeptionell war zunächst die Frage zu klären, ob vorausgesetzt werden soll, dass die Lernenden einen eigenen Google-Zugang besitzen bzw. sich selbst einrichten, oder ob ein Google-Konto durch das Portal geschaffen und ggf. wieder gelöscht wird. Zur Klärung dieser Frage wurde in einer Bachelorarbeit das Interface von Google zum Einrichten und Konfigurieren von Nutzeraccounts evaluiert und die API getestet [15]. Es zeigte sich, dass prinzipiell das Einrichten von Accounts machbar ist, aber das Löschen mit dem Verlust von Daten verbunden ist. Daher wurde für das Portal entschieden, dass die Lernenden selbst einen entsprechenden Account einrichten müssen. Auch die GoogleDrive und GoogleDocs Anbindung an Liferay wurde in Form eine Abschlussarbeit vorgenommen [11]. Das hier realisierte Portlet erwies sich aber im Nachhinein sowohl als unbrauchbar als auch als unnötig. Durch eine Veränderung der API und der Policy zur Nutzung der API war die initiale Version des Portlets schon nach wenigen Monaten des Einsatzes nicht mehr funktionsfähig, weshalb es aktuell nicht verwendet wird. Außerdem hatte ein anderes Entwicklerteam gleichzeitig dieselbe Idee und stellte etwa zeitgleich mit der Fertigstellung des MeinKOSMOS Portlets eine technologisch überlegene Lösung als Open Source Komponente bereit. Im Vergleich zur Einbindung der Google Plattform verlief die Skype Integration problemlos.

Die LDAP und Stud.IP Integration in das Portal kosteten deutlich mehr Zeit als geplant. Der wesentliche Grund hierfür lag nicht in unklaren Schnittstellen oder anderen technischen Problemen, sondern in der mangelnden Verfügbarkeit der entsprechenden Ansprechpartner und Kompetenzen innerhalb der Universität. Für zukünftige Projekte empfehlen wir, frühzeitig die Einbindung der entsprechenden Organisationseinheiten und ggf. auch die Finanzierung entsprechender Ressourcen in das Projekt einzuplanen.

Die Software-Komponente zur Verwaltung des Nutzerprofils [10] und Erfassung des Kontextes wurde hinsichtlich ihrer Komplexität unterschätzt. Obwohl bereits Erfahrungen im Bereich kontext-basierter Systeme in der Forschungsgruppe bestanden, war vor allem die Kombination aus Informations-getriebenen und Handlungs-getriebenen Kontextaktualisierungen eine Herausforderung. Informations-getrieben bezieht sich hier auf die Fortschreibung des Informationsbedarfs eines Nutzers auf Grundlage seiner Suchanfragen oder erstellten Inhalte. Hier ergibt sich eine ähnliche Fragestellung wie bei Recommender-Systemen: welche Inhalte sind relevant für einen Nutzer im Kontext seiner Aktivitäten? Handlungs-getriebene Aktualisierungen beziehen sich auf das aktive „Dazuschalten“ bzw. „Abwählen“ von Portlets. Hier sind Regeln zu definieren, unter welchen Bedingungen das Votum eines Nutzers „für“ oder „gegen“ ein Portlet in der Startkonfiguration des Portals anzunehmen ist.

4.2 Leitfaden für den Portaleinsatz

Bei der Realisierung des Portals zeigte sich, dass einerseits gewisse Voraussetzungen gegeben sein sollten, um das Portal einsetzen zu können, und dass das Portal andererseits vor dem Einsatz für konkrete Studienformate auf die Rahmenbedingungen in diesem Format vorbereitet werden muss. Dies führte zur Entwicklung eines Leitfadens, der in diesem Abschnitt kurz zusammengefasst ist. Die Struktur des Leitfadens orientiert sich an bewährten Mustern aus der Methodenentwicklung, da Methoden generell das Vorgehen und die Voraussetzungen zur Lösung eines Problems oder Bearbeitung einer Aufgabe beschreiben. Konkret wurde die Methodenstruktur von Goldkuhl [19] für diesen Leitfaden ausgewählt und leicht angepasst. Nach Goldkuhl besteht eine Methode aus den folgenden Bestandteilen:

- Methodenkomponenten: Konkrete Handlungsanweisungen für die Bearbeitung einer Aufgabe finden sich in den Methodenkomponenten, wovon es in einer Methode mindestens eine geben muss. Eine Methodenkomponente sollte aus Konzepten, Prozedur und Notation bestehen. Die Konzepte geben an, welche Begriffe wichtig sind und was diese bedeuten, d.h. die relevanten Konzepte sollten in der Methodenkomponente erläutert werden. Die Prozedur beschreibt das konkrete Vorgehen für die Bearbeitung der Aufgabe. Dies kann auch Voraussetzungen und Hilfsmittel umfassen. Die Notation gibt vor, wie das Ergebnis der Arbeiten zu dokumentieren ist, was z.B. in graphischer oder textueller Form erfolgen kann.

- Rahmenwerk: Das Rahmenwerk der Methode beschreibt den Zusammenhang zwischen den einzelnen Methodenkomponenten, d.h. welche Komponente unter welchen Bedingungen zu verwenden ist, und wie die Ergebnisse daraus für welche nachfolgende Komponente oder Komponenten zu benutzen sind. Wenn die Reihenfolge der Methodenkomponenten immer gleich ist, muss das Rahmenwerk nicht separat beschrieben werden, sondern ist implizit durch die Beschreibung der Methodenkomponenten gegeben.

- Kooperationsformen: Für viele Aufgaben ist das Vorhandensein unterschiedlicher Fachkompetenzen oder die Mitarbeit unterschiedlicher Rollen erforderlich. Diese erforderlichen Kompetenzen und Rollen müssen ebenso beschrieben werden, wie die Aufgabenverteilung zwischen den Rollen und die Kooperationsform, d.h. wer für welche Aufgabe oder Methodenkomponente die Verantwortung übernimmt.

- Perspektive: Jede Methode beschreibt das Vorgehen beim Bearbeiten einer Aufgabe aus einer bestimmten Perspektive, die Einfluss darauf hat, was bei der Bearbeitung als wichtig erachtet wird. Viele existierende Methoden beschreiben nicht explizit, welche Perspektive eingenommen wird, es ist aber implizit aus dem Rahmenwerk oder den Methodenkomponenten ersichtlich. Wenn die Perspektive explizit beschrieben wird, beinhaltet dies die Werte, Prinzipien und Kategorien, die der Methode zugrunde liegen, d.h. eine Perspektive ist die konzeptuelle und wertmäßige Basis der Methode.

Der Leitfaden [8] ist in Anlehnung an die oben beschriebene Methodenkonzeption gegliedert. Wir konzentrieren uns im Folgenden auf das Rahmenwerk, um die wesentlichen Schritte darzustellen. Das Ziel des vorliegenden Leitfadens ist es, eine systematische Vorgehensweise zu beschreiben, wie zum einen entschieden werden kann, ob das Portal MeinKOSMOS für ein Studienformat geeignet ist, und wie zum anderen die Anpassung des Portals für dieses Studienformat vorzunehmen ist. Der Leitfaden wurde mit Blick auf die fachlich Verantwortlichen für ein Studienformat erarbeitet, d.h. es werden keine Spezialkenntnisse in der Informationstechnik vorausgesetzt.

Schritt 1: Eignung des Portals bewerten
Der Einsatz des Portals macht keinen Sinn, wenn die inhaltliche und didaktische Konzeption des Studienformats die Nutzung von IT-gestützten Medien oder Lehr- und Lernplattformen nicht vorsieht oder gar explizit ausschließt. Der Portal-Einsatz ist dann besonders sinnvoll, wenn dadurch im Vergleich zur „Standard“ e-Learning Plattform Stud.IP ein Mehrwert entsteht. Stud.IP ist in MeinKOSMOS integriert, sodass dessen Funktion ohnehin bereitsteht. Um die Bewertung zu erleichtern, wurde als Hilfsmittel ein Fragenkatalog entwickelt. Sollte sich durch die Beantwortung der Fragen kein eindeutiges Bild ergeben, wird ein Gespräch mit dem Fachverantwortlichen für das Portal zwecks gemeinsamer Entscheidungsfindung empfohlen.

Schritt 2: Umfang des Portaleinsatzes festlegen
Da es prinzipiell möglich ist, die Portalnutzung nicht für den gesamten Verlauf des Studienformats sondern nur für ausgewählte Inhalte vorzusehen, muss in diesem Schritt der Umfang der Portalnutzung festgelegt werden. Der Umfang ist am leichtesten über

die Module des Studienformats zu definieren, die im Portal unterstützt werden sollen. Auf Grundlage der Modulliste lassen sich dann die einzubeziehenden Dozenten und Studierenden festlegen (für den Fall, dass nicht alle Teilnehmer am Studienformat auch an den Modulen teilnehmen müssen).

Schritt 3: Informationsbedarf analysieren
Eines der wichtigsten Ziele des Portaleinsatzes ist es, den Studierenden den Zugang zu Informationen zu erleichtern, die für die Bearbeitung von Aufgaben oder Themen im Rahmen ihres Studienformats wichtig sind. Diese Erleichterung wird zum einen dadurch erreicht, dass bei der Suche nach Informationen oder Literatur schon voreingestellt ist, welche Informationsquellen die höchste Relevanz für das Studienformat haben. Wenn die/der Studierende an dieser Voreinstellung nichts ändert, wird die in das Portal eingebaute Suchfunktionalität zunächst in diesen Informationsquellen suchen. Zum anderen können in die Portaloberfläche zusätzlich Anwendungen integriert werden, die benötigte Informationen bereitstellen. Dies könnten beispielsweise spezielle Informationsdienste oder –systeme sein, die mit allgemeiner Suche nicht zugreifbar sind.

Zur Ermittlung des Informationsbedarfs steht eine Methode zur Informationsbedarfsanalyse zur Verfügung, die ausgehend von Aufgaben und Verantwortlichkeiten den Informationsbedarf im Detail ermittelt. Diese Methode ist in [20] dokumentiert. Da die vollständige Durchführung einer solchen Analyse recht aufwändig werden kann, wird empfohlen, ein „vereinfachtes“ Verfahren zu benutzen. Dieses Verfahren ermittelt auf Grundlage des Studienformats sowie der in den einzelnen Modulen durchgeführten Aufgaben, welche Informationsquellen relevant sind, wie wichtig die Informationen aus diesen Quellen für die Aufgabe sind und welche Folgen ein Fehlen der Informationen hätte. Auf Grundlage dieser Bewertung der Informationsquellen werden vorrangige Informationsquellen ermittelt, die in das Profil des Studienformats aufgenommen werden und in Schritt 6 zur Konfiguration der Meta-Suche eingesetzt werden.

Schritt 4: Bedarf an Portalfunktionalität ermitteln
In diesem Schritt wird zum einen festgelegt, wie die initiale oder auch „Standard“-Konfiguration des Portal für das Studienformat aussehen muss. Dies umfasst u.a. welche Portlets zu integrieren und welches Layout zu realisieren ist. Weiterhin wird ermittelt, ob es für einzelne Teilgruppen der Studierenden Anpassungen in dieser Standardkonfiguration geben soll, um beispielsweise kollaboratives Lernen zu unterstützen. Für jede entwickelte Portalfunktionalität (siehe Abschnitt 3) wird dann ermitteln, ob diese gebraucht wird und wie die Grundkonfiguration sein soll.

Schritt 5: Erforderliche Portalanpassung zusammenstellen
Da die Ermittlung des Informationsbedarfs und die Ermittlung der erforderlichen Funktionalität des Portals möglicherweise unter Mitarbeit unterschiedlicher Beteiligter und zu verschiedenen Zeitpunkten erfolgt, wurde dieser Arbeitsschritt in den Ablauf integriert, um aus den Teilergebnissen eine Gesamtsicht zusammenzustellen. Im einfachsten Fall besteht dieser Schritt nur aus einem Zusammenfügen der Ergebnisdokumente der vorhergehenden Aktivitäten zu einem Gesamtdokument. Hierbei sollte die Stimmigkeit des Gesamtbilds geprüft werden. In wenigen Fällen wird bei dieser Zusammenstellung deutlich werden, dass es weiteren Informationsbedarf bzw. zusätzlich

benötigte Portalfunktionen gibt, was erst aus der Gesamtsicht erkennbar ist. Für diesen Fall wird empfohlen, die Arbeit in dem entsprechenden Teilschritt erneut aufzunehmen.

Schritt 6: Portalanpassung anstoßen
Das zentrale Ziel dieser Aktivität ist es, die Umsetzung der erforderlichen Portalanpassungen anzustoßen, um eine rechtzeitige Bereitstellung des Portals zu garantieren. Ein Teil dieses Arbeitsschrittes ist es auch, die Umsetzbarkeit aller Anforderungen zu prüfen und – falls erforderlich, diese zu präzisieren. In vielen Fällen wird die eigentliche Portalanpassung keine Programmieraufgaben erfordern, sondern nur das Konfigurieren des Portals umfassen und daher schnell durchzuführen sein. Dazu gehört in der Regel das Einrichten eines sogenannten „Profils“ für das Studienformat in der Portal-Suchfunktion, wobei die relevanten, mit Priorität zu durchsuchenden Informationsquellen dem System bekanntgemacht werden. In seltenen Fällen muss hier auch noch die technische Zugriffsschnittstelle zu den Informationsquellen eingerichtet werden, was eine Programmierschnittstelle erfordern kann. Weiterhin gehört zum Konfigurieren des Portals, die Grundeinstellung der Funktionen vorzunehmen, die im Studienformat bereitstehen sollen. Wenn funktionale Erweiterungen erforderlich sind, wie beispielsweise bei der Integration zusätzlicher Anwendungen oder Portlets, ist auch hierfür die Vorgehensweise zu klären.

5 Zusammenfassung und Ausblick

Der Aufsatz beschreibt im Kontext des Projekts KOSMOS die grundlegende Idee, das Konzept und Erfahrungen aus der Realisierung des Portals MeinKOSMOS, dessen zentraler Ansatz eine bedarfsgerechte Informationsversorgung und nutzerindividuelle Bereitstellung von Funktionalität im e-Learning ist. Als zukünftige Erweiterung des Portals wird das Anlegen, Erfassen und Weiterführen des Portfolios der Lernenden gesehen, welches sich zum Teil aus den systemischen Angaben zusammenstellt aber um persönliche Angaben des Teilnehmers erweitert wird. So ist es hier denkbar, die Ausbildungsstufe bzgl. der Grundlagenfächer (z.B. Mathematik) oder den Berufsstatus mit festzuhalten, um entsprechende Ergänzungsmaterialien bereit zu stellen. Diese Informationen müssten durch die Teilnehmer eingepflegt werden. Entsprechend der zur Verfügung gestellten Daten können Zusatzmaterialien ermittelt und Empfehlungen ermittelt und dargestellt werden.

Die bisher beschriebenen Arbeiten am Portal betrachten den einzelnen Nutzer isoliert von anderen Nutzern. Doch gerade aufgrund von einer bestehenden Ähnlichkeit der Interessen von Nutzern, ist es sinnvoll die Nutzer nicht isoliert voneinander zu betrachten, sondern als Gruppe von Personen mit ähnlichen Interessen anzusehen. So wird ein Benutzer des Portals, der sich aktuell auf eine Prüfung vorbereitet, ein ähnliches Interesse für bestimmte Informationen haben wie sein Kommilitone, der sich auf dieselbe Prüfung vorbereitet. Dieser Sachverhalt kann informationstechnisch in dem Portal „MeinKOSMOS“ mit einem Empfehlungssystem unterstützt werden. Ein Teil der weiterführenden Arbeiten ist daher die Integration solcher „recommendation systems“.

Weiterhin ist die Frage zu untersuchen, inwieweit für MeinKOSMOS die Einbindung einer virtuellen Lernumgebung mit entsprechenden Tutoren von Belang ist. Im Rahmen einer solchen Lernumgebung ist ebenfalls das Abhalten sogenannter Webinare zu diskutieren, bei denen die Lehrveranstaltung live entsprechend übertragen wird. Dieser Punkt fokussiert also weniger auf den Nutzerkontext an sich, sondern lediglich das Angebot eines solchen Tutoring sollte an den Studenten weitergegeben werden. Dynamischen Kontext bildet es dann, wenn das Tutoring Inhalte bzw. Anmerkungen bereitstellt, die dem Nutzer wiederum sichtbar gemacht werden müssen. Hier ist ähnlich wie bereits zuvor beschrieben eine Empfehlungsmöglichkeit denkbar, aber auch die veränderte Darstellung von neuen Dokumenten. Um dies zu ermöglichen, müsste das individuelle Verhalten im Nutzerprofil mit abgelegt werden, um dann noch nicht gelesene Artikel/ Einträge/ Anmerkungen besonders hervorheben zu können.

Danksagung: Die im Rahmen dieses Beitrags dargestellten Arbeiten wurden am Mitteln des BMBF und des ESF Programms der EU durch das Projekt KOSMOS finanziert.

Literatur

[1] Hepper, S., und Hesmer, S. (2003): Introducing the Portlet specification. Java World Journal.

[2] Jastram, S. (2013): Analyse und Vergleich von Portal-Entwicklungstools am Beispiel des E-Learning. Bachelorarbeit im Studiengang Wirtschaftsinformatik. Universität Rostock.

[3] Sandkuhl, K. (2005): Wissensportale. Informatik-Spektrum, 28(3), 193-201.

[4] Coates, H., James, R., & Baldwin, G. (2005): A critical examination of the effects of learning management systems on university teaching and learning. Tertiary education and management, 11, 19-36.

[5] Watson, W. R., & Watson, S. L. (2007): What are learning management systems, what are they not, and what should they become?. TechTrends, 51(2), 29.

[6] Atkins, D. (2003): Revolutionizing science and engineering through cyberinfrastructure: Report of the National Science Foundation blue-ribbon advisory panel on cyberinfrastructure.

[7] Borchardt U., Sandkuhl K. (2014): Wahrnehmung der Online-Unterstützung in der wissenschaftlichen Weiterbildung in: KOSMOS (Hrsg.) "Die Wahrnehmung der wissenschaftlichen Weiterbildung an der Universität Rostock. Eine qualitative Untersuchung der Studienformate Gartentherapie und Inklusive Hochbegabtenförderung.".

[8] Sandkuhl, K.; Stamer, D.; Borchardt, U. (2015): Portaleinsatz in der wissenschaftlichen Weiterbildung: Erfahrungen und Leitfaden. Universität Rostock, März 2015.

[9] Jastram, S. (2013): Analyse und Vergleich von Portal-Entwicklungstools am Beispiel des E-Learning. Bachelorarbeit im Studiengang Wirtschaftsinformatik. Universität Rostock.

[10] Ackermann, S. (2014): Bildung von Nutzerprofilen aus dynamischen Nutzungsdaten im Lernportal des Kosmos Projektes. Masterarbeit im Studiengang Wirtschaftsinformatik. Universität Rostock.

[11] Weigel, T. (2014): Joint Editing support in academic further education – Cloud Linkage for myKosmos. Masterarbeit im Studiengang Wirtschaftsinformatik. Universität Rostock.

[12] Dillenbourg, P. (1999).:What do you mean by collaborative learning?. Collaborative-learning: Cognitive and Computational Approaches., 1-19.

[13] Laal, M. and S. Ghodsi (2012): Benefits of collaborative learning, Procedia - Social and Behavioral Sciences, Volume 31, 2012, Pages 486-490.

[14] Wilson, J.; Goodman, P. and M. Cronin (2007): Group Learning. The Academy of Management Review, Vol. 32, No. 4 (Oct., 2007) , pp. 1041-1059.

[15] Weigel, T. (2013): Analyse und Konzeption des Identity Managements für Cloud Services am Beispiel iSM. Bachelorarbeit im Studiengang Wirtschaftsinformatik. Universität Rostock.

[16] Bellas, F. (2004): Standards for Second-Generation Portals. IEEE Internet Computing, March/April, pp. 54–60.

[17] Schelp, J., Winter, R. (2002): Enterprise Portals und Enterprise Application Integration. HMD 225, pp. 6–20.

[18] Sandkuhl, K. (2008): Information Logistics in Networked Organizations: Selected Concepts and Applications. Enterprise Information Systems, 9th International Conference, ICEIS 2008. LNBIP, Springer.

[19] Goldkuhl, G.; Lind, M. and Seigerroth U. (1998) : Method integration: the need for a learning perspective. IEEE Software 145(4):113–118.

[20] Lundqvist, M.; Sandkuhl, K.; Seigerroth, U. and E. Holmquist (2010) : IDA User Guide - Handbook for Information Demand Analysis. Version 2.0. InfoFLOW project deliverable. Technical Report. Jönköping University, Sweden.

[21] Frank, U. (2009): Die Konstruktion möglicher Welten als Chance und Herausforderung der Wirtschaftsinformatik. In: Becker, J.; Krcmar, H.; Niehaves, B. (Hrsg.) Wissenschaftstheorie und gestaltungsorientierte Wirtschaftsinformatik. Physica-Verlag: Heidelberg 2009, S. 167-180.

[22] Hevner, A. R.; March, S. T.; Park, J.; Ram, S. (2004): Design Science in Information Systems Research. In: MIS Quarterly 28 (2004), Nr. 1, S. 75-105. und Peffers, K.; Tuunanen, T.; Rothenberger, M.A. (2007); Chatterjee, S.: A Design Science Research Methodology for Information Systems Research. In: Journal of Management Information Systems, Volume 24 Issue 3, Winter 2007-8, pp. 45-78.

[23] Borchardt U., Sandkuhl K. (2014) : Wahrnehmung der Online-Unterstützung in der wissenschaftlichen Weiterbildung in: KOSMOS (Hrsg.) "Die Wahrnehmung der wissenschaftlichen Weiterbildung an der Universität Rostock. Eine qualitative Untersuchung der Studienformate Gartentherapie und Inklusive Hochbegabtenförderung in KiTa und Grundschule.".

[24] Sandkuhl, K. and U. Borchardt (2014): How to identify the relevant elements of “context” in Context-aware Information Systems? 13th International Conference on Business Informatics Research (BIR 2014), September 22-24, 2014, Lund (Sweden). LNBIP, Springer Verlag.

[25] Grabis, J.; Sandkuhl, K.; Stamer, D. (2015): Collaborative Teaching of ERP-Systems in International Context. International Conference on Enterprise Information Systems (ICEIS 2015). Accepted as full paper for publication in proceedings. INSTICC. May 2015, Barcelona, Spain.

[26] Melinat, P., Kreuzkam, T., & Stamer, D. (2014) : Information Overload: A Systematic Literature Review. In B. Johansson, B. Andersson, & N. Holmberg, (Vol. 194, pp. 72–86). 13th International Conference on Perspectives in Business Informatics Research BIR, Lund, Sweden.

[27] Stamer, D., Ponomarev, A., Sandkuhl, K., Shilov, N., & Smirnov, A. (2014): Collaborative Recommendation System for Improved Information Logistics: Adaption of Information Demand Pattern in E-Mail Communication. In K. Sandkuhl & U. Seigerroth, (pp. 35–48). Presented at the Proceedings of the 7th International Workshop on Information Logistics and Knowledge Supply co-located with the International Conference on Perspectives in Business Informatics Research BIR, Lund, Sweden.

Flexible Process-Aware Information Systems Deficiency Management in Construction

Sarah Gessinger, Ralph Bergmann

University of Trier, Business Information Systems II, 54296 Trier, Germany
{gessinge,bergmann}@uni-trier.de
www.wi2.uni-trier.de

Abstract. Deficiency management (DM) is an important subfield of the construction domain which is characterized by a high demand for immediate and flexible reactions to unexpected problems. Thus, there is a high potential for flexible process-aware information systems. We propose a deficiency management system (DMS) to support the DM process in a flexible manner supported by knowledge-sharing of best-practice processes. We acquired a set of requirements concerning process support and knowledge sharing for DMS and present first steps towards the development of a working prototype.

1 Introduction

To retain the competitive advantage of today's companies, the streamlining of business processes is increasingly important to develop new performance-enhancing features, to accelerate the internal efficiency, and to reduce costs [1,17]. Moreover, the economic success of a company heavily depends on its ability to flexibly respond to changes in its environment and to take advantage from arising opportunities. Hence, the ability to quickly change processes or to deviate from a pre-set course of action is essential. As a consequence process-aware information systems (PAISs) are a desirable technology in many domains as these systems support the operational business of a company based on models of the organisation and its processes [17]. PAISs include traditional workflow management (WFM) systems as well as modern business process management (BPM) systems. Current research particularly addresses approaches for increasing the flexibility of PAIS [1,4]. Recent research in PAIS has also recognized the need for knowledge management through process reuse from best-practice process collected in repositories [1]. Knowledge sharing and reuse becomes a central prerequisite for enabling process flexibility, in order to address the increased need for decision making on the process level.

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

Deficiency Management (DM) is an important subfield of construction domain that particularly needs to deal with unforeseen changes, demanding high flexibility by all involved parties. Generally speaking, a deficiency in construction is a negative deviation of the actual state of construction of a building from the specified or expected conditions [15, p. 5]. Thus, a deficiency is always unexpected and requires immediate remedial actions, which lead to changes of the current plans. Therefore, we expect that there is a high potential for flexible PAIS in the field of DM [7]. In this paper, we derive a set of requirements concerning process flexibility and process reuse in deficiency management systems (DMS) that are relevant for future more advanced approaches. We present a related concept as well as first steps towards the development of a working prototype.

2 Flexibility in Process-Aware Information Systems

A PAIS is "a software system that manages and executes operational processes involving people, applications, and/or information sources on the basis of process models" [4]. In order to operationalize process models, a PAIS typically includes a WFMS as a generic component for the execution of workflows. Traditional WFMSs strictly separate build time and run time of a workflow. During build time, a workflow definition is created to operationalize a business process (or a part of it). During run time, this workflow definition is repeatedly instantiated to execute the occurring business cases in exactly the same manner over and over again. For about 15 years, various approaches are discussed to address the flexibility needs of PAIS [5,16,13,17]. Schonenberg et al. [17] present a classification of flexibility approaches into four types: flexibility by definition, flexibility by change, flexibility by deviation, and flexibility by under-specification. *Flexibility by definition* refers to the ability to consider alternative execution paths in the process model during the modelling (build time). Traditional WFMS already support this type of flexibility. It can only take into account foreseen and predictable events and changes. *Flexibility by change* describes approaches that permit changes of process definitions and/or instances during run-time while maintaining consistency. *Flexibility by underspecification* refers to the ability to execute process descriptions which are not fully specified. Thus, certain decisions can be deferred to an appropriate point in time during process execution. Late modelling and late binding are two techniques used for this purpose. *Flexibility by deviation* refers to the forth and so far only rarely explored class of approaches that offers the possibility that the real-world process execution differs from the modelled process without the need to modify the process definition in advance.

An essential characteristic of all flexibility approaches is the fact that process modelling and execution are not strictly separated any more as in classical workflow systems [1,13]. Thus, a modification or a late modelling during runtime can be considered a re-modelling of the workflow that immediately effects its execution. However, modifying a workflow requires significant skills in the domain as well as in process modelling. Decisions must be taken concering how the workflow is modified and how this modification is formalized in the underlying workflow modelling language. Hence, methods are required that support users in performing such modifications. ADEPT/AristaFlow [16] and CAKE [3,13] are two advanced workflow systems which support flexibility by change and underspecification and which include methods to support users in reusing best-practice workflows.

3 Introducing Process Flexibility into Deficiency Management Systems

Based on an analysis of current DMS in construction available on the German market in summer 2014 [6] and four interviews with construction experts, we now derive a set of requirements for future more advanced DMS. We mainly focus on those requirements that related to the support by flexible PAIS. Additionally, we show how these requirements can be implemented based on the generic framework CAKE for integrated process and knowledge management. To provide a general understanding of the DM, we sketch the main processes and characteristics of this domain firstly.

3.1 Deficiency Management in Constructions

The entire DM process consists of several sub-processes addressing different steps of the overall DM process. It begins with logging a deficiency. General information like issuer name and company are recorded, as well as specific information such as deficiency description, floor, space, required action, and additional issuer notes. Motzko and Racky [14] point out the importance of a comprehensive record keeping of the deficiency in a centralized and dedicated area due to regulatory demands and further business needs. The entered information is verified and complemented by a responsible person inside the construction company. Next, the particular kind of deficiency must be identified. For example, deficiencies like wall cracks are described by different criteria such as appearance (e.g. a single or a bundle of cracks), crack width, or possible impacts such as impairment of the structural integrity. Depending on the reported deficiency, a visual on-site inspection is necessary in order to define the correct deficiency type and to decide an applicable rectification method. Further, it has to be check if it is eligible for deficiency rectification (DR) under warranty, for instances, if the deficiency was caused by liable a subcontractor. In addition the DR has to be assigned to construction workers for execution. The processing of the DR has to be controlled until its completion to ensure the timely finalization in accordance with the contract. Therefore, the accurate tracking of all open DRs is important. Overall, DM must incorporate all involved parties, including all subcontractors [14]. DMS aims at supporting DM by providing IT support for the relevant activities involved in DM. Today, a large variety of DMS exists, with different strengths and weaknesses. Thus, we analysed the process flexibility approaches implemented by these systems and determined that more advanced flexibility approaches than flexibility by definition are not implemented in any of the investigated systems [6].

3.2 User Requirement

The below described requirements have been derived based on the results of the interviews, also considering general regulatory requirements in DM [8] where necessary. We illustrate the most important requirements by examples, providing typical use cases of an envisioned future DMS.

R1: Support for process flexibility. Future DMS must enable flexibility by change and by underspecification. In particular, ad-hoc changes of workflow instances must be supported as well as late binding. An additional argument in favour of these methods is the fact that process instances must remain aligned with the activities in the real world in order to support tracing of DM processes as well as their accurate documentation. Further, flexibility by definition is required due to the high process variability. *Example:* An assigned construction worker receives a work order to fix a wall crack based on a process for elastic sealing. On the construction site s/he investigates the cracks and after cleaning s/he recognises that there is no need for an elastic sealing as due to drying shrinkage it is likely that the cracks will stabilize. Therefore, a structural strengthening non-elastic injection system is more effective. Using a mobile device, the worker immediately modifies the running process. S/he deletes those activities that are not relevant any more, i.e., “Prepare elastic sealing material” and “Apply two layers with the injection system”. Then, s/he adds the activity “Prepare non-elastic injection material” followed by the activity “Inject material into crack” at the appropriate position in the workflow instance.

R2: Collaboration support and role-based access control. A process-oriented collaboration platform is needed for coordinating all activities involved in DM and for supporting the necessary communication and documentation needs. As a large variety of parties and persons are involved in DM, such a platform requires a role-based access control. The access control must enable a detailed control of access rights for all resources, in particular on the level of individual tasks. *Example:* The investigation of a reported deficiency yields that it was caused by a liable subcontractor. Hence, several tasks of the DR process must be assigned to some of the subcontractor’s employees. Therefore, the responsible project leader from the subcontractor accesses the DMS. Due to her/his access rights s/he is able to make the respective assignments, but only for those tasks, his company is in charge of.

R3: Knowledge-sharing of best-practise processes. Future DMS supporting process flexibility should also support knowledge-sharing of best-practice workflows. Successfully finished DM processes should be captured and stored in a repository. Further, the reuse of best-practice workflows should be supported, thus asking for appropriate means for navigation and search in the repository. This requirement particularly arose in the interviews and is considered a means to improve efficiency and quality in the context of the large variability in DM. *Example:* After the successful termination of the DR process for repairing a crack, the project leader can store the particular workflow instance as a best-practise workflow in a repository. During this process, all case-specific data is removed and the workflow is generalised towards a workflow definition. At some later point in time, a similar type of crack must be fixed. By search in the repository, the previously stored workflow definition is found. It can be instantiated (and adapted if necessary).

In addition to these three requirements addressing process flexibility and closely related aspects, several further, more general requirements must be met as well by future DMS. For example, the usability of such a system must be particularly ensured by an intuitive user interface enabling to control the flexibility functions of the PAIS. Also a simple graphical modelling language for workflows is required to enable staff from construction companies to perform workflow modelling without the need to involve their IT personnel. Further, providing access to the DMS via mobile devices is important, as deviations from planned DR processes are mostly discovered at the construction site.

3.3 CAKE – Collaborative Agile Knowledge Engine

We now briefly describe CAKE, a generic framework for integrated process- and knowledge management [3] and explain how it could be used to fulfil the identified requirements. CAKE integrates an agile workflow engine with a so-called knowledge engine that supports process reuse as a particular kind of knowledge-sharing. The agile workflow engine is used for the enactment of agile workflows and supports their collaborative modelling and adaptation in a consistent manner. The workflow engine enables flexibility by change as it allows users to model and change workflow definition and instances at any time, provided that the user is granted the respective access right. Further, a simple graphical modelling language is available that allows modelling, execution monitoring, and adaptation of workflows within a browser-based editor. The modelling language includes placeholder task, thus late binding is supported. So, CAKE is in line with requirement R1.

The purpose of the knowledge engine is to support users in finding, defining, and adapting workflows according to their current needs. The knowledge engine implements a process-oriented case-based reasoning (CBR) method [12]. CBR is an established AI methodology for problem solving based on the assumption that similar problems have similar solutions [2]. The CAKE knowledge engine maintains a repository of workflows which can be semantically annotated using terms from a domain ontology. It supports workflow reuse by similarity-based retrieval of workflows. Thus CAKE is in line with requirement R3.

Finally, the CAKE framework consists of a storage layer that implements a role-based access control mechanism for all resources managed by CAKE, in particular workflows, tasks, documents, services, etc. The access control mechanism is a decentralized discretionary access control with subject-object relationships specified in access control lists. Thus CAKE is in line with requirement R2. The overall CAKE software is implemented as Web-based system. The client user interfaces enable access to all workflow related functions such as workflow modelling, execution, similarity-based retrieval, and adaptation using a standard browser. Further, the CAKE Server API also allows mobile applications to directly connect to CAKE, e.g., to support the mobile execution of tasks on an Android-based device.

3.4 Applying CAKE for Building a Prototype for Deficiency Management

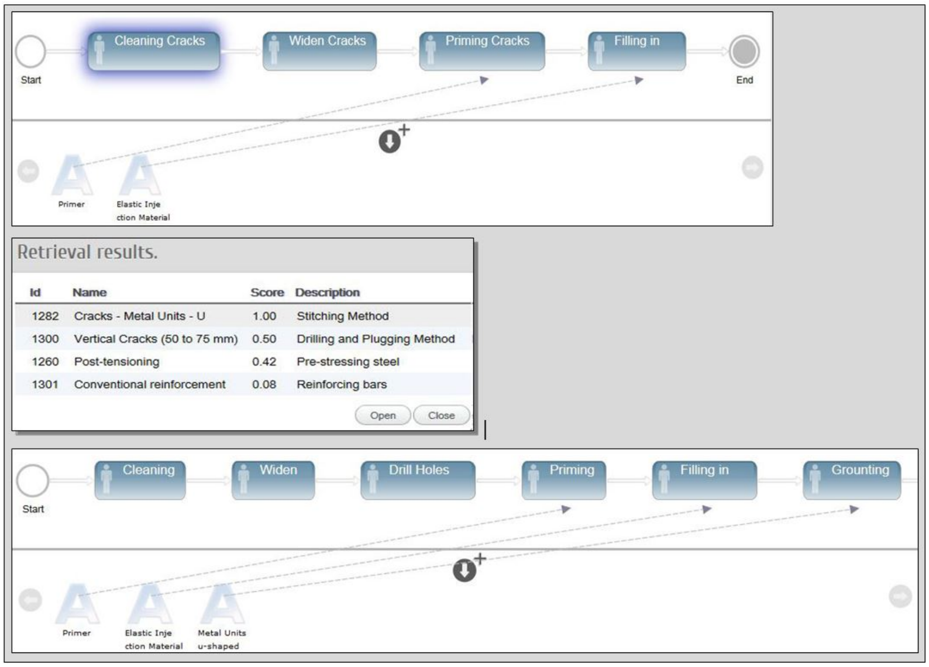
We build the first version of a prototype for demonstrating and evaluating the benefits of process flexibility in DM. For this purpose, we selected a subfield of frequently occurring deficiencies in construction, namely cracks in facades and masonry[1]. We collected technical background documentation and a set of process descriptions of respective DR processes from the construction companies contacted during the interviews. Based on this documentation, we developed an ontology of the relevant DM tasks and building materials, as well as a deficiency ontology to classify and describe different kinds of cracks. Further, we formalized the provided process descriptions using the CAKE workflow editor and thereby we created a repository of initial best-practice workflows. The resulting domain-specific CAKE instance can then be used to support the DR process as illustrated in the following use case.

Use Case: A series of cracks is a reported to the project manager of a construction company. An initial assessment of the cracks takes place, leading to a description of the cracks w.r.t. the deficiency ontology. Based on this description, the project manager searches for applicable workflows in the best-practice repository. S/he selects a DR workflow for “elastic sealing of cracks”, starts a new workflow instance for this case, and assigns a construction worker to the repair tasks. While performing the first activity of the workflow, which is the cleaning of the cracks, the worker recognises that most of the cracks exceed the maximum size allowed for sealing with a flexible injection system. To clarify which alternative method could be used, s/he searches for similar workflows in the best-practice repository. The knowledge engine of CAKE retrieves several workflows which are similar to the present workflow but which are in addition suitable for larger cracks (see Fig. 1). After inspecting the proposed workflow s/he decides to apply the method of stitching instead of sealing for repairing the crack. Thus, the workflow editor of CAKE is used to adapt the workflow to include the new activities “drill holes on both sides of the crack” and “grout in u-shaped metal units”. The worker now follows the adapted workflow to complete the repair, guided by the CAKE workflow engine.

4 Conclusion, Related, and Future Work

In this paper, we have revealed that DM in construction is a new and very promising application area for flexible PAIS. Due to the fact that deficiencies

[1] See http://theconstructor.org/concrete/methods-of-crack-repair/886/.



Retrieval results.

Id	Name	Score	Description
1282	Cracks - Metal Units - U	1.00	Stitching Method
1300	Vertical Cracks (50 to 75 mm)	0.50	Drilling and Plugging Method
1260	Post-tensioning	0.42	Pre-stressing steel
1301	Conventional reinforcement	0.08	Reinforcing bars

Fig. 1. Screenshots from the DM prototype built using CAKE

always occur unplanned and often require an immediate remedial action, it is essential that resulting process changes are performed during runtime. High process flexibility ask for flexibility approaches by change and/or underspecification. We have presented a concept and an initial prototype for a PAIS that addresses these needs based on the prototypical generic software system CAKE for integrated process and knowledge management and briefly demonstrate its feasibility by a first use case.

The application of process management and workflow systems in the construction industry is discussed for more than ten years in the scientific literature. Rüppel and Klauer [9] developed a workflow application that supports construction projects, but they do not consider the needs of process flexibility related to foreseen and/or unforeseen changes. Körten [11] comprehensively investigates the use of information technology to support construction processes. The BauVoGrid project aims at developing a grid-based framework for supporting construction processes. It also includes workflow components as well as semantic technologies. The project results demonstrated in the field of DM [10]. However, process flexibility is not addressed by this project. Although the investigation of unforeseen changes within process execution in construction was already considered in 2005 [9, p. 118], we are not aware of any in-depth research on this topic so far.

Future work will focus on tailoring CAKE for DM and on developing a full prototype that can be demonstrated to the construction industry. Based on such a prototype, more detailed case studies concerning the usability and potential benefits in DM can be performed. Another potential direction of future research is the investigation of new approaches for flexibility by deviation, which have the potential to void the need for explicit workflow adaptation. Such approaches might be able to better support processes in DM where only the purpose is known in advance, but not the precise order of steps that need to be executed.

References

1. Van der Aalst, W.: Business process management: A comprehensive survey. ISRN Software Engineering (2013)
2. Bergmann, R.: Experience Management: Foundations, Development Methodology, and Internet-Based Applications, LNAI, vol. 2432. Springer (2002)
3. Bergmann, R., Gessinger, S., Görg, S., Müller, G.: The Collaborative Agile Knowledge Engine CAKE. In: Proceedings of the 18th International Conference on Supporting Group Work. pp. 281–284. GROUP '14, ACM, New York, NY, USA (2014)
4. Dumas, M., van der Aalst, W., ter Hofstede, A.: Process-aware Information Systems: Bridging People and Software Through Process Technology. John Wiley & Sons, Inc., New York, NY, USA (2005)
5. Fleischmann, A., Schmidt, W., Stary, C., Augl, M.: Agiles Prozessmanagement mittels Subjektorientierung. HMD Praxis der Wirtschaftsinformatik 50(2), 64–76 (2013)
6. Frei, K.: Mängelmanagement im Bauwesen – Marktübersicht von Software zum Mängelmanagement und Nutzenpotenziale agiler Prozessunterstützung. Master thesis, University of Trier (2014)
7. Gessinger, S., Bergmann, R.: Potentialanalyse des prozessorientierten Wissensmanagement für die Baubranche. In: Henrich, A., Sperker, H.C. (eds.) LWA 2013. Lernen, Wissen & Adaptivität ; Workshop Proceedings. pp. 212–219 (2013)
8. Hechtl, A.: Kooperation im Bauwesen. Bautechnik - Zeitschrift für den gesamten Ingenierbau 79(6), 396–401 (2002)
9. Klauer, T.: Eine prozessorientierte Kooperationsplattform für Bauprojekte auf Basis eines internetbasierten Workflow-Managements. Ph.D thesis, Shaker (2005)
10. Klose, S., Angeli, R., Dollmann, T., Ernst, T., Fellmann, M., Hilbert, F., Hoheisel, A.: Spezifikation der mobilen Workflow Enactment Engine, der entwickelten Referenzprozessmodelle und der mobilen Dienste. Tech. Rep. BauVOGrid-Bericht A-5.1 (2010)
11. Körtgen, M.: Optimierungsansätze zur prozessorientierten Abwicklung komplexer Baumaßnahmen unter Einsatz neuer Informations- und Kommunikationssysteme. Ph.D thesis, Kassel University Press, Kassel (2010)
12. Minor, M., Montani, S., Recio-García, J.A.: Process-oriented case-based reasoning. Information Systems 40, 103–105 (2014)
13. Minor, M., Tartakovski, A., Schmalen, D., Bergmann, R.: Agile Workflow Technology for Long-Term Processes - Enhanced by Case-Based Change Reuse. In: Methodological Advancements in Intelligent Information Technologies: Evolutionary Trends, pp. 279–298. IGI Global (2010)
14. Motzko, C., Racky, P.: Erfolgsfaktoren für eine reibungslose Abnahme. Baumarkt und Bauwirtschaft 101, 30–33 (2002)

15. Oswald, R., Abel, R.: Leitfaden über hinzunehmende Unregelmässigkeiten bei Neubauten. Bau-Verlag (1996)
16. Reichert, M., Weber, B.: Enabling Flexibility in Process-Aware Information Systems: Challenges, Methods, Technologies. Springer, Berlin-Heidelberg (2012)
17. Schonenberg, H., Mans, R., Russell, N., Mulyar, N., Aalst, W.v.d.: Process Flexibility: A Survey of Contemporary Approaches. In: Advances in Enterprise Engineering I, pp. 16–30. No. 10 in LNBIP, Springer Berlin Heidelberg (2008)

IR: Workshop on Information Retrieval

Introducing a Gamification Approach for Enhancing Web Search Literacy

Ioannis Karatassis and Sebastian Dungs

University of Duisburg-Essen, Germany
{karatassis,dungs}@is.inf.uni-due.de

Abstract. Web search engines provide a rich feature set to users that allows efficient satisfaction of information needs. Nevertheless, recent studies show that Internet users do not know how to use Web search engines effectively for satisfying information needs. The overall level of Web search literacy leaves a lot to be desired and most users tend to overestimate their abilities in the domain of Web search. In this paper, we introduce a gamification approach with the aim of promoting search literacy as well as the current state of our prototype application. We present plans for future work to answer whether gamification is a viable means to improve Web search literacy. Our goals include finding indicators to differentiate between low and high literacy users and running long-term user studies to investigate the sustainability of search literacy improvements.

