Development of Precedents Searching Methods Based on Decision Trees

I I Astahova[1], **M V Fomina**[1] **and V N Shcherbakova**[1]

[1] Department of Computing Engineering, Moscow Power Engineering Institute, Moscow, Krasnokazarmennaya street, 14, Russia

Abstract. The problem of increasing the effectiveness of decision-making on the basis of precedents in intelligent systems is considered. The method of preliminary case base clustering and the subsequent construction of the decision trees set for the obtained clusters is proposed. The experimental results of the proposed algorithms application are given.

1. Introduction

The modern development of intelligent systems is closely connected with the development of intelligent (expert) decision support systems (IDSS), in particular, IDSS of real-time (IDSS RT), oriented to open and dynamic subject areas [1, 2]. The presence of such reasoning modeling methods (inductive, plausible, argumentation, and those based on analogies and cases) in IDSS RT designed for monitoring and management of complex objects (systems) and various processes allows to diagnose problem situations and assists decision making persons (DMPs) in finding effective managing actions aimed to normalize the situation. One of the methods of decision making in expert systems is the reasoning based on precedents (further called use case base). This is an approach based on the use and adaptation of a solution to an already known problem, to find a solution to a new, unknown problem. Thus, the accumulated experience of solving similar problems is used in the search for solutions to new problems.

The success of intelligent systems working based on precedents is related to the representativeness of the use case base and the availability of convenient means of finding analogues among the available examples. The paper proposes to investigate methods for improving the use of precedents by pre-structuring the database of precedents.

2. Problem statement

The main ways of presenting precedents can be divided into the following groups:

- parametric;
- object-oriented;
- special (in the form of trees, graphs, logical formulas, etc.).

In most cases, a simple parametric representation is sufficient to represent precedents, i.e. the presentation of a precedent in the form of a set of parameters with specific values and a decision that contains the diagnosis and recommendations for the decision-maker:

$$\mathrm{CASE}(x_1,\ldots, x_n, R),$$

where $x_1,\ldots,x_n$ – the parameters of the situation describing this precedent; $x_1 \in DOM(x_1),\ldots, x_n \in DOM(x_n)$, n – number of parameters describing the precedent, $DOM(x_1),\ldots, DOM(x_n)$ - the range of admissible values of the corresponding parameters, R – decision and recommendations for the decision-maker. Additionally, there may be a description of the result of applying the solution found and additional comments [2].

The choice of a particular solution from the use case base (CB) is performed according to the following rule: among the set of precedents stored in the use case base, an example is sought that is closest to the one presented. A measure of proximity is the distance between parameters of vectors $<x_1, ..., x_n>$ of the precedent stored in the CB and the set case for which a solution is to be found. The disadvantage of this method is that it is possible to determine the most accurately the precedent nearest to the presented one on the basis of comparing the vectors of numerical parameters. In the case when not only quantitative but also qualitative characteristics are used to describe the situation, one of the approaches to finding a solution can be to use a decision tree construction algorithm, which is one of the most successful in generalizing problems [3].

3. Algorithms for building decision trees

The most known methods of building decision trees are methods and algorithms ID3, C4.5 [4, 5]. The ID3 algorithm (Induction of Decision Trees, developed by R. Quinlan) forms the decision tree based on examples presented in the learning sample. The algorithm starts working with all the learning examples in the root node of the tree. To separate a set of examples of the root node, one of the attributes is selected based on the information criterion, and for each value received by this attribute, a branch is constructed and a child node is created. Then all the examples are distributed to the child nodes according to the value of the attribute. The algorithm is recursively repeated until at the nodes only examples of one class are left, after which the nodes will be declared leafs and the partition will stop [4].

Algorithm C4.5 is an improved version of the ID3 algorithm. C4.5 works better than ID3 and has several advantages:

- numerical (continuous) attributes are introduced;
- nominal (discrete) values of a single attribute may be grouped to perform more complex checking;
- subsequent pruning after building inductive tree based on the test set allows increasing of the classification accuracy [5].

After the decision tree is built, the exam is conducted - the decision tree is used to classify new examples that are not included in the learning sample. If the number of classification errors is large, it is recommended that a further examination be conducted with a second examination. To work with a real Database, its size becomes essential. Building a decision tree can be time consuming, but when the decision tree is finally built, the classification of new of new precedents is fast.

The building of the decision tree is difficult in the case when the values of the parameters describing the precedent may be inaccurate, and sometimes absent. Inaccurate values can arise from measurement errors, out-of-date data, several repeated measurements, etc.