Keywords: gamification, search literacy, Web search

1 Introduction

Search literacy denotes the ability to locate and access desired information with efficiency and effectiveness. It is, therefore, a subset of the much broader concept *information literacy* which also encompasses evaluation, reuse of information, and information synthesis. Instead of putting Web search on a level with information retrieval, we look upon it as a lifelong learning process that we aim to support in order to enable users orienting themselves in modern societies. Users employ Web search engines not only for answering trivial information needs but also trust in the machines and their own abilities when it comes to serious decisions, e.g., health related issues or financial concerns.

In this paper, we draw attention to the problems arising from deficiencies in aforementioned Web search literacy and introduce a potential means that aims at increasing Web search literacy beyond traditional training methods like courses or tutorials. Only few approaches exist in this regard. One notable exception being *A Google a Day*[1] which features a fact finding search quiz. Users are

[1] http://www.agoogleaday.com

encouraged to employ advanced features of the search engine to solve the tasks. While the system includes a scoring system and rewards fast task completion, it lacks key gamification elements like levels, achievement badges, or leaderboards. Users solve tasks themselves and can not compete with other users.

We developed a gamification framework that features different types of tasks (e.g., search and educational) to give users a deeper understanding of the functioning of Web search engines. At the same time, our users are to learn and develop skills that should help them in mastering their daily search tasks efficiently. Furthermore, we plan to use the presented system as a basis for long-term studies. The goal of these studies will be 1) to identify key factors that make a user actually Web search literate, 2) to measure whether Web search literacy was improved and by how much, and 3) to evaluate how sustainable these effects are.

The remainder of this paper is structured as follows: First, we take a closer look at related work regarding search literacy and gamification in Section 2. The gamified application is presented in Section 3, covering game modes and later focusing on the employed game design elements. In Section 4, we draw a conclusion and provide an outlook for future work.

2 Related Work

2.1 Web Search Literacy

A recent study by Stark et al. [1] revealed that Internet users tend to overestimate their capabilities in the domain of Web search. In fact, the overall Web search literacy leaves a lot to be desired and common Web search engine users even have problems with finding answers to yes-no questions [2]. Kogadoga et al. [3] refer to the problems that arise from being low literate based on a study: Participants with deficient Web search skills used to spend significantly more time to complete a search task in comparison to high literate users and were significantly less accurate. The main problem is that users do not know how to use Web search engines effectively for satisfying their information needs [4]. Referring to this, Web search engines offer no feedback for users beyond query completion or expansion that would help them in improving their skills.

In his recent talk at GamifIR'15 [17], Azzopardi raised the idea to make a nature-nurture distinction when it comes to search behaviour. While this is an interesting research approach, we expect nurture to a play considerable role in users' behaviour, allowing for potential improvements by promoting key search skills.

According to Fuhr [5], a Web search literate user needs to know appropriate search tactics and strategies in order to satisfy information needs effectively. Users should be aware of the basic functioning of Web search engines as well as the following key aspects:

Searchability In some cases where users try to find a specific open document in the browser through Web search engines, they fail since not all online resources are indexed. The language used in the search query, Website owner restrictions (e.g., the robots.txt), the document type, and the recency of Web pages are some of the reasons why resources can not be found.

Linguistic Functions A crucial problem in information retrieval is the language itself since every natural language is both vague and ambiguous. To deal with that problem and to represent a user's information needs as best as possible, Web search engines apply linguistic functions such as word normalization, lemmatization, and phrase identification to search queries and take composites and synonyms into account.

Query Language A specially designed language allows users to express complex information needs and leads to more specific results since the latter are restricted to a limited set. Search operators (e.g., Boolean operators, number ranges, facets, fields, and URL predicates) and search options (e.g., for restricting the time, place, language, and document type of result items) are commonly employed search features.

Ranking One search query leads to a set of results where the ranking is of utmost importance. Hochstotter et al. [6] found out that users tend to look at items on the first search result page and especially click on the first or second item. Result items below the fold are seldom clicked on. Hence, users have to formulate precise search queries to let search engines produce result sets where the most relevant items are located on the first page and preferably are visible without the need to scroll.

Strategies and Tactics The main goal of Web search engine users is to satisfy their current information need. Complex information needs require a series of search queries. Strategies are plans for performing a complex search whereas tactics denote single operations to advance searches. Bates [7] distinguishes between the following types of tactics: monitoring, file structure, search formulation, and term.

2.2 Gamification

In the book by Zichermann et al. [8] the term gamification is defined as the process of game-thinking and game mechanics to engage users and solve problems. The integration of game mechanics into non-game contexts invokes gameful and ludic experiences to motivate users in solving monotonous tasks or for training users in complex systems. Beyond that, the concept is a viable means to shape users' behavior and to enhance online services with motivational affordances [9]. At its worst, gamification is a "mindless slapping of points, badges and leaderboards [...] onto any boring and irrelevant activity in vain attempt to increase the corporate bottom-line" [15]. When a person performs activities driven by internal rewards, we say she is intrinsically motivated due to the enjoyment of the activity itself. In contrast, users acting based upon extrinsic motivation aim to earn external rewards or to avoid punishments. We focus on enhancing the intrinsic motivation of users since it is known to be associated with the quality of effort that people put into activities [16].

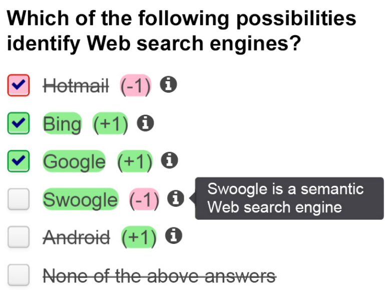
In [10, 11] university courses were gamified with great success. Gamification helped in improving lecture attendance, content understanding, problem solving skills, and general engagement of students. Achievement badges have been used by Hulinen et al. [12] to reward students for solving interactive tasks. Results show that the students' motivation has been enhanced even when the badges have had no impact on grading. Although there is still a lack of empirical evidence on the side effects of the employed game elements, these findings lead to the conclusion that gamification does not harm the intrinsic motivation at all if gamification is meaningful enough to the user and applied in a user-centered fashion [13]. Nevertheless, gamification designers should take social and contextual factors into account as they may determine whether the employed game elements diminish [14] or even suppress intrinsic motivation.

3 The Gamification Framework

3.1 Game Modes

Following the insights gained by literature review, we developed an application for improving search skills which appears to the user in the form of a game. We introduce the notion of *game mode* which emphasizes the playful character of the framework and summarizes a set of tasks of a specific nature. In total, we developed three game modes each of which aiming at a different aspect of Web search literacy: Quiz, Search Hunt, and Query Tuning.

Quiz The quiz mode features single and multiple choice question answering tasks (see Fig. 1). They allow for a deeper understanding of the functioning of search engines. The main goal of this game mode is to familiarize a user with advanced search engine functionality in a series of tasks. Furthermore, the quiz acts as a means to measure a priori search knowledge.



Which of the following possibilities identify Web search engines?

☑ ~~Hotmail~~ (-1) ℹ

☑ Bing (+1) ℹ

☑ Google (+1) ℹ

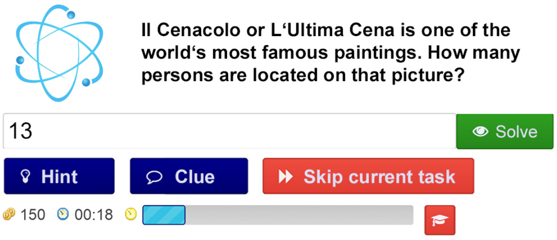
□ Swoogle (-1) ℹ Swoogle is a semantic Web search engine

□ ~~Android~~ (+1) ℹ

□ ~~None of the above answers~~

Fig. 1. The demonstration quiz task asks the user to select all items identifying Web search engines. Additional answer related information can be accessed by clicking on the respective information icon.

Search Hunt This mode comprises typical fact finding tasks (see Fig. 2). Users are asked to complete a task by issuing queries to one of the world's leading search engines which is directly included in the game by a proxy solution in order to find the solution being sought. Search hunt primarily trains users to formulate precise search queries, to identify relevant results, and to find the desired content within the document. Furthermore, it promotes the ability to judge accuracy of results. We exploit the search engine's rich feature set to provide a complete interface that contains all commonly employed search functions to our users. The interface allows us to train users on how to use specific features and can have more or less importance depending on the task. In addition, tasks will be designed in a way to familiarize users with commonly neglected search engine features and search strategies.



Il Cenacolo or L‘Ultima Cena is one of the world‘s most famous paintings. How many persons are located on that picture?

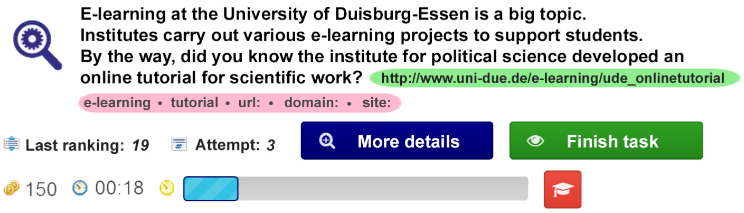
13

Fig. 2. In this task, users have to formulate a search query and to use certain functions of the search interface in order to find the required image that includes the solution. The task interface as seen in this figure offers hints and clues that users can request in exchange of points as well as a function for skipping the current task if desired. The hint can be a single word or a phrase that provides additional information and the clue reveals the first and the last letter as well as the length of the solution.

Query Tuning is comprised of precision oriented tasks (see Fig. 3). Users are again required to interact with the search engine but this time to produce a result set that contains a specific site at a top position. Along with the target site and a summary of the content comes a small set of search terms that are not allowed to be used in the query to avoid trivial solutions (e.g., querying for the URL of the site). Users formulate and reformulate queries until either the given site is ranked at the top position or the search performance can not be improved further. Hence, required skills for formulating precise queries are enhanced within a step-by-step refining process. The main goal of this mode is to form the understanding of ranking and to get a feeling how small changes in the query can yield to major differences in the result set.

3.2 Game Design Elements

The core game mechanics of the application consist of points, levels, badges, and leaderboards. Points are received for (partial) successful completion of tasks. The amount is determined by the degree of correctness, the current user level as well as the time needed to complete a task. Points act as the main performance indicator in the application. Levels are used to define the user's current state and represent a task's complexity. The next level up can be reached by exceeding



E-learning at the University of Duisburg-Essen is a big topic.
Institutes carry out various e-learning projects to support students.
By the way, did you know the institute for political science developed an online tutorial for scientific work? http://www.uni-due.de/e-learning/ude_onlinetutorial

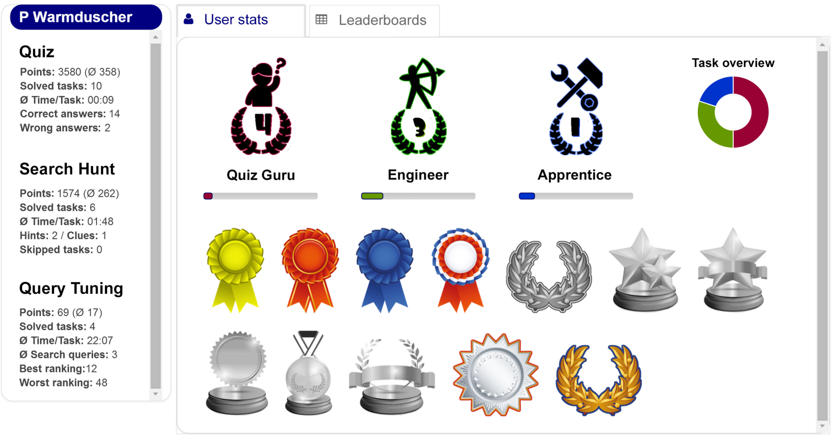
e-learning • tutorial • url: • domain: • site:

Last ranking: *19*
Attempt: *3*

150 00:18

Fig. 3. Users enter a search query in a text field which is forwarded via a search proxy to the connected Web search engine to produce a result set that holds the URL marked in green at a top position. The words marked in red are terms or search operators that must not appear in the search query. A user is free so close the task at any time. Points are calculated based on the last search process: The position of the given Website within the result list and the number of attempts have the biggest impact on scoring.

the corresponding point threshold. Badges are special rewards that are acquired either for reaching a certain state or for various actions. They may come as a surprise and with varying frequency and act as a motivator to explore the application. Furthermore, badges can be used to "show off" individual user skills to other users via a profile page (see Figure 4).



P Warmduscher

Quiz

Points: 3580 (Ø 358)
Solved tasks: 10
Ø Time/Task: 00:09
Correct answers: 14
Wrong answers: 2

Search Hunt

Points: 1574 (Ø 262)
Solved tasks: 6
Ø Time/Task: 01:48
Hints: 2 / **Clues:** 1
Skipped tasks: 0

Query Tuning

Points: 69 (Ø 17)
Solved tasks: 4
Ø Time/Task: 22:07
Ø Search queries: 3
Best ranking:12
Worst ranking: 48

Fig. 4. The user's profile page comprises game mode related statistics on the left side. The main area gives an overview about the effort achieved in each game mode by displaying the current level, a corresponding level description and the current score in the form of a progress bar respectively. While already collected achievement badges are depicted in color, the application displays all available badges to promote transparency.

Leaderboards are overviews of the top performing players in each game mode and are represented as ordered lists with a points score beside each name to allow simple comparisons and to engage users in competition. Again, these boards act as a motivational means for continuing as well as an instrument along with levels to indicate that users have more or less status or achievement in the game. Besides the core mechanics, the application features different sound effects to guide users and to introduce events, e.g., the beginning/completion of tasks and the receipt of awards. A comprehensive logging system collects user data in the background. The log data gives an insight into a user's behavior and thus can be used to create user profiles that reveal strategies and techniques being used to solve tasks.

4 Conclusion and Future Work

In this paper, we presented a gamification framework for Web search. The system in its current state features key gamification elements like points, badges, and leaderboards. Furthermore, three different game modes, i.e., types of tasks, are included. The system was tested regarding usability in a small user study ($N = 15$) with great success. The main goal of the system is to improve Web search literacy among general Web search users. We believe that this will allow for more time efficient and effective search sessions, which will lead to a higher task completion rate. To accomplish this goal, we will address various challenges that we are still facing with our prototype:

1. The actual search tasks and quizzes need to be tailored for the goal of improving literacy. Therefore, tasks will be created that are challenging for an average search engine user. Ideally, tasks should promote specific learning goals, e.g., search strategies or search engine features.
2. After a larger collection of tasks has been created, we will run a long-term study with a larger user base. This will not only allow us to tune game balance. The results will also act as a ground truth for future experiments. Observing many users completing the same tasks will allow us to "pool" solution attempts and to generate an ideal solution for every task. Individual users will then be scored by the closeness of their solution to the ideal one.
3. In the end, we will isolate key factors that make a user *Web search literate* and find means to specifically promote these skills in a gamified environment. The secondary goal of the long-term study will be to test the sustainability of literacy improvements. Therefore, we will invite participants to reuse the application with new tasks of similar complexity after a specific time period and compare the outcomes of their endeavors.

References

1. Stark, B., Dörr, D., Aufenanger, S.: The Googleization of information search - Search engines in the field of tension between usage and regulation. Management Summary (in German), 2014. Available online at `http://www.ifp.uni-mainz.de/Bilder_allgemein/Suchmaschinen_Management_Summary.pdf`; accessed 15-June-2015.
2. White, R.: Beliefs and biases in web search. In Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '13, pages 312, New York, NY, USA, 2013. ACM.

3. Kodagoda, N., Wong., B.L.W.: Effects of low & high literacy on user performance in information search and retrieval. In Proceedings of the 22Nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction - Volume 1, BCS-HCI '08, pages 173181, Swinton, UK, UK, 2008. British Computer Society.
4. Bateman, S., Teevan, J., White, R.W.: The search dashboard: How reflection and comparison impact search behavior. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '12, pages 17851794, New York, NY, USA, 2012. ACM.
5. Fuhr, N.: Internet search engines - Lecture script for the course in SS 2014 (in German), 2014. Available online at `http://www.is.inf.uni-due.de/courses/ir_ss14/ISMs_1-7.pdf`; accessed 15-June-2015.
6. Höchstötter, N., Lewandowski, D.: What users see – structures in search engine results pages. Inf. Sci., 179(12):17961812, May 2009. ISSN 0020-0255.
7. Bates, M.J.: Information search tactics. Journal of the American Society for Information Science, 30(4):205214, 1979.
8. Zichermann, G., Cunningham, C.: Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps. O'Reilly Media, Inc., 1st edition, 2011.
9. Deterding, S., Dixon, D., Khaled, R., Nacke, L.: From game design elements to gamefulness: Defining gamification. In Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments, MindTrek '11, pages 915, New York, NY, USA, 2011. ACM.
10. Iosup, A., Epema., D.: An experience report on using gamification in technical higher education. In Proceedings of the 45th ACM Technical Symposium on Computer Science Education, SIGCSE '14, pages 2732, New York, NY, USA, 2014. ACM.
11. O'Donovan, S., Gain, J., Marais, P.: A case study in the gamification of a university-level games development course. In Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference, SAICSIT '13, pages 242251, New York, NY, USA, 2013. ACM.
12. Hakulinen, L., Auvinen, T., Korhonen, A.: Empirical study on the effect of achievement badges in trakla2 online learning environment. In Proceedings of the 2013 Learning and Teaching in Computing and Engineering, LATICE '13, pages 4754, Washington, DC, USA, 2013. IEEE Computer Society.
13. Nicholson, S.: A User-Centered Theoretical Framework for Meaningful Gamification. Paper Presented at Games+Learning+Society 8.0, Madison, WI, June 2012. Available online at `http://scottnicholson.com/pubs/meaningfulframework.pdf`; accessed 15-June-2015.
14. Mekler, E.D., Brühlmann, F., Opwis, K., Tuch, A.N.: Do points, levels and leaderboards harm intrinsic motivation?: An empirical analysis of common gamification elements. In Proceedings of the First International Conference on Gameful Design, Research, and Applications, Gamification '13, pages 6673, New York, NY, USA, 2013. ACM.
15. Shovman, M.: The game of search: What is the fun in that? In Proceedings of the First International Workshop on Gamification for Information Retrieval, GamifIR '14, pages 4648, New York, NY, USA, 2014. ACM.
16. Ryan, R., Deci, E.: Intrinsic and extrinsic motivations: Classic definitions and new directions. Contemporary educational psychology 25, 1 (2000), 5467.
17. Kazai, G., Hopfgartner, F., Kruschwitz, U., Meder, M.: ECIR 2015 Workshop on Gamification for Information Retrieval (GamifIR'15). SIGIR Forum 49(1): 41-49 (2015)

A Machine Learning Framework to Detect And Document Text-based Cyberstalking

Zinnar Ghasem[1], Ingo Frommholz[1], and Carsten Maple[2]

[1] University of Bedfordshire,UK
[2] University of Warwick, UK
{zinnar.ghasem,ingo.frommholz}@beds.ac.uk
carsten.maple@warwick.ac.uk

Abstract. Cyberstalking is becoming a social and international problem, where cyberstalkers utilise the Internet to target individuals and disguise themselves without fear of any consequences. Several technologies, methods, and techniques are used by perpetrators to terrorise victims. While spam email filtering systems have been effective by applying various statistical and machine learning algorithms, utilising text categorization and filtering to detect text- and email-based cyberstalking is an interesting new application. There is also the need to gather evidence by the victim. To this end we discuss a framework to detect cyberstalking in messages; short message service, multimedia messaging service, chat, instance messaging and emails, and as well as to support documenting evidence. Our framework consists of five main modules: a detection module which detects cyberstalking using message categorisation; an attacker identification module based on cyberstalkers' previous messages history, personalisation module, aggregator module and messages and evidence collection module. We discuss our ongoing work and how different text categorization and machine learning approaches can be applied to identify cyberstalkers.

Keywords: Cyberstalking, digital forensics, email filtering, data mining, cyberharassment, machine learning, text categorisation

1 Introduction

With the proliferation of the use of the Internet, cyber security has become a major concern for users and businesses alike. While communication technologies have undoubtedly positively changed the way we communicate, it also provides cybercriminals with methods and techniques to be used for illegitimate purposes such as the distribution of offensive and threatening materials [25], spamming, phishing, cyberbullying, viruses, harassment and cyberstalking [18]. Cyberstalking is a complicated and pervasive problem, which affects and targets a huge

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

number of individuals [4], and unlike many other cybercrimes, cyberstalking does not occur on a single occasion [24], rather victims experience repeated, systematic and multiple attacks. Cyberstalking has been identified as a growing social problem [7], and a global issue [13], to an extent in which it is envisaged that almost twenty percent of people at one stages of their lives will become a victim of cyberstalking, where women will more likely become a victim than men [11]. In [12] Maple *et al.* have defined cyberstalking as a “course of actions that involves more than one incident perpetrated through or utilising electronic means that cause distress, fear or alarm”. There is evidence that cyberstalking will increase in both frequency and intensity [16].

While cybercriminals such as cyberstalkers utilise an array of technologies, tools and techniques like chat rooms, bulletin boards, newsgroups, instant messaging (*IM*), short message service (*SMS*), multimedia messaging service (*MMS*), and trojans, email is one of the most commonly used methods of cyberstalking [21, 13, 20, 15]. A cyberstalker can send emails, SMS, IM, MMS, and chat to threaten, insult, harass, or disrupt e-mail communications by flooding a victim’s e-mail inbox with unwanted mail [23, 22] anywhere at any time anonymously or pseudonymously, without fear of prosecution. This creates a new challenge for law enforcement and in digital forensic investigation. Anonymity in communication is one of the main issues exploited by cybercriminals [10]. Therefore, cyberstalkers could easily disguise themselves by spoofing email, and creating different pseudonym accounts mostly from free web mail providers. Similarly web based gateways are utilised to spoof *SMS* [5], and different anonymous chat IDs are easily created.

These techniques, coupled with the availability of remailers, unauthorised networks, public library’s computers, internet cafs, and free anonymous communications through websites, in addition to free and unregistered mobile SIM cards, inexpensive and unregistered mobile handsets, give an upper hand to cyberstalkers in their attack and complicate the investigation of cyberstalking cases [20]. Cyberstalking prevention with text messages filtering might not be as effective as required, because it does not always hold cyberstalkers accountable for their misuse of emails, and other text-based messages communication. Therefore identifying the original sender of emails, SMS, MMS, and chat is an important factor in the prosecution of an attacker [25].

The discussion so far shows that it is imperative to deal with cyberstalking on the very earliest stage. We will therefore in the remainder of the paper discuss a framework to detect cyberstalking in text-based messages and to support the collection of evidence for law enforcement. Text categorization and analysis plays a crucial role in our framework.

2 The Need to Detect and Document Text-based Cyberstalking

Text-based cyberstalking includes sending abusive, hate, threatening, harassing and obscene emails, SMS, chats, MMS, IM, including video and photo; sending email or MMS with the intention to spread viruses to a victim's device, either with attachments containing viruses or directing victims to a malicious website through a hyperlink; taking over victim's email account; sending high volumes of junk emails, SMS, MMS, Chat and IM. To minimise the effect of text-based cyberstalking, we propose a system that monitors, detects, captures and documents evidence. Such a system requires that we provide means to analyse textual documents like messages and gather some information from this analysis, for instance to determine the true authorship of emails when we cannot trust any email header information. We therefore discuss how text categorization and processing can be utilized for several aspects of this task.

Our work is inspired by [2] where the authors propose a system that simply records data within a session, that is duration of victim's computer connection to and disconnection from Internet. However, their system has major limitations in handling text-based cyberstalking. We therefore propose a framework to detect and filter messages and to collect and analyse evidence.

A further aim of our system is to assist the victims in documenting evidence for the initial complain process, as well as to help law enforcement in early stages of their investigation. In order to persuade authorities to investigate or prosecute a cyberstalker, the responsibility is often on the victim to produce such evidence [21, 3], thus, it is imperative that the victims save and keep all copies of communication whether email or other communications and with all their headers available and readable to be given to law enforcement [8, 14]. Such documentation will clearly demonstrate the course of incident and provide valuable information for both the investigation and prosecution process [17].

Therefore an automated system will not only make the initial complaint process and investigation easier but will also speedup investigations with less effort. Furthermore it will encourage victims to come forward and complain to prosecute a cyberstalker, because "cyberstalking and stalking's victim reporting is an important consideration for the criminal justice system, not only to guarantee that offenders are held accountable for their actions, but also to ensure that crime victims receive the support and services needed"[19].

3 The ACTS Framework

Our proposed framework is called *Anti Cyberstalking Text-based System (ACTS).* To the best of our knowledge it is the first framework that specialises on the automatic detection and evidence documentation of text-based cyberstalking. A prototypical implementation of the framework is under development, and the data collection process is ongoing. ACTS will, e.g., run on a user's device to

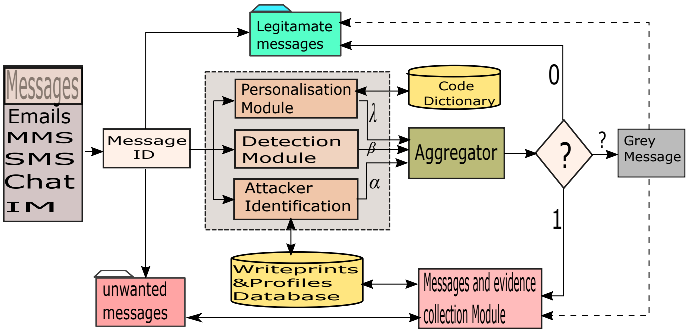


Fig. 1. ACTS Framework

detect text-based cyberstalking. The architecture of ACTS is depicted in Figure 1.

The proposed system utilises text mining, statistical analysis, text categorization and machine learning to combat cyberstalking. It consists of five main modules: detection, attacker identification, personalisation, Aggregator and messages and evidence collection.

ACTS first tries to detect cyberstalking based on a message ID list (the lists is optional for users and initially empty), which is automatically updated by the system. Messages whose IDs do not appear in the list are examined by the identification, personalisation, and detection modules; the results from three modules are passed to the aggregator for final decision.

Similar to some text categorization based email detection systems which can identify unwanted email, the *detection module* is employed to detect potential cyberstalking text based on their content. The received message is preprocessed by appying tokenisation, stop-word removal, stemming and presentation. Text mining techniques are utilised to extract required patterns from the message; a corresponding supervised algorithm like neural network/support vector machines is employed to detect and categorise emails to compute a value β based on three outputs, labelled as (00) not cyberstalking, (10) cyberstalking, and (01) grey email.

The *attacker identification module* is employed to identify whether received anonymous or pseudonymous messages are written by a cyberstalker or not, and to detect those messages from cyberstalkers where the message does not contain any known unwanted words. For this purpose, cyberstalker's writeprints including lexical, syntactic, structural and content-specific features [26] will be utilised. Unfortunately, due to character limitation of short messages like Twitter tweets, for e.g. SMS is limited to 160 $characters$ per message, writeprints might not always provide enough information to identify the author of the message. Nevertheless, because of their characters limitation, people tend to use unstandardised and informal language abbreviations and other symbols, which mostly depend on user's choice, subject of discussion and communities [9], where some of these abbreviations and symbols could provide valuable information in identify the sender. Thus to overcome this shortcoming and to enhance the identification process, we combines cyberstalker's writeprints with cyberstalker's profile including linguistic and behavioural profiles, utilising the existing cyberstalker's writeprints and profiles history in database.

Above considerations are based on the premise that, by definition, the victim must receive more than one attack to constitute cyberstalking. Thus there exist n messages where $n \in C_E\{m_1, ...m_n\}$ and $n \geqslant 2$ which will be used to check any new arriving message. Intuitively C_E will increase as the attack continues. A number of supervised algorithms have been used in authorship identification, where both authors and a set of their work are known prior to the identification process. Unfortunately, this is not the case in our approach. Identifying attackers is more challenging, firstly, because it will be implemented on user's device to detect messages, secondly, because we need to identify and detect the sender without prior knowledge. The system needs to decide whether a received message is written by a cyberstalker or not, thus supervised algorithms are not applicable. For this purpose, principal component analysis (PCA) could be employed to detect messages based on stylometrics and profiles; the new message's data is projected on PCA, and compared to the data matrix of all cyberstalking messages in C_E. The result is represented by the value α based on three outputs: not cyberstalking ($\alpha \geq r_2$), cyberstalking ($\alpha \leq r_1$) and grey ($r_1 < \alpha < r_2$). The α value is passed to the aggregator component. r_1 and r_2 are pre-defined threshold values in attacker identification (which have to be determined experimentally).

We also have to take into account that cyberstalking is often highly personalised; a bare word(s) or a phrase(s) of a message might have no inclination whatsoever towards bad feeling almost to anyone, but it might cause fear and distress to a cyberstalking victim. For instance, sending child birthday wishes may commonly be considered as positive, but not in case of somebody who lost their child or had undergone abortion. This complicates the process of developing a general tool to combat text-based cyberstalking. For this reason we define a *personalisation module* which is employed to enhance overall victim's control over incoming messages, where each victim can outline and define their own rule preferences. Therefore the personalisation module consists of rule based components and a code dictionary. The rule based component is optional, where rules are defined based on words, date and phrases provided by the user. A typical rule could be $if((date_A < today < date_B) \wedge (message\ contains\ "abc"))\ return\ true.$ A code dictionary is created from sentiment and affect word(s) or phrases, which are commonly used in cyberstalking. Furthermore, the code dictionary could also be updated by the user. The received message would be preprocessed, for this purpose *k-shingling* [6] could be utilised. Where each k-length shingle is run against the dictionary, probabilistic disambiguation[1] is another possible method to be used; both the dictionary's returned result and rule-based result are represented by the value λ: either cyberstalking (1) or not cyberstalking (0) (when both returned results are negative).

The final decision whether a received message is cyberstalking or not is made in the *aggregator module*, utilising the outcome from the previous modules. α, β and λ are the final calculated result values for each individual received email by the attacker identifier and detection module, respectively. Messages are identified as either grey (?), cyberstalking (1) or not cyberstalking (0) based on $\psi(\beta,\ \alpha,\ \lambda)$ as follows

$$\psi(\beta,\ \alpha,\ \lambda) = \begin{cases} 0 & \text{if } (\beta = 00 \wedge (\ \alpha \geq\ r_2)\ \wedge (\ \lambda = 0)\ )\ , \\ 1 & \text{if } (\beta = 10 \vee \alpha\ \leq\ r_1\ \vee\ \lambda = 1)), \\ ? & \text{if } (\beta = 01 \wedge (r_1\ < \alpha < r_2)\ \wedge\ \lambda = 0). \end{cases}$$

The final module is the *messages and evidence collection module*, which collects evidence from a newly arriving cyberstalking message, for instance, in the case of email the c source IP address or, if it is not available, the next server relay in the path, and the domain name (both addresses are automatically submitted to WHOIS and other IP geolocation website). The information with timestamp and email headers is saved for instance in the evidence database on the victims' device. The module also regularly updates and adds stylometrics, profiles and related information of the cyberstalking message to the database. Furthermore it will utilise statistical methods like multivariate Gaussian distribution and PCA to analyse the writeprint and profiles of cyberstalking, and text mining to extract similar features, attacker behavioural, greeting, farewell, etc, specifically between anonymous message and non anonymous message. The integrity and authenticity of a cyberstalking message, gathered evidence information whether saved on computer or through transmission are preserved using hash functions and asymmetric encryption keys.

4 Conclusions and Future Work

Combating cyberstalking is a challenging task, where technical solutions are a cornerstone in its prevention and mitigation. We therefore presented the ACTS framework that filters, detect and documents text-based cyberstalking. In the context of our framework we in particular discussed the potential use of textual machine learning approaches and discussed the difference to classical email categorization. The aim of our solution is not only to mitigate cyberstalking, but also to help victims in documenting evidence, which is required for law enforcement. Future work includes the implementation and evaluation of ACTS, in particular by measuring the effectiveness of the proposed text categorization methods in this emerging problem area.

References

1. A. Abbasi and H. Chen. Affect Intensity Analysis of Dark Web Forums. In *Intelligence and Security Informatics, 2007 IEEE*, pages 282–288. IEEE, 2007.

2. S. Aggarwal, M. Burmester, P. Henry, L. Kermes, and J. Mulholland. Anti-Cyberstalking: The Predator and Prey Alert (PAPA) System . In *Systematic Approaches to Digital Forensic Engineering, 2005. First International Workshop*, number iv, pages 195—-205. IEEE-CPS, 2005.
3. T. K. and et al Logan. Research on Partner Stalking: Putting the Pieces Together. *Lexington, KY: Department of Behavioral Science and Center on Drug and Alcohol Research, University of Kentucky*, pages 1–27, 2010.
4. M. Baer. Cyberstalking and the Internet Landscape We Have Constructed. *Virginia Journal of Law & Technology*, 15(154):153—-227, 2010.
5. A. Bose and K. G. Shin. On mobile viruses exploiting messaging and Bluetooth services. In *2006 Securecomm and Workshops*, pages 1–10. IEEE, 2006.
6. M. Chang and C. K. Poon. Using phrases as features in email classification. *The Journal of Systems & Software*, 82(6):1036–1045, 2009.
7. B. L. Ellison and Y. Akdeniz. Cyber-stalking: the Regulation of Harassment on the Internet. *Criminal Law Review*, 29:29–48, 2001.
8. J. Finn. A survey of online harassment at a university campus. *Journal of interpersonal violence*, 19:468–483, 2004.
9. J. M. Gómez Hidalgo, G. C. Bringas, E. P. Sánz, and F. C. García. Content based SMS spam filtering. In *Proceedings of the 2006 ACM symposium on Document engineering - DocEng '06*, pages 1–8. ACM, 2006.
10. R. Hadjidj, M. Debbabi, H. Lounis, F. Iqbal, A. Szporer, and D. Benredjem. Towards an integrated e-mail forensic analysis framework. *Digital Investigation*, 5(3-4):124–137, Mar. 2009.
11. D. A. Jurgens, P. D. Turney, and K. J. Holyoak. SemEval-2012 Task 2: Measuring Degrees of Relational Similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1*, pages 356–364. Association for Computational Linguistics, 2012.
12. C. Maple, E. Short, A. Brwon, C. Bryden, and M. Salter. Cyberstalking in the UK: Analysis and Recommendations. *International Journal of Distributed Systems and Technologies*, 3(4):34–51, 2012.
13. A. Maxwell. Cyberstalking. Technical Report 7, Auckland University, July 2001.
14. R. Mccall. Online Harassment and Cyberstalking: Victim Access to Crisis , Referral and Support Services in Canada Concepts and Recommendations. *Canada: Victim Assistance Online Resources*, page 17, 2003.
15. E. Ogilvie. The internet and cyberstalking. (December), 2000.
16. N. Parsons-pollard and L. J. Moriarty. Cyberstalking: Utilizing What We do Know. *Victims and Offenders*, 4(4):435–441, 2009.
17. D. A. Pinals. *Stalking, Psychiatric Prespective and Practical Approach*. 2007.
18. K. Reynolds, A. Kontostathis, and L. Edwards. Using Machine Learning to Detect Cyberbullying. In *Proceedings ICMLA 2011*, pages 241–244. IEEE, Dec. 2011.
19. B. W. Reyns and C. M. Englebrecht. The stalking victim's decision to contact the police: A test of Gottfredson and Gottfredson's theory of criminal justice decision making. *Journal of Criminal Justice*, 38(5):998–1005, Sept. 2010.
20. D. Robert and J. Doyle. Study on Cyberstalking: Understanding Investigative Hurdles. *FBI Law Enforcement Bulletin*, 72(3):10–17, 2003.
21. L. Roberts. Jurisdictional and definitional concerns with computer-mediated interpersonal crimes: An Analysis on Cyber Stalking. *International Journal of Cyber Criminology*, 2(1):271–285, 2008.
22. L. L. Sheridan and T. D. Grant. Is cyberstalking different? *Psychology, Crime & Law*, 13(6):627–640, Dec. 2007.

23. C. Southworth, J. Finn, S. Dawson, C. Fraser, and S. Tucker. Intimate partner violence, technology, and stalking. *Violence against women*, 13(8):842–56, Aug. 2007.
24. J. L. Truman. *Examining intimate partner stalking and use of technology in stalking victimization.* PhD thesis, University of Central Florida Orlando, Florida, 2010.
25. O. D. Vel, A. Anderson, M. Corney, and G. Mohay. Mining E-mail Content for Author Identification Forensics. *ACM Sigmod Record*, 30(4):55–64, 2001.
26. R. Zheng, J. Li, H. Chen, and Z. Huang. A Framework for Authorship Identification of Online Messages: Writing-Style Features and Classification Techniques. *JASIST*, 57(3):378–393, 2006.

Optimierung von Analytischen Abfragen über Statistical Linked Data mit MapReduce

Sébastien Jelsch[1], Benedikt Kämpgen[1] und Stefan Igel[2]

[1] FZI Forschungszentrum Informatik
sebastien.jelsch@fzi.de, kaempgen@fzi.de
[2] inovex GmbH
stefan.igel@inovex.de

Zusammenfassung. In den letzten Jahren ist die Menge der verfügbaren Linked Data im Web stetig gestiegen. Daher veröffentlichen immer mehr Provider ihre statistischen Datensätze als Linked Data, um sie mit weiteren Informationen anzureichern. Wir möchten in diesem Kurzbeitrag zu einer laufenden Arbeit eine Extract-Transform-Load (ETL) Pipeline vorstellen, die extrem große Mengen an Linked Data automatisiert in ein horizontal skalierbares Open Source OLAP-System bereitstellen kann.

Schlüsselwörter: Linked Data, Data Cube, Parallelisierung, MapReduce

1 Einleitung

In den letzten Jahren ist die Menge der verfügbaren Linked Data stetig gestiegen und immer mehr numerische Datensätze werden im Web mittels des RDF Data Cube Vokabulars (QB) als Linked Data veröffentlicht. Ein Vorteil besteht darin, die Bedeutung der numerischen Daten durch Verlinkung mit Zusatzinformationen näher zu bestimmen. Somit können beispielsweise Provenance-Informationen oder weitergehende Informationen (z.B. Anzahl der Mitarbeiter) hinzugefügt werden. Darüber hinaus können auch interne Daten mit den numerischen Daten verlinkt und zur Analyse verwendet werden.

Bevor Analysten jedoch in der Lage sind, Unternehmensleistungen vergleichen zu können, verbringen sie unverhältnismäßig viel Zeit mit der Identifizierung, Erfassung und Aufbereitung der relevanten Daten. Der Aufwand steigt mit der Anzahl der Datenquellen und damit unterschiedlichen Formaten oder Bezeichnungen für identische Objekte. Diese Prozesse müssen daher optimiert und möglichst automatisiert werden. Für entscheidungsunterstützende Analysen numerischer Datensätze bietet das Konzept OLAP (Online Analytical Processing) eine multidimensionale Betrachtung des Datenbestands.

In der Arbeit von Kämpgen und Harth [4] wurde ein Extract-, Transform- und Load-Prozess (ETL-Prozess) vorgestellt, der die statistischen Linked Data aus unterschiedlichen RDF Stores, unter Anwendung der Abfragesprache SPARQL und dem Cube-Vokabular QB, in ein multidimensionales Datenmodell transformiert. Ferner wurden die Informationen in diesem ETL-Prozess in einem relationalen Data Warehouse gespeichert. Auf diese Weise konnte mit der OLAP-to-SQL-Engine Mondrian [6] die Vorteile der multidimensionalen Abfragemöglichkeit und erweiterten Selektierbarkeit von OLAP-Anfragen mit MDX (Multidimensional Expressions) genutzt werden. Dieser Ansatz beinhaltet jedoch drei wesentliche Probleme:

(V1) Die Dauer des ETL-Prozesses bei großen Datensätzen mit vielen Zusatzinformationen ist nicht zufriedenstellend, da innerhalb der RDF-Daten die nötigen Informationen für das multidimensionale Datenmodell (Metadaten und Daten) herausgezogen werden müssen.

(V2) Bei einer Aktualisierung des Datenbestands oder bei neu hinzukommende Informationen muss der ETL-Prozess komplett neu durchgeführt werden.

(V3) Zusatzinformationen an den Datensätzen werden bei der Erstellung des multidimensionalen Datenmodells gefiltert und können bei analytischen Abfragen nicht berücksichtigt oder als Zusatzinformation abgefragt werden. Auch wenn das multidimensionale Datenmodell dementsprechend erweitert wird, können heterogene Zusatzinformationen nicht berücksichtigt werden.

In einer vorherigen Arbeiten [5] wurden die Daten in einem Triple-Store geladen, um analytische Abfragen mittels der graphenbasierten Sprache SPARQL auszuführen. Es zeigte sich, dass der Triple Store mit beliebigen RDF-Daten weniger effizient für analytische Abfragen geeignet ist als ein RDBMS mit Sternschema. Eine weitere Arbeit [3] beschäftigte sich mit der Optimierung eines Triple Stores durch horizontale Skalierung. Da NoSQL-Systeme für komplexe Operationen weniger geeignet sind, war die Ausführung der analytischen Abfragen nicht effizient genug. In der Arbeit von Abelló et al. [1] wurden analytische Abfragen auf MapReduce-basierten Systemen evaluiert. Dabei wurden die Vorteile von Big-Data-Technologien bei der Generierung eines OLAP Cubes für analytische Abfragen überprüft, jedoch ohne eine horizontale Skalierung durchzuführen. Grundlegend kann gesagt werden, dass diese drei Ansätze für die Analyse einer beliebigen Menge an RDF-Daten eine Herausforderung darstellen.

Bei der Analyse großer Datenmengen sind daher Big-Data-Technologien notwendig. Sowohl relationale Datenbanken, RDF Stores als auch OLAP Engines skalieren in der Regel nicht horizontal und besitzen daher eine natürliche Grenze bzgl. ihrer Datenspeicher- und Datenverarbeitungskapazität. Wir glauben, dass sich diese Beschränkungen mittels Parallelisierung über viele Rechner überwinden lassen. Mit Apache Hadoop sind derartige Technologien in einem Open Source Software Stack verfügbar. Bislang wurde nicht erforscht, ob eine enorme RDF-Datenmenge in einem automatisierten ETL-Prozess durch eine Umsetzung der Architektur von Kämpgen und Harth mit Hadoop-Komponenten für OLAP-Analysen bereitgestellt werden kann.

2 Aktueller Ansatz

Der hier präsentierte Lösungsansatz überführt Kämpgen und Harths Konzept in eine horizontal skalierende Architektur auf der Basis von Hadoop. Die nichtskalierbaren Komponenten, wie die RDF-Datenbank, die Abfragesprache SPARQL und die relationale Datenbank, werden dabei durch Technologien und Frameworks aus dem Hadoop-Ökosystem ersetzt. Abbildung 1 veranschaulicht die neue Gesamtarchitektur.

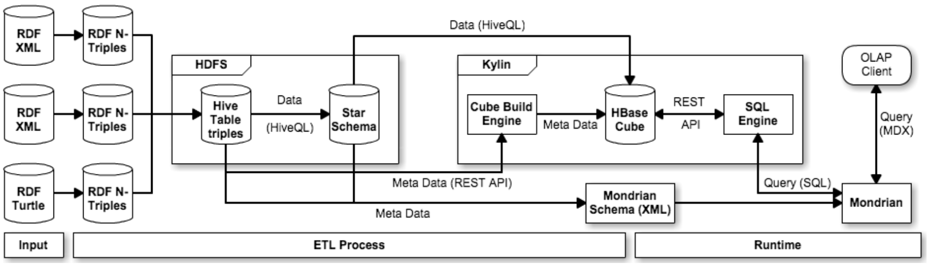


Abb. 1. Parallelisierungsarchitektur für analytische Abfragen auf Statistical Linked Data mit MapReduce-basiertem ETL-Prozess.

Die in verschiedenen RDF Stores abgelegten Linked Data werden in das N-Triples-Format umgewandelt und in das Hadoop Distributed File System (HDFS) geladen. Das zeilenbasierte N-Triples-Format ist besonders gut geeignet um die Daten in die Hive-Tabelle „triples“ mit den Spalten „subject“, „predicate“ und „object“ zu transformieren. Im Vergleich zur Arbeit von Cudré-Mauroux et al. [3] werden in unserem ETL-Prozess keine Property Tables generiert. Die vorkommenden Predicates werden beim Befüllen der Hive-Tabelle in Ordnern partitioniert. Dies optimiert Hive-Abfragen bei der Suche nach bestimmten Predicates, z. B. nach Measures (Predicate qb:measure). Unter Anwendung solcher Hive-Abfragen und MapReduce Jobs findet auf Basis der Definition des Cubes im QB-Vokabular eine Transformation der triples-Tabelle in ein relationales Datenmodell im Sternschema mit einer Fakten- und mehreren Dimensionstabellen statt. Die Cube Build Engine in Apache Kylin[2] erstellt aus dem Hive-Sternschema in mehreren MapReduce Jobs den OLAP Cube. Für die Speicherung der Cuboids ist die NoSQL-Datenbank HBase verantwortlich. In dieser spaltenorientierten NoSQL-Datenbank werden die verschiedenen Aggregationen der Cuboids gespeichert. HBase eignet sich, aufgrund der robusten Verarbeitung sehr großer Datenmengen und durch redundante, horizontale Verteilung durch das Ablegen der Daten ins HDFS, besonders gut als Speicherort der Cuboids.

Die SQL Engine in Kylin erlaubt das Absetzen von SQL-Anfragen an den Cube. Eine Ausführung von OLAP-Anfragen mit MDX ist bislang jedoch nicht möglich. Daher liegt ein weiteres Augenmerk dieser Arbeit in der Anpassung der OLAP-to-SQL Engine Mondrian. Im Mittelpunkt dieser Betrachtung steht die Kommunikation zwischen dem OLAP Client und Kylin, besonders vor dem Hintergrund, dass Kylin lediglich eine ANSI-SQL-Teilmenge verarbeiten kann.

3 Vorläufige Evaluation

Grundlage der Evaluation ist die von Kämpgen und Hart vorgestellten Arbeit [5], die wiederum auf den Star Schema Benchmark (SSB) aufbaut, um analytische Abfragemethoden über Statistical Linked Data zu evaluieren. Die Generierung einer beliebigen Datenmenge im Sternschema wird durch das TPC-H sichergestellt. Dies erlaubt eine Untersuchung größerer RDF-Datenmengen im Hinblick auf die geplante Architektur. Zusätzlich stellt SSB unterschiedliche analytische SQL-Queries zur Verfügung, die eine detaillierte und vergleichbare Evaluierung der Architektur ermöglicht. In [5] wurden diese SQL-Queries aufgelistet, diskutiert und in vergleichbare MDX-Queries umgewandelt.

In dieser Arbeit werden mithilfe der TPC-H Benchmark-Datengenerierung verschieden große Datenmengen sowohl hinsichtlich der Ausführungsdauer des ETL-Prozesses als auch der Anfragedauer bei analytischen Queries untersucht. Für die erste Evaluation verwenden wir SSB mit der Skalierung 1 (ca. 6.000.000 Datensätze). Dies entspricht einem Umfang von 4,4GB an RDF-Daten im QB-Vokabular. Unser Cluster besteht aus drei virtuellen Rechnern mit Ubuntu 12.04 LTS und jeweils 32GB Ram, wobei jeder MapReduce Job max. 8GB zugewiesen bekommt. Der Hauptknoten hat dabei vier CPUs mit 2,5GHz, die restlichen Knoten besitzen jeweils zwei CPUs mit 2,4GHz.

Die Umwandlung der RDF-Daten in das zeilenbasierte N-Triples-Format dauert auf dem Hauptrechner durchschnittlich 1147s. Die resultierenden N-Triples-Dateien haben eine Gesamtgröße von 16,6GB und der Umzug ins HDFS dauert durchschnittlich 224s (75,92 MB/s). Der ETL-Prozess zur Bewirtschaftung des Sternschemas dauert durchschnittlich 5748s und die Generierung des OLAP-Cubes in Kylin benötigt auf diesem Cluster 2257s.

Bei unserer vorläufigen Evaluation findet, bis auf die Umwandlung der RDF-Daten in das N-Triples-Format, bereits an jeder möglichen Stelle eine Parallelisierung statt. Nach dem Umzug der Daten in das HDFS werden alle restlichen Schritte durch verschiedene MapReduce Jobs parallel ausgeführt. Die Speicherung des OLAP Cubes in HBase führt zu einer horizontalen Verteilung der Daten.

Aufgrund der Verwendung von Calculated Members und der Einschränkung der SQL-Syntax ist zum jetzigen Zeitpunkt eine Evaluation der MDX-Abfragen Q1, Q2 und Q3 in Kylin nicht möglich. Ein Lösungsansatz dieses Problems besteht darin, die Multiplikation der Measures zur Laufzeit des ETL-Prozesses zu berechnen und in die Faktentabelle als neue Spalte zu speichern.

Die durchschnittliche Ausführungsdauer der MDX-Abfragen ist in Tabelle 1 aufgelistet. Obwohl die vorläufige Evaluation auf einem Cluster mit virtuellen Rechnern ausgeführt wird, lässt sich erkennen, dass alle MDX-Abfragen nach dem ETL-Prozess mit Kylin schneller ausgeführt werden als bei einer traditionellen Datenbank wie MySQL. Eine systematische Evaluation in Abhängigkeit der Clustergröße sollte einen noch deutlicheren Unterschied bei der Ausführung

	Q1.1	Q1.2	Q1.3	Q2.1	Q2.2	Q2.3	Q3.1	Q3.2	Q3.3	Q3.4	Q4.1	Q4.2	Q4.3	Total
MySQL	42,4	40,9	41,7	16,2	17,3	15,2	12,4	10,6	9,6	7,3	15,4	10,7	10,7	250,4
Kylin	N/A	N/A	N/A	4,0	9,2	1,9	3,1	4,0	3,1	2,2	5,3	3,8	5,1	41,7

Tabelle 1. Ergebnisse der Prä-Evaluationen mit Ausführungsdauer pro MDX-Abfrage

der Abfragen aufzeigen. Ferner soll der ETL-Prozess und die Analysedauer bei größeren und mit Hintergrundinformationen angereicherten Datensätzen untersucht werden.

4 Zusammenfassung

Die vorgestellte, durchgängig horizontal skalierende Architektur auf Basis von Hadoop liefert eine vielversprechende Lösung für die effiziente Speicherung und performante Verarbeitung großer Linked Data Volumina mit Hadoop. Sie beinhaltet einen Ansatz zur skalierbaren Transformation dieser Daten mittels MapReduce und Hive. Aufsetzend auf Hive und HBase stellt Kylin multidimensionale Datenstrukturen zur Verfügung, die die Analyse großer Volumina numerischer Linked Data mittels etablierter OLAP-Methoden und Tools ermöglichen. Eine folgende systematische Evaluation bezüglich Skalierbarkeit und Performanz muss die ersten Ergebnisse allerdings noch bestätigen. Die beschriebene Architektur erscheint grundsätzlich auch geeignet zur effizienten Aktualisierung des Datenbestandes und zur Ergänzung der numerischen Daten um heterogene Zusatzinformationen. Dies wird Gegenstand zukünftiger Forschungsarbeiten sein.

Literatur

1. Abelló, A., Ferrarons, J., Romero, O.: Building Cubes with MapReduce. In: Proceedings of the ACM 14th international workshop on Data Warehousing and OLAP. pp. 17–24. ACM (2011)
2. Apache: Kylin – An Open Source Distributed Analytics Engine (2015), http://kylin.incubator.apache.org/, aufgerufen am 11. September 2015
3. Cudré-Mauroux, P., Enchev, I., Fundatureanu, S., Groth, P., Haque, A., Harth, A., Keppmann, F.L., Miranker, D., Sequeda, J.F., Wylot, M.: NoSQL databases for RDF: an empirical evaluation. In: The Semantic Web–ISWC 2013, pp. 310–325. Springer (2013)
4. Kämpgen, B., Harth, A.: Transforming Statistical Linked Data for Use in OLAP Systems. In: Proceedings of the 7th international conference on Semantic systems. pp. 33–40. ACM (2011)
5. Kämpgen, B., Harth, A.: No Size Fits All – Running the Star Schema Benchmark with SPARQL and RDF Aggregate Views. In: The Semantic Web: Semantics and Big Data, pp. 290–304. Springer (2013)
6. Pentaho: Mondrian - Open Source Business Analytics Engine (2015), http://community.pentaho.com/projects/mondrian/, aufgerufen am 11. September 2015

Company Search — When Documents are only Second Class Citizens

Daniel Blank, Sebastian Boosz, and Andreas Henrich

University of Bamberg, D-96047 Bamberg, Germany,
firstname.lastname@uni-bamberg.de,
WWW home page: http://www.uni-bamberg.de/minf/team

Abstract. Usually retrieval systems search for documents relevant to a certain query or—more general—information need. However, in some situations the user is not interested in documents but other types of entities. In the paper at hand, we will propose a system searching for companies with expertise in a given field sketched by a keyword query. The system covers all aspects: determining and representing the expertise of the companies, query processing and retrieval models, as well as query formulation and result presentation.

Keywords: Domain specific search solutions, expertise retrieval

1 Motivation

The idea for the presented search engine came up when the association of IT companies in Upper Franconia tried to figure out the opportunities to set up a register of competencies for their members and associated companies. The basic idea was: Whenever a company with a demand for IT solutions is looking for a potential provider of services or solutions the “register of competencies” will easily expose good matches.

The first idea was to conduct a survey among the companies and derive a brochure or website. However, the foreseeable problems with low return rates as well as update and maintenance problems led to a broader consideration of potential approaches. The next thing that came into mind was “search engines”. A closer look at commercial and freely available candidates made clear that these approaches were not convincing for the intended target group. The main reason based on small experiments was the inappropriate result presentation. The results consisted of documents and in many situations it was rather unclear what qualified the documents for top ranks in the result list and—even more important—which company was associated with the document.

Consequently, the next step was the design of a special purpose search engine optimized for the “company search task”. In fact, this scenario can be envisaged

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

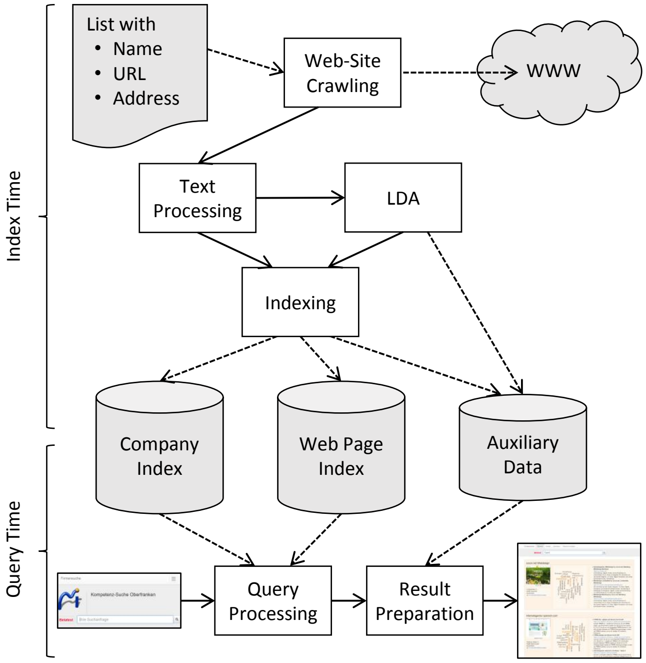


Fig. 1. System overview

as a special case of expertise retrieval—a topic intensely considered in literature [3]. The contribution of the paper at hand in this context is the design and reflection of a system based on state-of-the-art models and components adapted for a real world application scenario. As we will see this requires some peculiar design decisions and user interface aspects.

The remainder of the paper is organized as follows: In the two subsections of section 1 below we present a rough overview of the proposed system and we shortly address related work. Thereafter we discuss the four components identified to make up an expertise retrieval system according to Balog et al. [3, p. 145]: *modeling and retrieval* (section 2), *data acquisition* (section 3), *preprocessing and indexing* (section 4), as well as *interaction design* (section 5). Finally section 6 concludes the paper.

System Overview The basic assumption of the system is that a manually defined set of companies should be searchable in the system. These are the members and associated companies of the IT-Cluster Upper Franconia. A further assumption is that all these companies have a more or less expressive website. Hence, the starting point for the system is a list of companies, consisting of the name of the company, the URL of the corresponding website and the office address (represented in the upper left corner of Fig. 1).

The URLs are used to crawl the websites of the companies. Roughly spoken, each company c is represented by the concatenation of the content blocks of its web pages, called d_c. Of course, some text processing is necessary here for noise elimination, tokenizing, stemming, and so forth.

Since the corpus (consisting of about 700 companies at present) is rather small and queries might be specific (for example a search for “Typo3”) we incorporated topic models (using Latent Dirichlet Allocation currently) to boost companies with a broader background in the topic of the query. Using the terms and LDA-based boost-factors two indexes are build: In the first index the companies are indexed based on the pseudo documents d_c. In the second index the single web pages are indexed because we also want to deliver the best landing pages for the query within the websites of the ranked companies in the result. Finally some auxiliary data (for example the topic models generated via LDA) is stored since it is needed during query processing or result presentation.

When a query is issued the query is processed on the company index and on the web page index. Then a result page is generated which represents companies as first class citizens. For each company a home page thumbnail, a characterization of its competencies and its relationship to the query, as well as up to three query related landing pages are presented. All these aspects will be considered in more detail in the remainder of this paper, but beforehand we want to shortly address related work.

Related Work To our best knowledge, there is no directly related work on company search. The two most related areas are expertise retrieval [3] and entity search [10]. Many lessons can be learned from these areas. Nevertheless, there are some peculiarities with our company search scenario. In expert finding scenarios the identification of the experts is often a hard problem (see the expert search in the TREC 2007 enterprise track as an example [1]). Another aspect is the ambiguity of names or the vague relationship between persons and documents. On the other hand, representing experts by pseudo documents is also an established approach in expert search [2] and an elaborate result presentation is important here as well.

2 Modeling and retrieval

When thinking about the retrieval model for the given scenario on a higher level, a model of the competencies of a company has to be matched with the query representing the user’s information need. From the requirements it was defined that a keyword query should be used. With respect to the representation of the company profiles interviews showed that an automatic extraction process is preferable to the manual definition of profiles because of the sheer creation effort and update problems. Due to the addressed domain of IT companies it can be assumed that all companies worth to be found maintain a website depicting their competencies. Of course other sources of evidence could also be addressed—such as newspaper reports, business reports, or mentions of the companies on other pages in the Internet. These approaches are surely worth consideration in the future. Nevertheless, the concentration and restriction to the company's own website also has the advantage of predictability and clear responsibilities. Put simply, if a company complains that it is not among the best matches for a particular query, we can pass the buck back and encourage them to improve their website—what they should do anyway because of SEO considerations.

To avoid our arguments eventually turning against us, we have to exploit the information on the websites as good as we can. Besides crawling and preprocessing aspects addressed in the following sections 3 and 4, in particular we have to use an appropriate retrieval model. As a first decision we have to choose between a *company-based approach* (each company is represented by a pseudo document used directly for ranking) and a *document-based approach* (the documents are ranked and from this ranking a ranking of the companies is derived). A comparison of these approaches can, for instance, be found in [3], however not showing a clear winner. We plan to test both approaches in the future but we started with the *company-based approach* where each company c is represented by a pseudo document d_c generated as the concatenation of the content blocks of the web pages crawled from the respective website. In the future, we plan to test weighting schemes based on the markup information, the depth of the single pages, and other parameters.

For a more formal look we use the following definitions:

- $q = \{w_1, w_2, \dots w_n\}$ is the query submitted by the user (set of words)
- $C = \{c_1, c_2, \dots c_m\}$ is the set of companies
- d_c is the concatenation of the documents representing company c
- f_{w,d_c} is the number of times w appears in d_c
- cf_w is the number of companies for which d_c contains w
- λ and μ are design parameters

Following [6] we use a candidate generation model and try to rank the companies by $P(c|q)$, the likelihood of company c to be competent for query q. As usual, by invoking Bayes' Theorem, this probability can be refactored as follows [3]:

$$P(c|q) = \frac{P(q|c)P(c)}{P(q)} \overset{\text{rank}}{=} P(q|c)P(c) \approx P(q|d_c)P(c)$$

Currently, we use a constant for the company prior $P(c)$. However, it turned out that this will be an interesting point for future optimizations highly correlated with aspects of document length normalization for the pseudo documents d_c. For test purposes we inserted big national and international IT companies in the list. In our current implementation these companies did not make it to the absolute top ranks even for queries simply consisting of a registered product name of the company. Instead, small service providers which have specialized in support for this product were ranked higher. Interestingly, this problem is already an issue with expert finding, but an even bigger challenge in company search because of the heterogeneous company sizes.

Another point which became obvious in first experiments was the well known vocabulary mismatch. For example with the query "Typo3" the ranking did not consider the broader competence of the companies in the topics of web applications or content management systems. As proposed in [4, 5] we decided to use Latent Dirichlet Allocation (LDA) to address this problem. An independence assumption to calculate the probabilities wordwise by $P(q|d_c) = \prod_{w \in q} P(w|d_c)$ and a combination of the word-based perspective with a topic-based one would then lead to:

$$P(w|d_c) = \lambda \left(\frac{f_{w,d_c} + \mu \frac{cf_w}{|C|}}{|d_c| + \mu} \right) + (1 - \lambda) P_{lda}(w|d_c)$$

$P_{lda}(w|d_c)$ stands for the probability that a word w is generated by a topic which is generated by d_c (see [5, 8] for more details). To simplify things further, we employed an idea presented in [9]. Here the Lucene Payloads are used to boost terms via LDA. The payload $lda(w, d_c)$ assigned to word w is determined as the weight of w according to the topic distribution of d_c. This means that $lda(w, d_c)$ is high when w fits well with the broader topics dealt with in d_c. Combining this boost factor with term frequency and inverse document frequency information we yield the following scoring function:

$$score(c, q) = \sum_{w \in q} tf\,(w, d_c) \cdot idf(w) \cdot lda(w, d_c)$$

Of course, this is only a first pragmatic starting point and the above considerations point out various interesting aspects for future comparisons and optimizations.

3 Data acquisition

As a prerequisite documents representing companies have to be obtained first. For crawling company websites we chose to employ crawler4j, a lightweight Java web crawler (`https://github.com/yasserg/crawler4j`).

The crawling of each company is an individual process which allows us to crawl multiple companies at once. We start with the company's home URL as a seed and use a truncated form of that URL as a pattern to discard all links to external domains found during the process. For our first investigation, we crawled a maximum amount of 2000 documents per company in a breadth first manner, where each document is a web page. We plan to leverage additional document types, such as PDF, in the future.

For each page the corresponding company, page title and full URL are stored in a database. This information is reused later when creating the web page indexes. To obtain d_c (the pseudo document describing company c) the contents of all crawled pages of the company are concatenated. To reduce noise, we apply Boilerpipe [7] (`https://code.google.com/p/boilerpipe`) to all documents in order to extract the main textual content from those pages first. This step aims to eliminate those page elements which do not contribute to the actual content of a page and are repeated very often: navigation menus, footer information, etc.

4 Preprocessing and indexing

Early experiments have shown that data quality plays a seminal role for the quality of a topic model learned. That is why we utilize a customized Lucene Analyzer before applying the LDA to d_c or indexing the company documents. The analyzer filters German stop words, applies a Porter Stemmer for the German language and uses a series of regular expressions to remove or modify tokens. As an example, digit-only tokens are removed, while tokens of the form *word1:word2* are split into two tokens, *word1* and *word2*. Consistently, incoming user queries are processed by the same analyzer.

After the analyzing step, an LDA topic model of all company representations d_c is created, utilizing the jgibbsLDA (`http://jgibblda.sourceforge.net/`) implementation. The resulting model is represented in a Java class hierarchy, which enables us to directly access the distribution of topics for each company, as well as the word probability distributions within topics. Therefore the payload function $lda(w|d_c)$ for each word w in d_c can be computed immediately. Another representation of d_c is created, where each term is enriched with its determined LDA payload. The generated LDA model is reused for result preparation.

The company index is created from all pseudo documents d_c enriched with payloads. When executing a query it is examined by the index searcher and consequently determines the ranks of the companies in the result set. To be able to show a query's top documents for a given company, we also create an index for the companies' web pages. All crawled web pages are considered and for each page we also preserve the information of the corresponding company. Both types of indexes are based on Lucene (`https://lucene.apache.org/core/`). Prior to indexing we apply the analyzing process described above.

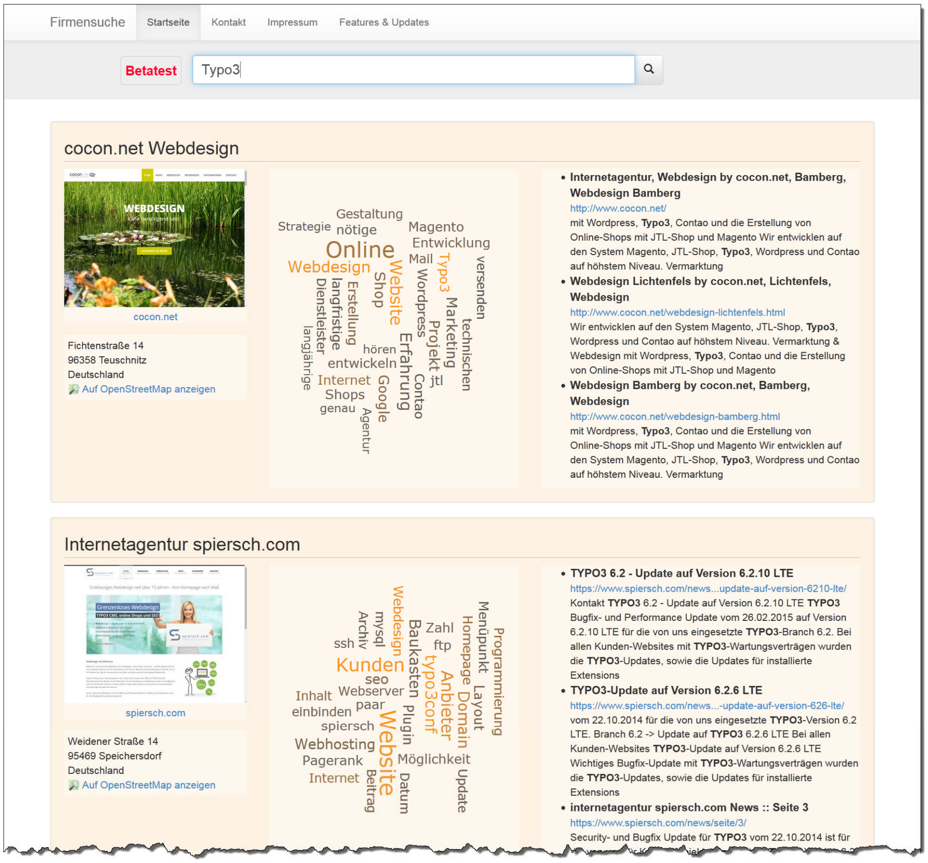
With the creation of a company index representing companies and their competencies, a web page index for the companies as well as the overall topic model, all steps necessary to enable searching are completed.

5 Interaction design

As usual the query is given as a simple keyword query. In the future more sophisticated variants are conceivable, for example allowing for geographic filter constraints. Nevertheless, the simple and familiar keyword solution has its advantages and the use of geographic filter constraints is debatable as long as only companies located in Upper Franconia are listed, anyway.

With respect to the result presentation the situation is more demanding. Discussions with potential users disclosed the following requirements: (1) Companies are the main objects of interest. (2) Address information, a first overview, and a visual clue would be nice. (3) The general company profile as well as the relationship to the query should become obvious. (4) Entry points (landing pages) for the query within the website of the company are desired.

The result page depicted in Fig. 2 directly implements these requirements. Companies are ranked with respect to the retrieval model described in section 2.



cocon.net Webdesign

cocon.net

Fichtenstraße 14
96358 Teuschnitz
Deutschland
Auf OpenStreetMap anzeigen

- **Internetagentur, Webdesign by cocon.net, Bamberg, Webdesign Bamberg**
  http://www.cocon.net/
  mit Wordpress, **Typo3**, Contao und die Erstellung von Online-Shops mit JTL-Shop und Magento Wir entwicklen auf den System Magento, JTL-Shop, **Typo3**, Wordpress und Contao auf höhstem Niveau. Vermarktung
- **Webdesign Lichtenfels by cocon.net, Lichtenfels, Webdesign**
  http://www.cocon.net/webdesign-lichtenfels.html
  Wir entwicklen auf den System Magento, JTL-Shop, **Typo3**, Wordpress und Contao auf höhstem Niveau. Vermarktung & Webdesign mit Wordpress, **Typo3**, Contao und die Erstellung von Online-Shops mit JTL-Shop und Magento
- **Webdesign Bamberg by cocon.net, Bamberg, Webdesign**
  http://www.cocon.net/webdesign-bamberg.html
  mit Wordpress, **Typo3**, Contao und die Erstellung von Online-Shops mit JTL-Shop und Magento Wir entwicklen auf den System Magento, JTL-Shop, **Typo3**, Wordpress und Contao auf höhstem Niveau. Vermarktung

Internetagentur spiersch.com

spiersch.com

Weidener Straße 14
95469 Speichersdorf
Deutschland
Auf OpenStreetMap anzeigen

- **TYPO3 6.2 - Update auf Version 6.2.10 LTE**
  https://www.spiersch.com/news...update-auf-version-6210-lte/
  Kontakt **TYPO3** 6.2 - Update auf Version 6.2.10 LTE **TYPO3** Bugfix- und Performance Update vom 26.02.2015 auf Version 6.2.10 LTE für die von uns eingesetzte **TYPO3**-Branch 6.2. Bei allen Kunden-Websites mit **TYPO3**-Wartungsverträgen wurden die **TYPO3**-Updates, sowie die Updates für installierte Extensions
- **TYPO3-Update auf Version 6.2.6 LTE**
  https://www.spiersch.com/news...-update-auf-version-626-lte/
  vom 22.10.2014 für die von uns eingesetzte **TYPO3**-Version 6.2 LTE. Branch 6.2 -> Update auf **TYPO3** 6.2.6 LTE Bei allen Kunden-Websites **TYPO3**-Update auf Version 6.2.6 LTE Wichtiges Bugfix-Update mit **TYPO3**-Wartungsverträgen wurden die **TYPO3**-Updates, sowie die Updates für installierte Extensions
- **internetagentur spiersch.com News :: Seite 3**
  https://www.spiersch.com/news/seite/3/
  Security- und Bugfix Update für **TYPO3** vom 22.10.2014 ist für

Fig. 2. Result page for the query “Typo3”

For each company in the result list a row with the name and three information blocks is shown. The company name is directly taken from the input data as well as the address information (Fig. 1 upper left corner). A screenshot of the homepage (captured with Selenium; `http://www.seleniumhq.org/`) and a prepared link to OpenStreetMap complete the left overview block for each company. In the middle block a word cloud is given. Here the size of the terms represents the importance of the terms for the company profile (based on $tf \cdot idf$ information). The color represents the relationship of the terms to the query. Orange represents a strong relationship. The relationship is calculated based on a company's prevalent LDA topics. Currently, we consider the five top terms of the five topics with the highest correlation to the query. At most thirty terms are shown in the word cloud taking terms important for the company profile and important for the relationship to the query in a round robin procedure. Finally, the right block consists of up to three most relevant landing pages within the company website represented by their title, the URL, and a query-dependent snippet.

6 Conclusion

In this paper we have described the company search problem and presented a solution based on pseudo document ranking, the use of LDA to incorporate topical relevance, and a suitable result presentation. Currently the prototype implementation is tested. It turned out that the effectiveness and the efficiency are promising in preliminary interviews with representatives of local IT companies. Current response times of the system are below two seconds. The most obvious challenges are the appropriate ranking of companies with different sizes, the visualization of the company profiles in the result page, and a reasonable modeling and presentation of topics (number of topics in LDA and also alternative approaches). The current prototype is available on the project web page[1].

References

1. Bailey, P., De Vries, A.P., Craswell, N., Soboroff, I.: Overview of the TREC-2007 enterprise track. In: The Sixteenth Text REtrieval Conference (TREC 2007) Proceedings. NIST Special Publication: SP 500-274 (2007)
2. Balog, K., Azzopardi, L., de Rijke, M.: Formal models for expert finding in enterprise corpora. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '06, pp. 43–50. ACM, New York, NY, USA (2006). DOI 10.1145/1148170.1148181
3. Balog, K., Fang, Y., de Rijke, M., Serdyukov, P., Si, L.: Expertise retrieval. Found. Trends Inf. Retr. **6**(2–3), 127–256 (2012). DOI 10.1561/1500000024
4. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. The Journal of Machine Learning Research **3**, 993–1022 (2003)
5. Croft, W.B., Metzler, D., Strohman, T.: Search Engines: Information Retrieval in Practice. Pearson Education (2009)
6. Fang, H., Zhai, C.: Probabilistic models for expert finding. In: Proceedings of the 29th European Conference on IR Research, ECIR'07, pp. 418–430. Springer-Verlag, Berlin, Heidelberg (2007). URL http://dl.acm.org/citation.cfm?id=1763653.1763703
7. Kohlschütter, C., Fankhauser, P., Nejdl, W.: Boilerplate detection using shallow text features. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining, WSDM '10, pp. 441–450. ACM, New York, NY, USA (2010). DOI 10.1145/1718487.1718542
8. Wei, X., Croft, W.B.: LDA-based document models for ad-hoc retrieval. In: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 178–185. ACM (2006)
9. Zhang, M., Luo, C.: A new ranking method based on latent dirichlet allocation. Journal of Computational Information Systems **8**(24), 10,141–10,148 (2012)
10. Zhou, M.: Entity-centric search: querying by entities and for entities. Dissertation, University of Illinois at Urbana-Champaign (2014). URL http://hdl.handle.net/2142/72748

[1] `http://www.uni-bamberg.de/minf/forschung/firmensuche`

Similarity-Based Cross-Media Retrieval for Events

Piroska Lendvai and Thierry Declerck

Dept. of Computational Linguistics, Saarland University
Saarbrücken, Germany
piroska.r@gmail.com,declerck@dfki.de

Abstract. Our goal is to link social media content to contextually relevant information in complementary media in the domain of daily news. Web links from tweets with user-included URLs are transferred to URL-less tweets, using manually annotated events. The new cross-media ties establish authoritative feedback documents for unsupported social media content, and enable extracting an improved set of event-denoting terms based on longest common subsequences between tweets and documents.

Keywords: social media, information contextualization, similarity-based retrieval, cross-media feedback documents, term extraction

1 Introduction

We aim to create a cross-media (CM) linking algorithm in the *PHEME* project[1] to connect User-Generated Content (UGC) to topically relevant information in complementary media. Media that is complementary to UGC (in our pilot study, a tweet) is defined to be authoritative news releases on the web.

Recent natural language processing studies present some CM approaches with the purpose of aligning UGC and authoritative content. The goal of [5] is to collect information about emergency situations from tweets that are complementary to mainstream media reports. First, relevant keywords are determined from a centroid news article in a topically connected article cluster, and used in various query constructions to retrieve event-related tweets. The direction of linking is motivated by the need to boost retrieval precision on established events, which is orthogonal to the mission of the PHEME project – our targeted starting point is events that first emerge in social media and only later or not at all are covered in mainstream news releases. The algorithm of [5] is reused and extended in [2]: based on a centroid article in an event cluster, related tweets that contain URLs are mined, using custom-threshold-based term vector similarity. Then, relevance ranking takes place on these tweets, using platform-specific

[1] www.pheme.eu

indicators (number of mentions, retweets, etc). New, related articles on the web are retrieved based on the URLs of top-ranked tweets. [2] do not report on the proportion of web articles found that were already seen in the query-originating news cluster. Such information would evaluate the retrieval of complementary sources more transparently, and it forms an important part of our CM algorithm.

To implement CM linking for PHEME, our core assumption was that URL presence in tweets is a relevance feedback analogous to landing page information in click data, utilizable to develop retrieval functions from observed user behavior (see e.g. [3]). Referring to external sources is a multi-purpose activity in social media practices that may amalgamate among others intents of content framing (i.e., quoting authoritative sources) and content enrichment (i.e., guiding to extended information). Based on URLs that are present in tweets and point to web documents, we devised a method that transfers this explicit, user-included relevance signal to a collection of tweets that do not include explicit web links. The transfer is based on Events that have been manually annotated; each tweet is annotated with exactly one Event. Events are manually annotated situations or stories that describe smaller scale episodes than hashtag-denoted topics.

Our goal is to link URL-less tweets to a ranked list of web documents, where topic relevance is bootstrapped from event-based similarity between URL-including tweets and URL-less tweets, and ranking is based on aggregated n-gram similarity between tweet text and web document text. To this end, we extract and rank key phrases based on document–tweet similarity, and associate them with the Event the referring tweet is annotated with. As we focus on related content discovery and its use for rumour[2] verification purposes, our setup and results are more specific than the INEX tweet contextualization tasks (see e.g. [1]) to support a human reader.

2 Data and Algorithm

We worked with a dataset that consists of tweets relating to two broad events: (G) the Gurlitt art collection[3] and (O) the Ottawa shooting[4]. Tweets were precollected by filtering on event-related keywords (e.g. '*gurlitt*'), selecting events that meet the characteristics of a rumour. Each tweet was manually annotated for situations/stories (henceforth: Events[5]) that correspond to specific rumours, as described in [6]; for characteristics of the data see the top section of Table 1.

2.1 String similarity-based term extraction

For each URL-containing tweet within each Event, a tweet – document similarity calculation cycle is run. Similarity in the current implementation is based on

[2] defined in PHEME as *a circulating story of questionable veracity*

[3] https://de.wikipedia.org/wiki/Schwabinger_Kunstfund

[4] https://en.wikipedia.org/wiki/2014_shootings_at_Parliament_Hill,_Ottawa

[5] e.g. (G): `'The Bern Museum will accept the Gurlitt collection'`, `'Gurlitt was mentally unfit when he wrote his will'`; (O): `'There are snipers on the roof of the National Art Gallery'`, `'Shooter is still on the loose'`.

	Gurlitt	**Ottawa**
languages	DE, FR, EN	EN
events	3	51
tweets without URL	43	182
tweets with URL	147	341
unique URLs	143	187
fetchable web documents [by authoritative sources]	61 [61]	107 [107]
terms extracted from URLed tweets	110	169
terms extracted from URLless tweets	96	190
terms unseen in URLed tweets	83	143

Table 1. Characteristics of tweet data and of terms extracted from fetched web documents.

the Longest Common Subsequence (LCS) metric (cf. [4]). LCS is a language-independent, flexible-length skip-gram matching method that we apply on the token level for each tweet – document sentence pair[6]. No linguistic information is used, except for stopword filtering by the NLTK toolkit[7]. The process produces a ranked list of tweets based on LCS similarity with their linked document (which is in effect a user-coded feedback document) for all URL-providing tweets for a given Event, and outputs the longest common subsequence tokens between tweet and document body.

In the second pass, the cycle is applied to the same feedback web document set, now paired with tweets that did *not* link external documents but are hand-labeled with the same Events as the tweets from which web documents are referred from. This boosts the pool of linked authoritative[8] documents and tweets by 105% for G and 294% for O; extracted top-5 LCS phrases[9] grow qualitatively[10] by 75% for G and by 85% for O; cf. the bottom section of Table 1. An example output is provided below for the focus Event '`The Bern Museum will accept the Gurlitt collection`'.

Focus document's **headlines:** "Bestätigt: Kunstmuseum Bern nimmt das Erbe des Kunstsammlers Cornelius Gurlitt an - KURIER.at"

Top tweet with URL to focus document: *Bestätigt: Sammlung Gurlitt geht nach Bern http://t.co/FRCSHTU5hL*

LCS term of top URL-ed tweet and focus document: `'bestätigt sammlung gurlitt geht bern'`; Similarity **score:** 1.00

Top URL-less tweet labeled with focus Event: *RT @SWRinfo: Das Kunstmuseum Bern nimmt das Erbe des Kunstsammlers Cornelius #gurlitt an.*

[6] Casing is normalized, the retweet token, screen names and punctuation are removed

[7] nltk.org

[8] Based on a list of 25k authoritative news sources collected by PHEME.

[9] We keep the 5 most similar LCS phrases for each tweet–web document pair.

[10] I.e., in terms of obtaining new phrases that were unseen in the pool of URL-ed tweets–linked web documents.

LCS term for top URL-less tweet and focus document: `'kunstmuseum bern nimmt erbe kunstsammlers cornelius gurlitt'`; Similarity **score:** 0.79

3 Evaluation and Outlook

We presented a pilot study on transferring feedback document relevance for social media posts, based on manually annotated, fine-grained events. We used the LCS similarity metric to extract descriptive phrases for each Event; the obtained multi-word terms implicitly encode token proximity and word order, valuable for query- and document language modeling and indexing. LCS was also used to assign term-, respectively document weights to each Event, independent of a fixed document collection. Tweets with unsupported claims could be linked to authoritative web documents by utilizing hand-coded tweet–tweet similarity information; automatically obtaining this information is currently ongoing.

The findings suggest that LCS is advantageous when working with big data across languages and domains, as foreseen in the PHEME project. In future work we plan to compare LCS with other similarity metrics, as well as evaluate the obtained term, respectively document rankings in a retrieval scenario for information verification purposes. The major impact of Event-based bootstrapping of cross-media links is that we obtain a much larger set of cross-media context pairs, enabling the extraction of an improved list of event descriptors that can be put to use in fact checking and contextual document ranking, on which we plan to report in follow-up studies.

Acknowledgments We are grateful to two anonymous reviewers for their insightful comments. Work presented in this paper has been supported by the PHEME FP7 project (grant No. 611233).

References

1. Bellot, P., Moriceau, V., Mothe, J., Sanjuan, E., Tannier, X.: Overview of INEX tweet contextualization 2013 track. CLEF (2013)
2. Balahur, A., Tanev, C.: Detecting Event-Related Links and Sentiments from Social Media Texts. ACL Conference System Demonstrations (2013)
3. Joachims, T.: Optimizing search engines using clickthrough data. Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (2002)
4. Lin, Ch. Y.: Rouge: A package for automatic evaluation of summaries. In: Text summarization branches out: Proceedings of the ACL-04 workshop. Vol. 8 (2004)
5. Tanev, H., Ehrmann, M., Piskorski, J., Zavarella V.: Enhancing Event Descriptions through Twitter Mining. In: Proceedings of ICWSM (2012)
6. Zubiaga, A., Liakata, M., Procter, R. N., Bontcheva, K., Tolmie, P.: Towards detecting rumours in social media. In: AAAI Workshop on AI for Cities (2015)

How to Stay Up-to-date on Twitter with General Keywords

Mandy Roick, Maximilian Jenders, and Ralf Krestel

Hasso Plattner Institute
Prof.-Dr.-Helmert-Str. 2-3, 14482 Potsdam, Germany

Abstract. Microblogging platforms make it easy for users to share information through the publication of short personal messages. However, users are not only interested in sharing, but even more so in consuming information. As a result, they are confronted with new challenges when it comes to retrieving information on microblogging platforms. In this paper we present a query expansion method based on latent topics to support users interested in topical information. Similar to news aggregator sites, our approach identifies subtopics to a given query and provides the user with a quick overview of discussed topics within the microblogging platform. Using a document collection of microblog posts from Twitter, we compare the quality of search results returned by our algorithm with a baseline approach and a state-of-the-art microblog-specific query expansion method. We introduce a novel, innovative semi-supervised evaluation strategy based on expert Twitter users. In contrast to existing query expansion methods, our approach can be used to aggregate and visualize topical query results based on the calculated topic models, while achieving competitive results for traditional keyword-based search.