4. Method of precedents searching using decision trees

To solve the problem of building a decision tree, it is suggested to use the values of the parameters stored in the CB in the form of tables. When the decision tree is built, one or more precedents can be associated with each leaf of this tree. Because of the large volume of CB, the constructed tree can be cumbersome, and with its reduction, some of the information describing the precedent is inevitably lost. It is proposed to use the procedure for preliminary processing of precedents in the CB according to the following principle.

Pre-clustering. Preliminary clustering of objects is carried out on the basis of data analysis and obtaining groups of similar objects. Clustering can be performed using one of the metric algorithms, such as k-means, c-means and others [6, 7]. In this case, the result of the partitioning should be a set of clusters, commensurate with the number of solutions presented in the CB. Since the k-means algorithm splits the object space into a predetermined number of clusters, we will use a clustering algorithm, in which the division of objects into clusters (at least two) is performed iteratively, and the number of clusters can increase from iteration to iteration. When the resulting partition is stable, the algorithm stops. [8]

Construction of decision trees. Let the clusters have been built at the previous stage $M_1, M_2, \ldots, M_p$. Each cluster M_i ($i = \overline{1,p}$) is a group of compactly located objects (it is possible that for various precedents included in the cluster, various solutions R were adopted). Let K be a non-empty set of objects such that $K=K^+ \cup K^-$, where $K^+ \subseteq M_i$ and $K^- \subseteq M_j$ $i\neq j$. Decision tree $Tree_i$, constructed by algorithm C 4.5 on the basis on learning sample K, is used to find decision R when a new precedent appears, which is absent in the CB. As a result of this step, a set **TR** = {$Tree_1, Tree_2, \ldots, Tree_p$ } of decision trees will be received.

Extraction of precedents. Each solution tree from **TR**, constructed by the algorithm C4.5, is a classifier that is used for the precedents of a particular cluster. The search for a CB precedent, closest to the new situation, is performed as follows. The new situation refers to a particular cluster based on the closest proximity to its center. When a cluster is defined, a further solution is specified using the decision tree associated with the cluster. As a result, the precedents assigned to the leaf node of the decision tree will be selected.

5. The MAXMIN algorithm and its modification

At the stage of preliminary clustering, an iterative MAXMIN algorithm is used to divide CB objects into classes, this algorithm is described in detail in [8]. The initial data for the operation of the algorithm is a sample $\tilde{X}$ which contains precedents X from the CB, ($X \in \tilde{X}$, $X = < x_1, \ldots, x_n >$).
Objects of this sample should be divided into classes whose number and characteristics are unknown in advance.

The assignment of each object to one of the classes is performed on the basis of the criterion of the minimum distance from the prototype points of these classes (precedents are initially chosen as prototype points, for which various solutions R were adopted). Then in each class the object most remote from its prototype is selected. If it is removed from its prototype for a distance exceeding the threshold, such an object becomes the prototype of a new class. Note that in this algorithm the threshold distance T is not fixed, but is determined based on the average distance between all points-prototypes, that is, it is corrected during the algorithm operation. If new prototypes were created during the distribution of sampling objects by classes, the distribution process is repeated. Thus, in the MAXMIN algorithm, the partition is final, for which in each class the distance from the prototype point to all objects of this class does not exceed the final value of the threshold T.

To calculate the T - threshold distance between K prototype points we use the following formula.

$D(Z_i, Z_j)$ denotes the distance between the prototype points Z_i and Z_j . For an arbitrary number of classes K the threshold distance is considered as half the average distance between the prototype points, that is,

$$T = \frac{1}{2}\sum_{i=1}^{K-1}\sum_{j=i+1}^{K} D(Z_i,\ Z_j)\cdot\frac{1}{L} \quad \text{where } L = \frac{K(K-1)}{2}.$$

Let's note the main features of using the clustering algorithm for the search for precedents. The vector X, contains only the parameters describing the situation. Solution R is not considered at this stage. Numerical parameters describing the situation can have different nature, therefore preliminary normalization is required. For qualitative parameters, when calculating the distance between two vectors, two values are used: 0 for complete coincidence, and 1 for non-coincidence of qualitative values.

An important point is also the problem of choosing the most informative parameters (attributes) of the situation for clustering [9]. When using $x_1,\ldots,x_n$ – all the parameters of the situation that describe each precedent X, splitting CB into clusters can lead to a large set of small clusters. To search for the most important parameters, the following procedure was implemented. Algorithm C 4.5 is used to build a decision tree on a learning sample of CB, where each $X = <x_1, \ldots, x_n>$. The most significant are considered 2-3 attributes located on the upper levels of the decision tree. The partitioning of the whole

CB into clusters is performed by the MAXMIN algorithm, and only these most significant parameters are used to calculate distances.