1 Searching Microblog Posts

Along with the development of Web 2.0, users have increasingly become content providers. A good example of this trend are microblogging platforms. These platforms allow users to share short text messages, images, or links with interested observers (followers) [5]. Microblogging platforms, such as Facebook, Tumblr, or Twitter, report constantly increasing numbers of users. According to Twitter's website, e.g., the platform has 284 million active users monthly and 500 million shared microblog posts daily, averaging 6,000 tweets per second. However, not all of Twitter's users share content. 44% of the users have never posted anything[1]. These users are only interested in consuming content, thus filtering and searching microblog posts becomes an increasingly important task.

[1] Digital Insights
http://blog.digitalinsights.in/social-media-users-2014-stats-numbers/05205287.html

In 2011, Twitter's search engine processed about 1.6 billion search queries daily. An analysis of the search behavior [10] shows that 49% of Twitter users search for timely information, such as trending topics or information related to news, 26% describe an interest in social information about other users, and 36% report a search for specific topics, such as "astronomy". Since then, searching microblog posts has become part of the research agenda. The Text REtrieval Conference[2] (TREC) opened a Microblog track in 2011 addressing a real-time search task on microblogging platforms. In 2014, Twitter expanded its search service to allow users to search for all tweets ever posted[3].

In contrast to Web search, searching microblogs displays some characteristic challenges [10]. To cope with the restricted length of tweets, Twitter users not only use abbreviations and emoticons, but also employ hashtags, which are explicit, user-specified topic markers. Another means to artificially condense information to fit in tweets is using a link to another web page with more information on the topic. Hence, many tweets contain URLs. However, these instruments are user-specified and their quality and usability for search depends on how users adopt them. URLs for instance often link to images or videos, which are difficult to interpret for a machine. The given hashtags are very inconsistent through different spellings and different interpretations of users; "#4YearsAgo5StrangersBecame5Brothers", "#ThankYou1DYouChangedOurLives", and "#4YearsOf1D" all refer to the four year anniversary of the band One Direction. For a user who does not follow this content on Twitter every day, it is difficult to pose queries that match the language used in tweets. The massive number of tweets every day constitutes an additional challenge to new users who are interested in an overview of the content on Twitter. To overcome the differences in the language used by users who post tweets and users who pose queries, we introduce a new query expansion approach to allow topic-based searching. This improves the search experience for people searching topical and news-like information on Twitter using rather general keywords such as "politics" or "basketball".

While many researchers propose query expansion algorithms for microblogging platforms [9], [11], [1], [4], none of them deal with the search for specific topics. Currently, Twitter presents search results in a list view showing the content of tweets and their authors, the time that has passed since the tweets were posted, and, if the tweets link to a news page, a short summary of the news page. The ranking is mainly based on exact query term matching, on recency, and on popularity. While query expansion can help to overcome the problems of exact query term matching, topical queries usually include many subtopics that a user might be interested in. Gaining an overview of these results is difficult using ranked lists. Given the fact that Twitter behaves similar to news media [6], we propose to use our results for query expansion to cluster tweets about similar topics. An application could display a user interface similar to platforms such as Google News[4], where individual news articles are aggregated and categorized.

[2] TREC http://trec.nist.gov

[3] Twitter https://blog.twitter.com/2014/building-a-complete-tweet-index

[4] Google News http://news.google.com

2 Related Work

There are many approaches that use topic models for query expansion in classic information retrieval [13], not so many for microblog posts. Yan et al. [12] present an alternative to LDA specially for short texts: the biterm topic model (BTM). Instead of generating documents, BTM models the generation of biterms (unordered word-pairs that co-occur in short texts) and assumes that each biterm is drawn from one topic. One work similar to ours describes the automatic topic-focused monitor (ATM) [7], which is able to monitor tweets relevant to a given topic. While the strength of ATM lies in the monitoring of tweets over time, our search approach selects keywords firsthand and does not need to know the search query in advance for correct sampling.

Several approaches for query expansion and document expansion have been proposed in the context of the Microblog Track at TREC. For example, Wang et al. [11] use a query expansion by accessing pseudo-relevance feedback and a document expansion through given URLs that some tweets contain. They use this expansion to break ties between tweets that display the same retrieval score, meaning that only tweets with the same retrieval score are considered. In that context, Wang et al. showed that the expansions did not support the ranking but lead to worse results. Bandyopadhyay et al. [1] aim to improve weak queries (e.g., short tweets with different spelling and grammar than a regular search query would exhibit) and present a query expansion algorithm which is based on pseudo-relevant web documents. The algorithm transfers the original queries to the Google search API and expands the query with the most frequent terms in the resulting titles and snippets which are returned by the search API. Irrespective of the TREC Conference, Massoudi et al. [9] developed a retrieval model for queries that contain trending topics. They extend the model by taking quality indicators, like recency and followers, into account as well as a query expansion through co-occurrence of terms. An approach for document expansion has been described by Efron et al. [4] using a language model which includes a weighted probability for a word given the expanded document. An expansion is achieved by using the document as pseudo-query on the corpus of documents. Liang et al. [8] use pseudo-relevance feedback query expansion based on language models and employ temporal re-ranking to discover recent but relevant information for a query in microblogs. Topic models have been used by Chua et al. [3] to extract representative tweets from a stream for event summarization.

The presented approaches mostly aim to expand the given query to match the language which is used in the short microblog posts [1], [11] or to expand the microblog posts to match the language which is used in a query [11], [4]. In this paper, we concentrate on queries which are intentionally very general and we aim to expand those queries to provide a good overview of the trending subtopics at different levels of granularity.

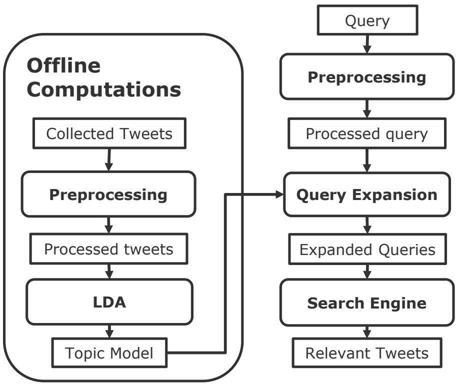


Fig. 1: System architecture

3 Topic-Based Query Expansion

We want to support users searching for general topics, such as "politics" or "Ukraine". To this end, we propose a query expansion approach based on topic modelling. These models are learned on a daily basis from a data set of crawled and preprocessed tweets and are later used to expand user-specified queries. Figure 1 displays our system's architecture. The crawling of tweets and topic model construction is handled offline, while the topic model is being used to expand queries in an online fashion at query time. If a new, unkown query term is used which is not present in our offline-computed topic model, we fall back to standard keyword search. However, this essentially does not happen for our targeted general queries. Furthermore, we address recency and popularity in Twitter indirectly via computing new topic models daily so our model reflects trends accordingly.

Topic Model Construction We used latent Dirichlet allocation (LDA) [2] to compute a topic model[5] prior to search, e.g. once a day. The resulting topic model can then be used to infer a topic distribution for a new tweet d, Θ_d. Given a query, the most probable topics can be determined using Φ, the topic-word-distributions. Table 1 shows the 10 most probable topics for a one-day topic model together with the probability of the topic given the query word "politics".

Using LDA, the number of topics K has to be specified in advance. A larger K leads to splitting of topics, allowing for the separation of ambiguous topics. However, if no ambiguous topics are left, homogeneous ones are split up. For the purpose of query expansion, it is important that different topics can be found for a term, and that the topics found are not ambiguous, as this could lead to topic shifts. We evaluated different values for K on a validation set, which is described in Section 4.

[5] We use Mallet http://mallet.cs.umass.edu

Table 1: Top 10 topics from October 20, 2014 for the query *"politics"*

$\hat{p}(i\|\boldsymbol{q})$	**8 most probable stemmed words**
0.167	obama ebola tcot speech presid reason net ban
0.155	ukip vote ward tori parti peopl nh elect
0.115	bjp part scienc india modi biblic read congress
0.089	vote elect voter earli blue texa gop todai
0.072	gate gamer gamerg peopl women stop game bulli
0.069	presid indonesia jokowi minist presiden japan russia
0.052	isi turkei kurd koban fight kill iran syria
0.044	ari support pakistan ban stand pti khan wesupportari
0.043	energi price compani tax loan pai servic power
0.036	class question teacher answer write english word learn

Query Expansion We are interested in the most probable topics for all words of a query $\boldsymbol{q}$, i.e., we search for topics i where $p(z = i|\boldsymbol{q})$ (in the following $p(i|\boldsymbol{q})$) is maximal. During Gibbs sampling, we sample values for $\boldsymbol{z}$ for each word w in the vocabulary $\boldsymbol{w}$. We use these samples of $\boldsymbol{z}$ to estimate $\hat{p}(i|\boldsymbol{q})$ with $\frac{n(i,w)}{n(w)}$. In other words, $\hat{p}(i|\boldsymbol{q})$ is estimated by the number of times the query words $\boldsymbol{q}$ were assigned to topic i divided by the total number of occurrences of words $\boldsymbol{q}$ in the corpus. Note that, although our test queries only contain a single term, this formulation also holds for queries with multiple words. For the query expansion, we then use the topics' best representatives, i.e., for a topic i the most probable words based on $p(w|i) = \phi_i^w$. The quality of the query expansion is heavily influenced by the number of topics the query is expanded with, as well as the number of words chosen from each topic for expansion. We optimized these model parameters on a validation set (see Section 4). Best results were achieved setting K, the number of topics, to 200; the number of terms to use for query expansion to 10, and the threshold to include a topic for an expansion to $\hat{p}(i|\boldsymbol{q}) > 0.05$. For our example in Table 1 the top 7 topics would be used for query expansion, while the rest are disregarded.

4 Experiments

To assess the ability of our algorithm to retrieve topically relevant tweets, we propose a novel, semi-automatic evaluation strategy that produces high-quality labeled data by utilizing expert Twitter users. In addition, we present some example queries together with the expanded queries based on our topic model as anecdotal evidence demonstrating how our algorithm can help users to get a topical overview of subtopics for a given general query.

Data Set Most existing annotated data sets are focused on detailed information needs, such as the Tweets2011 corpus used for the TREC Microblog Track[6]. General topical queries are not included. Therefore, we created our own data set with

[6] TREC microblog data http://trec.nist.gov/data/tweets

semi-automatic annotations. We chose 2 general topical queries:“sports”, “politics” and for each general query 2 more specific ones: “baseball”, “basketball”, “Ebola”, “Ukraine”. To find relevant tweets for each of the queries, we handpicked 10 expert twitter users who primarily tweet on the topic corresponding to the query. Together with the relevance of tweets we used popularity and the number of tweets to select these users. For “politics”, e.g., these users were: @BBCPolitics, @CNNPolitics, @NicRobertsonCNN, @KevinBohnCNN, @TheWhiteHouse, @politico, @thehill, @HuffPostPol, @CBSPolitics, @BarackObama. We then crawled these users’ tweets together with the 1% of general tweets available through the Twitter API. We annotated only our expert users’ tweets as relevant for the respective queries, leading to small values in precision, because some tweets marked as non-relevant are actually relevant. Yet, tweets marked as relevant are in large part actually relevant. Thus, we estimate a method’s tendency for the actual precisions. We constructed two data sets, one for validation and one for testing. Each set includes a training set of one day of twitter data to learn the topic model and the subsequent day to validate or test (Oct. 21st and Dec. 4th 2014, each 1.4m tweets (1% of all tweets)). On average, our expert users published 196 tweets per query per day.

Baseline Approach As baseline approach BL, we search for the given queries without query expansion. Similar to Twitter’s search engine, we search for the query terms in tweets as well as in linked content using BM25. In contrast to Twitter’s search, our ranking is not incorporating recency or popularity.

Next to the baseline approach, we compare our search results with a competing query expansion algorithm that is designed for microblogging platforms and based on word co-occurance [9]. It shows improved search results against a standard query expansion with pseudo relevance feedback.

Topic-Based Approach Our topic-based approach results in a set of expanded queries for each initial query according to our topic model. We set α asymmetric and choose the initial value $\alpha_i = K \cdot 0.01$ for all $i \in \{1, 2, \ldots, K\}$. In contrast to α, we set β symmetric with initial value $\beta_i = 0.05$. We run Gibbs sampling for 500 iterations. Each topic i in our model that contains the query term q (i.e. $\hat{p}(i|q) > 0.05$) forms the basis for one query. To compare our search results with other search algorithms and the baseline, we merge the tweets resulting from each expanded query q into one ranking. We calculate a ranking score $sc_q(i, d)$ for each tweet d that was found for a query q. The score depends on the topic (=expanded query) i for which the tweet was found and the tweet d itself. The score combines the probability $\hat{p}(i|q)$ of the query term q belonging to the topic i, the topic’s proportion θ_d^i for tweet d, and the BM25 score for the tweet $BM25(d)$:$sc_q(i, d) = \hat{p}(i|q) \cdot \theta_d^i + BM25(d)$ This score allows to combine the results of all expanded queries for a query term into one ranking, which is needed to compare the precision with other approaches.

Results The results differ from query to query. Mean average precision (MAP) is 0.101 for the baseline approach (BL), 0.152 for the co-occurance-based approach

Table 2: Average precision for various algorithms for particular queries

	sports	**baseball**	**basketball**	**politics**	**Ukraine**	**Ebola**
BL	0.0035	0.0033	0.0038	0.0000	0.2730	**0.3232**
CB	0.0057	0.2578	0.0106	0.0158	**0.4595**	0.1617
TB	**0.0150**	**0.3175**	**0.0158**	**0.0166**	0.3068	0.2403

Table 3: Example expanded queries for topic-based approach (TB) and co-occurance-based approach (CB) [9] for queries "sports" and "Ebola"

sports				**Ebola**		
CB	**TB**			**CB**	**TB**	
girls	sports	sports	sports	outbreak	ebola	ebola
cespedes	united	hurt	game	americans	dallas	nigeria
boston	goals	head	football	free	health	free
football	game	butt	week	officially	save	big
betting	score	error	win	declared	hospital	plan
sports	mufc	vixx	state	virus	nurse	reason
pretend	west	body	nba	nigeria	patient	declared
smh	liverpool	button	team	health	care	immediate
females	man	touch	play	ebola	disease	someone's
yahoo	manchester	work	season	obama	africa	capricorn

(CB), and 0.152 for the topic-based approach[7]. The co-occurrence-based query expansion and our topic-based approach improve the results decidedly over the baseline. CB outperforms the topic-based approach only for the query "Ukraine", which results in similar MAP scores, see Table 2. Less general queries, such as Ebola, are less likely to benefit from query expansion since most tweets contain the keyword itself, whereas tweets about baseball are much more likely to contain words such as "MLB" instead of the word "baseball".

The expanded queries give an overview of the topic. The co-occurance-based approach only produces one expanded query, whereas our topic-based approach finds multiple topics for a given keyword and thus can create multiple expanded queries representing subtopics. Table 3 shows how our approach identifies different subtopics related to sports: English soccer, injuries, and American sports, while the co-occurance based approach fails to give a good overview and mixes various sports-related terms. The results are similar for the query Ebola. Here our approach identifies a topic related to Ebola in the U.S. vs. Africa.

Discussion The co-occurance-based expansion is calculated specifically for each query, therefore it benefits from the expansion terms being well suited. Yet, especially for the more general queries, the expanded queries can become ambiguous, i.e., contain more than one specific topic with considerable topic shifts. In contrast to the co-occurance approach, our topic-based approach discovers more

[7] To create comparable MAP scores, each ranking is restricted to 500 tweets

relevant terms for a given query. Thus, the focus of the search can transform to a broader topic than the original one. A strength of our topic-based approach is also the flexibility allowing to expand the query with a variable number of topics and visualize the inherent subtopics.

5 Conclusion

We have analyzed the usage of topic models to support general keyword queries in microblog search. We proposed a query expansion method using latent Dirichlet allocation to find relevant tweets and to group them based on latent topic information. Our experiments have shown that our approach outperforms standard keyword-based search and further demonstrated competitive results compared to a state-of-the-art microblog-specific query expansion algorithm. While standard search algorithms do not by default cluster search results, our approach returns tweets from various subtopics and the topics itself can be inspected to get a quick overview of what is currently discussed in Twitter related to general keywords. Besides a further, large-scale evaluation, for future work we are interested in the development of topics over time. Since Twitter is a highly dynamic platform, we hope to capture trending subtopics for general keywords by substituting LDA with a dynamic topic model.

References

1. Bandyopadhyay, A., Ghosh, K., Majumder, P., Mitra, M.: Query expansion for microblog retrieval. In: TREC. vol. 1, pp. 368–380. NIST (2012)
2. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of Machine Learning Research 3, 993–1022 (2003)
3. Chua, F.C.T., Asur, S.: Automatic summarization of events from social media. In: ICWSM. pp. 81–90. AAAI (2013)
4. Efron, M., Organisciak, P., Fenlon, K.: Improving retrieval of short texts through document expansion. In: SIGIR. pp. 911–920. ACM (2012)
5. Kaplan, A.M., Haenlein, M.: The early bird catches the news: Nine things you should know about micro-blogging. Business Horizons 54(2), 105–113 (2011)
6. Kwak, H., Lee, C., Park, H., Moon, S.: What is twitter, a social network or a news media? In: WWW. pp. 591–600. ACM (2010)
7. Li, R., Wang, S., Chang, K.C.C.: Towards social data platform: Automatic topic-focused monitor for twitter stream. In: VLDB. pp. 1966–1977. VLDB End. (2013)
8. Liang, F., Qiang, R., Yang, J.: Exploiting real-time information retrieval in the microblogosphere. pp. 267–276. JCDL, ACM (2012)
9. Massoudi, K., Tsagkias, M., de Rijke, M., Weerkamp, W.: Incorporating query expansion and quality indicators in searching microblog posts. In: ECIR, pp. 362–367. Springer (2011)
10. Teevan, J., Ramage, D., Morris, M.R.: #twittersearch: a comparison of microblog search and web search. In: WSDM. pp. 35–44. ACM (2011)
11. Wang, Y., Darko, J., Fang, H.: Tie-breaker: A new perspective of ranking and evaluation for microblog retrieval. In: TREC. NIST (2013)

12. Yan, X., Guo, J., Lan, Y., Cheng, X.: A biterm topic model for short texts. In: WWW. pp. 1445–1456. ACM (2013)
13. Yi, X., Allan, J.: A comparative study of utilizing topic models for information retrieval. In: ECIR. pp. 29–41. Springer (2009)

A New Approach For Selecting Informative Features For Text Classification

Zinnar Ghasem[1], Ingo Frommholz[1], and Carsten Maple[2]

[1] University of Bedfordshire, UK
[2] University of Warwick, UK
{zinnar.ghasem,ingo.frommholz}@beds.ac.uk
carsten.maple@warwick.ac.uk

Abstract. Selecting useful and informative features to classify text is not only important to decrease the size of the feature space, but as well for the overall performance and precision of machine learning. In this study we propose a new feature selection method called Informative Feature Selector (IFS). Different machine learning algorithms and datasets have been utilised to examine the effectiveness of IFS, and it is compared to well-established methods, namely Information Gain, Odd Ratio, Chi Square, Mutual Information and Class Discriminative Measure. Our experiments show that IFS is able to outperform aforementioned methods and to produce effective and efficient results.

Keywords: Feature selection, text classification, text preprocessing

1 Introduction

Automatic text classification is increasingly imperative for managing a substantial amount of data and information on the Web, for instance for search, spam filtering, malware classification, etc. It has been well studied over the past half century [1, 2], and a number of machine learning and statistical algorithms have extensively been used in various types of classification. In text classification, each document needs to be represented as a vector of features, for which various methods have been employed – Bag-of-Words is one of the major methods, where all unique features (in this case terms) of documents are utilised. This approach results in a high dimensional vector space of features, particularly in the case of a big dataset or where we find a large volume of document content. This large set of features is one of main issues of text classification, therefore it is highly desirable to reduce the size of the vector space by selecting useful features.

For this purpose, we propose a probabilistic *Informative Feature Selector (IFS).* The IFS is compared using ten folds cross validation to some of the

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

well-known methods, namely Chi Square *(chi)*, Odd Ratio*(OR)*, Information Gain *(IG)*, Class Discriminative Measures *(CDM)* and Mutual Information *(MI)*. The results of our experiment show that IFS is comparable to some and superior to others in term classification accuracy. The rest of this paper is organised as follows: Section 2 lists all feature selection methods which have been compared with IFS. Section 3 introduces IFS method. Section 4 outlines the experiments; datasets and performance measures, while results are analysed in section 5 and conclusion is the last section.

2 Related Work

As aforementioned, there exist a number of feature selection methods for text classification; among them IG, OR, CDM, and Chi have been claimed to perform well in experimental studies [3–6, 11, 8]. Thus, specifically those methods have been selected to be evaluated against IFS. In the following subsections, a brief description of each method is provided.

2.1 Information Gain (IG)

Information gain evaluates the usefulness of a feature for classification based on absence and presence of that feature in documents [9]. Information Gain defines the relationship between class c_j and feature t_i in the following formula.

$$\begin{aligned} IG(t_i, c_j) = -\sum_{j=1}^{m} P(c_j)\log(P(c_j)) + P(t_i)\sum_{j=1}^{m} P(c_j|t_i)\log(P(c_j|t_i)) \\ +P(\bar{t_i})\sum_{j=1}^{m} P(c_j|\bar{t_i})\log(P(c_j|\bar{t_i})), \qquad (1) \end{aligned}$$

where $P(c_j)$ is the probability of c_j, $P(t_i)$, and $P(\bar{t_i})$ is the probability of t_i occurring and not occurring in c_j correspondingly , while $P(c_j|t_i)$ is the probability of c_j given the probability of t_i , and $P(c_j|\bar{t_i})$ is the probability of c_j given the probability of $\bar{t_i}$ and m is number of classes in the training set.

2.2 Mutual Information (MI)

Mutual information evaluates the association between a feature and a class as

$$MI(t_i, c_j) = \log\left(\frac{P(t_i \wedge c_j)}{P(t_i)P(c_j)}\right) = \log\left(\frac{P(t_i|c_j)}{P(t_i)}\right). \qquad (2)$$

However, using the the two ways contingency table shown in Table 1, Eq. 2 can be represented in terms of the number of documents in a class as shown in Eq. 3 where α and λ represent the number of documents containing and not containing term t_i in class c_j respectively, similarly β and δ are the number of documents

Table 1. Two ways contingency table

	C_j	$\neg C_j$
term t_i	α	β
term $\bar{t_i}$	λ	δ

containing and not containing t_i in $\neg C_j$. If $\eta = \alpha + \beta + \lambda + \delta$ is total number of documents in all classes, MI can be estimated as

$$MI(t_i, c_j) = \log \left(\frac{\alpha \times \eta}{(\alpha + \lambda)(\alpha + \beta)} \right) \tag{3}$$

2.3 Chi Square (χ^2)

χ^2 or *chi* has been reported as one of top feature selection methods and it measures the correlation between feature t_i and class c_j [3]. Using Table 1, χ^2 is defined as

$$\chi^2(t_i, c_j) = \frac{\eta \times (\alpha\delta - \beta\lambda)^2}{(\alpha + \lambda)(\beta + \delta)(\alpha + \beta)(\lambda + \delta)} \tag{4}$$

The goodness of t_i decreases as the independence between t_i and c_j increases to the point where the value is zero, that is, when the number of documents containing term t_i in classes is equal.

2.4 Odd Ratio (OR)

Odd Ratio was initially developed for a binary classification; it has been reported by [6] that this value has additional advantages to other classification methods. Odd Ratio can be computed as follow:

$$OR(t_i) = \log \left(\frac{odd(t_i|pos)}{odd(t_i|neg)} \right) = \log \left(\frac{P(t_i|pos)(P(1 - P(t_i|neg)))}{P(t_i|neg)(P(1 - P(t_i|Pos)))} \right) \tag{5}$$

where both “pos” and “neg” represent a positive and negative class, respectively.

2.5 Class Discrimination Measure (CDM)

The CDM is an improved version of Multi-class Odd Ratio (MOR), where MOR is originally based on Odd Ratio. CDM is introduced in [10] and has reportedly outperformed IG.

$$CMD(t_i, c_j) = \sum_{j=1}^{m} \left| log \frac{P(t_i|c_j)}{P(t_i|\bar{c_j})} \right| \tag{6}$$

where $P(t_i|\bar{c_j})$ is the probability that t_i occurs in another class than c_j.

3 Informative Feature Selector (IFS)

The principal aim of a feature selection method is to differentiate between informative and uninformative features by giving highest values to most informative and lowest values to least informative features, which subsequently facilitates the process of reducing the size of the feature vector space. However, the following issues must be taken into account by a feature selection method in the process of weighting the features for text classification. A feature that appears in all documents of a class and does not appear in any documents of other class(es) is a good discriminator and should be given the highest value. Consequently, a feature which equally appears in all classes should be given a lower value. The value assigned to a feature should reflect its degree of discrimination and its correlation between the classes.[3].

Based on these consideration, the IFS method is formulated as follows:

$$\begin{aligned} IFS(t_i, c_j) = \log(&|(P(t_i|c_j)P(\bar{t_i}|\bar{c_j}) - P(t_i|\bar{c_j})P(\bar{t_i}|c_j))| \\ &\tfrac{|P(t_i|c_j) - P(t_i|\bar{c_j})|}{\min(P(t_i|c_j), P(t_i|\bar{c_j}))+1} + 1) \end{aligned} \tag{7}$$

where $P(t_i|c_j)$ is the probability of t_i appearing in class c_j and $P(\bar{t_i}|c_j)$ is the probability of t_i not appearing in class c_j. Similarly $P(t_i|\bar{c_j})$ is the probability of t_i appearing in another class $\bar{c_j}$ and $P(\bar{t_i}|\bar{c_j})$ is probability of t_i not occurring in $\bar{c_j}$.

IFS has been formulated in a way so that the overall value given to a feature is sensitive to changes in the number of features which are absent, shared or present in classes. In order for IFS to assign a value which reflects the usefulness of a feature for classification, and to adhere to above considerations, IFS makes use of the difference between the probability of a feature that appears in both classes, then dividing it by the smallest value between $P(t_i|c_j)$ and $P(t_i|\bar{c_j})$; 1 is added in case the smallest value is zero (this is the 2nd part of the equation). This makes sure the feature is assigned an appropriate value not only according to the differences between $P(t_i|c_j)$ and $P(t_i|\bar{c_j})$ but also according to the probability of t_i occurring in the intersection between c_j and $\bar{c_j}$. However, the calculation so far does not consider the number of documents which do not contain features in all classes; therefore, both the absence and presence of features in classes $P(t_i|c_j)$, $P(\bar{t_i}|c_j)$, $P(t_i|\bar{c_j})$ and $P(\bar{t_i}|\bar{c_j})$ are used in the calculation to reflect the probability of a feature being absent or present in classes. The whole formula makes sure the feature is assigned a value which reflects its usefulness for classification. This is not always the case in some other approaches such as OR, in which the intersection or number of shared features between classes is not taken into account. The maximum and minimum values are between zero and one, and a feature which equally appears in both classes is assigned a minimum value,

[3] We find similar considerations in classical probabilistic information retrieval models (where often the probability that a term occurs in the relevant/non-relevant documents is used) and also in the well-known concept of the inverse document frequency (terms appearing in less documents are deemed good discriminators).

which is zero in case of balanced binary classes. The method adheres to all above mentioned considerations.

4 Evaluation

A 10-fold cross-validation of text classification on both unbalanced and balanced datasets of spam emails and SMS have been carried out to test the performance and sensitivity of the IFS method to the number of documents in classes and the distribution ratio of documents between classes. In this experiment we followed [11] and used binary classification, because the results of binary classification reflect the effectiveness of the used measure more directly than that of multi-class classification [11]. Moreover, resolving the binary text classification also means resolving multi-classes [2]. For this experiment, two supervised algorithms namely Support Vector Machines (SVM) and Neural Network (NN) have been employed, as both algorithms are performing well in text classification [1, 12].

4.1 Dataset and Performance Measure

Two sets of data have been used in our experiments, namely the enron1 email collection [13], in total 5172 emails, in which 1500 are spam email and 3672 are legitimate emails, and an SMS collection, which was introduced in[14, 15] and consists of 5574 SMS messages, where 4827 are legitimate SMS and 747 spam SMS. From these two datasets we have created 3 test datasets, to enable us to test the sensitivity of the methods with respect to allocation and distribution of documents in classes. The first (*balanced*) test dataset consist of 3000 emails with 1500 spam and randomly chosen 1500 from legitimate emails. The 2nd set (*unbalanced*) consists of 4500 emails in which 1500 are spam emails and randomly chosen 3000 legitimate emails. Finally third test dataset (*unbalanced*) is based on the SMS dataset, which consists of 747 spam and randomly chosen 2576 legitimate ones from a total of 3323 SMS messages. The top n subset of features with highest weight were used in experiment, and n was set to 10, 15, 25, 50, 100, 200, 300, 400 and 500.

The performance of all methods was assessed using the F-measure (F_1) based on precision and recall as defined in [2] using micro-averaging.

5 Result Analysis

5.1 Unbalanced datasets

The F-measures of the 10-fold cross validation of the email and SMS datasets based on SVM and NN classifiers are shown in Figures 1, 2, 3, and 4 respectively. Based on the results, IFS is performing well with both classifiers. Across all used feature sets, IFS is either superior or comparable to others, while in some instances it is shown as second best.

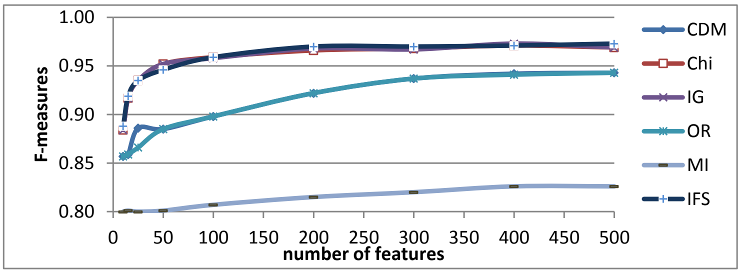


Fig. 1. F-measures of SVM with unbalanced emails dataset

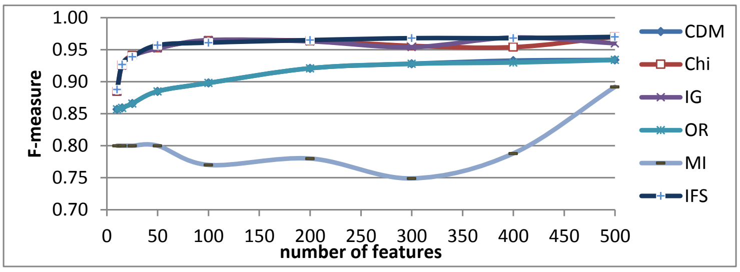


Fig. 2. F-measures of NN with unbalanced emails dataset

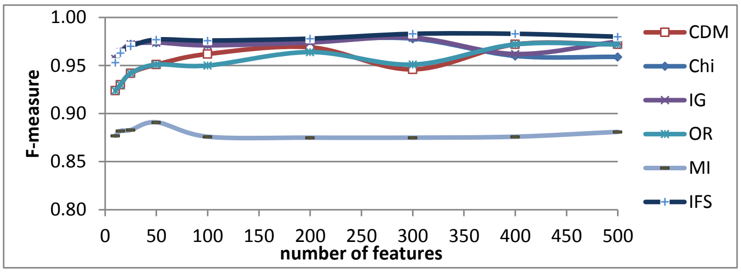


Fig. 3. F-measures of NN with unbalanced SMS collection

5.2 Balanced datasets

The outcome of F-measures of the 10-fold cross validation for above mentioned methods using SVM and the NN classifier with datasets, 3000 emails with ratio of 1:1 respectively, is shown in Figures 5 and 6. Considering F-measure, IFS yet again is performing well in balanced datasets, and it could be noticed from the results that IFS is superior to others and only in some cases slightly scoring behind. In general, we can conclude that IFS is robust in performing well in both SVM and NN classifiers, compared to other methods.

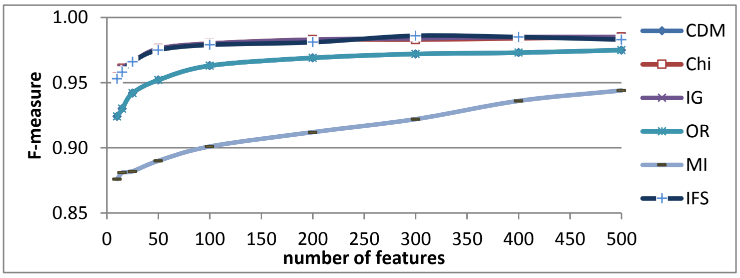


Fig. 4. F-measures of SVM with unbalanced SMS collection

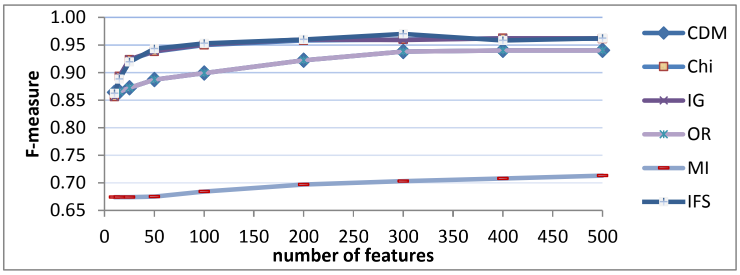


Fig. 5. F-measures of SVM with balanced emails dataset

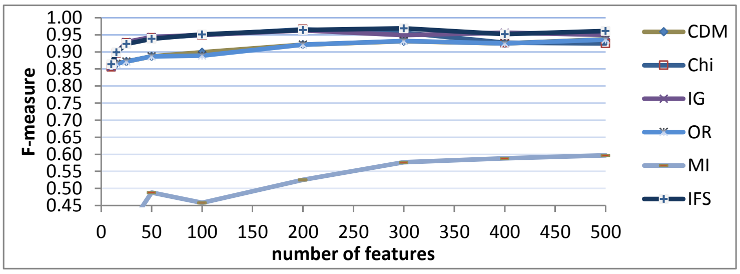


Fig. 6. F-measures of NN with balanced emails dataset

6 Conclusion

This paper has introduced a new method called IFS to select informative features for classification. The method has been evaluated against some well established feature space reduction methods. Throughout the F-measure values of the 10-fold cross validations result, and processing time, IFS has shown to be robust and often superior, at least competitive compared to other methods. In future work, we aim to test IFS on multi classes datasets. We also aim to improve our method by considering and calculating the frequency of a feature in document and add its weight to the overall usefulness of the feature.

References

1. Alan S. Abrahams, Eloise Coupey, Eva X. Zhong, Reza Barkhi, and Pete S. Manasantivongs.: Audience targeting by B-to-B advertisement classification: A neural network approach. Expert Systems with Applications, Elsevier, 40(8): 2777-2791, (2013)
2. Fabrizio Sebastiani.: Machine Learning in Automated Text Categorization. ACM computing surveys (CSUR), 34(1):1-47, (2002)
3. Yiming Yang and Jan O Pedersen.: A Comparative Study of Features selection in text categorization. The Fourteenth International Conference on Machine Learning (ICML), 97:412-420 (1997)
4. Jingnian Chen, Houkuan Huang, Shengfeng Tian, and Youli Qu.: Expert Systems with Applications Feature selection for text classification with Naive Bayes. Expert Systems With Applications, Elsevier, 36(3):5432-5435 (2009)
5. George Foreman.: An Extensive Empirical Study of Feature Selection Metrics for Text Classification. Journal of Machine Learning Research, 3:1289-1305 (2003)
6. Jun-Ming Xu, Xiaojin Zhu, and Amy Bellmore. Fast learning for sentiment analysis on bullying.: In Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining - WISDOM, New York, USA, ACM Press, 1-6 (2012)
7. Hiroshi Ogura, Hiromi Amano, and Masato Kondo.: Feature selection with a measure of deviations from Poisson in text categorization. Expert Systems with Applications, Elsevier, 36(3):6826-6832 (2009)
8. Dunja Mladenic and Marko Grobelnik.: Feature selection for unbalanced class distribution and Naive Bayes. In Proceedings of the Sixteenth International Conference on Machine Learning, 258-267 (1999)
9. Yan Xu. A Comparative Study on Feature Selection in Unbalance Text Classification. In Information Science and Engineering (ISISE), 2012 International Symposium, IEEE-CPS, 44-47 (2012)
10. Matthew Chang and Chung Keung Poon.: Using phrases as features in email classification.The Journal of Systems and Software, Elsevier, 82(6): 1036-1045 (2009)
11. Hiroshi Ogura, Hiromi Amano, and Masato Kondo.: Feature selection with a measure of deviations from Poisson in text categorization. Expert Systems with Applications, Elsevier, 36(3):6826-6832 (2009)
12. Yang, Y. and Liu, X., A re-examination of text categorization methods. Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval SIGIR 99, pp.4249 (1999)
13. Vangelis Metsis, I Androutsopoulos, and G Paliouras.: Spam filtering with naive bayes - which naive bayes?. CEAS, the third conference on Email and Anti-Spam, 27-28 (2006)
14. Gordon V. Cormack, Jose Mara Gomez Hidalgo, and Enrique Puertas Sanz.: Feature engineering for mobile (SMS) spam filtering. In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, 871-872 (2007)
15. Tiago A Almeida, Jose Maria G Hidalgo, and Akebo Yamakami.: Contributions to the study of SMS spam filtering: new collection and results. In Proceedings of the 11th ACM symposium on Document engineering, 259-262 (2011)

FGDB: Workshop on Database Systems - Database Support for Big Data Applications

Towards Programmability of a NUMA-aware Storage Engine

Dirk Habich, Johannes Schad, Thomas Kissinger, and Wolfgang Lehner

Technische Universität Dresden, Database Systems Group,
{dirk.habich,johannes.schad,thomas.kissinger,wolfgang.lehner}@tu-dresden.de
WWW home page: https://wwwdb.inf.tu-dresden.de

Abstract. The SQL database language was originally intended for application programmers. However, after more than 20 years of language extensions, SQL can only be generated by software components and is no longer suitable for an increasing user base like knowledge workers or data scientists, who want to work with data in an interactive fashion. The original idea of declarative query languages, telling the system what information to retrieve and not how to retrieve it, is still relevant. However, procedural elements are extremely worthwhile and have to be part of a next generation database programming language without compromising performance and scalability. To tackle this challenge, we are going to present our overall approach consisting of a highly-scalable NUMA-aware storage engine *ERIS* and a novel appropriate procedural programming approach on top of *ERIS* in this paper.

1 Introduction

In the era of *Big Data*, the requirements for data management systems are manifold, whereas performance and scalability are still the two most important technical requirements. Aside from these technical requirements, the relevance of the usability aspect increases. Generally, all these aspects are not new and already well-established in the database community over a long period. While most research work has been focused on solutions for satisfying the performance and scalability aspects, the usability aspect has been singularly addressed in various work as well e.g., the map-reduce programming concept or the PACT programming model [3] come to mind. From our point of view, the most challenging issue for tomorrow's data management systems is the consolidation of technical and usability aspects, taking programmability into account.

As already mentioned in the Claremont [2] as well as in the Beckman [1] report on database research, the user base for DBMS is rapidly growing and new users bring new expectation with regard to programmability or programming

Copyright © 2015 by the paper's authors. Copying permitted only for private and academic purposes. In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

abstractions against very large data sets. Today, most of the users are unhappy (i) with the offered user interfaces and (ii) with the heavyweight system architecture. While the declarative way of SQL is often perceived as too restricted, procedural opportunities like stored procedures are too complex. Furthermore, the extensibility of traditional database systems using procedural opportunities is limited, which has an influence on usability as well as performance. Therefore, traditional DBMS are often degraded to storage units today and enhanced functionality is implemented on top or in competing data processing systems like Hadoop. However, with increasing data volumes, exporting massive data sets from the database and conduction data-intensive processing in user applications is no longer a valid opportunity. Tomorrow's applications will have to push-down their procedural logic into the database to work close to the data.

In order to tackle technical and usability requirements for database systems in a unified way to satisfy a growing user base like knowledge workers and data scientists, we are pursuing a holistic approach with novel and unique features:

NUMA-aware Storage Engine: The ever-growing demand for more computing power forces hardware vendors to put an increasing number of multiprocessors into a single server system, which usually exhibits a non-uniform memory access (NUMA). Based on that hardware foundation, we developed a new NUMA-aware in-memory storage engine *ERIS* that is based on a data-oriented architecture [7]. In contrast to existing approaches that focus on transactional workloads on a disk-based DBMS, *ERIS* aims at tera-scale analytical workloads that are executed entirely in main memory. *ERIS* uses an adaptive data partitioning approach exploiting the topology of the underlying NUMA platform and significantly reduces NUMA-related issues. As we have shown in [5], we achieve *more than a linear* speedup for index lookups and scalable parallel scan operations.

Programmability: Our NUMA-aware in-memory storage engine currently supports three basic storage operations that are required to run analytical queries: scan, lookup, and insert/upsert. Aside from reading operations, fast writing operations are necessary, especially to materialize large intermediate results. Based on those storage operations, we are currently developing a new procedural user interface, so that users are able to program their analytical tasks. In this paper, our primary goal is to introduce the procedural user interface and show how it can be used to add support for SQL functionality on *ERIS*. On the one hand, this new user interface facilitates easy and abstract programming in a procedural fashion and for a broad user base. On the other hand, the procedural analytical tasks are efficiently executable on our NUMA-aware storage engine *ERIS*.

The remainder of the paper is structured as follows: In the following section, we briefly review our NUMA-aware in-memory storage system. Then, we introduce our programmability concept using partitioning schemes as first-class citizen operators in Section 3. In Section 4, we discuss some execution perspectives. We conclude the paper, with some remarks regarding related work, future work and short conclusion.

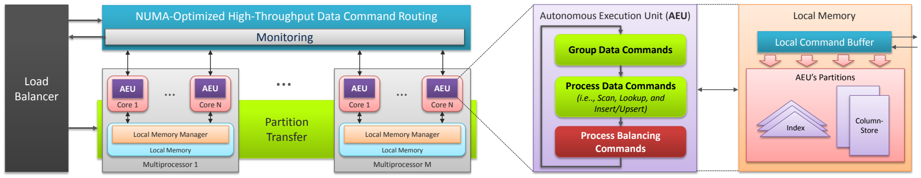


Fig. 1. Architectural Overview of ERIS and AEU Details.

2 The *ERIS* Storage Engine

In [5], we introduced our NUMA-aware all-in-memory storage engine for terascale analytical workloads. Fundamentally, NUMA systems consist of several interconnected multiprocessors (also denoted as nodes). Each multiprocessor contains multiple processing units (cores) and an integrated memory controller (IMC). Although the installed main memory is distributed across the IMCs of the different multiprocessors, each multiprocessor can access each memory location of the system. Therefore, latency and bandwidth of memory accesses depend on the distance between the requesting multiprocessor and the multiprocessor which has the data in local memory. The local memory associated with each multiprocessor is accessed with low latency at a high bandwidth. In contrast, remote memory is accessed via point-to-point connections between the multiprocessors that add latency and limit the achievable bandwidth. As we have shown in [5], the latency of remote access is approximately 10 times higher and the bandwidth is limited to about 11% compared to local accesses in the worst case.

2.1 *ERIS* Architecture Design

To efficiently tackle the NUMA-specific properties from a performance perspective for analytical workloads, our in-memory storage engine *ERIS* is treated like a distributed system using a data-oriented architecture [7] where each data object is logically partitioned. Figure 1 shows the overall architecture of *ERIS* and its individual components. The central components are the worker threads, which we call *Autonomous Execution Units (AEU)*. Each core, respectively hardware context, of a NUMA system runs exactly one AEU and an AEU never leaves its core. All AEUs that are pinned on the same multiprocessor use a common memory manager, because they share the same local main memory. Every AEU gets assigned a set of disjoint data partitions —each belonging to a different data object— which it stores in local memory. Further, every AEU holds exclusive access rights to its partitions. This approach restricts memory accesses of an AEU to the multiprocessor's local main memory and data objects do not have to be protected against concurrent accesses via latches. ERIS primarily uses range partitioning to split data objects (e.g., tables) into partitions.

On the right hand side of Figure 1, we depict the AEU loop as well as the local memory organization of an AEU. Each AEU maintains local data command buffers and the actual data object partitions (either stored as a column-store, a row-store, or an index). In the first stage of the loop, the AEU scans its data command buffer (i.e., scan, lookup, or insert/upsert), which is periodically filled by the routing layer, and groups commands by the accessed data object and the command type. This optimization step allows to coalesce the same type of access to the same partition. For instance, an AEU is able to execute multiple scan commands on the same partition with a single scan and is thereby implementing scan sharing in combination with MVCC to ensure isolation. Moreover, the command grouping allows us to execute multiple index lookup or insert/upsert operations in a single batch operation to hide the main memory latency. Following the grouping step, the AEU actually processes its data command buffer, which is the most time consuming part of the loop. Afterwards, the AEU checks its command buffer for pending balancing or transfer commands. Such commands force an AEU to grow or shrink its partition or to transfer a range of its partition to another AEU.

NUMA-Optimized High-Troughput Data Command Routing: As shown in Figure 1, the data command routing is the most essential part of ERIS, because AEUs have to be supplied with data commands just in time. Especially during the execution of analytical queries, large amounts of data commands have to be routed between AEUs (e.g., lookup operations during a join). Thus, the main goal of the data command routing is to distribute data commands at a high throughput. A data command consists of a storage operation type (i.e., scan, lookup, or insert/upsert), a data object identifier, a reference to a callback function, a data segment that contains all the necessary parameters for the storage operation (e.g., a batch of keys for the lookup or filters for a scan), and additional data that is necessary for the query processing.

Load Balancing Besides data command routing, *ERIS* owns a NUMA-aware load balancer component to adapt the data partitioning to a changing workload. Since *ERIS* aims at analytical workloads, the maximization of parallel processing is the main objective of the load balancer. Thus, there is no need for inter-data-object balancing, for instance to colocate certain partitions of different data objects on the same AEU as it is beneficial for transactional workloads. The *ERIS* adaption loop starts with the monitoring of different metrics on a per data object level. Based on the captured metrics, the load balancer periodically checks the load of *ERIS* for imbalances. If the standard deviation between the different AEUs exceeds a given threshold, the load balancer executes a load balancing algorithm that calculates a new target partitioning. With the help of the current and the targeted partitioning, the load balancer computes a series of balancing commands that are routed to the involved AEUs.

2.2 Summary

ERIS is a NUMA-aware purely in-memory storage engine for tera-scale analytical workloads that is based on a data-oriented architecture. *ERIS* uses a

NUMA-optimized high-throughput data command routing as well as a configurable NUMA-aware load balancing algorithm to achieve a maximum of parallelism to execute analytical queries with low latency. Our analysis in [5] shows that *ERIS* greatly improves memory locality and cache usage and thus scales even on large-scale NUMA platforms. Since *ERIS* only provides storage operation primitives, the next step is to implement a query processing framework on top of *ERIS* and to evaluate the performance of more complex queries. Query processing with *ERIS* requires techniques for distributed systems and poses additional challenges for load balancing. In particular, data partitioning is a key aspect to enable data parallelism for processing.

3 *ERIS*-Programmability for SQL Functionality

In order to support the creation of complex analytical workloads, *ERIS* needs a flexible but easy to use interface for its storage and processing capabilities. Instead of specializing *ERIS* to one specific database interface, we equip it with a general purpose programming interface which can be used as the basis for an implementation of different high-level interfaces like SQL. Many state-of-the-art databases rely on a set of hardcoded physical operators to execute SQL statements. Thereby, a SQL statement is transformed into a graph representation, the query plan, and that graph representation is subsequently interpreted to call adequate physical operators. *ERIS*, on the other hand, does not limit data processing to a fixed set of relational operators. Instead, *ERIS* provides a means to run user provided code on data but enforces a certain structuring of user code and control flow in order to enable efficient and parallel execution. In order to ideally exploit *ERIS*' data processing capabilities, the *ERIS* programming framework's main design objective is to support data locality and parallel execution of computations. To enable that, we require for each user function, an explicit description of the data partitions the user function has to be granted access to.

3.1 Partitioning Schemes

The *ERIS* programming framework is built around three data structures for the desired SQL functionality: `tuples`, `relations`, and `partitions`. Just like in typical databases, `tuples` are flat containers of data types like numbers, strings or dates. Both `relations` and `partitions` are sets of tuples with a fixed schema; but only `relations` can be named and loaded from *ERIS* by name. There is no direct way to access the tuples of a relation. Instead `relations` have to be partitioned first and subsequently the individual parts of the relation can be processed by accessing their tuples. Requiring relations to be partitioned before their tuples can be accessed, is the key to enabling data parallel processing in *ERIS*.

The current version of our *ERIS* programming framework supports three partitionings from a programming perspective as first-class citizens, which are necessary to support SQL functionality.

Individuals: The first partitioning scheme is called *individuals* partitioning and it simply turns every tuple of the target relation into its own partition. In equation 1, we define the individuals partitioning as a family of sets over a base relation $R_1 \times \ldots \times R_n$ where each element of the base relation forms its own subset.

$$\mathbf{Ind}(R) \ = \ \{\,\{t\}\,|\,t \in R\,\} \tag{1}$$

Equivalents: The second partitioning groups tuples having the same value in a fixed set of attributes. Again, the *equivalents* partitioning is a family of sets. This time, a subset contains all tuples which are equivalent to a representative tuple. The representative tuple is drawn from a selection on the base relation of the partitioning. Two tuples are considered equivalent if they have the same value in all attributes that they have in common.

$$\mathbf{Equiv}(R, a_1, \ldots, a_k) \quad = \quad \{\,\{t\,|\,t \ \in \ R \wedge t \ \equiv \ c\}\,|\,c \ \in \ \pi_{a_1,\ldots,a_k}(R)\,\} \tag{2}$$

All: The final partitioning does not partition its input relation at all but simply declares the whole relation as a single partition. *All* is needed because some algorithms do not lend themselves to the form of data parallelism promoted by partitionings. This is especially true for all kinds of aggregations that fold a complete relation into a single tuple or value.

$$\mathbf{All}(R) \ = \ \{\,R\,\} \tag{3}$$

3.2 Processing Functions for Partitioning Schemes

Once a relation is partitioned using one of our three partitioning schemes, the individual parts can be used for processing. The processing of tuples has to be encapsulated in a pure function, the so called *user function* which takes one or multiple partitions as parameter and returns a single partition. The framework implements this concept by embedding tuple processing capabilities in $process(P_1, \ldots, P_n, f\backslash n)$, a second-order function which is parametrized with a list of input partitionings and a first-order *user function*. The arity of the *user function* always has to match the number of partitionings supplied to *process*.

In the simple case, $process(P, f\backslash 1)$ is applied to a single relation P and a *user function* $f\backslash 1$ with arity 1. In that case, *process* applies $f\backslash 1$ to each subset of partition P, creates a new relation by unioning the results of each application of $f\backslash 1$, and finally returns the new relation as overall result of the second-order function.

$$process(P_1, \ldots, P_n, f\backslash n) = \bigcup_{p_1 \in \mathbf{P_1}} \ldots \bigcup_{p_n \in \mathbf{P_n}} f(p_1, \ldots, p_n) \tag{4}$$

Equation 4 shows the generalization of *process* to multiple input partitions. In this case, $process(P_1, \ldots, P_n, f\backslash n)$ applies $f\backslash n$ to all possible combinations of subsets of the input partitionings. Again, the results of all applications are

Listing 1.1. Function for computing the relational projection

```
def projection(r: Relation, attrs: Attribute*): Relation = {
  val d = process(r.individuals) { partition =>
    val tuples = for (tuple <- partition) yield {
      new Tuple(attrs.map(a => tuple(a))
    }
    return new Partition(tuples)
  }
  process(d.equivalents(attrs)) { partition =>
    // eliminate duplicates
    return new Partition(partition.randomTuple())
  }
}
```

unioned into a new relation which is the final result of *process*$(P_1, \ldots, P_n, f\backslash n)$. Independent of the number of partitionings, $f\backslash n$ has to map its inputs to a single output schema or it will break the unioning of output partitions. In this way, *process*$(P_1, \ldots, P_n, f\backslash n)$ completes the story of data-parallelism in *ERIS* by enabling the parallel processing of multiple elements of one or more partitionings.

3.3 Realization of the Relational Algebra for SQL functionality

To demonstrate the adequacy of our presented framework, we use it to outline an implementation of the relational algebra on *ERIS*, whereas our language approach is based on Scala.

Projection: $\mathbf{P} = \boldsymbol{\pi_{a_1,\ldots,a_k}(R)}$ The relational projection requires a two step approach: first unwanted attributes are removed from the individual tuples and then duplicates are removed from the resulting relation.

Listing 1.1 shows the Scala code for that operation. The first *process* is applied to the individuals of the target relation R. The user function simply iterates the partition, extracts the relevant attributes from each tuple and uses those attributes to create a new tuple. Finally, all tuples are combined into a new partition and that partition is returned as a result of the user function. It has to be noted that in this case, because the partition only ever contains a single tuple, the more convenient parameter to the user function would be a simple tuple instead of a partition. But for the sake of generality we avoid the introduction of special cases at this point.

The second *process* application is necessary to eliminate duplicates which can be introduced when attributes are removed from relations. Duplicate elimination is not an automatic feature of the underlying `relation` data structure as it mapped onto *ERIS* containers which do not exhibit this feature. The idea is to perform an equivalents partitioning on all attributes of the relation. In this setup each partition can only hold either a single tuple or multiple duplicated tuples

Listing 1.2. Function for computing the cartesian product

```
def cross(r: Relation, s: Relation): Relation =
  process(r.individuals, s.individuals) { (p1, p2) =>
    val tuples = for (tuple1 <- p1; tuple2 <- p2) yield {
      new Tuple(tuple1.attributes ++ tuple2.attributes)
    }
    return new Partition(tuples)
  }
```

as all tuples in the partition have to be equal in all attributes. Most of the work of removing duplicates is of course done by the partitioning function itself. The user function simply selects a random element of the input partition and creates a new partition with the one element as only content.

Selection: $S = \sigma_P(R)$ The selection can be performed on a per tuple basis and therefore, a single processing of individuals is sufficient. The user function tests for each tuple whether it satisfies the predicate and creates the output partition with all tuples that have passed the test.

Cartesian Product: $P = R \times S$ Due to *process*' support for multiple input partitionings, the implementation of the cartesian product in listing 1.2 is straightforward. *process* is applied to the *individuals* partitionings of the relations R and S. Therefore, the *user function* is applied once on every possible pair of input tuples and merely has to combine the attributes of these tuples into a single output tuple and pack that tuple into a new partition.

Union: $U = R \cup S$ Set union is a typical example of an operation that is hard to parallelize with the partitioning approach as there is minimal independent per element computation. Listing 1.3 shows a simple solution relying on the *all* partitioning and simply concatenates all tuples of R with all tuples of S. As with the relational projection, this process can introduce duplicate entries which have to be removed in a second step.

Difference: $D = R \backslash S$ The set difference operator can be implemented as multiple parallel set inclusion tests using an *individuals* and an *all* partitioning. In listing 1.4 each tuple of R is combined with all tuples of S and the user function performs a single set inclusion test to check if the tuple has to be included in the output relation.

4 Execution and Complex Query Example

Our *ERIS* programming framework follows recent proposals [4, 6] to connect best performance with high programmer productivity by relying on domain specific code generation. As already shown in the previous section, users of our framework write their programs in Scala and rely on a set of framework operations to interact with *ERIS*' storage and processing resources. The framework's operations are our partitioning schemes. In Section 4.1, we introduce our execution approach, followed in Section 4.2 by a short example of a complex query.

4.1 Execution

The overall compilation of programs written in Scala and using our *ERIS* framework is driven by the translation of partitionings and the $process(P_1, \ldots, P_n, f \backslash n)$ function as these constructs are directly mapped onto invocations of *ERIS*' native storage operations like `ScanTable` to scan a table. The `ScanTable` operation is responsible for all reading activitivies in *ERIS*, whereas the `ScanTable` can be parameterized with a single data container and a callback function being called whenever it has collected a chunk of records from the target container. The callback processes the records and pushes the resulting data to a new container using an insert operation. `ScanTable` invokes the callback with chunks of records until all records of the target container have been processed.

A single execution of `ScanTable` in its basic form is sufficient to translate an *execute* over a single *individuals* partitioning. The `ScanTable`'s callback is created with a loop that iterates the individual records of the input chunk. The body of the loop contains the translation of the *user function*, which is thereby applied on each individual record. At the end of each loop, an insert message materializes the result of the loop body into an output container.

The translation of an *execute* over an *equivalents* or an *all* partitioning of a single relation is of a similar character. `ScanTable` can be parametrized to call its callback either with all records that fulfill certain requirements or simply with all records of a container. These parametrization options exactly match the semantics of the *equivalents* and *all* partitionings, so the translation of both partitionings is a straightforward generation of `ScanTable` parameters. Requiring the grouping of certain or all records of a container has of course a negative impact on the locality of data access and should be used as sparingly as possible. Because they are already called with the correct set of records, the callback func-

Listing 1.3. Function for computing the set union

```
def union(r: Relation, s: Relation): Relation = {
  val d = process(r.all, s.all) { (p1, p2) =>
    return new Partition(p1.tuples ++ p2.tuples)
  }
  process(Ud.equivalents(r.attributes ++ s.attributes))
    { partition =>
      return new Partition(partition.randomTuple())
    }
}
```

Listing 1.4. Function for computing the set difference

```
def difference(r: Relation, s: Relation): Relation =
  process(r.individuals, s.all) { (p1, p2) =>
    val tuples = for (tuple <- p1 if !p2.contains(p1)) yield {
      tuple
    }
    new Partition(tuples)
  }
```

tions for *equivalents* and *all* do not need an inner loop to iterate over records. They simply contain the cross-compiled code of the respective user function and at the end an insert message to materialize the output tuples of the user function.

The code for a multi relation *execute* is a little bit more involved because a `ScanTable` will only ever touch one single *ERIS* container. Combining data from containers $C_1, \ldots, C_n$ requires n sequential `ScanTable` runs, one on each of the containers. As an example, we will examine a *process* over three *individuals* partitionings P_1, P_2, P_3. Processing will start with a `ScanTable` on P_1 with a callback that iterates over the individual records. For each record the callback will request a new `ScanTable` over P_2 and send the current record as additional data to the callback of the new scan. Each callback over P_2 will in turn iterate over the individual records of P_2, request a third `ScanTable` on P_3, and send the records from P_1 and P_2 as additional data to the callback of the scan over P_3. The callback of the final scan iterates once again over the records of its container, combines each one with the other two records, and hands them to the code of the user function in its iterator loop. Similar to the earlier examples, the result records produced by the user code have to be inserted into a new container at the end of the loop. The algorithm for three *individuals* partitionings can be easily generalized to other types of partitionings and relation counts, so we abstain from a more detailed description at this point.

4.2 Complex Query Example

We finish our discussion of the *ERIS* programming framework with a high level example that combines multiple relational operators to implement the simple SQL query depicted in listing 1.5.

Listing 1.5. A simple SQL example

```
SELECT student.name grade.course grade.grade
FROM student, grade
WHERE student.id = grade.studentID AND student.age > 30;
```

Listing 1.6 shows a straightforward translation from SQL to relational operators. Each operator is annotated with the *process* calls it requires and with the number of *ScanTable* operations necessary to execute the *process* calls in *ERIS*.

Listing 1.7. Optimized SQL translation. 3 x ScanTable

```
// Two input relations, 2 x ScanTable
val nonUnique = process(students.individuals, grades.individuals) {
    // join student and grade
    // select on joined
    // project on selected
}

// One input relation, 1 x ScanTable
val result = process(nonUnique.equals(name, course, grade) {
    // filter duplicates
}
```

In total, the direct translation of the SQL statement requires 5 *ScanTable* operations: 2 for the *cross*, 1 for the *select*, and 2 more for the *projection.*

Listing 1.6. Naive SQL translation. 5 x ScanTable

```
// 2 x ScanTable
// c = process(students.individuals, grades.individuals)
val c = cross(students, grades)

// s = process(c.individuals), 1 x ScanTable
val s = select(c, { t: Tuple =>
    return t(”student.id”) == t(”grade.studentID”)
        && t(”student.age”) > 30})

// nu = process(s.individuals), 1 x ScanTable
// result = process(nu.equivalents(...)), 1 x ScanTable
val attrs = [”student.name”, ”grade.course”, ”grade.grade”]
val result = project(s, attrs)
```

The straightforward translation provides a good opportunity for an important domain specific optimization. The first *process* joins the students and grades tuples and the two following *process* invocations process the *individuals* partitioning of the crossed relations. The individual tuples of the crossed relations are already materialized in the first *process* invocation. Therefore, we are able to append the bodies of the second and third *process* to the body of the first *process* and do all work in a single *process* invocation. This optimization is a good example of what is possible with a compiler that can be extended with domain specific knowledge. Listing 1.7 shows an outline of the optimized code, where the cross, select, and the first part of the project operations are fused into a single *process* invocation. The optimized version of the code reduces the number of required *ScanTable* operations to three and avoids two costly materializations of intermediate data containers.

5 Future Work and Conclusion

In this paper, we have presented our programmability idea for an in-memory storage engine which is based on a data-oriented architecture. Instead of relying on hardcoded physical operators, our approach introduces data partitionings as first-class citizens on the programming layer as second-order functions, which can be parameterized using any kind of first-order functions. Fundamentally, this approach is similar to the PACT programming model [3], whereas our concept is not limited to the key-value data format. Based on our overall idea, we want to extend our work in different directions. First, we want to build a more user friendly domain specific language (DSL) based on our partitioning schemes for SQL. Second, we are going to map different languages like SQL or Pig to our DSL to support these higher-level languages in an efficient way. Third, we will investigate further application domains to find additional specific partitioning schemes.

References

1. Abadi, D., Agrawal, R., Ailamaki, A., Balazinska, M., Bernstein, P.A., Carey, M.J., Chaudhuri, S., Dean, J., Doan, A., Franklin, M.J., Gehrke, J., Haas, L.M., Halevy, A.Y., Hellerstein, J.M., Ioannidis, Y.E., Jagadish, H.V., Kossmann, D., Madden, S., Mehrotra, S., Milo, T., Naughton, J.F., Ramakrishnan, R., Markl, V., Olston, C., Ooi, B.C., Ré, C., Suciu, D., Stonebraker, M., Walter, T., Widom, J.: The beckman report on database research. SIGMOD Rec. 43(3), 61–70 (Dec 2014)
2. Agrawal, R., Ailamaki, A., Bernstein, P.A., Brewer, E.A., Carey, M.J., Chaudhuri, S., Doan, A., Florescu, D., Franklin, M.J., Garcia-Molina, H., Gehrke, J., Gruenwald, L., Haas, L.M., Halevy, A.Y., Hellerstein, J.M., Ioannidis, Y.E., Korth, H.F., Kossmann, D., Madden, S., Magoulas, R., Ooi, B.C., O'Reilly, T., Ramakrishnan, R., Sarawagi, S., Stonebraker, M., Szalay, A.S., Weikum, G.: The claremont report on database research. SIGMOD Rec. 37(3), 9–19 (Sep 2008)
3. Alexandrov, A., Battré, D., Ewen, S., Heimel, M., Hueske, F., Kao, O., Markl, V., Nijkamp, E., Warneke, D.: Massively parallel data analysis with pacts on nephele. PVLDB 3(2), 1625–1628 (2010)
4. Brown, K.J., Sujeeth, A.K., Lee, H., Rompf, T., Chafi, H., Odersky, M., Olukotun, K.: A heterogeneous parallel framework for domain-specific languages. In: Parallel Architectures and Compilation Techniques (PACT), 2011 International Conference on. pp. 89–100. IEEE (2011)
5. Kissinger, T., Kiefer, T., Schlegel, B., Habich, D., Molka, D., Lehner, W.: ERIS: A numa-aware in-memory storage engine for analytical workload. In: International Workshop on Accelerating Data Management Systems Using Modern Processor and Storage Architectures. pp. 74–85 (2014)
6. Klonatos, Y., Koch, C., Rompf, T., Chafi, H.: Building efficient query engines in a high-level language. Proceedings of the VLDB Endowment 7(10), 853–864 (2014)
7. Pandis, I., Johnson, R., Hardavellas, N., Ailamaki, A.: Data-oriented transaction execution. PVLDB 3(1-2), 928–939 (2010)

Die Apache Flink Plattform zur parallelen Analyse von Datenströmen und Stapeldaten

Jonas Traub*, Tilmann Rabl*, Fabian Hueske†,
Till Rohrmann† und Volker Markl*§

*Technische Universität Berlin, FG DIMA, Einsteinufer 17, 10587 Berlin
†data Artisans GmbH, Tempelhofer Ufer 17, 10963 Berlin
§DFKI GmbH, Intelligente Analytik für Massendaten, Alt-Moabit 91c, 10559 Berlin
`{jonas.traub,rabl,volker.markl}@tu-berlin.de,`
`{fabian,till}@data-artisans.com  http://www.dima.tu-berlin.de`

Abstract. Die Menge an analysierbaren Daten steigt aufgrund fallender Preise für Speicherlösungen und der Erschließung neuer Datenquellen rasant. Da klassische Datenbanksysteme nicht ausreichend parallelisierbar sind, können sie die heute anfallenden Datenmengen häufig nicht mehr verarbeiten. Hierdurch ist es notwendig spezielle Programme zur parallelen Datenanalyse zu verwenden. Die Entwicklung solcher Programme für Computercluster ist selbst für erfahrene Systemprogrammierer eine komplexe Herausforderung. Frameworks wie Apache Hadoop MapReduce sind zwar skalierbar, aber im Vergleich zu SQL schwer zu programmieren.
Die Open-Source Plattform Apache Flink schließt die Lücke zwischen herkömmlichen Datenbanksystemen und Big-Data Analyseframeworks. Das Top Level Projekt der Apache Software Foundation basiert auf einer fehlertoleranten Laufzeitumgebung zur Datenstromverarbeitung, welche die Datenverteilung und Kommunikation im Cluster übernimmt. Verschiedene Schnittstellen erlauben die Implementierung von Datenanalyseabläufen für unterschiedlichste Anwendungsfälle. Die Plattform wird von einer aktiven Community kontinuierlich weiter entwickelt. Sie ist gleichzeitig Produkt und Basis vieler Forschungsarbeiten im Bereich Datenbanken und Informationsmanagement.

Stichwörter: Big-Data, Datenstromverarbeitung, Stapelverarbeitung, Datenanalyse, Datenbanken, Datenanalyseabläufe

1 Einleitung

Dank der stark fallenden Kosten für die Datenspeicherung, Cloud-Speicherangeboten und der intensivierten Nutzung des Internets, steigt die Menge an verfügbaren Daten sehr schnell an. Während der theoretische Wert der Analyse dieser Daten unbestritten ist, stellt die tatsächliche Datenauswertung eine

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

große Herausforderung dar. Konventionelle Datenbanksysteme sind nicht länger in der Lage mit den enormen Datenmengen und der dynamischen oder fehlenden Struktur der Daten umzugehen.

Das Forschungsprojekt Stratosphere[1] verfolgt das Ziel die Big-Data Analyseplattform der nächsten Generation zu entwickeln und damit die Analyse sehr großer Datenmengen handhabbar zu machen. Im Jahr 2014 wurde das im Stratosphere Projekt entwickelte System unter dem Namen Flink[1] zunächst ein Apache Incubator Projekt und später ein Apache Top Level Projekt.

Im Vergleich zu anderen verteilten Datenanalyseplattformen, reduziert Flink die Komplexität für Anwender durch die Integration von traditionellen Datenbanksystemkonzepten, wie deklarativen Abfragesprachen und automatischer Abfrageoptimierung. Gleichzeitig erlaubt Flink *schema-on-read*[2], ermöglicht die Verwendung von benutzerdefinierten Funktionen und ist kompatibel mit dem Apache Hadoop MapReduce Framework[3]. Die Plattform hat eine sehr gute Skalierbarkeit und wurde auf Clustern mit hunderten Maschinen, in Amazons EC2 und auf Googles Compute Engine erprobt.

Im Folgenden stellen wir in Abschnitt 2 die Architektur der Flink Plattform näher vor und zeigen Bibliotheken, Schnittstellen und ein Programmbeispiel in Abschnitt 3. Abschnitt 4 beschreibt Besonderheiten in der Datenstromanalyse von Apache Flink im Vergleich zu anderen Plattformen. Abschließend stellen wir in Abschnitt 5 weiterführende Publikationen vor.

2 Architektur

Abbildung 1 zeigt eine Übersicht der Architektur der Apache Flink Plattform. Die Basis von Flink ist eine einheitliche Laufzeitumgebung in der alle Programme ausgeführt werden. Programme in Flink sind strukturiert als gerichtete Graphen aus parallelisierbaren Operatoren, welche auch Iterationen beinhalten können [6]. Bei der Ausführung eines Programms in Flink werden Operatoren zu mehreren parallelen Instanzen segmentiert, welche jeweils einen Teil der Datentupel verarbeiten (Datenparallelität). Im Gegensatz zu Hadoop MapReduce, werden Programme in Flink nicht in nacheinander auszuführende Phasen (Map und Reduce) geteilt. Alle Operatoren werden nebenläufig ausgeführt, sodass die Ergebnisse eines Operators direkt zu folgenden Operatoren weitergeleitet und dort verarbeitet werden können (Pipelineparallelität). Neben der verteilten Laufzeitumgebung für Cluster, stellt Flink auch eine lokale Laufzeitumgebung bereit. Diese ermöglicht es Programme direkt in der Entwicklungsumgebung auszuführen und zu debuggen. Flink ist kompatibel mit einer Vielzahl von Clustermanagement-

[1] `http://flink.apache.org`

[2] Bei *schema-on-read* werden Daten in ihrer ursprünglichen Form gespeichert, ohne ein Datenbankschema festzulegen. Erst beim Lesen der Daten werden diese in ein abfragespezifisches Schema überführt, was eine große Flexibilität bedeutet.

[3] `http://hadoop.apache.org`

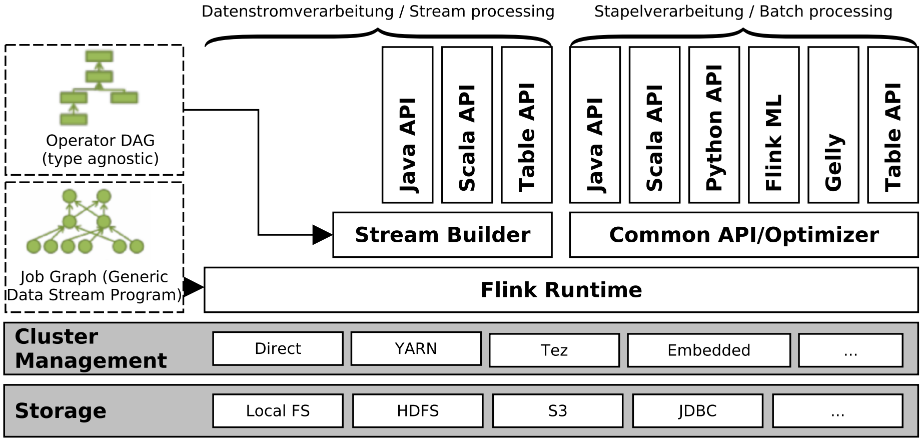


Abb. 1. Architektur- und Komponentenübersicht der Apache Flink Plattform.

und Speicherlösungen, wie Apache Tez[4], Apache Kafka[5] [9], Apache HDFS[3] [10] und Apache Hadoop YARN[3] [12].

Zwischen der Laufzeitumgebung und den Programmierschnittstellen (API), sorgen *Stream Builder* und *Common API* für die Übersetzung von gerichteten Graphen aus logischen Operatoren in generische Datenstromprogramme, welche in der Laufzeitumgebung ausgeführt werden. In diesem Schritt erfolgt auch die automatische Optimierung des Datenflussprogramms. Während der Anwender beispielsweise lediglich einen Join spezifiziert, wählt der integrierte Optimierer den für den jeweiligen Anwendungsfall besten konkreten Join-Algorithmus aus.

Der folgende Abschnitt gibt eine Übersicht über die oberste Schicht der Flink Architektur, welche aus einem breiten Spektrum von Bibliotheken und Programmierschnittstellen besteht.

3 Bibliotheken und Schnittstellen

Nutzer von Apache Flink können Abfragen in verschiedenen Programmiersprachen spezifizieren. Zur Analyse von Datenströmen und zur Stapelverarbeitung stehen jeweils eine Scala und eine Java API zur Verfügung. Stapeldaten können außerdem mit einer Python API verarbeitet werden. Alle APIs stellen dem Programmierer generischen Operatoren wie Join, Cross, Map, Reduce und Filter zur Verfügung. Dies steht im Gegensatz zu Hadoop Map Reduce wo komplexe Operatoren als Folge von Map- und Reducephasen implementiert werden müssen. Listing 1 zeigt eine Wordcount-Implementierung in der Scala Stream Processing API. Eine Implementierung zur Stapelverarbeitung ist analog zu diesem Beispiel unter Auslassung der Window-Spezifikation möglich.

[4] `http://tez.apache.org`
[5] `http://kafka.apache.org`

```
1 case class Word (word: String, frequency: Int)
2 val lines: DataStream[String] = env.fromSocketStream(...)
3 lines.flatMap{line => line.split(" ")}
4     .map(word => Word(word,1))}
5     .window(Time.of(5,SECONDS)).every(Time.of(1,SECONDS))
6     .groupBy("word").sum("frequency").print()
```

Listing 1. Eine Wordcount-Implementierung unter Verwendung der Scala Stream Processing API von Apache Flink.

In der ersten Zeile wird ein Tupel bestehend aus einem String und einer Ganzzahl definiert. Programmzeile 2 gibt einen Socketstream an von dem ein Textdatenstrom zeilenweise eingelesen wird. In Zeile 3 wird ein FlatMap-Operator angewendet, welcher Zeilen als Eingabe erhält, diese an Leerzeichen trennt und die resultierenden Einzelwörter in das zuvor definierte Tupelformat mit dem Wort als String und 1 als Zahlenwert konvertiert. Da es sich um eine Datenstromabfrage handelt, wird ein Fenster spezifiziert, hier ein gleitendes Fenster mit einer Länge von fünf Sekunden und einer Schrittweite von einer Sekunde. Abschließend erfolgt eine Gruppierung nach Wörtern und die Zahlenwerte werden innerhalb der Gruppen aufsummiert. Die Printmethode sorgt für die Ergebnisausgabe auf der Konsole.

Zusätzlich zu den klassischen Programmierschnittstellen, bietet die *Flink ML* Bibliothek eine Vielzahl an Algorithmen des maschinellen Lernens. *Gelly* ermöglicht die Graphenanalyse mit Flink. Die *Table API* bietet die Möglichkeit der deklarativen Spezifikation von Abfragen ähnlich zu *SQL* und steht als Java- und Scalaversion zur Verfügung. Listing 2 zeigt eine Wordcount-Implementierung mit der Java Table API zur Stapelverarbeitung.

```
1 DataSet<WC> input = env.fromElements(new WC("Hello",1),new WC("Bye",1),new WC("Hello",1));
2 Table table=tableEnv.fromDataSet(input).groupBy("word").select("word.count as count, word");
3 tableEnv.toDataSet(table, WC.class).print();
```

Listing 2. Eine Wordcount-Implementierung unter Verwendung der Java Table API zur Stapelverarbeitung von Apache Flink.

In Zeile 1 werden Eingabewörter explizit angegeben. Zeile 2 konvertiert das *DataSet* zunächst zu einer Tabelle, die anhand des Attributs *word* gruppiert wird. Die Selectanweisung wählt wie in SQL das Wort sowie die Summe der Zähler aus. Abschließen wird die Ergebnistabelle zurück zu einem *DataSet* konvertiert und ausgegeben.

4 Datenstromverarbeitung

Die Datenstromverarbeitung unterscheidet sich signifikant von der Stapelverarbeitung: Programme haben lange (theoretisch unendliche) Laufzeiten, konsumieren Daten kontinuierlich von Eingabeströmen und produzieren im Gegenzug Ausgabeströme. Aggregationen können jedoch nur für abgeschlossene Datenblöcke berechnet werden. Sie folgen in Datenstromprogrammen daher auf eine Diskretisierung, die einen Datenstrom in abgeschlossene, potentiell überlappende Fenster unterteilt. Eine Aggregation erfolgt dann fortlaufend per Fenster.

Im Gegensatz zu vielen anderen Datenanalyseplattformen ist Flink durch seine Laufzeitumgebung nicht an Limitationen gebunden, die aus Micro-Batching Techniken [14] entstehen. Beim Micro-Batching wird ein Datenstrom als Serie von Datenblöcken fester Größe interpretiert, die separat als Stapel verarbeitet werden. Die Größen aller Fenster müssen Vielfache der Blockgröße sein, sodass ein Gesamtergebnis aus den Blockergebnissen berechnet werden kann. Flink stellt weitaus flexiblere Diskretisierungsoptionen bereit, welche eine Generalisierung von IBM SPLs Trigger und Eviction Policies [7] sind. Eine Trigger Policy gibt an, wann ein Fenster endet und die Aggregation für dieses Fenster ausgeführt wird. Die Eviction Policy gibt an, wann Tupel aus dem Fernsterpuffer entfernt werden und spezifiziert so die größe von Fenstern. Anwender können aus einer Vielzahl von vordefinierten Policies wählen (z.B. basierend auf Zeit, Zählern oder Deltafunktionen) oder benutzerdefinierte Policies implementieren. Flink erreicht damit bei geringeren Latenzen eine größere Expressivität als micro-batch-abhängige Systeme und vermeidet die Komplexität von Lambda-Architekturen.

Operatoren in Flink können statusbehaftet sein. Ein Schnappschussalgorithmus stellt sicher, dass jedes Tupel auch im Fehlerfall exakt einmal im Operatorstatus repräsentiert ist und verarbeitet wird.

Flink bietet somit eine bei Open-Source-Systemen einmalige Kombination aus Stapelverarbeitung, nativer Datenstromverarbeitung ohne Beschränkungen durch Micro-Batching, statusbehafteten Operatoren, ausdrucksstarken APIs und *exactly-once* Garantien.

5 Weiterführende Publikationen

Flink ist sowohl Produkt als auch Basis vieler Forschungsarbeiten. Im Folgenden werden die wichtigsten Publikationen genannt. Warnecke et al. stellen die Nephele Laufzeitumgebung vor [13], auf der Flinks Laufzeit ursprünglich basierte. Battré et al. ergänzen sie mit dem PACT Modell [3], einer Erweiterung von MapReduce [4]. Alexandrov et al. geben eine detaillierte Beschreibung der Stratosphere Plattform [1]. Hueske et al. befassen sich mit der Optimierung von Programmen mit benutzerdefinierten Funktionen [8]. Ewen et al. führen die native Unterstützung von Iterationen ein [6]. Aktuelle Arbeiten befassen sich mit Fehlertoleranz [5] und implizitem Parallelismus mittels eingebetteter Sprachen [2]. Spangenberg et al. vergleichen die Performance von Flink und Spark für unterschiedliche Algorithmen [11].

6 Resumé

Flink vereinfacht die parallele Analyse großer Datenmengen durch die Anwendung klassischer Datenbanktechniken wie automatischer Optimierung und deklarativen Abfragesprachen. Ausdrucksstarke, intuitive APIs ermöglichen sowohl Stapel- als auch Datenstromverarbeitung. Flink ist skalierbar und durch seine große Kompatibilität vielseitig einsetzbar. Operatoren werden nebenläufig, frei von Limitierungen durch Micro-Batching-Techniken, in einer Pipeline ausgeführt.

7 Danksagung

Für die Entwicklung der Plattform gilt unser besonderer Dank der gesamten Flink Community. Dieses Forschungsprojekt wird unterstützt durch Mittel des BMBF für das Berlin Big Data Center (BBDC) unter der Förderungsnummer 01IS14013 sowie der DFG Forschergruppe Stratosphere (FOR 1306).

Referenzen

1. Alexandrov, A., Bergmann, R., Ewen, S., Freytag, J. C., Hueske, F., Heise, A., ... & Warneke, D. (2014). The Stratosphere platform for big data analytics. The VLDB Journal—The International Journal on Very Large Data Bases, 23(6), 939-964.
2. Alexandrov, A., Kunft, A., Katsifodimos, A., Schüler, F., Thamsen, L., Kao, O., ... & Markl, V. (2015, May). Implicit Parallelism through Deep Language Embedding. In Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data (pp. 47-61). ACM.
3. Battré, D., Ewen, S., Hueske, F., Kao, O., Markl, V., & Warneke, D. (2010, June). Nephele/PACTs: a programming model and execution framework for web-scale analytical processing. In Proceedings of the 1st ACM symposium on Cloud computing (pp. 119-130). ACM.
4. Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. Communications of the ACM, 51(1), 107-113.
5. Dudoladov, S., Xu, C., Schelter, S., Katsifodimos, A., Ewen, S., Tzoumas, K., & Markl, V. (2015, May). Optimistic Recovery for Iterative Dataflows in Action. In Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data (pp. 1439-1443). ACM.
6. Ewen, S., Tzoumas, K., Kaufmann, M., & Markl, V. (2012). Spinning fast iterative data flows. Proceedings of the VLDB Endowment, 5(11), 1268-1279.
7. Gedik, B. (2014). Generic windowing support for extensible stream processing systems. Software: Practice and Experience, 44(9), 1105-1128.
8. Hueske, F., Peters, M., Sax, M. J., Rheinländer, A., Bergmann, R., Krettek, A., & Tzoumas, K. (2012). Opening the black boxes in data flow optimization. Proceedings of the VLDB Endowment, 5(11), 1256-1267.
9. Kreps, J., Narkhede, N., & Rao, J. (2011, June). Kafka: A distributed messaging system for log processing. In Proceedings of the NetDB (pp. 1-7).
10. Shvachko, K., Kuang, H., Radia, S., & Chansler, R. (2010, May). The hadoop distributed file system. In Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on (pp. 1-10). IEEE.
11. Spangenberg, N., Roth, M., & Franczyk, B. (2015, June). Evaluating New Approaches of Big Data Analytics Frameworks. In Business Information Systems (pp. 28-37). Springer International Publishing.
12. Vavilapalli, V. K., Murthy, A. C., Douglas, C., Agarwal, S., Konar, M., Evans, R., ... & Baldeschwieler, E. (2013, October). Apache hadoop yarn: Yet another resource negotiator. In Proceedings of the 4th annual Symposium on Cloud Computing (p. 5). ACM.
13. Warneke, D., & Kao, O. (2009, November). Nephele: efficient parallel data processing in the cloud. In Proceedings of the 2nd workshop on many-task computing on grids and supercomputers (p. 8). ACM.
14. Zaharia, M., Das, T., Li, H., Shenker, S., & Stoica, I. (2012, June). Discretized streams: an efficient and fault-tolerant model for stream processing on large clusters. In Proceedings of the 4th USENIX conference on Hot Topics in Cloud Ccomputing (pp. 10-10). USENIX Association.

The GOBIA Method: Towards Goal-Oriented Business Intelligence Architectures

David Fekete[1] and Gottfried Vossen[1,2]

[1] ERCIS, Leonardo-Campus 3, 48149 Münster, Germany,
firstname.lastname@ercis.de
[2] University of Waikato Management School, Private Bag 3105, Hamilton 3240,
New Zealand, vossen@waikato.ac.nz

Abstract. Traditional Data Warehouse (DWH) architectures are challenged by numerous novel Big Data products. These tools are typically presented as alternatives or extensions for one or more of the layers of a typical DWH reference architecture. Still, there is no established joint reference architecture for both DWH and Big Data that is inherently aligned with business goals as implied by Business Intelligence (BI) projects. In this paper, a work-in-progress approach towards such custom BI architectures, the GOBIA method, is presented to address this gap, combining a BI reference architecture and a development process.

1 Introduction

Big Data has generated widespread interest among both academia and practitioners [9]. Several new products (such as Apache Hadoop) and approaches have emerged that allow to store or process Big Data, which was not feasible or efficient before. Big Data is larger, more diverse, and speedier than it was with data in established traditional technologies. Big Data challenges often exceed an organization's capability to process and analyze data in a timely manner for decision making [9]. On the other hand, traditional Data Warehouse (DWH) architectures are an established concept for Business Intelligence based on a common reference architecture (e.g., [8]). Nevertheless, with the plethora of novel Big Data products, the question arises which impact these have on analytic architectures and which form a reference architecture for both Big Data and DWH could have. Especially Apache Hadoop distributions such as MapR[1] offer so many products that building a customized architecture is rendered an increasingly complex task. Thus, additional clarity on the process of deriving a customized architecture from a reference architecture is required as well. The goal of this work is to design artifacts that address these questions following a

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

[1] https://www.mapr.com/products/mapr-distribution-including-apache-hadoop

design science approach [6]. To this end, a theoretical background on the foundations of the solution artifacts is given in Sec. 2. The various solution artifacts are described in Sec. 3. Finally, the work is summarized and next research steps are outlined in Sec. 4.

2 Fundamentals

The following fundamentals explain the basic concepts regarding architectures and Business Intelligence required for the proposed solution and outline the problem statement to be addressed. Definitions of and further reading on the basic terms Data Warehouse and Big Data can be found in [3,4,8] and in [10,9], resp.

The term *Business Intelligence* (BI) is used to describe a holistic enterprise-wide approach for decision support that integrates analytics systems (e.g., a DWH), but also strategy, processes, applications, and technologies in addition to data [2, p. 13]. Besides that, BI is also said to postulate the generation of knowledge about business status and further opportunities [2, p. 13]. More importantly, a crucial aspect of BI is its alignment to its business area of application [2, p. 14]. This implies that BI and also its parts (including an analytics system) should be aligned to the respective business in order to support decision making for business operations.

While a traditional DWH architecture has well-defined layers such as the staging area (Extract-Transform-Load, ETL) or data marts [8,4], several examples for Big Data attached to DWH architectures (e.g., with Big Data tools used for ETL) can be found. Typically, these represent specific setups (e.g., [10, p. 23], [5, p. 18]), but cannot be generalized into a reference architecture. Other attempts include more general (reference) architectures (e.g., [7, p. 62], [5, p. 12], [9]), yet the question remains of how to allocate (which) products to specific roles in an architecture, especially with several alternatives to traditional DWH architectures and products available. This is exacerbated by the fact that some of these new product offerings can be used for multiple purposes inside such an architecture. For example, MapReduce as a generic tool can be used for data preprocessing (e.g., performing large-scale cleansing operations) as well as for an actual analysis (e.g., basic word count statistics or sentiment analyses).

However, no BI reference architecture has been established yet that is inherently technology-independent, i.e., usable for both DWH and Big Data, and addresses the business-alignment of BI. Such *goal-orientation* aids the selection of customized architectures, since specific goals can be considered in the process.

As several combinations of technologies and products can be placed in an analytics architecture nowadays, the potential complexity of architectures is increased. For instance, certain Apache Hadoop distributions (e.g., by MapR[2] or Hortonworks[3]) present all of their offered product options in a single package,

[2] `http://doc.mapr.com/display/MapR/Architecture+Guide`

[3] `http://hortonworks.com/building-an-enterprise-data-architecture/`

where no process to a customized architecture is outlined and the necessary architectural choices are left to the implementer. For instance, if a weather prediction BI application should be implemented using these Hadoop distributions the fitting products have to be chosen. While these choices could possibly be made with certain effort, e.g., Apache Storm for streaming weather data processing and MapReduce for batch analytics, the process of arriving at these decisions cannot be supported best solely by considering a (simple) classical layered view as with the DWH reference architecture before. Previously, this view was sufficient as typical products were located mainly in the DWH sphere, but to match todays complexity and heterogeneity from an architectural point of view, the classical layered view needs to be further refined.

Reference architectures used in computer applications typically exhibit a layering of services [1]. The various layers interact through well-defined interfaces, and their structure commonly follows an abstraction process. Indeed, the top layer comprises the most coarse (high-level) services, which are refined at the next lower layer, and this is often repeated until a layer of most basic functions is reached. In other words, in a system representing a service hierarchy, higher-level services are realized by lower-level services.

An example for a service hierarchy is a high-level telecommunication service provided to an end-user that can be comprised of several lower-level services in the back-end. In a data analytics scenario, high-level analytical services could be placed in a core analytics layer (e.g., "Cluster customer groups" or "Sentiment analysis of product-related tweets") and be consumed by BI applications on top, possibly supplied to by a middle-ware (e.g., data marts). These services are provided for by data preprocessing services (such as "Cleanse customer data" or "Filter tweets") at a lower layer and are ultimately based on several data sources (e.g., "Twitter" or "ERP"). Each of these services can be allocated, respectively be backed, by a novel or traditional product. However, the mentioned challenge of actually allocating these heterogeneous products to layers or services in a specific scenario remains and needs to be addressed (cf. Fig 1). We do so using a service hierarchy within a layered architecture that serves as a guide towards a final implementation of a customized architecture, since it allows for a clear structuring of complex architectures in a modern heterogeneous product landscape.

Abeck et al. see a layered architecture as a foundation for (software) reference architectures, as software systems development would be based on layering [1]. Employing a layered architecture for a BI reference architecture could use these properties during customization and place adequate BI-related services at the appropriate architectural layer, which adhere to the intended level of abstraction. When BI is seen in this way, a general reference architecture can individually be customized and hence aligned to the goals and requirements of a specific business scenario or application. Goal orientation and layered architecture should hence be part of the solution artifacts to be designed, which will be elaborated upon in the following.

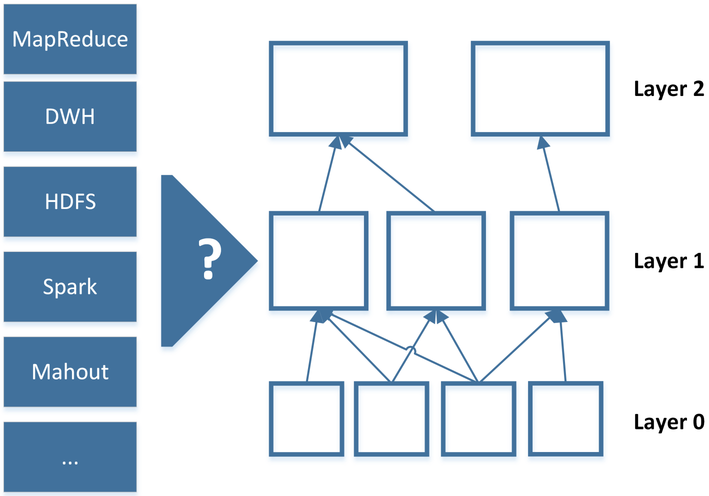


Fig. 1. A layered architecture with a service hierarchy (right) with the to be addressed gap of allocating heterogeneous products and technologies to it.

3 Goal-Oriented Business Intelligence Architectures (GOBIA)

The proposed approach is termed the "Goal-oriented Business Intelligence Architectures" (GOBIA) method and consists of a BI reference architecture (GOBIA.REF) and development process (GOBIA.DEV). In the following, both artifacts are briefly presented.

GOBIA.REF aims to address the architectural gap outlined above and is intended as a layered, technology-independent BI reference architecture. It is accompanied by a development process (GOBIA.DEV) that aids in its customization, so that the outcome is aligned to the goals and requirements of a specific scenario or application. This inherently supports the principle of BI to be business-aligned. The resulting architecture is a high-level conceptual model resembling a service hierarchy, which is not yet focused on technical details, but aims to alleviate the challenge of implementing the architecture (i.e., assigning specific products to the defined roles and functions).

3.1 BI reference architecture

The proposed BI reference architecture (see Fig. 2 on the right) as a layered architecture generalizes DWH and Big Data in the analytics layer as "BI functionality" as common denominator. The customized architecture is built based on this reference architecture and should be seen as a service hierarchy.

Data sources of the architecture reside at the bottom of the reference architecture. These can be located internally or externally (e.g., in a cloud). While this is comparable to other architectures, no restrictions are imposed on data formats or delivery and persistence modes. For instance, data source blocks could simply be "Mapping data" or "Transportation routes". The "Data Preparation and Preprocessing" above it fulfills a similar purpose as the staging area in a DWH, but the tasks should be more coarse-grained and mostly omit technical details. For instance, a task in this layer could be to "Transform mapping data" or to "Complete disease data". Instead of having a DWH and/or Big Data tools in the core analytics layer, this layer contains BI functionality in general, which is technology-independent and focused on the results of BI. For instance, BI functionality could include high-level functionalities such as "Classify customer into types" or "Identify sales patterns". Data marts, as in a DWH, can fulfill the role to provide subsets of data to the BI-specific applications. As the layers are conceptual, a decision whether to materialize any of these subsets is not made at this point.

BI-specific applications consume the BI functionalities delivered through the data mart layer to deliver applications to a client or end-user, much as in many other architectures. The difference, however, is that GOBIA.REF aims to clarify on the actually needed BI functionality so that the choice of selecting suitable technological artifacts afterwards becomes less complex.

3.2 Development process

The proposed development process of the customized architecture, GOBIA.REF (see Fig. 2), is designed so that actual goals and requirements on a target BI system are derived from a more coarse-grained strategy, which is assumed to be already defined. The latter, indicated as (0) in Fig. 2, allows to derive application domain(s) and scenario(s) (use cases) from it in step (1). The underlying domain should define the playing field laid out by the strategy (e.g., finance, health care...). The scenarios, set in the domains, define the requirements and goals of the customized architecture (2), and business-relevant information such as costs, expected value, or revenue. A defined goal could, for example, be to "Analyze customer behavior to map his characteristics to products that the he might find interesting". At this point, the BI-specific applications required at the top-most layer are determined.

This is followed by a co-alignment step (3). The main outcomes are BI functionalities to be placed in the architecture, as well as necessary data preparation or preprocessing tasks, and data properties of suitable data. For this, requirements and goals are aligned together with BI functionality and data properties. The result should be that, eventually, suitable BI functionalities realize the goals and adhere to the requirements set before and that these BI functionalities and data preparation tasks fit the data properties. If, e.g., a goal was to differentiate groups of customers, BI functionality for a suitable clustering method must be defined.

Data properties are characterized by using the "V's" [10,9], which are typically used in Big Data context, but should be applied to any data in this method. For instance, if the quality of a data source is poor (e.g., low validity or high vagueness), but the set goal is to work on higher quality data, proper data cleansing or enrichment tasks have to be conducted.

Notably, co-alignment can also mean that requirements and goals are adjusted as well in the course of an iterative definition process. For example, if data properties for an initial set of requirements and goals are characterized and the data is of higher quality than expected , goals could be refined to explicitly exploit this data. This refinement, then, could lead to a further adjustment of BI functionality or data preprocessing tasks.

Input to the requirements and goals in step (2) is a BI reference architecture (GOBIA.REF). Also, there can be a direct strategic impact on it, e.g., a decision not to have any data marts in the final architecture. Moreover, domain-specific template reference architectures could be possible like, e.g., a set of typical finance-algorithms as BI functionality templates.

Finally, in step (4), the customized architecture is assembled by assigning the outcomes of the co-alignment (e.g., BI functionalities) to the respective layers and by building a service hierarchy. This high-level conceptual output can be used further in the implementation of the target BI system.

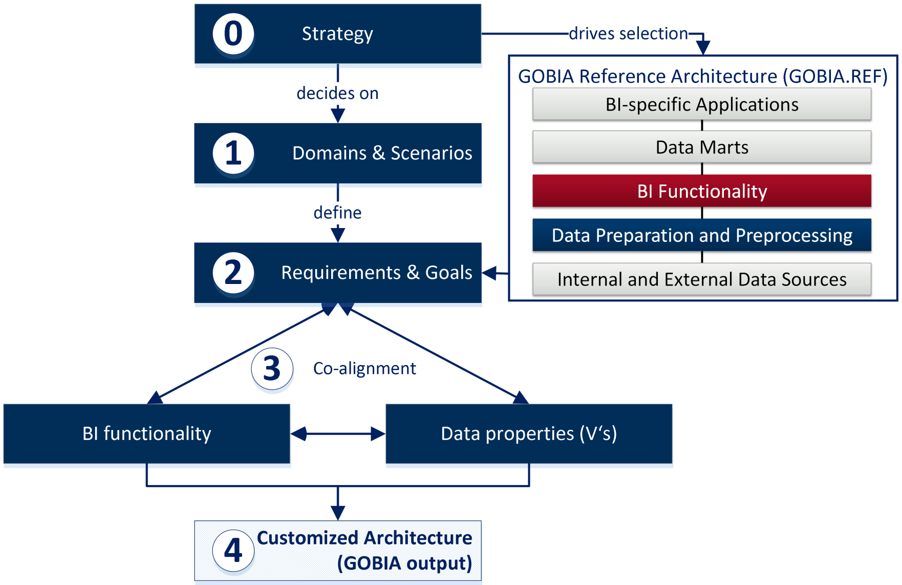


Fig. 2. Customized BI architecture development process proposal for the GOBIA method (GOBIA.DEV) and the BI reference architecture proposal (GOBIA.REF) on the right.

3.3 Sample case

For illustration purposes, a sample use case is briefly discussed and its outcome presented (cf. Fig. 3). This fictitious case is tailored towards a global organization concerned with the health of people, e.g. the World Health Organization (WHO). Firstly, the GOBIA.DEV process is executed to determine the goals and requirements and to present the functionality at the different architectural layers. Then, GOBIA.REF is used to assemble this into a layered and hierarchical form that can be used as a blueprint for a subsequent implementation.

The set strategy (0) could be the objective of the WHO, which "is the attainment by all people of the highest possible level of health" [4]. Naturally, healthcare is set as application domain (1). In this scenario, global disease management should be the specific application that allows for disease monitoring and pattern recognition to facilitate appropriate mitigation or containment procedures and that supports the set objective of the WHO.

The goal (2) of the BI system should be recognizing disease patterns around the globe to allow description and comparison of current disease spread patterns to eventually allow for enhanced monitoring. To achieve this, several data sources are required. These could be confirmed disease cases ("disease occurrence data"), incoming reports of potential, unconfirmed diseases ("emergent disease information") and knowledge about diseases like symptoms ("disease data"). To create a map and to predict endangered areas in the future location data — ranging from basic maps to more refined data as health infrastructure and environmental data, which could influence disease spreadings and potential dangers — is needed as well as, e.g., public and private transportation routes such as flight routes or roads. Furthermore, appropriate algorithms are required to map health dangers of certain levels to appropriate mitigation or containments procedures.

The results of the subsequent co-alignment lead to a service hierarchy or layered architecture as shown in Fig. 3. To conduct co-alignment, the properties of the required data sources should be assessed. For instance, because of the extensive mapping data, potential data volumes could be regarded as "high". Besides technical properties (volume, variety, velocity), qualitative properties as "value" can be assessed. In this case, value could be rather low for unconfirmed, emergent disease reports due to the uncertainty and poor initial data quality and be potentially high for actually confirmed disease cases from, e.g., hospitals. With this, the required BI-specific functionality can be formulated that supports the goals of the application as well as the necessary tasks that, e.g., deal with the data properties (such as the cleansing of unconfirmed disease cases). These tasks could include a classification of emergent disease information (i.e., if its actually a confirmed case or an irrelevant report). By clustering the cleansed and completed diseases occurrences, these can be integrated into the preprocessed mapping data. Lastly, disease patterns could be recognized and eventually plotted on an interactive map or assembled into regional health reports.

[4] `http://apps.who.int/gb/DGNP/pdf_files/constitution-en.pdf`

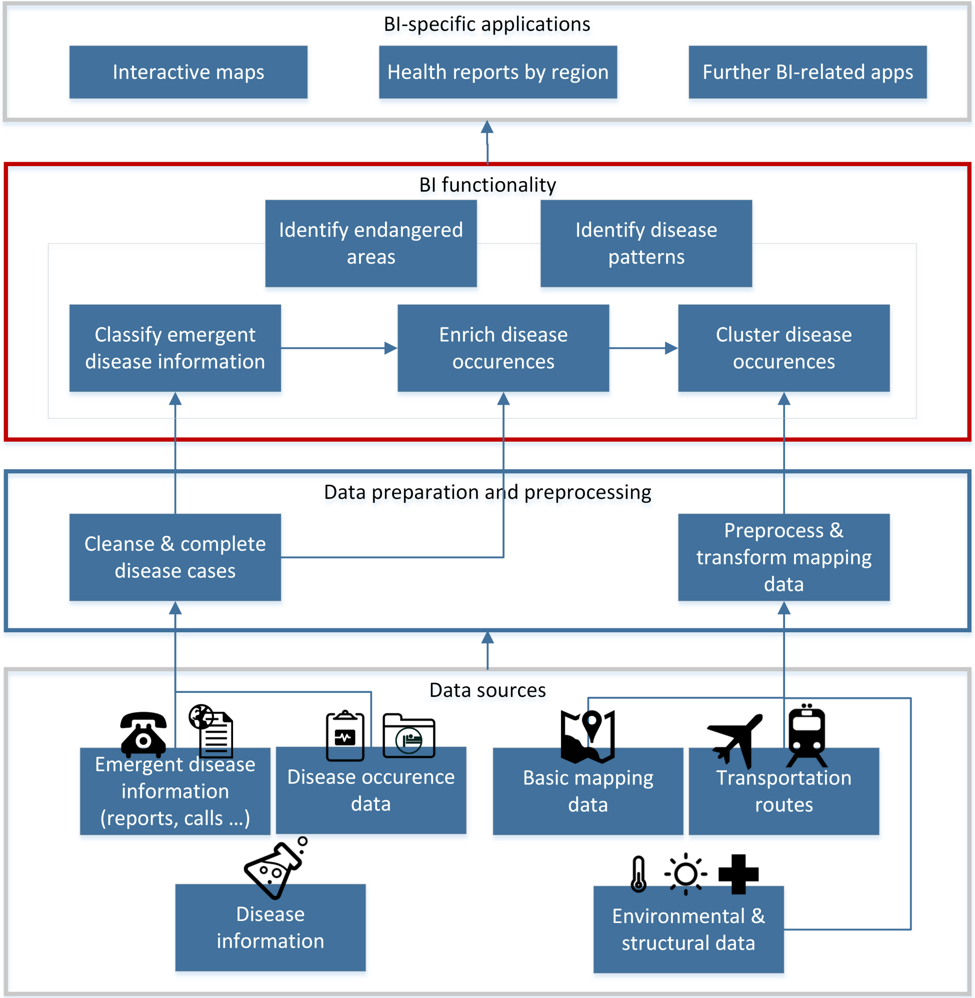

BI-specific applications
Interactive maps
Health reports by region
Further BI-related apps
BI functionality
Identify endangered areas
Identify disease patterns
Classify emergent disease information
Enrich disease occurences
Cluster disease occurences
Data preparation and preprocessing
Cleanse & complete disease cases
Preprocess & transform mapping data
Data sources
Emergent disease information (reports, calls ...)
Disease occurence data
Basic mapping data
Transportation routes
Disease information
Environmental & structural data

Fig. 3. GOBIA output example: Customized layered, hierarchical architecture for a fictitious public health use case.

4 Summary and Future Work

This work has tried to outline gaps in "universal" reference architectures that did arise as a result of moving into the age of Big Data. A proposal for a BI reference architecture based on a basic concept in Computer Science was made. A development process has been proposed to support a goal-oriented creation of a customized BI architecture, yielding a possible prerequisite for choosing suitable analytic tools.

Future work should address various parts of the proposed method. Firstly, the proposal is to be refined. For example, the semantics in the development process needs to be elaborated upon, and inputs and outputs be specified in more detail.

Secondly, the steps following an execution of the development process are to be elaborated, since the high-level conceptual model output cannot be directly operationalized. The challenge to select technological artifacts (e.g., from a Hadoop distribution) and connect these to realize the concept is not resolved yet. Here, best practices or generalizations of architecture setups could be derived in order to address this challenge. It should also be elaborated how findings from these can be generalized to templates to enhance the method itself. For instance, best practices could be used to derive domain-specific reference architectures or predefined building blocks for the co-alignment step in GOBIA.DEV (e.g., common data processing tasks that address certain data properties or specific BI functionalities).

Thirdly, both reference architecture and development process should be evaluated empirically to test if they fit their intended usage. Such an evaluation should build, for example, a customized architecture based on a Hadoop framework (e.g., MapR) to verify whether the process is indeed less complex when using the GOBIA method. Also, such evaluation should include a comparison to other existing approaches (e.g., for reference architectures) to better assess to which extent GOBIA.REF and GOBIA.DEV can utilize the proposed advantages in practice.

References

1. Abeck, S., Lockemann, P.C., Schiller, J., Seitz, J.: Verteilte Informationssysteme: Integration von Datenübertragungstechnik und Datenbanktechnik. dpunkt, Heidelberg (2003)
2. Bauer, A., Günzel, H.: Data Warehouse Systeme. dpunkt, Heidelberg, 3rd edn. (2009)
3. Inmon, W.: Building the Data Warehouse. John Wiley & Sons Inc., New York, New York, USA, 2nd edn. (1996)
4. Lehner, W.: Datenbanktechnologie für Data-Warehouse-Systeme. d.punkt Verlag, Heidelberg (2003)
5. Oracle: Oracle Information Architecture: An Architect's Guide to Big Data (2012), `http://www.oracle.com/technetwork/topics/entarch/articles/oea-big-data-guide-1522052.pdf`

6. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems 24(3), 45–77 (Dec 2007), `http://www.tandfonline.com/doi/full/10.2753/MIS0742-1222240302`
7. Thiele, M., Lehner, W., Habich, D.: Data-Warehousing 3.0 Die Rolle von Data-Warehouse- Systemen auf Basis von In-Memory-Technologie. In: Innovative Unternehmensanwendungen mit In-Memory Data Management (IMDM). pp. 57–68. Wolfgang Lehner, Gunther Piller, Mainz (2011)
8. Vossen, G.: Datenmodelle, Datenbanksprachen und Datenbankmanagementsysteme. Oldenbourg, München, 5th edn. (2008)
9. Vossen, G.: Big data as the new enabler in business and other intelligence. Vietnam Journal of Computer Science 1(1), 3–14 (Feb 2014)
10. Zikopoulos, P., Eaton, C., DeRoos, D., Deutsch, T., Lapis, G.: Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data. McGraw-Hill, New York, USA, 1st edn. (2012)

Data Quality Adjustments for Pricing on Data Marketplaces

Florian Stahl[1] and Gottfried Vossen[1,2]

[1] ERCIS, Leonardo-Campus 3, 48149 Münster, Germany,
{Stahl,Vossen}@ercis.de
[2] University of Waikato Management School,Private Bag 3105,
Hamilton, 3240, New Zealand

Abstract. Currently, information has become an increasingly important production factor which has led to the emergence of data marketplaces that leverage big data technologies. However, value attribution of data is still difficult. This work suggests to discuss what role data quality can play in this context, particularly: what quality measures are relevant in the context of big data, how they can be measured, and how the quality of a data product can be efficiently modified to create different versions[3] of a data product.

1 Introduction

Information has become an important production factor [6]. This has led to a point at which data – as the basic unit in which information is exchanged – is increasingly being traded on data marketplaces, extensively described in [4, 9] and put on the database research agenda by Balazinska et al. [2, 1]. Basically, data marketplaces are platforms levering big data technologies that allow providers and consumers of data and data-related services, such as data mining algorithms, to interact with each other. One prominent German example of such a data marketplace is MIA[4] which employs large computer clusters to crawl substantial parts of the German Web and to provide an analysis infrastructure for the gathered data. This is particularly beneficial for small and medium-sized enterprises as they would otherwise not be able to access and analyse such data. This paper suggests to discuss what role data quality modifications can play in the context of data marketplaces and big data applications building on them.

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

[3] In this work, the term *version* will be used in its economic sense, i. e., to refer to different variants of a data product; this is not to be confused with versions as known from temporal databases.

[4] http://mia-marktplatz.de/

2 Pricing on Data Marketplaces

Given that the value for data and data-related services is subjective to its consumers [7], it is not surprising that little sense for its value exists in the database community [2, 1]; this is mainly owing to the fact that data is an information good with peculiarities, such as resemblance to public goods [12].

One approach to reduce the uncertainty for data providers is to apply reverse pricing mechanisms that allow customers to suggest prices, which – if well-designed – allow for a revelation of the customer's true willingness to pay. Reverse pricing has the advantage that customers participate in the pricing process, which is generally seen as positive, even if used for price discrimination – i. e., asking different prices of different customers [3].

Name Your Own Price is such a pricing mechanism which is often employed in auctions, for instance, EBAY'S *make offer* option. In contrast to established physical goods, digital goods, such as data, can be sold multiple times because of the low cost of reproduction. Thus, in order to avoid fierce price wars, it is recommendable to adapt a data product to a customer's preferences, which can also be seen as a further benefit for customers. Although not discussing the economic intuition behind it, TANG ET AL. suggested to use a Name Your Own Price mechanism in the context of data marketplaces. In [10] they suggested to adapt the completeness of XML data and in [11] they focused on the accuracy of relational data based on a customer's bid. Concretely, the provider advertises a price and customers may suggest a price they are willing to pay. If the bid is lower than the ask price, completeness or accuracy of the data product will be lowered to match the offered price. Furthermore, the argument can be made that the threshold can also be hidden from the buyer. In this case the profit increases if the suggested price is higher than the requested price.

3 Discussion

The previously mentioned works focus on only one quality dimension. Therefore, the question remains how this can be adapted to multiple quality criteria, which has been extensively discussed in [8]. As a starting point we suggest to model the distribution of discounts to different quality criteria as a multiple-choice knapsack problem. Given 1) a set of quality criteria, 2) a function that creates versions for all quality criteria, 3) a function that attributes the ask price to these versions, and 4) customer as well as vendor preferences for certain quality criteria, an optimal combination can be calculated even on commodity hardware for a limited number of quality criteria such as those identified by NAUMANN [5].

Having made these calculations, the quality of data products has to be adjusted to match a customer's suggested price. However, at the moment it is not quite clear what data quality dimensions are relevant in the context of big data analysis applications. Thus, a number of questions arise. Consequently, this work suggests to discuss the following questions:

- What quality dimensions are relevant for big data applications?

– How can they be practically applied to large data sets and how can they be measured efficiently?
– And most importantly: how can big data architectures be utilised to adapt the quality of big data products efficiently in order to meet a customer's requirements?

References

[1] M. Balazinska, B. Howe, P. Koutris, D. Suciu, and P. Upadhyaya. "A Discussion on Pricing Relational Data". In: *In Search of Elegance in the Theory and Practice of Computation*. Ed. by V. Tannen, L. Wong, L. Libkin, W. Fan, W.-C. Tan, and M. Fourman. Vol. 8000. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013, pp. 167–173.

[2] M. Balazinska, B. Howe, and D. Suciu. "Data Markets in the Cloud: An Opportunity for the Database Community". In: *PVLDB* 4.12 (2011), pp. 1482–1485.

[3] O. Hinz, I.-H. Hann, and M. Spann. "Price discrimination in e-commerce? An examination of dynamic pricing in name-your-own price markets". In: *MIS quarterly* 35.1 (2011), pp. 81–98.

[4] A. Muschalle, F. Stahl, A. Löser, and G. Vossen. "Pricing Approaches for Data Markets". In: *Proceedings of the Workshop Business Intelligence for the Real Time Enterprise*. Istanbul, Turkey, 2012.

[5] F. Naumann. *Quality-Driven Query Answering for Integrated Information Systems*. Vol. 2261. Lecture Notes in Computer Science. Springer, 2002.

[6] K. North. *Wissensorientierte Unternehmensführung*. 5th edition. Gabler, 2011.

[7] C. Shapiro and H. Varian. *Information Rules: A Strategic Guide to the Network Economy*. Strategy/Technology / Harvard Business School Press. Harvard Business School Press, 1999.

[8] F. Stahl. "High-Quality Web Information Provisioning and Quality-Based Data Pricing". PhD thesis. University of Münster, 2015.

[9] F. Stahl, A. Löser, and G. Vossen. "Preismodelle für Datenmarktplätze". In: *Informatik-Spektrum* 37.1 (2014).

[10] R. Tang, A. Amarilli, P. Senellart, and S. Bressan. "Get a Sample for a Discount". In: *Database and Expert Systems Applications*. Ed. by H. Decker, L. Lhotská, S. Link, M. Spies, and R. R. Wagner. Vol. 8644. Lecture Notes in Computer Science. Springer International Publishing, 2014, pp. 20–34.

[11] R. Tang, H. Wu, Z. Bao, S. Bressan, and P. Valduriez. "The Price Is Right". In: *Database and Expert Systems Applications*. Ed. by H. Decker, L. Lhotská, S. Link, J. Basl, and A. Tjoa. Vol. 8056. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013, pp. 380–394.

[12] L. Vomfell, F. Stahl, F. Schomm, and G. Vossen. *A Classification Frameworik for Data Marketplaces*. Tech. rep. 23. Münster: ERCIS, 2015.

Polyglot database architectures
= polyglot challenges

Lena Wiese

Institute of Computer Science
University of Göttingen
Goldschmidtstraße 7
37077 Göttingen, Germany
lena.wiese@uni-goettingen.de

Abstract. We categorize polyglot database architectures into three types (polyglot persistence, lambda architecture and multi-model databases) and discuss their advantages and disadvantages.

1 Polyglot Database Architectures

When designing the data management layer for an application, several database requirements may be contradictory. For example, regarding access patterns some data might be accessed by write-heavy workloads while others are accessed by read-heavy workloads. Regarding the data model, some data might be of a different structure than other data; for example, in an application processing both social network data and order or billing data, the former might usually be graph-structured while the latter might be semi-structured data. Regarding the access method, a web application might want to access data via a REST interface while another application might prefer data access with query language. It is hence worthwhile to consider a database and storage architecture that includes all these requirements. We describe three option for polyglot database architectures in the following three sections.

1.1 Polyglot Persistence

Instead of choosing just one single database management system to store the entire data, so-called **polyglot persistence** could be a viable option to satisfy all requirements towards a modern data management infrastructure. Polyglot persistence (a term coined in [4]) denotes that one can choose as many databases as needed so that all requirements are satisfied. Polyglot persistence can in particular be an optimal solution when backward-compatibility with a **legacy application** must be ensured. The new database system can run alongside the legacy

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

database system; while the legacy application still remains fully functional, novel requirements can be taken into account by using the new database system.

An implementation of a data processing system that connects to several data sources and integrates and merges data from these sources is Apache Drill [2]. Apache Drill is inspired by the ideas developed in Google's Dremel system [6].

It should obviously be avoided to push the burden of all of these query handling and database synchronization task to the application level – that is, in the end to the programmers that maintain the data processing applications. Instead it is usually better to introduce an integration layer. The integration layer then takes care of processing the queries – decomposing queries in to several subqueries, redirecting queries to the appropriate databases and recombining the results obtained from the accessed databases; ideally, the integration layer should offer several access methods, and should be able to parse all the different query languages of the underlying database systems as well as potentially translate queries into other query languages. Moreover, the integration layer should ensure **cross-database consistency**: it must synchronize data in the different databases by propagating additions, modifications or deletions among them.

Polyglot persistence however comes with severe disadvantages:

- Uniform access: There is no unique query interface or query language, and hence access to the database systems is not unified and requires knowledge of all needed database access methods.
- Consistency: Cross-database consistency is a major challenge because referential integrity must be ensured across databases (for example if a record in one database references a record in another database) and in case data are duplicated (and hence occur in different representation in several databases at the same time) the duplicates have to be updated or deleted in unison.
- Interoperability: The underlying database systems are developed independently. Newer versions of databases may not be interoperable with the integration layer and the administrator has to keep track of frequent updates.
- Logical Redundancy: Logical redundancy can only be avoided with a database design that strictly assigns non-intersecting subsets of the data to different databases. This might contradict some access requirements of users.
- Security: Access control must be enforced by the integration layer and all connected databases have to be configured to only allow restricted access.

1.2 Lambda Architecture

When real-time (stream) data processing is a requirement, a combination of a slower batch processing layer and a speedier stream processing layer might be appropriate. This architecture has been recently termed **lambda architecture** [5]. The lambda architecture processes a continuous flow of data in the following three layers:

Speed Layer: The speed layer collects only the most recent data. As soon as data have been included in the other two layers (batch layer and serving layer), the data can be discarded from the speed layer dataset. The speed layer incrementally computes some results over its dataset and delivers these results in several **real-time views**; that is, the speed layer is able to adapt his output based on the constantly changing data set. Due to the relatively small size of the speed layer data set, the runtime penalty of incremental computations are still within acceptable limits.

Batch Layer: The batch layer stores all data in an append-only and immutable fashion in a so-called master dataset. It evaluates functions over the entire dataset; the results are delivered in so-called **batch views**. Computing the batch views is an inherently slow process. Hence, recent data will only be gradually reflected in the results.

Serving Layer: The serving layer makes batch views accessible to user queries. This can for example be achieved by maintaining indexes over the batch views.

User queries can be answered by merging data from both the appropriate batch views and the appropriate real-time views.

An open source implementation following the ideas of a lambda architecture is Apache Druid [3] that processes streaming data in real-time nodes and batch data in historical nodes.

In practice, the lambda architecture often relies on external storage ("deep storage" in case of Druid) or stream processors (on the input side). Due to this it only has slight advantages over the polyglot persistence approach. Moreover this architecture is mostly geared towards real-time processing of data and less to ad-hoc querying.

1.3 Multi-Model Databases

Relying on different storage backends increases the overall complexity of the system and raises concerns like inter-database consistency, inter-database transactions and interoperability as well as version compatibility and security. It might hence be advantageous to use a database system that stores data in a single store but provides access to the data with different APIs according to different data models. Databases offering this feature have been termed **multi-model** databases. Multi-model databases either support different data models directly inside the database engine or they offer layers for additional data models on top of a single-model engine.

Two open source multi-model databases are OrientDB [7] and ArangoDB [1]. OrientDB offers a document API, an object API, and a graph API; it offers extensions of the SQL standard to interact will all three APIs. Alternatively, Java APIs are available. The Java Graph API is compliant with Tinkerpop [8]. ArangoDB is a multi-model database with a graph API, a key-value API and a document API. Its query language AQL (ArangoDB query language) resembles SQL in parts but adds several database-specific extensions to it.

Several advantages come along with this single-database multi-model approach:

- Reduced database administration: maintaining a single database installation is easier than maintaining several different database installations in parallel, keeping up with their newest versions and ensure inter-database compatibility. Configuration and fine-tuning database settings can be geared towards a single database system.
- Reduced user administration: In a multi-model database only one level of user management (including authentication and authorization) is necessary.
- Integrated low-level components: Low-level database components (like memory buffer management) can be shared between the different data models in a multi-model database. In contrast, polyglot persistence with several database systems requires each database engine to have its own low-level components.
- Improved consistency: With a single database engine, consistency (including synchronization and conflict resolution in a distributed system) is a lot easier to ensure than consistency across several different database platforms.
- Reliability and fault tolerance: Backup just has to be set up for a single database and upon recovery only a single database has to be brought up to date. Intra-database fault handling (like hinted handoff) is less complex than implementing fault handling across different databases.
- Scalability: Data partitioning (in particular "auto-sharding") as well as profiting from data locality can best be configured in a single database system – as opposed to more complex partitioning design when data are stored in different distributed database systems.
- Easier application development: Programming efforts regarding database administration, data models and query languages can focus on a single database system. Connections (and optimizations like connection pooling) have to be managed only for a single database installation.

2 Conclusion

Data come in different formats and data models. Modern data stores support advanced data management in the native data models [9]. Polyglot database architectures can handle several different data models at a time.

Polyglot persistence can respond to differing user demands; however it comes at the cost of increased administration overhead and more complex configuration (in particular in terms of security). Hence, polyglot persistence can only be recommended if several diverse data models have to be supported and the maintenance overhead can be managed.

The lambda architecture is a good choice for real-time data analytics but also relies on external data storage with similar disadvantages as polyglot persistence.

Multi-model databases are a good choice if only a limited set of data models is required by the accessing applications. Multi-model excel in terms of administration effort and security and hence are optimal, when only the limited set of data formats supported by the multi-model database are needed.

References

1. ArangoDB: Https://www.arangodb.com/
2. Drill: Http://drill.apache.org/
3. Druid: Http://druid.io/
4. Fowler, M.J., Sadalage, P.J.: NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence. Prentice Hall (2012)
5. Marz, N., Warren, J.: Big Data: Principles and best practices of scalable realtime data systems. Manning Publications Co. (2015)
6. Melnik, S., Gubarev, A., Long, J.J., Romer, G., Shivakumar, S., Tolton, M., Vassilakis, T.: Dremel: interactive analysis of web-scale datasets. Proceedings of the VLDB Endowment 3(1-2), 330–339 (2010)
7. OrientDB: Http://orientdb.com/
8. Tinkerpop: Http://tinkerpop.incubator.apache.org/
9. Wiese, L.: Advanced Data Management – for SQL, NoSQL, Cloud and Distributed Databases. DeGruyter/Oldenbourg (2015)

Visualisierung von NoSQL-Transformationen unter der Verwendung von Sampling-Techniken

Stefan Braun, Johannes Schildgen, Stefan Deßloch

{s_braun10, schildgen, dessloch}@cs.uni-kl.de
AG Heterogene Informationssysteme, Fachbereich Informatik, TU Kaiserslautern

Zusammenfassung. Analysen auf NoSQL-Datenbanken sind oft langdauernd und die Ergebnisse für den Benutzer häufig schwer verständlich. Wir präsentieren eine Möglichkeit, Datenmengen aus Wide-Column Stores mittels der Transformationssprache NotaQL zu transformieren sowie zu aggregieren und die Ergebnisse in Form von Diagrammen dem Benutzer darzustellen. Dabei kommen Sampling-Techniken zum Einsatz, um die Berechnung auf Kosten der Genauigkeit zu beschleunigen. Das von uns verwendete iterative Samplingverfahren sorgt für eine kontinuierliche Verbesserung der Berechnungsgenauigkeit und bietet zudem Möglichkeiten zur Genauigkeitsabschätzung, die in Form von Konfidenzintervallen in den Diagrammen dargestellt werden kann.

1 Einführung

Big-Data-Analysen sind Prozesse, die aufgrund großer Datenvolumina oft viele Minuten oder Stunden in Anspruch nehmen. Zudem sind die Ergebnisse solcher Analysen für den Menschen oft schwierig zu interpretieren, da sowohl die Datenbasis als auch das Analyseergebnis im Wesentlichen unstrukturierte Daten oder die Aggregation vieler nummerischer Werte sind. Um dem Benutzer die Datenmengen adäquat zu visualisieren, sind oft einfache Werkzeuge wie Kreis-, Balken- oder Liniendiagramme eine gute Wahl. Sie geben einen Überblick über die Datenverteilung, Zusammenhänge und über zeitliche Verläufe.

Um die Analysen verteilt auf großen Rechenclustern auszuführen, wird oft auf Verarbeitungs-Frameworks wie MapReduce [6] (bzw. Hadoop [1]) oder Spark [19] zurückgegriffen. Die verteilte Speicherung übernimmt in diesen Fällen entweder ein verteiltes Dateisystem oder ein NoSQL-Datenbanksystem. Ersteres eignet sich besonders für die Analyse von Log-Dateien, letzteres für heterogene Datensammlungen ohne fixes Schema. Die verschiedenen Arten von NoSQL-Systemen [4] bieten unterschiedliche Datenmodelle, von Sammlungen einfacher Schlüssel-Wert-Paare (*Key-Value Stores*), über Tabellen mit flexiblen Spalten (*Wide-Column Stores*) bis hin zur Speicherung komplexer Dokumente (*Document Stores*). Diese drei Arten werden die Aggregat-orientierten Datenbanken

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

genannt und haben gemeinsam, dass jeder Eintrag über eine eindeutige ID identifiziert wird. Über diese ID erfolgt auch die Partitionierung; ein impliziter Index darauf erlaubt effiziente Lese- und Schreibzugriffe. Da viele NoSQL-Datenbanksysteme keine komplexen Anfragen erlauben, werden oft Datentransformationen mittels MapReduce oder höheren Sprachen wie Pig [13], Hive [18], Phoenix [3] oder NotaQL [15] durchgeführt. Während Hive und Phoenix einen SQL-artigen Zugriff auf die Daten erlauben, bieten Pig und NotaQL weitere Transformationsmöglichkeiten, die aufgrund der Schema-Flexibilität in NoSQL-Datenbanken vonnöten sind.

Diese Arbeit beschäftigt sich mit der Visualisierung von Daten, die im Wide-Column Store *HBase* [2] gespeichert sind. Mithilfe von Transformationsskripten, die in der Sprache NotaQL formuliert werden, können Eingabedaten zunächst gefiltert, transformiert und aggregiert werden. Das Resultat wird in Form von Diagrammen dem Benutzer präsentiert. Da die Transformationen sehr lange dauern können und in Diagrammen oft keine hundertprozentige Genauigkeit erforderlich ist, schlagen wir die Verwendung von Sampling-Techniken vor. Durch das Ermitteln von Zufallsstichproben in den Eingabedaten wird die Berechnung beschleunigt, sodass der Benutzer bereits nach kurzer Zeit das Resultat in Form von Kreis-, Balken- oder Liniendiagrammen sehen kann. Der Hauptfokus dieser Arbeit liegt auf der Anwendung von iterativen Samplingprozessen bei NoSQL-Datentransformationen sowie dem Zusammenspiel von Sampling-, Visualisierungs- und Ungenauigkeitsbestimmungstechniken.

Im folgenden Kapitel stellen wir Sampling-Ansätze vor und erläutern, wie sich mit statistischen Methoden Abschätzungen über die Genauigkeit machen lassen. Weiterhin präsentieren wir die Sprache NotaQL, mit der Transformationen auf der HBase-Datenbank ausgeführt werden können. In Kapitel 3 stellen wir ein Visualisierungswerkzeug vor, welches mittels Sampling-Methoden NotaQL-Transformationen durchführt und in Form von Diagrammen visualisiert. Dort wird erläutert, wie mithilfe von Whiskers die Berechnungsgenauigkeit im Diagramm dargestellt werden kann und wie ein iterativer Transformationsprozess diese Genauigkeit kontinuierlich steigern lässt. Kapitel 4 beinhaltet Ergebnisse von Experimenten, die die Performanz des iterativen Samplingprozesses analysieren. Nach einer Vorstellung verwandter Arbeiten in Kapitel 5 folgt eine Zusammenfassung in Kapitel 6.

2 Grundlagen

Dieses Kapitel beinhaltet die mathematischen und technischen Grundlagen zur Ausführung von Sampling-basierten Tabellentransformationen, die für die Visualisierung genutzt werden. Wide-Column Stores bieten eine simple API, um komplette Tabellen zu scannen und bestimmte Zeilen anhand ihrer ID (*row-id*) abzurufen. Wegen des flexiblen Datenmodells kann keine Aussage über die Spaltennamen einer Tabelle gemacht werden. Aus diesen Gründen kommt im Rahmen dieser Arbeit kein SQL zum Einsatz, sondern die Transformationssprache NotaQL.

In Tabelle 1 wird der zeitliche Verlauf von Spritpreisen gespeichert. Die Tabelle besteht aus zwei sogenannten Spaltenfamilien, die beim Erstellen der Tabelle definiert werden. Die erste Spaltenfamilie „Spritpreise“ zeigt die Spritsorten, die eine Tankstelle anbietet, sowie deren Preise. Die zweite Spaltenfamilie „Informationen“ hingegen listet den Tankstellennamen sowie die Straße auf.

row_id	**Spritpreise**		**Informationen**	
	Diesel	SuperE10	Tankstelle	Straße
2014-11-07 12:32:00	1,319	1,489	FillItUp	Rue de Gaulle
	Diesel		Tankstelle	Straße
2014-11-07 21:30:00	1,409		FillItUp	Rotweg
	Diesel	SuperE10	Tankstelle	Straße
2014-11-08 04:30:00	1,409	1,509	FillItUp	Rue de Gaulle
	Diesel		Tankstelle	Straße
2014-11-08 05:30:00	1,389		FillItUp	Rotweg

Tabelle 1. Wide-Column Tabelle mit zwei Spaltenfamilien.

2.1 NotaQL

In [15] wird die Datentransformationssprache NotaQL vorgestellt. Sie ermöglicht das Ausdrücken vieler MapReduce-Algorithmen anhand von zwei oder drei Zeilen Code. Im Gegensatz zu Phoenix oder Hive kann direkt auf den Tabellen eines Wide-Column Stores gearbeitet werden, ohne dass vorher ein Tabellenschema definiert werden muss. NotaQL dient im Grunde zur Erstellung einer Vorschrift, wie eine *Output-Zelle* anhand des Inputs gebildet werden soll, wobei eine Zelle die Kombination aus einer row-id und einem Spaltennamen $(\_r, \_c)$ bildet. Da jede von ihnen einen atomaren Wert besitzt, repräsentiert das Tupel $(\_r, \_c, \_v)$ die Verknüpfung der Zelle mit ihrem Wert. Existieren mehrere Spaltenfamilien in der Tabelle, können die Namen der Spaltenfamilien als Präfix für den Spaltennamen verwendet werden, z.B. `Informationen:Straße` anstatt `Straße`.

Die Möglichkeit zur Selektion von Zeilen wird durch die *IN-FILTER*-Klausel gegeben. Sie definiert einen optionalen Filter am Anfang eines NotaQL-Skripts.

Folgendes ist ein NotaQL-Skript, welches angewandt auf Tabelle 1 zur Berechnung des Durchschnittspreises für die einzelnen Spritsorten der Tankstellenkette „FillItUp“ dient:

```
IN-FILTER: Informationen:Tankstelle = 'FillItUp',
OUT._r <- IN.Spritpreise:_c,
OUT.aggr:AVGPreis <- AVG(IN.Spritpreise:_v);
```

Das Skript ist wie folgt zu verstehen: Die erste Zeile führt eine Zeilenselektierung durch. Es werden nur Zeilen mit dem Wert „FillItUp“ in dem Spaltennamen „Tankstelle“ der Spaltenfamilie „Informationen“ ausgewählt. In der zweiten Zeile wird beschrieben, dass die Spaltennamen der Spaltenfamilie „Spritpreise“ (`IN.Spritpreise:_c`) die neuen row-ids (`OUT._r`) sind. Es wird also für jeden distinkten Spaltennamen in dieser Spaltenfamilie eine Zeile in der Ausgabetabelle erzeugt. Die dritte Zeile beschreibt nun die Anwendung der Aggregatfunktion *AVG* auf die Werte der Spaltenfamilie „Spritpreise“ (`AVG(IN.Spritpreise:_v)`); also eine Ermittlung des Durchschnittspreises je Sorte. Dieser Wert wird dann in der Spaltenfamilie „aggr“ unter dem Spaltennamen „AVGPreis“ (`OUT.aggr:AVGPreis`) abgelegt. Das Ergebnis der Transformation ist in Tabelle 2 zu sehen.

row_id	**aggr**
	AVGPreis
Diesel	1,3815
	AVGPreis
SuperE10	1,499

Tabelle 2. Transformationsergebnis

2.2 Sampling

Durch die Verwendung von *Sampling* lassen sich Datentransformationen auf Kosten der Berechnungsgenauigkeit um einen beliebigen Faktor beschleunigen. Sampling wird verwendet, um eine *Stichprobe*, also eine Teilmenge, aus einer *Grundgesamtheit* zu ziehen und anhand dieser Statistiken, wie Ergebnisse von Aggregatfunktionen, der Grundgesamtheit abzuschätzen. Je repräsentativer eine solche Teilmenge ist, also je mehr ihr prozentualer Aufbau dem der Grundgesamtheit gleicht, desto genauere Hochrechnungen und Schätzungen erlaubt sie. Es existieren verschiedene Techniken zum Ziehen von Stichproben [16]. Im Folgenden wird die *einfache Zufallsstichprobe* und das *iterative Sampling* kurz erläutert.

Einfache Zufallsstichprobe Auch bekannt als *Simple Random Sampling (SRS)*. Diese Technik kann mit und ohne Zurücklegen der Elemente durchgeführt werden. Wir betrachten letzteres, da durch das Vermeiden von Duplikaten in der Stichprobe im Allgemeinen eine höhere Genauigkeit erreicht wird. Jedes Element der Datenmenge hat die gleiche Wahrscheinlichkeit $p\%$, um in die Stichprobe aufgenommen zu werden. Laut dem *Gesetz der großen Zahlen* beträgt der Stichprobenumfang n somit $N \cdot p/100$, wobei N dem Umfang der Datenmenge entspricht. Die Vorteile dieser Technik liegen in der einfachen Umsetzbarkeit, und dass keine weitere Informationen, wie die Häufigkeitsverteilung des Merkmals, über die Datenmenge vorliegen müssen. Außerdem ist diese Technik *unbiased*, das heißt, dass kein Element bei der Auswahl bevorzugt wird, was zu einer Verzerrung der Häufigkeitsverteilung in der Stichprobe führen könnte.

Iteratives Sampling Der Samplingprozess besteht aus mehreren Iterationen von einfachen Zufallsstichproben. Es empfiehlt sich eine kleine Startgröße zu wählen, um bereits nach kurzer Zeit Ergebnisse zu sehen. Die iterative Ausführung mit immer größer werdender Stichprobengröße sorgt nicht nur für die kontinuierliche Verbesserung der Berechnungsgenauigkeit, sondern liefert auch die notwendigen Informationen, um ebendiese Genauigkeit mathematisch berechnen zu können. Der Prozess kann abgebrochen werden, wenn für den Nutzer eine ausreichende Genauigkeit erzielt wurde. Alternativ ist der Prozess dann beendet, sobald die Stichprobengröße 100% der Grundmenge beträgt. In [12] wird erläutert, wie das Ergebnis der vorherigen Iteration für die darauf folgende wiederverwendet werden kann. Dieses Verfahren beschleunigt die Berechnung, erhöht jedoch die Ungenauigkeit, da es sich in diesem Fall um Sampling mit Zurücklegen handelt.

Hochrechnung Bei der Verwendung der Aggregatfunktionen `SUM` und `COUNT` ist das Ergebnis einer Berechnung nicht direkt aussagekräftig für die Grundgesamtheit, wenn die Berechnung nur eine Teilmenge betrachtet hatte. Das Ergebnis muss somit zuerst mit dem Faktor $100/p$ hochgerechnet werden, wobei p der Samplingwahrscheinlichkeit entspricht.

Konfidenzintervalle Durch die Verwendung von Sampling und beim Hochrechnen der Ergebnisse von Aggregatfunktionen ergibt sich eine gewissen Berechnungsungenauigkeit. Um diese Ungenauigkeit auszudrücken, werden *ci%-Konfidenzintervalle* verwendet. Dabei gilt, dass je größer *ci*% ist, desto größer ist auch das Intervall. Beispielsweise wird Tabelle 1 als Stichprobe betrachtet und anhand dieser Daten der Durchschnittspreis für Diesel geschätzt. Dieser Durchschnittswert, genannt *Stichprobenmittel* $\bar{x}$, ist in Tabelle 2 gelistet. Die Berechnung eines 50%-Konfidenzintervalles liefert nun das Intervall [1,3513, 1,4117], ein 95%-Konfidenzintervall liefert hingegen die etwas breiteren Grenzen [1,286, 1,477]. Die Interpretation ist wie folgt: Würde das Stichprobenziehen und Anwenden derselben Transformation immer wieder wiederholt werden und jedes mal ein *ci*%-Konfidenzintervall berechnet werden, so würden *ci*% der Intervalle den wahren Wert beinhalten. Wobei *ci* während den Wiederholungen immer den gleichen Wert hat.

Für die Berechnung der Intervalle wird neben dem Stichprobenmittel $\bar{x}$ auch die Varianz der Stichprobe, genannt *Stichprobenvarianz* s_n^2, benötigt. Dieser Wert ist ein Maß für die Streuung der Daten in der Stichprobe.

$$\text{Mittelwert:} \quad \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \text{Varianz:} \quad s_n^2 = \frac{1}{n}\sum_{i=1}^{n} x_i^2 - \bar{x}^2 \tag{1}$$

Dabei entspricht n dem Stichprobenumfang und x_i dem i-ten Element in der Stichprobe. Die Stichprobenvarianz weist weiterhin eine systematische, nichtzufällige Abweichung auf. Das Ausmaß dieser Abweichung wird *Bias* oder Verzerrung genannt. Zur Korrektur wird der Faktor $n/(n-1)$ verwendet. Dieser Faktor wird als *Bessel-Korrektur* bezeichnet. Daraus berechnet sich dann die *korrigierte Stichprobenvarianz* s^2.

$$s^2 = \frac{n}{n-1} s_n^2 \tag{2}$$

Die Quadratwurzel aus der korrigierten Stichprobenvarianz liefert die *Stichprobenstandardabweichung* σ. Diese ist ein Maß für die Streuung der Daten um das Stichprobenmittel.

$$\sigma = \sqrt{s^2} \tag{3}$$

Die Berechnung der Konfidenzintervalle erfolgt nun nach der *Chebyshev-Formel* [14]:

$$[\bar{x} - \sqrt{(1/\alpha)} \cdot \frac{\sigma}{\sqrt{n}}, \bar{x} + \sqrt{(1/\alpha)} \cdot \frac{\sigma}{\sqrt{n}}] \tag{4}$$

α dient dabei als Genauigkeitsregulator. Für ein 95%-Konfidenzintervall beträgt $\alpha = 1$ - $0.95 = 0.05$, für ein 90% entsprechend $\alpha = 1$ - $0.90 = 0.10$. Allgemein ausgedrückt:

$$\alpha = 1 - ci\% \tag{5}$$

Aktuell gilt die Formel 4 nur für die Aggregatfunktion *AVG*. Eine Erweiterung dieser für die Aggregatfunktion SUM ist möglich durch die Multiplikation der Grenzen mit der Anzahl N an Datensätzen in der Grundgesamtheit. N kann bei Verwendung der einfachen Zufallsstichprobe aus der Stichprobengröße n und der Sampling-Wahrscheinlichkeit p hergeleitet werden.

$$n = N \cdot \frac{p}{100} \implies N = n \cdot \frac{100}{p} \tag{6}$$

Die Chebyshev-Formel gilt als eine konservative Abschätzung, da sie meist Intervallgrenzen berechnet, die nicht nur das gewünschten Konfidenzniveau abdecken, sondern auch überschreiten. Es wird also meist ein zu großes Intervall berechnet. Jedoch kann es auch zu einer Unterschreitung des Niveaus kommen, wenn die geschätzte Standardabweichung stark von der wirklichen abweicht und diese darüber hinaus noch sehr groß ist. Der Vorteil dieser Formel ist die einfache Berechenbarkeit und die Anwendbarkeit auf jede beliebige Werteverteilung, solange das Stichprobenmittel sowie die Stichprobenvarianz berechenbar sind.

3 Visualisierung von Transformationsergebnissen

Das Ziel dieser Arbeit ist die Visualisierung von Daten aus einem Wide-Column Store, welche mit Hilfe von NotaQL-Skripten zuerst transformiert wurden. Der entsprechende Workflow wird in Abbildung 1 dargestellt. Gestrichelte Linien zeigen dabei Prozesse an, die nur im Zusammenhang mit einem iterativen Samplingprozess ausgeführt werden.

Zuerst erfolgt eine einfache Zufallsstichprobe mit der eingestellten Sampling-Wahrscheinlichkeit aus der Eingabetabelle. Die HBase API stellt hierfür die Klasse *RandomRowFilter* bereit. Die Daten werden anschließend gemäß der gegebenen NotaQL-Vorschrift transformiert [15] und - falls erforderlich - hochgerechnet. Außerdem werden die Ergebnisse für einen iterativen Samplingprozess, in Verbindung mit den Aggregatfunktionen, in einem *Uncertainity-Computer* abgelegt. Dieser dient unter anderem zur Berechnung von Konfidenzintervallen. Vor der

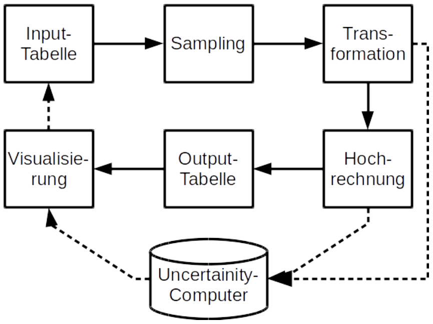


Abb. 1. Arbeitsworkflow

Visualisierung werden die Ergebnisse sortiert in der Ausgabetabelle abgelegt. Den letzten Schritt, die Erzeugung von Diagrammen, übernimmt die *Visualisierungskomponente*.

3.1 Uncertainity-Computer

Die Aggregatfunktionen AVG, COUNT, SUM, MAX, MIN werden in zwei Gruppen aufgeteilt. Gruppe eins besteht aus den beiden Funktionen MAX und MIN, deren Resultat ein bestimmtes Element der zu aggregierenden Werte darstellt. Die zweite Gruppe besteht aus den Funktionen AVG, COUNT, SUM, auf deren Resultat alle Werte einen direkten Einfluss haben.

MAX und MIN Die Funktionen der ersten Gruppe berechnen im Gegensatz zur zweiten Gruppe keinen Wert, sondern suchen den größten beziehungsweise kleinsten Wert aus einer Datenmenge heraus. Der gesuchte Wert ist also auf jeden Fall ein Bestandteil der Datenmenge. Werden diese Funktionen nun anhand einer Stichprobe ausgewertet, so kann man über den gefunden Wert nur sagen, dass der wahre Wert nicht kleiner (im Falle von MAX) oder größer (im Falle von MIN) ist. Weitere Aussagen anhand der Standardabweichung oder anderen Werten sind meist nicht aussagekräftig, da diese Funktionen die Ausreiser einer Verteilung als Ziel haben. Verringert sich jedoch ein mit der MAX-Funktion berechnetes Ergebnis in Iteration $i + 1$ gegenüber der vorherigen Iteration i, wird die vorherige Schätzung beibehalten, um eine Verschlechterung der Schätzwerte zu verhindern.

AVG, COUNT und SUM Die Funktionen der zweiten Gruppe berechnen anhand einer Datenmenge eine Statistik. Im vorherigen Kapitel wurden die Konfidenzintervalle bereits erklärt und gezeigt, wie diese für die Aggregatfunktionen AVG und SUM berechnet werden können. Um auch Konfidenzintervalle für die Aggregatfunktion COUNT angeben zu können, wird die Berechnung dieser Intervalle mit dem iterativen Samplingprozess verbunden. Zur Berechnung der Konfidenzintervalle verwaltet der Uncertainity-Computer für jede Output-Zelle die Anzahl der gespeicherten Werte sowie deren Summe und Quadratsumme. Daraus können bei Bedarf in konstanter Zeit die Konfidenzintervalle mit der Chebyshev-Formel (Gleichung 4) berechnet werden. Es werden somit mindestens zwei Werte, entsprechend mindestens zwei Samplingschritte, zur Berechnung benötigt, da sonst die Varianz 0 ist und somit kein Intervall berechnet werden kann. Dieses Berechnungsverfahren wird für alle Aggregatfunktionen dieser Gruppe verwendet.

3.2 Visualisierungskomponente

Diagramme wie Linien- oder Balkendiagramme dienen zur adäquaten Darstellung großer Datenmengen und bieten großen Modifikationsspielraum. So lassen sich Ungenauigkeiten von Daten, also Konfidenzintervalle, mit Hilfe von Whiskers (Abbildung 2) oder Deviation Areas (Abbildung 3) darstellen. Erstere nutzen dafür einen kleinen vertikalen Balken und letztere heben das Konfidenzintervall farblich vom Hintergrund und anderen Linien ab. Während Whiskers auf beiden Diagrammtypen angewandt werden können, sind die Devation Areas nur für Liniendiagramme sinnvoll, da sie eine Interpretation des Zwischenraums erlauben. Zur Umsetzung der Diagramme haben wir die Java Bibliothek *JFreeChart*[1] verwendet und erweitert [17].

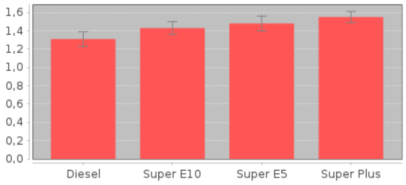


Abb. 2. Balkendiagramm mit einem Whisker pro Statistik.

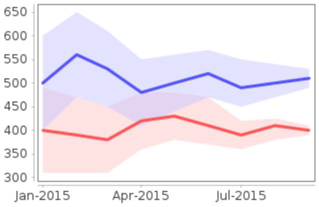


Abb. 3. Liniendiagramm mit einer Deviation Area pro Kurve.

4 Experimente

In diesem Kapitel demonstrieren wir anhand von Experimenten zum einen die Genauigkeit, die sich durch die Verwendung von iterativen Samplingtechniken erreichen lässt. Zum anderen analysieren wir die Performanz dieses Prozesses und vergleichen die Laufzeiten mit der einer vollständigen Berechnung. Dazu betrachten wir zuerst anhand eines Tests mit Startsamplinggröße 1%, die pro Iteration um 1% steigt, den Verlauf der Konfidenzintervallgrenzen und im darauffolgenden Test wird die Verwendung verschiedener Startsamplinggrößen mit entsprechender Erhöhung pro Iteration evaluiert.

Wie in Abbildung 4 zu sehen ist, wächst mit steigendem Iterationsdurchlauf, also auch steigender Sampling-Wahrscheinlichkeit, die Berechnungsgenauigkeit der verwendeten Aggregatfunktion. Dadurch sinkt nicht nur im Mittel die Abweichung zwischen berechnetem und wahrem Wert, sondern auch die berechnete Standardabweichung, wodurch die Chebyshev-Formel anwendbar ist. Der berechnete Wert konvergiert also gegen den wahren Wert und somit konvergieren auch die Intervallgrenzen gegen den wahren Wert. Die Abbildung stellt einen iterativen Samplingprozess mit Startwahrscheinlichkeit 1% auf der Tabelle 1 dar. Die verwendete Transformationsvorschrift sorgt für die Berechnung des Durchschnittspreises für Diesel über den kompletten Zeitraum und alle Tankstellen: `OUT._r <- 'Diesel', OUT.Preis <- AVG(IN.Diesel)`. Die grüne und die gelbe Linie stellen den Werteverlauf der oberen bzw. unteren Grenze des 95%-Konfidenzintervalls dar. Die blaue Linie markiert den tatsächlichen Durchschnittspreis und die rote Linie den Verlauf der berechneten Durchschnittspreise. Auffällig ist, dass manche berechneten Werte außerhalb des Konfidenzintervalls liegen. So zum Beispiel der Wert, der mit einer Sampling-Wahrscheinlichkeit von

[1] `http://www.jfree.org/jfreechart/index.html`

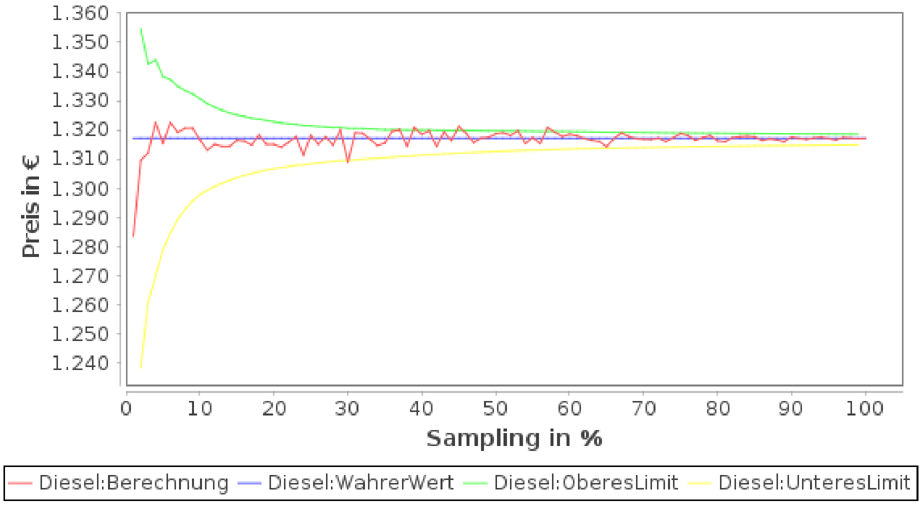


Abb. 4. Darstellung des Verlaufs der Werte des berechneten Dieseldurchschnittpreises (rot), des 95%-Konfidenzintervalls (grün, gelb) und des wahren Wertes (blau) nach Anwendung eines iterativen Samplingprozess auf Tabelle 1.

30% entstanden ist. Die Abweichung dieses Wertes zur Ideallinie ist, im Vergleich zu seinen Vor- und Nachgängern, wesentlich größer. Dies lässt sich dadurch erklären, dass die Stichprobe die Verteilung der Datenmenge ungenauer dargestellt hat. Es wurden also prozentual wesentlich mehr unterdurchschnittliche Werte in die Stichprobe eingelesen, als wirklich in der Datenmenge vorhanden sind. Es ist also eine zufällige Verzerrung aufgetreten. Obwohl der berechnete Wert außerhalb des Konfidenzintervalls liegt, enthält dieses in jedem Iterationsschritt dennoch den wahren Wert. Weiterhin ist zu erkennen, dass für den ersten Iterationsschritt und für das komplette Auslesen der Datenmenge, wie bereits erwähnt, kein Konfidenzintervall berechnet wurde.

Für das nächste Experiment wurde die folgende Transformationsvorschrift verwendet: `OUT._r <- IN._c, OUT.avg <- AVG(IN._v);` Diese führt für jeden Spaltennamen eine Durchschnittswertberechnung aus.

Eine HBase-Tabelle wurde von einem Datengenerator mit zehn Million Zeilen gefüllt, von denen jede drei Spalten mit Zufallszahlenwerten im Intervall [0, 1500] haben. Zudem wurden weitere Testtabellen generiert und dabei die Anzahl der Zeilen und Spalten variiert. Alle Tests wurden auf einem Zweikern-Prozessor (je 2,1 GHz) sowie 3,9 GB RAM durchgeführt. Als Wide-Column Store wurde eine HBase Standalone Datenbank (Version 0.98.7) verwendet.

Abbildung 5 zeigt die Abhängigkeit der Berechnungsdauer von der Samplinggröße. Es fällt

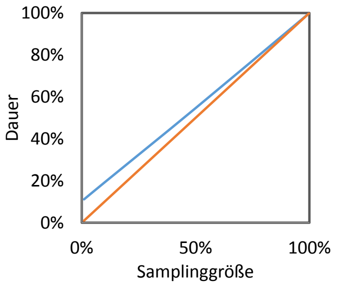


Abb. 5. Berechnungsdauer relativ zur vollständigen Berechnung (blaue Linie). Zum Vergleich die Optimallinie (orange).

auf, dass bereits eine 1% Samplingberechnung zehn Prozent der Dauer im Vergleich zur vollständigen Berechnung benötigt. Mit zunehmender Samplinggröße nähert sich der Zusammenhang nahezu der Optimallinie.

Im Folgenden wird der iterative Samplingprozess auf der gleichen Tabelle (10 Mio. Zeilen, 3 Spalten) genauer untersucht. Dazu wurden auf dieser Tabelle vier iterative Samplingprozesse mit den Startwahrscheinlichkeiten 1%, 5%, 10% und 20%, mit entsprechender prozentualen Erhöhung pro weiterem Iterationsschritt, durchgeführt (siehe Abbildung 6). Zum Vergleich wurde eine vollständige Transformation ohne die Verwendung von Sampling ausgeführt. Diese dauerte knapp zehn Minuten und ist als rote Linie in Abbildung 6 eingezeichnet. Ein iterativer Samplingprozess besitzt gegenüber der Verarbeitung ohne Sampling einen zeitlichen Vorteil, solange seine kumulierte Dauer geringer ist als die Dauer für die komplette Verarbeitung. Die Verwendung der Startwahrscheinlichkeit 1% liefert zwar am schnellsten die ersten Ergebnisse, jedoch sind diese recht ungenau und liefern somit nur eine grobe Approximation. Nach acht Schritten hat der Sampling-Prozess seinen Zeitvorteil verloren und nur 8% der Datenmenge im letzten Schritt verarbeitet. Mit steigender Startwahrscheinlichkeit erhöht sich die Dauer bis zum ersten Ergebnis, dafür steigt aber auch die maximal mögliche Anzahl an ausgelesenen Daten. So ist mit einer Startwahrscheinlichkeit von 20% die Ausführung von bis zu zwei Iterationsschritten sinnvoll und somit werden 40% der Datenmenge im zweiten Schritt ausgelesen. Wie in Abbildung 4 zu sehen ist, ist die Berechnungsgenauigkeit bereits bei einer Samplinggröße von 10 bis 15% für viele Anwendungen ausreichend. Diese Größe ist nach zwei bis drei Iterationen zu je 5% erreicht, was in der Hälfte der Zeit gegenüber einer vollständigen Berechnung ausgeführt werden kann. Die direkte Wahl einer größeren Samplinggröße würde zwar die gleichen Ergebnisse bereits nach kürzerer Zeit und nur einer Iteration liefern. Dafür kann dem Benutzer allerdings keine Information über die Berechnungsgenauigkeit mittels Whiskers und Deviation Areas gegeben werden.

Die Wahl der Startwahrscheinlichkeit hängt somit von der gewünschten Berechnungsdauer für das erste Ergebnis und der Approximationsgenauigkeit ab. Dabei bezieht sich die Approximationsgenauigkeit auf die maximale Sampling-Wahrscheinlichkeit und etwaige berechnete Konfidenzintervalle.

5 Verwandte Arbeiten

Sowohl Datenvisualisierungen als auch Sampling-Verfahren kommen oft zum Einsatz, wenn es um die Analyse großer Datenmengen geht. In [11] wird auf die Wichtigkeit hingewiesen, bei der Verwendung von Sampling darauf zu achten, dass die Stichproben die Charakteristika der Originaldaten widerspiegeln. Der in diesem Artikel vorgestellte Ansatz verwendet zufällige Stichproben anhand der Zeilen auf einer HBase-Tabelle. Gegenüber komplexeren Strukturen wie Graph-Datenbanken sowie Tabellen, die über Join-Pfäde verbunden werden müssen, können hier Datensätze weitestgehend unabhängig voneinander betrachtet werden, was eine höhere Genauigkeit zur Folge hat.

EARL [10] ist eine auf Hadoop-basierende Sampling-Bibliothek, die Bootstrapping [7–9] verwendet, um Aussagen über die Genauigkeit einer Berechnung

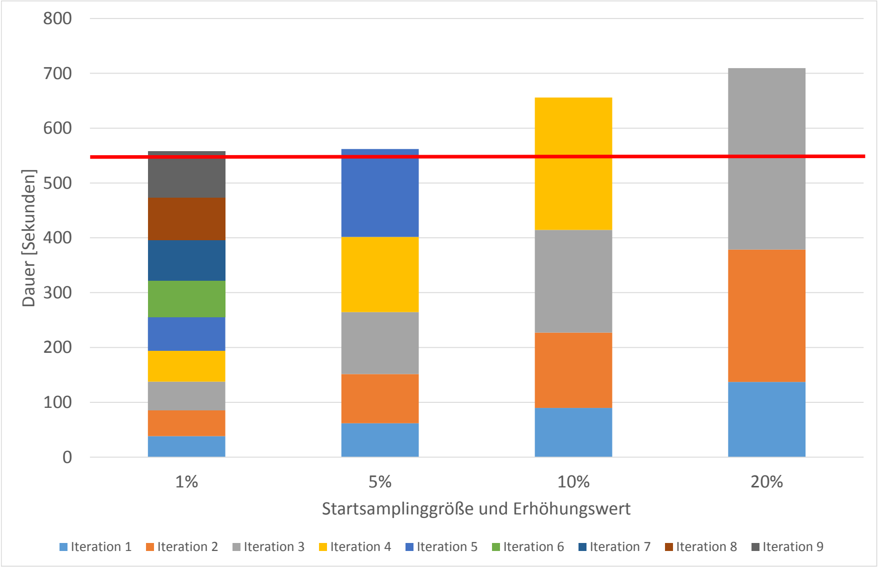


Abb. 6. Ausführung mehrerer Iterationen von Transformationen mit unterschiedlichen Samplinggrößen sowie einer vollständigen Berechnung (rote Linie).

zu machen. EARL führt dabei Transformationen kontinuierlich durch. Unser Ansatz dagegen ist iterativ, was den Vorteil hat, dass auch ohne Bootstrapping eine Genauigkeitschätzung möglich ist, nämlich über die Varianz der Ergebnisse verschiedener Iterationen. Beim Bootstrapping wird eine Stichprobe in Unterstichproben zerlegt, sodass die die aggregierten Resultate auf diesen kleineren Mengen basieren.

In [5] werden Techniken zur Visualisierung von SQL-Anfrageergebnissen vorgestellt. Der Autor verwendet verschiedene Darstellungsformen wie Whiskers und Deviation Areas, um die Berechnungsunsicherheit darzustellen.

6 Zusammenfassung

Wir haben gezeigt, dass bei der Ausführung von Tabellentransformationen Sampling-Techniken dazu beitragen können, früh erste Ergebnisse zu sehen. Eine Visualisierungssystem ist in der Lage, HBase-Tabellen als Linien-, Balken- oder Kreisdiagramm darzustellen sowie weitere Iterationen auf vergrößerten Stichproben im Hintergrund weiterlaufen zu lassen. Die Ergebnisse mehrerer Iterationen können für die Genauigkeitsberechnung verwendet werden, die mittels Whisters und Deviation Areas in die dargestellten Diagramme eingezeichnet werden können. Dadurch erhält der Benutzer nicht nur nach kurzer Zeit erste Ergebnisse einer Berechnung, sondern auch eine Information über deren Genauigkeit und die Möglichkeit, eine Berechnung vorzeitig abzubrechen, wenn die gewünschte Genauigkeit erreicht ist. Wir haben mittels Experimenten gezeigt, dass dieses Vorgehen deutlich schneller ist als eine vollständige Berechnung, und dass die Ungenauigkeit bereits bei geringen Stichprobengrößen so minimal ist, dass die

Diagramme, die dem Benutzer gezeigt werden, im Wesentlichen so aussehen, als würden sie auf der kompletten Datenbasis basieren.

Literatur

1. Apache Hadoop project. `http://hadoop.apache.org/`.
2. Apache HBase. `http://hbase.apache.org/`.
3. Apache Phoenix. `http://phoenix.apache.org/`.
4. Rick Cattell. Scalable sql and nosql data stores. *ACM SIGMOD Record*, 39(4):12–27, 2011.
5. Danyel Fisher. Incremental, Approximate Database Queries and Uncertainty for Exploratory Visualization. In *IEEE Symposium on Large Data Analysis and Visualization*. IEEE, October 2011.
6. Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.
7. B. Efron. Bootstrap methods: Another look at the jackknife. *Ann. Statist.*, 7(1):1–26, 01 1979.
8. Michael I. Jordan. Divide-and-conquer and statistical inference for big data. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 4–4, New York, NY, USA, 2012. ACM.
9. Ariel Kleiner, Ameet Talwalkar, Purnamrita Sarkar, and Michael Jordan. The big data bootstrap. *arXiv preprint arXiv:1206.6415*, 2012.
10. Nikolay Laptev, Kai Zeng, and Carlo Zaniolo. Early accurate results for advanced analytics on mapreduce. *Proceedings of the VLDB Endowment*, 5(10):1028–1039, 2012.
11. Shuai Ma, Jia Li, Chunming Hu, Xuelian Lin, and Jinpeng Huai. Big graph search: challenges and techniques. *Frontiers of Computer Science*, pages 1–12, 2014.
12. Marc Schäfer, Johannes Schildgen, Stefan Deßloch. Sampling with Incremental MapReduce. *Workshop on Big Data in Science (BigDS), BTW*, 2015.
13. Christopher Olston, Benjamin Reed, Utkarsh Srivastava, Ravi Kumar, and Andrew Tomkins. Pig latin: a not-so-foreign language for data processing. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1099–1110. ACM, 2008.
14. S. Acharaya, P. B. Gibbons, V. Poosala , S. Ramaswamy. Join Synopses for Approximate Query Answering. Technical report, Bell Laboratories, Murray Hill, New Jersey, 1999. Full version of the paper appearing in SIGMOD'99.
15. Johannes Schildgen and Stefan Deßloch. NotaQL Is Not a Query Language! It's for Data Transformation on Wide-Column Stores. In *British International Conference on Databases - BICOD 2015*, 7 2015.
16. Seymour Sudman. Applied sampling. *Academic Press New York*, 1976.
17. Stefan Braun. Visualisierung von NotaQL-Transformationen unter der Verwendung von Sampling-Techniken. Bachelorarbeit TU Kaiserslautern, 2015.
18. Ashish Thusoo, Joydeep Sen Sarma, Namit Jain, Zheng Shao, Prasad Chakka, Suresh Anthony, Hao Liu, Pete Wyckoff, and Raghotham Murthy. Hive: a warehousing solution over a map-reduce framework. *Proceedings of the VLDB Endowment*, 2(2):1626–1629, 2009.
19. Matei Zaharia, Mosharaf Chowdhury, Michael J Franklin, Scott Shenker, and Ion Stoica. Spark: cluster computing with working sets. In *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, volume 10, page 10, 2010.

Kontrolliertes Schema-Evolutionsmanagement für NoSQL-Datenbanksysteme

Uta Störl[1], Meike Klettke[2], Stefanie Scherzinger[3]

[1] Hochschule Darmstadt, uta.stoerl@h-da.de
[2] Universität Rostock, meike.klettke@uni-rostock.de
[3] OTH Regensburg, stefanie.scherzinger@oth-regensburg.de

Zusammenfassung. In der agilen Entwicklung von Anwendungen werden neue Software-Versionen häufig und regelmäßig veröffentlicht. Relationale Datenbanksysteme mit ihrem rigiden Schema-Management werden dabei oft als unflexibel empfunden. Schemalose NoSQL-Datenbanksysteme bieten zwar die nötige Flexibilität, unterstützen aber kein systematisches Release- und Schema-Evolutionsmanagement.
Dieser Artikel stellt entsprechende Konzepte vor: Schema-Evolutionsschritte werden deklarativ spezifiziert, ihre Umsetzung erfolgt für die Anwendung transparent *eager* oder *lazy*. Während eine *eager* Migration sämtliche Datensätze erfasst, werden *lazy* persistierte Objekte nur bei Zugriff durch die Anwendung aktualisiert. Wir diskutieren eine effiziente *lazy* Migration selbst für den Fall, dass eine Migration über mehrere Evolutionsschritte und mehrere persistierte Objekte hinweg erfolgt.

1 Einführung

NoSQL-Datenbanksysteme werden in der Anwendungsentwicklung nicht nur bei sehr großen Datenmengen eingesetzt: Die Flexibilität in der Verwaltung heterogen strukturierter Daten macht NoSQL-DBMS gerade in der agilen Entwicklung attraktiv [5]. Das Schema wird typischerweise in der Anwendungsschicht mit Hilfe von Objekt-NoSQL Mapper Bibliotheken deklariert. Diese unterstützen mitunter auch weitere Aufgaben des Schema-Managements, wie etwa die *lazy* Migration von vorhandenen Daten im Produktionssystem [11]. Letztlich stellen Mapper aber nur eine Programmierschnittstelle bereit, die Ausimplementierung bleibt Aufgabe der Entwickler. Während sich Objekt-NoSQL Mapper in der Entwickler-Community großer Beliebtheit erfreuen, findet aus Sicht der Datenbank-Community eine gravierende Schichtverletzung statt.

Eine Schichtverschiebung des Schema-Managements aus der Anwendung in die Datenbank ist (nicht nur aus Gründen der Performance) wünschenswert. Das NoSQL-DBMS F1 [6] ist ein Schritt in diese Richtung: F1 verwaltet ein relationales Schema und implementiert ein rigides Protokoll, um hochfrequent

Copyright © 2015 by the paper's authors. Copying permitted only for private and academic purposes. In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

Schemaänderungen in einem verteilten System zu propagieren. Schemaänderungen werden hier zwar asynchron, aber *eager* ausgeführt.

Database-as-a-Service Kunden sind allerdings sehr daran interessiert, unnötige (kostenpflichtige) Lese- und Schreiboperationen gegen die Datenbank zu vermeiden. Das macht eine *lazy* Datenmigration besonders interessant, da persistierte Objekte nur dann migriert werden, wenn die Anwendung auch auf sie zugreift.

KVolve [7] vollzieht *lazy* Schemaänderungen in NoSQL-DBMS mit einem nachweislich niedrigen Overhead. Allerdings werden nur einfache Operationen unterstützt, wie das Hinzufügen und Entfernen von Attributen. Da die meisten NoSQL-DBMS keine Join-Operationen unterstützen, stellen Denormalisierungsoperationen, und damit komplexere Schema-Änderungen wie `copy` oder `move` Operationen, wichtige Schema-Evolutionsschritte dar.

Unsere deklarative Evolutionssprache aus [8] unterstützt entsprechend das Kopieren von Attributen zwischen persistierten Objekten. Wir zeigen in diesem Artikel, dass sich dadurch neue Herausforderungen an die Korrektheit einer *lazy* Migration stellen (Kapitel 2). In Kapitel 3 präsentierten wir das *Darwin* Projekt[4] mit der *lazy* Implementierung unserer Evolutionssprache. Die Zusammenfassung und ein Ausblick auf weitere Vorhaben folgen am Ende des Artikels.

2 Lazy Migration

Bei der *lazy* Migration wird ein Entity (d.h. ein persistiertes Objekt) erst zum Zeitpunkt seiner Verwendung in das aktuelle Schema migriert. Dabei bleibt die Datenbank für die Anwendung verfügbar. Aus Sicht der Anwendung muss transparent bleiben, ob die Daten *eager* oder *lazy* migriert werden; das stellt eine Herausforderung bei der Entwicklung von *lazy* Migrationsprotokollen dar.

Beispiel: Abbildung 1 zeigt die Daten eines Online-Rollenspiels über mehrere Versionen der Anwendung hinweg. In der NoSQL-DB werden Player und ihre Missionen persistiert. Das Schema entwickelt sich mit der Anwendung, so wird in Version 2 ein neues Attribut SCORE zur Klasse Player hinzugefügt. Bei einer *lazy* Migration werden persistierte Entities nicht unmittelbar bei der Veröffentlichung einer neuen Anwendungsversion aktualisiert. Erst wenn Player Lisa von Version 2 der Anwendung geladen wird, erfolgt das Hinzufügen des Attributes SCORE. Beim Übergang zu Schema-Version 3 soll das Attribut SCORE von der Klasse Player zur Klasse Mission kopiert[5] werden. Diese Operation wird für Mission 100 erst dann ausgeführt, wenn diese in die Anwendung geladen wird.

Die analoge Vorgehensweise führt bei Mission 101 zu einem inkorrekten Ergebnis: In Abbildung 1 wird die `copy` Operation mit einer noch nicht migrierten Version von Player Bart ausgeführt. Dementsprechend wird kein SCORE-Attribut kopiert. Das geladene Objekt unterscheidet sich von dem Objekt, das

[4] In einer früheren Implementierung wurde unsere Sprache aus [8] in der *Cleager* Konsole *eager* mit Hilfe von MapReduce Prozessen umgesetzt [9].

[5] Wie in Abbildung 1 zu sehen, erfolgt die Auswahl der *target* Entites bei der `copy` Operation durch die Angabe einer geeigneten `where`-Klausel (analoges gilt für `move`).

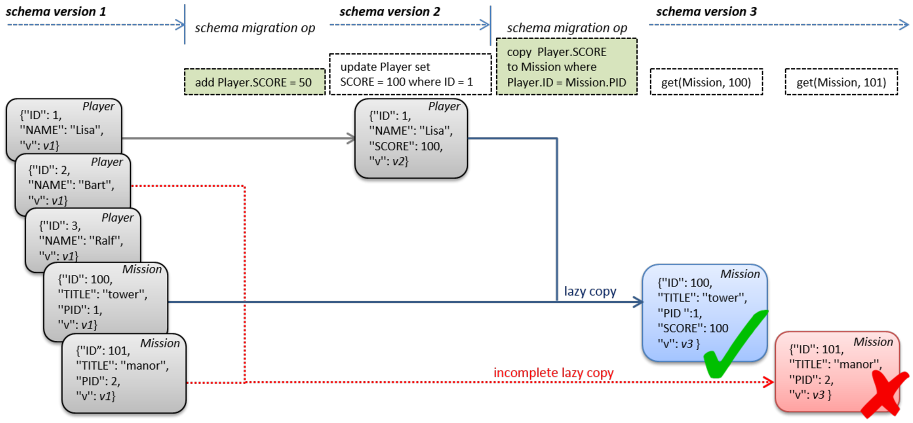

schema version 1
schema migration op
schema version 2
schema migration op
schema version 3
add Player.SCORE = 50
update Player set SCORE = 100 where ID = 1
copy Player.SCORE to Mission where Player.ID = Mission.PID
get(Mission, 100)
get(Mission, 101)
Player {"ID": 1, "NAME": "Lisa", "v": v1}
Player {"ID": 2, "NAME": "Bart", "v": v1}
Player {"ID": 3, "NAME": "Ralf", "v": v1}
Mission {"ID": 100, "TITLE": "tower", "PID": 1, "v": v1}
Mission {"ID": 101, "TITLE": "manor", "PID": 2, "v": v1}
Player {"ID": 1, "NAME": "Lisa", "SCORE": 100, "v": v2}
lazy copy
Mission {"ID": 100, "TITLE": "tower", "PID ":1, "SCORE": 100 "v": v3 }
incomplete lazy copy
Mission {"ID": 101, "TITLE": "manor", "PID": 2, "v": v3 }

Abb. 1. Mission 100 wird *lazy* migriert, indem das SCORE-Attribut des Spielers kopiert wird. Bei Mission 101 führt diese Vorgehensweise zu einem inkorrekten Ergebnis.

durch eine *eager* Migration geladen worden wäre. Wenn mehrere Evolutionsschritte *lazy* nachzuvollziehen sind und mehr als ein Entity an der Migration beteiligt ist (etwa bei `copy` oder `move` Operationen), stellen sich Herausforderungen an die Korrektheit einer *lazy* Migration.

Abbildung 2 zeigt eine korrekte, zweistufige Migration von Mission 101, bei der Player Bart zunächst migriert wird, bevor sein SCORE kopiert wird.

Kaskadierender Implementierungsansatz: Ein erster Ansatz für die korrekte Ausführung der *lazy* Migration basiert auf folgender Vorgehensweise: Bei einer `copy` oder `move` Operation werden alle korrespondierenden *source* bzw. *target* Entities, die in der gleichen oder einer früheren Version im Vergleich zum zu migrierenden Entity vorliegen, ebenfalls in die aktuelle Version des Entity migriert. Sofern dabei eine weitere `copy` oder `move` Operation ausgeführt werden muss, wird diese analog durchgeführt und ggf. rekursiv fortgesetzt. Damit wird nachträglich der Zustand einer *eager* Migration für die betroffenen Entities sichergestellt.

Dieser *kaskadierende* Ansatz stellt die Korrektheit der *lazy* Migration sicher, führt allerdings dazu, dass beim Laden eines einzelnen Entity ggf. weitere, unbeteiligte Entities migriert werden, was zu Einbußen in der Laufzeit führt. Im Folgenden skizzieren wir erste Ideen für die Optimierung der *lazy* Migration.

Optimierungsansätze: Bei der *kaskadierenden* Implementierung werden bei der Migration von Entities, die *source* einer `copy` oder `move` Operation sind, auch die *target* Entities (kaskadierend) migriert, da sonst ggf. die Informationen der *source* Entities später nicht mehr zur Verfügung stehen. Sind hingegen alte Versionen der Entities verfügbar (wie in vielen NoSQL-DBMS implementiert), kann die Migration der *target* Entities *lazy* ausgeführt werden, also erst beim Zugriff. Dies reduziert die Anzahl der (zu einem Zeitpunkt) zu migrierenden Entities.

Bei einer *lazy* Migration liegen Entities, die über längere Zeit nicht verwendet wurden, in einer älteren Version vor (Version *i*). Werden diese von der Anwen-

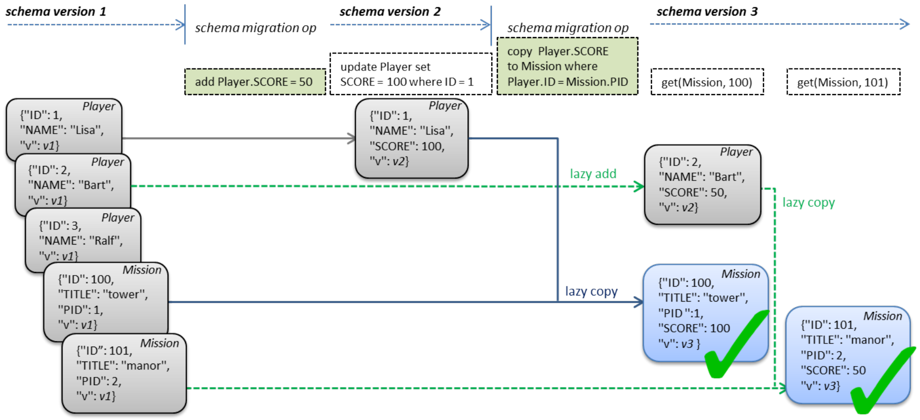

schema version 1
schema migration op
schema version 2
schema migration op
schema version 3
add Player.SCORE = 50
update Player set SCORE = 100 where ID = 1
copy Player.SCORE to Mission where Player.ID = Mission.PID
get(Mission, 100)
get(Mission, 101)
Player
{"ID": 1, "NAME": "Lisa", "v": v1}
Player
{"ID": 2, "NAME": "Bart", "v": v1}
Player
{"ID": 3, "NAME": "Ralf", "v": v1}
Mission
{"ID": 100, "TITLE": "tower", "PID": 1, "v": v1}
Mission
{"ID": 101, "TITLE": "manor", "PID": 2, "v": v1}
Player
{"ID": 1, "NAME": "Lisa", "SCORE": 100, "v": v2}
lazy add
Player
{"ID": 2, "NAME": "Bart", "SCORE": 50, "v": v2}
lazy copy
lazy copy
Mission
{"ID": 100, "TITLE": "tower", "PID ":1, "SCORE": 100 "v": v3 }
Mission
{"ID": 101, "TITLE": "manor", "PID": 2, "SCORE": 50 "v": v3}

Abb. 2. Korrekte Ausführung der *lazy* Migration von Mission 101.

dung gelesen, dann erfolgt die Migration in die aktuelle Version $(i+x)$. Über der *Folge von Update-Operationen* $u_{i+x}(u_{i+x-1}(..(u_{i+1}(entity_i))))$ sind äquivalente Zusammenfassungen möglich [1]. In der NoSQL-DB wird dann nur das Ergebnis der Migration (das Entity in der Version $i + x$) persistiert. Entities, die durch Zwischenschritte entstanden sind, werden nicht dauerhaft gespeichert, sodass die Anzahl der Schreiboperationen erheblich reduziert werden kann.

In [10] präsentieren wir einen Ansatz, der die Migration durch Datalog-Regeln spezifiziert. Eine inkrementelle top-down Auswertung stellt sicher, dass die Ergebnisse einer *lazy* Migration aus Sicht des Anwendungsprogramms mit dem Ergebnis übereinstimmt, das bei der Durchführung der *eager* Migration (bzw. der äquivalenten bottom-up Auswertung) entsteht.

3 Schema-Evolutionsmanagement mit *Darwin*

In [3] wurden als Anforderungen für eine Schema-Management-Komponente die *Definition eines Schemas*, die *Validierung von Entities* gegen ein Schema sowie die Unterstützung der *Schema-Evolution inklusive Datenmigration* definiert. Die dort vorgeschlagene Schema-Management-Komponente wurde inzwischen prototypisch implementiert: *Darwin* ist eine Schema-Management-Komponente, die zwischen der Applikation bzw. dem Objekt-NoSQL Mapper und dem NoSQL-DBMS angesiedelt ist und die oben stehenden Funktionalitäten unterstützt.

Das Schema wird als JSON-Schema [2] gespeichert. Damit lassen sich sowohl Schemata von Dokumentenorientierten als auch Column-Family-Datenbanksystemen verwalten. Aktuell unterstützt *Darwin* die NoSQL-DBMS MongoDB und Couchbase. Durch die bereitgestellte abstrakte Datenbank-Schnittstelle ist es aber einfach möglich, weitere DBMS anzubinden.

Die Schema-Evolutionsoperationen können in *Darwin* direkt auf einer Konsole (CLI) eingegeben oder über eine Web-Applikation generiert werden. Die Migration der Daten erfolgt *eager* oder *lazy*. *Darwin* ist damit die erste uns bekannte Schema-Management-Komponente für NoSQL-DBMS, die ein kontrolliertes Schema-Management für NoSQL-DBMS (inklusive `copy` und `move` Operationen) und *lazy* Migration unterstützt.

4 Zusammenfassung und Ausblick

In der vorgestellten Schema-Management-Komponente werden verschiedene Datenbanktechniken für NoSQL-Datenbanksysteme eingesetzt, die Schema-Evolution in hochverfügbaren Anwendungen orchestrieren:

– Eine deklarative Sprache zur Schemaevolution
– Definition der Semantik der Datenmigrations-Operationen über Datalog
– Versionierung von Daten zur Konsistenzsicherung bei *lazy* Migration

Es wurden weitere Datenbanktechniken für NoSQL-Daten adaptiert, wie die Schema-Extraktion aus vorhandenen Datensätzen über Strukturgraphen [4]. Die Integration dieser Implementierung in *Darwin* ist einer der nächsten Schritte.

Um die Schema-Management-Komponente komfortabler für den Anwendungsentwickler zu gestalten, ist die Entwicklung eines IDE Plugins geplant, das bei Veränderungen an der Klassenstruktur die korrespondierenden Schema-Evolutionsoperationen automatisch generiert.

Danksagung: Wir danken den Studierenden O. Haller, T. Landmann, T. Lehwalder, D. Müller, H. Nkwinchu, M. Richter und M. Shenavai der Hochschule Darmstadt für die Implementierung von *Darwin*.

Literatur

1. M. Arenas, P. Barceló, L. Libkin, and F. Murlak. *Relational and XML Data Exchange*. Synthesis Lectures on Data Management. Morgan & Claypool, 2010.
2. JSON Schema Community. *JSON Schema*, June 2015. `http://json-schema.org`.
3. M. Klettke, S. Scherzinger, and U. Störl. "Datenbanken ohne Schema? - Herausforderungen und Lösungs-Strategien in der agilen Anwendungsentwicklung mit schema-flexiblen NoSQL-Datenbanksystemen". *Datenbank-Spektrum*, 14(2), 2014.
4. M. Klettke, U. Störl, and S. Scherzinger. "Schema Extraction and Structural Outlier Detection for JSON-based NoSQL Data Stores". In *Proc. BTW'15*, 2015.
5. Z. H. Liu and D. Gawlick. "Management of Flexible Schema Data in RDBMSs - Opportunities and Limitations for NoSQL". In *CIDR'15*, 2015.
6. I. Rae, E. Rollins, J. Shute, S. Sodhi, and R. Vingralek. "Online, Asynchronous Schema Change in F1". In *Proc. VLDB'13*, 2013.
7. K. Saur, T. Dumitra, and M. Hicks. "Evolving NoSQL Databases without Downtime". Technical report, University of Maryland, College Park, Apr. 2015. `http://www.cs.umd.edu/~ksaur/pubs/kvolve-submitted.pdf`.
8. S. Scherzinger, M. Klettke, and U. Störl. "Managing Schema Evolution in NoSQL Data Stores". *Proc. DBPL'13*, arXiv:1308.0514 [cs.DB], 2013.
9. S. Scherzinger, M. Klettke, and U. Störl. "Cleager: Eager Schema Evolution in NoSQL Document Stores". In *Proc. BTW'15*, 2015.
10. S. Scherzinger, U. Störl, and M. Klettke. "A Datalog-based Protocol for Lazy Data Migration in Agile NoSQL Application Development". In *Proc. DBPL'15*, 2015.
11. U. Störl, T. Hauff, M. Klettke, and S. Scherzinger. "Schemaless NoSQL Data Stores Object-NoSQL Mappers to the Rescue?". In *Proc. BTW'15*, 2015.

Exploiting Social Judgements in Big Data Analytics

Christoph Lofi, Philipp Wille

Institut für Informationssysteme
Technische Universität Braunschweig
38106 Braunschweig, Germany
{lofi, wille}@ifis.cs.tu-bs.de

Abstract. Social judgements like comments, reviews, discussions, or ratings have become a ubiquitous component of most Web applications, especially in the e-commerce domain. Now, a central challenge is using these judgements to improve the user experience by offering new query paradigms or better data analytics. Recommender systems have already demonstrated how ratings can be effectively used towards that end, allowing users to semantically explore even large item databases. In this paper, we will discuss how to use unstructured reviews to build a structured semantic representation of database items, enabling the implementation of semantic queries and further machine-learning analytics. Thus, we address one of the central challenge of Big Data: making sense of huge collections of unstructured user feedback.

Keywords: Big Data; User Generated Content; Data Mining; Latent Semantics

1 Introduction

The recent years have brought several changes in how the Web is used by both individual users and companies alike. Especially, the Social Web had a strong impact and has now become a major innovator of technology. Users got accustomed to an active and contributive usage of the Web, and feel the need to express themselves and connect with like-minded peers. As a result, social networking sites like Facebook amassed over 940 million active users. At the same time, there are countless special-interest sites for music, movies, art, or anything that is of interest to any larger group. But the real revolution lies in the way people interact with these sites: Following their social nature, millions of people discuss, rate, tag, review, or vote content and items they encounter on the Web. Therefore, "I Like" buttons, star scales, or comment boxes are omnipresent on today's Web. Of course, recognizing the value of the exploitation of such activities, many companies encourage the creation of such user-generated feedback [1] and exploit it in order to analyze their user base, provide better meta-data and user interaction, or to optimize their marketing strategies. Storing and querying the huge amount of data related to these socially-driven Web activities and also supporting the subsequent analysis are among the central concerns in the current discussions about Big Data and Cloud Computing systems. From a database research point of view, there is a clear challenge given by Big Data applications: huge amounts of data need to be stored and served efficiently and flexibly in a distributed fashion. This results in many interesting database-like systems which have to decide on tough trade-offs with respect to possible database features, efficiency, and scalability, e.g., [2]. However, beyond storage, there is another at least equally challenging problem: How can all that data, and especially user-generated judgements and feedback, be put to a practical use? Here, a core problem is that user contributions in the Social Web are often very hard to control and usually do not follow strict schemas or guidelines. For example, a user finding an interesting online news article might vote for that article on her preferred social site, while a user leaving the cinema after a particular bad movie experience may log onto her favorite movie database, rating the movie lowly, and venting her disappointment in a short comment or a more elaborate review.

In this paper, we discuss the challenge of building structured, but latent representations of "experience items" stored in a database (like movies, books, music, games, but also restaurants or hotels) from unstructured user feedback. Such representations should encode the consensual perception of an item from the perspective of a large general user base. If this challenge could be solved, established database techniques like SQL-queries, similarity queries, but also several data mining techniques like clustering could be easily applied to user-generated feedback. In the following, we will use movies as an example use case. However, the described techniques can easily be transferred to any other domain which has user ratings or reviews available. In detail, our outline is:

- We explain our perceptual space, an established state-of-the-art latent representation of items based on user-item ratings. While this technique is accepted and proven, the required data is hard to obtain and carries some privacy concerns.
- Instead of using ratings, we discuss how to use user-provided natural language reviews to derive a semantically meaningful item representation. As reviews are easier to obtain, they could serve as an attractive alternative to rating-based approaches. We evaluate three different approaches and compare them to an established rating-based approach. Especially, we will focus on the use of neural language embeddings, a currently emerging technique from the natural language processing community.
- As our evaluation will show, there are still quality problems with directly turning review texts into latent item representations. We will discuss the potential sources of these problems, and will outline possible solutions and remedies to be explored by later research. Furthermore, we will briefly discuss the challenge of making some of the latent dimensions explicit and explainable.

2 Experience Items, User Content and Latent Representations

In this paper, we use an e-commerce scenario where users can browse and buy frequently consumable experience. This scenario is a prime application for user-generated judgements: user-friendly interaction with experience items is notoriously difficult, as there is an overwhelming number of those items easily available, some of them being mainstream, vastly popular, and well-known, but most of them being relatively unknown long tail products which are hard to discover without suitable support. Even more, the subjective user experience those products will entail (which, for most people, is the deciding factor for buying the product) are difficult to describe by typically available meta-data like production year, actor names, or even rough genre labels. Due to this problem, web services dealing with experience products enthusiastically embraced techniques for motivating the creation of user-generated judgements in the form of ratings, comments or reviews. In its most naïve (but very common) implementation, rating and review data are simply displayed to users without any additional processing. Querying and discovering items still relies on traditional SQL-style queries and categorizing based on non-perceptual meta-data (e.g., year, actor list, genre label, etc.). Manually ingesting these user judgements may help potential new customers to decide if they will like or dislike a certain item, but it does not really help them to discover new items beyond their expertise (i.e., this approach works fine if a user knows exactly what she is looking for, but has not yet come to a final buying decision). This led to the development of recommender systems [3, 4], which proactively predict which items a user would enjoy. Often, this relies on collaborative filtering techniques [3] which exploit a large number of user-item ratings for predicting a user's likely ratings for each yet-unrated item. While collaborative filtering recommender systems have been proven to be effective [5], they have only very limited query capabilities (basically, a recommender system is just a single static query for each user).

For enabling semantic queries like similarity exploration [6], the first step is to find semantically meaningful representations of database items going beyond available structured meta-data. It has been shown that experience items are generally better characterized by their *perceived properties*, e.g. their mood, their style, or if there are certain plot elements – information which is rarely explicitly available and expensive to obtain.

Therefore, we aim at extracting the most relevant perceptual aspects or attributes describing each item from user-generated judgements automatically, resulting in a vector representing these attributes. Some existing systems like Pandora's Music Genome [7] and Jinni's Movie Genome [8] already worked on this challenge, but these systems are proprietary and rely on strong human curation and expert taxonomies. In contrast, we try to use fully automatic approaches. Therefore, we investigate *dense latent representations* of each item, i.e. for each item, we mine all values with respect to each perceptual attribute. However, while these attributes might have a real-world interpretation, that interpretation is unknown to us (for example, one attribute might represent how scary a movie is, but this attribute will simply have a generic name and we do not know that it indeed refers to scariness). Basically, when creating a dense latent representation, each item is embedded in a high-dimensional vector space (therefore, such

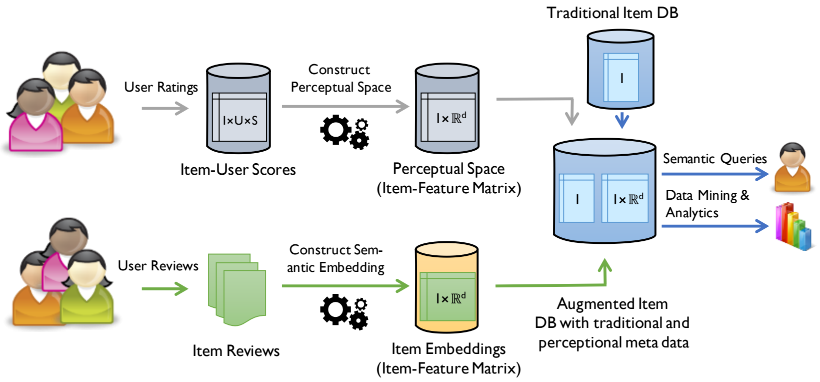


Figure 1: Augmenting Item Meta Data with User Generated Information

Extracted latent vector representations can be used side-by-side for supporting, e.g., similarity queries or different data mining techniques like clustering or automatic labeling

techniques are also sometimes called "embeddings") with usually 100-600 automatically created dimensions where each dimension represents an (unlabeled) perceptual aspect (like scariness, funniness, quality of special effects, or even the presence of certain plot elements like "movie has slimy monsters").

Even without explicitly labeling the dimensions, latent representations can already provide tremendous benefits with respect to the user experience. They can directly be used by most state-of-the art data analytics and machine learning algorithms like clustering, supervised labeling, or regression. Also, from a user's perspective, such representations can be used with great effect to allow for semantic queries as we have shown in [6] for movies. Here, having a meaningful implementation for measuring the semantic similarity (derived from the vector distance between movies) has been exploited to realize personalized and user-friendly queries using the query-by-example (QBE) paradigm. In that work ([6]), we relied on Perceptual Spaces, a latent representation derived from a large collection of user-item ratings which we will use as a reference implementation in this paper. Unfortunately, such ratings are hard to obtain and also come with some privacy concerns. Therefore, a core contribution of this paper is exploring alternative techniques for building item embeddings using much more accessible reviews (see section 2.2) instead. The resulting overall workflow of our aproach is summarized in figure 1.

2.1 Perceptual Spaces: Latent Representations from Ratings

There are several (somewhat similar) techniques for building latent semantic representations based on rating data, which mostly differ with respect to the chosen basic assumptions and decomposition algorithms. Our Perceptual Spaces introduced in [9] and [6] rely on a factor model using the following assumptions:

Perceptual Spaces use the established assumption that item ratings in the Social Web are a result of a user's preferences with respect to an item's attributes [10]. Using movies as an example, a given user might have a bias towards furious action; therefore, she will see movies featuring good action in a slightly more positive light than the average user who cares less for action. The sum of all these likes and dislikes, combined with a user's general rating bias and the consensual quality of a movie will lead to the user's overall perception of that movie, and will therefore ultimately determine how she rates it on a social movie site. The challenge of perceptual spaces is to reverse a user's rating process: For each item which was rated, commented, or discussed by a large number of users, we approximate the actual characteristics (i.e., the systematic bias) which led to each user's opinion as numeric features. This process usually only works if there is a huge number of ratings available, with each user rating many items and each item being rated by many users (this does of course also apply to all other techniques using user-item ratings for latent representations or recommendations). The Perceptual Space is then a consensual view of the item's properties from the perspective of the average user, and one can claim that it therefore captures the "essence" of all user feedback. A similar reasoning is also successfully used by other latent, e.g. [5, 11].

As our experiments in [9] showed, quality of perceptual spaces increase with the involvement and activity of users: rating data obtained from a restaurant data set (where users in a large "lazy" community rated only few restaurants each) produced worse results than using more active Netflix users. Very strong results could be achieved using an enthusiast community focused on discussing board games, here an even smaller group of highly active users rate a huge collection of board games each.

In the experiments presented in section 3, we rely on the dataset released during the Netflix Prize challenge [4] in 2005 (as an alternative, the MovieLens dataset [12] could be used which contains fewer ratings for a larger number of more recent movies). The Netflix dataset is still one of the largest user-item-rating datasets available to the research community. This fact is also the central problem limiting the value of rating-based approaches. While large web companies like Amazon, Google, or Netflix have large user-item rating datasets available in-house, these datasets are usually neither accessible nor shared. One reason for this problem is that it is very hard to foster a community active and large enough to reliably provide a huge number of ratings, and therefore companies which were able to overcome these challenges often consider their rating data as valuable business assets which are kept protected. Furthermore, user ratings can be problematic from a privacy perspective: as shown by [13], this type of rating data can be de-anonymized surprisingly well, opening legal concerns for sharing rating datasets. In fact, there have indeed been problems with bad publicity and legal issues with respect to de-anonymizing the Netflix dataset after its release, e.g., [14]. But even in a closed in-house environment, the possibility of de-anonymization and user profiling fosters discomfort in the user base – and users are less motivated to actively contribute if they feel that their privacy could be compromised [15]. In contrast, reviews are clearly public, so users are not surprised (and angry) by the fact that somebody reverse-engineered their seemingly anonymous rating. Furthermore, if datasets are shared, all references to actual users can be fully removed (in rating data sets, user ids can only be obfuscated, but not removed entirely. Reversing this obfuscation is the core of de-anonymization techniques.)

2.2 Neural Word Embeddings: Latent Representations from Reviews

In the last section, we argued that obtaining the rating data required for building latent representations of items can be very challenging. Therefore, in this section we introduce latent representations based on reviews. A good review dataset is significantly easier to create and share: it is acceptable if users have only a brief period of activity (e.g., writing only few reviews and then turn inactive again) as long as there are enough reviews overall (in contrast, rating-based approaches usually can only consider ratings from users who have rated many different items). Furthermore, reviews can be anonymized effectively.

Using reviews to build semantic representations seems to be an alluring idea: in a good review, a user will take the time to briefly summarize the content of an item, and then expresses her feelings and opinions towards it. Transforming the essence of a large number of reviews into a latent representation of items promises to be a semantically valuable alternative to ratings. In this paper, we will discuss latent fixed-length vector representations for this task. These approaches have the advantage that they are particularly easy to use in machine learning algorithms, and can naïvely be utilized to measure similarity between items which benefits explorative queries. As an alternative route, one could also use opinion mining techniques which use additional natural language analysis techniques to explicitly extract features and opinions from texts (see [16]). Here, the core challenge lies in how to match the extracted features into a uniform representation, which is a problem we will investigate in a later work.

In the following, we discuss three different fixed-length approaches for creating latent item representations from reviews. They share a similar core workflow: a) first, we represent a single review as a fixed-length feature vector, and then we b) combine all review vectors of a given item into a single latent item representation. In this work, we use the centroid vector of all review vectors for combining.

The simplest but due to its surprising efficiency and accuracy still very popular technique for representing a given text (e.g., a review) as a fixed length vector is the bag-of-words model (BOW) [17]. In its basic version, the BOW model counts the number of occurrences of each term/word in a document, and each document in a given collection is represented by a word count vector with all words in the whole collection (the vocabulary) as dimensions. It is an accepted assumption that most machine learning tasks work better when term weightings are used instead of simple word counts. We therefore use the popular TF-IDF weighting scheme [18], in which words commonly appearing in documents have generally a lower weight than specific words appearing only in few documents. As a preprocessing step, we also remove all stop words (i.e., words without any particular semantics for the purpose of latent representation).

As a result, most of these TF-IDF document vectors will be very sparse as each document only covers a fraction of vocabulary words. In many real life tasks, dense vector representations have been shown to achieve better results (as, e.g., in [19] for semantic clustering). The core idea of many dense vector representations is to apply dimension reduction techniques to the matrix of all document vectors M, reducing it to its most dominant (latent) dimensions (usually around 100 to 600 dimensions). This process usually relies on matrix decomposition, i.e. the matrix M is decomposed into at two matrixes with a significantly reduced number of latent dimensions such that their product approximates M as closely as possible. This can be achieved by approaches like principal component analysis (PCA) [20], or latent semantic analysis (LSA) [21]. In our evaluations, we will apply LSA to the TF-IDF representation.

In the last few years there has been a surge of approaches proposing to build dense word vectors not by using matrix factorization, but by using neural language models which have the training of a neural network at their core. Early neural language models were designed to predict the next word given a sequence of initial words of a sentence [22] (as for example used in text input auto-completion) or to predict a nearby word given a cue word [23]. While neural language models can be designed for different tasks and trained with a variety of techniques, most share the trait that they internally create a dense vector representation of *words* (note: not documents!). This representation is often referred to "neural word embeddings". The usefulness of these embedding may vary with respect to the chosen tasks, but it has been shown that they have surprising (and hard to explain) properties when it comes to modelling the semantics and perceived similarities of words (like being able to represent rhetoric analogies [25]). The common process of training a neural language model is to learn real-valued embeddings for words in a predefined vocabulary. In each training step, a score for the current training example is computed based on the embeddings in their current state. This score is compared to the model's objective function, and then the error is propagated back to update both the model and the weights. At the end of this process, the embeddings should encode information that enables the model to optimally satisfy its objective [26].

Early approaches like [22] used rather slow multi-layer neural networks, but current approaches adopted a significantly simpler technique using non-linear hidden layer neural networks (like the popular skip-gram negative sampling approach (SGNS) [23, 27]). These models are trained using 'windows' extracted from a natural language corpus (i.e. an unordered set of words which occur nearby in a text sequence in the corpus). The model is trained to predict, given a single word from the vocabulary, those words which will likely occur nearby it (i.e. share the same windows).

Most neural language models focus on vector representations of single words. In order to represent a whole review as a latent vector, we will use a novel neural document embedding technique described in [28], a multi-word extension of the skip-n-gram model introduced in [23]. This technique brings several unique new features. In contrast to BOW or simply applying LSA, neural document embeddings have a sense of similarity between different words occurring in a text: e.g., in BOW-models words like "funny", "amusing", and "horrifying" are treated as equidistant in their semantics, while in reality "funny" and "amusing" are semantically very close. Another new feature of document embeddings is that they will consider the order of words in texts, i.e. all BOW-based approaches and also simple aggregates of neural word embeddings will create the same latent representation of a document regardless of the word order. In [28], it has been shown that these new features result in an tremendous increase of result quality for different semantic tasks like sentiment analysis, classification, or information retrieval. Therefore, we assume that using document embeddings will also result in an increase of quality when creating latent item representations in a Big Data environment.

3 Evaluation

In the following, we evaluate different review-based embeddings in comparison with our rating-based perceptual space [9] as a baseline. As the rating data for building a perceptual space can be hard to get, the following experiments investigate how well latent representation of movies mined from reviews can be used as replacements for the perceptual space. Our perceptual space is built from the Netflix dataset [4] which consists of 103M ratings provided by 480k users on 17k video and movie titles (all titles from 2005 and older). We filtered out all TV series and retained only full feature movies for our evaluation, leaving 11,976 movies. The initial construction of the 100-dimensional space took slightly below 2 hours on a standard notebook computer.

For the latent representations built from reviews, we used the Amazon Review dataset introduced in [29]. The full review dataset consists of 143.7 million reviews from May 1996 up to July 2014, covering all items sold at Amazon.com. We only used the "Movies and TV" category, leaving 64,835 products and 294,333 reviews. We applied each of the three review-based techniques described in section 2.2 to this corpus (the simple TF-IDF model, a standard LSA model using the previous TF-IDF with varying dimensionality, and a neural document embedding model [28] also with varying dimensionality). After the model generation, for the final experiments, we dropped all items with less than 20 reviews, and all reviews which have less than 2,000 characters (or 500 alternatively). (a similar cleaning procedure was also applied by Netflix to rating-based dataset, dropping items with very few ratings). Finally, we consider only movies which are in the Netflix dataset and which we also could reliably match to the Amazon review dataset by exact title matches. For many titles this is not easily possible as there is no uniform naming scheme across datasets (e.g., "Terminator 2: Judgement Day" vs. "Terminator 2: Ultimate Edition" - both referring to the same movie), and overall data is very dirty and ambiguous. To a certain extent, a better matching would have to rely on manually comparing the movie cover arts as this is the only meta-data available in both datasets besides the (ambiguous) title name. This finally leaves us with 3,284 movies, and each of these movies has an average of 8.58 reviews. With respect to computation time (using a standard notebook computer), building the BOW model took us 67 minutes, the LSA model 28 hours, and the neural embeddings take roughly 2 hours.

We consider this paper as a work-in-progress report of our current research efforts, and in the following, we will focus on the performance of the aforementioned approaches with respect to similarity computation, i.e. we will evaluate if the similarity measured between all pairs of movies in one of the review-based models correlates (using Spearman rank correlation) with the measured similarity of the same movies in the perceptual space. On one hand, we choose this evaluation design because all four different latent representation will likely choose different dimensions (including different dimensionalities), so vectors cannot be compared directly. On the other hand, we do not require the vector spaces to be equivalent as we only need them to behave comparably in application, and for most applications being able to measure item similarity is the only required feature for, e.g., explorative queries and cluster analysis. An in-depth evaluation of other aspects will follow at a later stage.

All in all, the measured correlations paint a discouraging picture at first: the correlation coefficient of the different techniques is rather low varying between 0.05 (BOW) and 0.30 (Neural Embedding). Considering the really low performance of BOW, the neural embeddings fared surprisingly well, as shown in figure 2. In figure 3, we show the correlation for neural embeddings with varying dimensions, and a minimal text length of 2,000 characters (as used in the experiments in figure 2), and a smaller minimal text length of 500 characters. Interestingly, the correlation increases when less dimensions are chosen. This is likely due to the fact that the neural model has a higher degree of abstraction with a lower number of dimensions. Also, as expected, result correlation decreases when a smaller minimum text length is chosen.

4 Issues and Outlook

As we have shown in the last section, similarity measured using latent representations built from reviews do not convincingly correlate with similarity in perceptual spaces. However, it is unclear what this low correlation means for practical applications: To our current knowledge, there is no study which examined how well rating-based representations like our perceptual space approximate real user perceptions from a psychological perspective in a quantitative way. We still used the perceptual space as a baseline because of the popularity of such item-rating based approaches, and their effectiveness has been shown in actual systems on many occasions (i.e. it is unclear in how far rating-based approaches are indeed "correct" and "complete"; however, they "work well", e.g. see [6]). Now, it could be possible that review-based representations are still semantically meaningful, but simply focus on different aspects of items: i.e. for ratings, people simply provide an overall judgement while in reviews, often certain dominant aspects are highlighted and discussed. This should indeed lead to different but still correct similarity semantics. In order to shed light on this problem, we would need to perform a study with a large group of users focusing on the performance and correctness of systems using either representation, which definitely will be a part of a later research

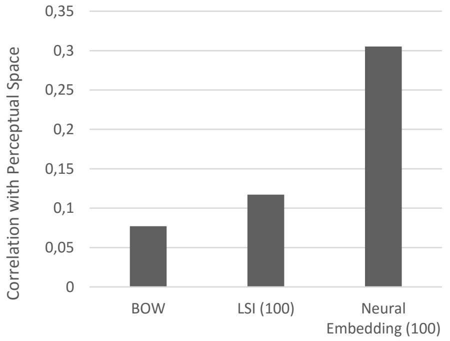


Figure 2: Correlation between review-based techniques and Perceptual Space
(number of dimensions in parenthesis)

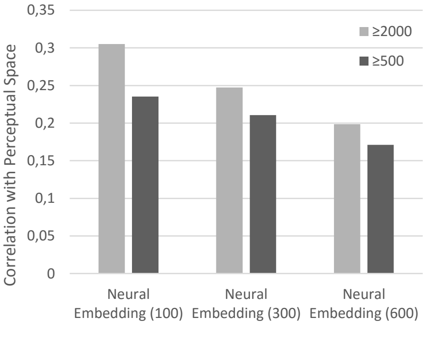


Figure 3: Neural Word Embedding with varying number of dimensions and minimal text length
(number of dimensions in parenthesis)

work. In any case, the workflow described in this paper leaves room for improvement (like introducing proper data cleaning), and we will discuss such issues below.

Data Cleaning and Review Quality: We manually inspected a selection of movies and their supposedly most similar titles as suggested by the neural document embedding technique. Here, it turned out that there are indeed some good matches in the similarity list, both confirmed by the perceptual space and the authors. However, there are also some random-looking titles suggested (e.g., for the movie "Terminator 2", both "Robocop" (a good match) and "Dream Girls Private Screenings" (a surprisingly bad match). The reason for this irritating behavior seems to be that there are many "bad" reviews (as for example most of the reviews of the second movie). "Bad" reviews are not discussing the movie itself, but other issues and do therefore not contribute to a meaningful representation. Typical examples are "I had to wait 5 weeks for delivery of the item! Stupid Amazon!", "Srsly?! Package damaged on delivery?", "I ordered the DVD version, got the Blue Ray!". For "Dream Girls", reviewers seem to be mostly concerned with the bad quality of the DVD version in comparison to the older VHS release. A similar thing happens in several reviews of the original DVD release of Terminator 2. Therefore, a next step would be excluding all reviews which do not discuss the content of the movie per se, but have other topics (e.g., print quality, delivery time, quality of customer service, etc.). However, this task is not trivial. It could be realized by training a machine classifier detecting topics, or by generic topic-modelling techniques like LDA [30]. This quality problem does not occur with our rating data, as in Netflix, it was made clear that users are supposed to rate only the content of a movie.

Overall, it seems that Amazon reviews are of rather mediocre detail and quality. In contrast, there are some online enthusiast communities like the aforementioned board game community which mostly consists of highly motivated members. There, user reviews are usually quite detailed and verbose, and it could be that our approach will yield significantly stronger results in that scenario. Another factor to consider is how we train our neural embedding models: we only used the Amazon review corpus for training. However, it is quite possible (and likely) that overall accuracy could be increased by also incorporating common knowledge corpora like the popular Wikipedia or Google News dumps [23] in the training process as this should result in better word sense semantics, which in turn should also benefit the representation of a whole review.

Additionally, we experimented only with one techniques for combining review vectors. Instead of simply computing an average vector, a weighted combination could improve quality considerably. In this sense, we also tried to combine reviews before computing the vector representations. However, this approach has prohibitive runtimes for training the document embedding, and we therefore stopped investigating it further.

Latent Representations and Explicit Properties: While latent representations of items can be used in a variety of machine learning tasks and can also be used for example-based user queries, we have no explicit real-world interpretation of the semantics of the different latent attributes. In [9], we have shown that certain perceived properties (like the degree of funniness) can be made explicit with only minimal human input using crowdsourcing-based machine regression. The core idea is that by providing few examples of items strongly exhibiting an interesting trait, and a few items which do not exhibit that trait at all, this trait can be approximated also for all other items as long as the trait is somewhere covered in a combination of attributes of our latent representation. In our future work, we will focus on the challenge of how to find interesting traits automatically and how to minimize the required human input, which could either rely on opinion mining [16] or user-generated item tags [31].

5 Summary

In this paper, we discussed building dense sematic vector representations of database items from user-provided judgements. These representations can be used in many different applications like semantic queries and a multitude of data analytics tasks. Especially, we focused on neural language models, a new emerging technique from the natural language processing community which has shown impressive semantic performance for a wide variety of language processing problems. We compared how these review-based approaches compare to an established state-of-the-art rating-based technique using similarity measurements as a benchmark. Unfortunately, the results are less conclusive than we hoped for. Basically, measurements based on neural document embeddings correlate only weakly with those from our baseline. However, a brief qualitative inspection into the results reveal some obvious shortcomings of our current approach which will be fixed in future works. Especially, a central problem of review-based approaches in general seems to be low review quality, i.e. many reviews are off-topic or simply uninformative. Categorizing, filtering and weighting different reviews, among some other optimizations, should yield significantly better results in future works. In general, we can conclude that it is significantly more challenging to extract latent semantic attributes from reviews than from ratings, and therefore this challenge requires additional study.

References

1. Liu, Q. Ben, Karahanna, E., Watson, R.T.: Unveiling user-generated content: Designing websites to best present customer reviews. Bus. Horiz. 54, 231–240 (2011).
2. Chang, F., Dean, J., Ghemawat, S., Hsieh, W.C., Wallach, D.A., Burrows, M., Chandra, T., Fikes, A., Gruber, R.E.: Bigtable: A Distributed Storage System for Structured Data. ACM Trans. Comput. Syst. 26, (2008).
3. Linden, G., Smith, B., York, J.: Amazon.com recommendations: item-to-item collaborative filtering. IEEE Internet Comput. 7, 76–80 (2003).
4. Bell, R.M., Koren, Y., Volinsky, C.: All together now: A perspective on the Netflix Price. CHANCE. 23, 24–24 (2010).
5. Koren, Y., Bell, R.: Advances in Collaborative Filtering. Recommender Systems Handbook. pp. 145–186 (2011).
6. Lofi, C., Nieke, C.: Exploiting Perceptual Similarity: Privacy-Preserving Cooperative Query Personalization. Int. Conf. on Web Information System Engineering (WISE). , Thessaloniki, Greece (2014).
7. John, J.: Pandora and the music genome project. Sci. Comput. 23, 40–41 (2006).
8. Jinni Movie Genome, http://www.jinni.com/discovery/.

9. Selke, J., Lofi, C., Balke, W.-T.: Pushing the Boundaries of Crowd-Enabled Databases with Query-Driven Schema Expansion. Proc. VLDB. 5, 538–549 (2012).
10. Kahneman, D., Tversky, A.: Psychology of Preferences. Sci. Am. 246, 160–173 (1982).
11. Hofmann, T.: Latent semantic models for collaborative filtering. ACM Trans. Inf. Syst. 22, 89–115 (2004).
12. Miller, B.N., Albert, I., Lam, S.K., Konstan, J.A., Riedl, J.: MovieLens unplugged: experiences with an occasionally connected recommender system. Int. Conf. on Intelligent User Interfaces (IUF). , Miami, USA (2003).
13. Narayanan, A., Shmatikov, V.: Robust De-anonymization of Large Sparse Dataset. IEEE Symposium on Security and Privacy. , Oakland, USA (2008).
14. Soghoian, C.: AOL, Netflix and the end of open access to research data, http://www.cnet.com/news/aol-netflix-and-the-end-of-open-access-to-research-data/.
15. Knijnenburg, B.P., Willemsen, M.C., Gantner, Z., Soncu, H., Newell, C.: Explaining the user experience of recommender systems. UMUAI. 22, 441–504 (2012).
16. Liu, B., Zhang, L.: A Survey of Opinion Mining and Sentiment Analysis. Mining Text Data. pp. 415–463 (2012).
17. Harris, Z.: Distributional Structure. Word. 10, 146–162 (1954).
18. Robertson, S.E., Jones, K.S.: Relevance weighting of search terms. Document retrieval systems. pp. 143–160 (1988).
19. Zhanga, W., Yoshidab, T., Tang, X.: A comparative study of TF*IDF, LSI and multi-words for text classification. Expert Syst. Appl. 38, 2758–2765 (2011).
20. Wold, S.: Pattern recognition by means of disjoint principal components models. Pattern Recognit. 8, 127–139 (1976).
21. Dumais, S.T.: Latent semantic analysis. Annu. Rev. Inf. Sci. Technol. 38, 188–230 (2004).
22. Mnih, A., Hinton, G.E.: A scalable hierarchical distributed language model. Adv. Neural Inf. Process. Syst. 1081–1088 (2009).
23. Mikolov, T., Yih, W., Zweig, G.: Linguistic Regularities in Continuous Space Word Representations. Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language (NAACL-HLT). , Atlanta, USA (2013).
24. Cho, K., van Merrienboer, B., Gulcehre, C., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. Empirical Methods in Natural Language Processing (EMNLP). , Doha, Qatar (2014).
25. Levy, O., Goldberg, Y.: Neural Word Embedding as Implicit Matrix Factorization. Conf of the Neural Information Processing Foundation (NIPS). , Quebec, Canada (2014).
26. Hill, F., Cho, K., Jean, S., Devin, C., Bengio, Y.: Not All Neural Embeddings are Born Equal. NIPS Workshop on Learning Semantics. , Montreal, Canada (2014).
27. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. Conf. on Empirical Methods on Natural Language Processing (EMNLP). , Doha, Qatar (2014).
28. Le, Q., Mikolov, T.: Distributed Representations of Sentences and Documents. Int. Conf. on Machine Learning (ICML). pp. 1188–1196 (2014).
29. McAuley, J., Targett, C., Shi, J., Hengel, A. van den: Image-based recommendations on styles and substitutes. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR). , Santiago de Chile, Chile (2015).
30. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. Mach. Learn. Res. 3, 993–1022 (2003).
31. Sen, S., Harper, F.M., LaPitz, A., Riedl, J.: The quest for quality tags. ACM Conf. on Supporting Group Work. , Sanibel Island, Florida, USA (2007).