6. Evaluation of clustering results

Computer modeling of the above methods and algorithms has been carried out. For the experiment, the data set from the UCI Repository of Machine Learning Datasets [10] was used.

Let us consider in detail an example of clustering and obtaining classification rules. We used a set of database data with information on the level of knowledge of students (trainees) in the discipline “DC electric machines.” The data set “User Knowledge Modeling” from the repository includes 258 examples, characterized by 5 attributes (parameters) and belonging to one of 4 solutions (classes):1 - very low, 2 - low, 3 - medium and 4 - high.

The result is influenced by the following attributes:

- **STG** (the degree of study time for goal object materials) - the proportion of study time spent studying materials on the discipline (the range of the given parameter [0, 1]);
- **SCG** (the degree of repetition of the number of users for goal object materials) - the fraction of the number of repetitions when studying materials on the discipline (the range of the given parameter [0, 1]);
- **STR** (The degree of study time of the user for related objects with goal object) - the proportion of study time spent for studying materials on related disciplines (the range of the given parameter [0, 1]);
- **LPR** (results of examinations in related disciplines (range of the given parameter [0, 1]);
- **PEG** (the exam performance of the user for goal objects) - the results of examinations in the discipline (the range of the given parameter [0, 1]).

The first step in the experiment was to split the learning sample into clusters in accordance with the above algorithm. The clustering, taking into account all the above attributes, resulted in an excessively large number of clusters (over 40). The analysis of the decision tree constructed with the C 4.5 algorithm on the whole training sample made it possible to distinguish two of the most powerful attributes from the five - PEG and LPR. Clustering based on these attributes has divided the training sample into 4 clusters, as shown in Figure 1

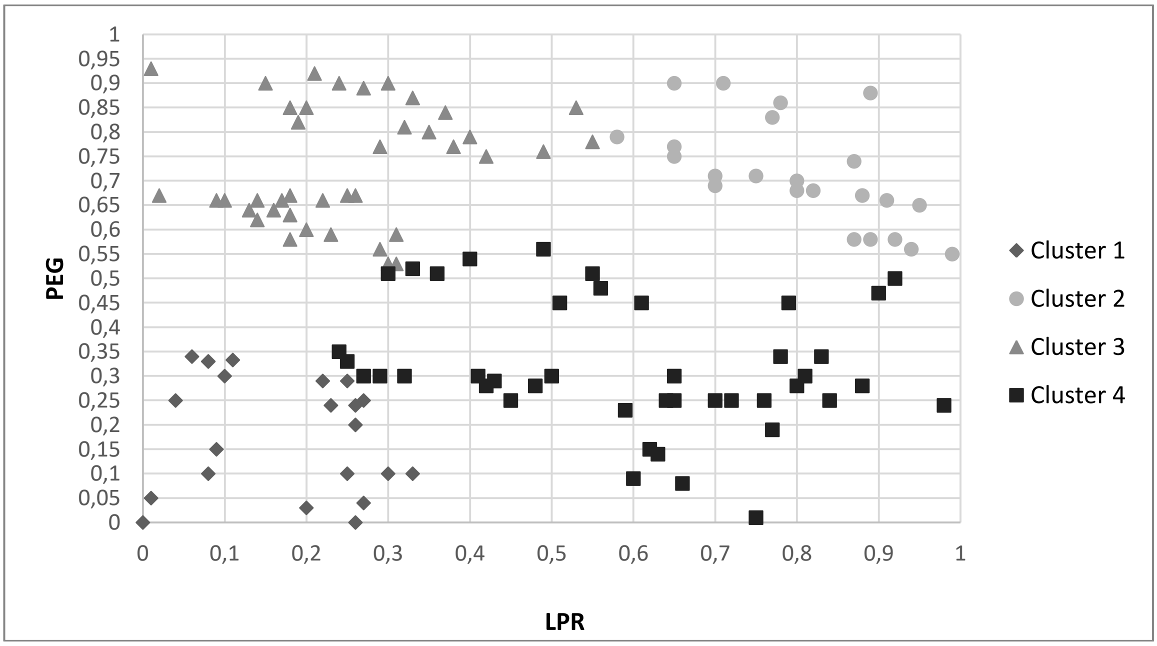

Figure 1. The results of clustering learning sample.

The second stage was the construction of decision trees using built-in clusters as training samples. Figure 2 shows how the constructed clusters correspond to the membership of the elements of the learning set to one of the classes of solutions.

For Clusters 1 and 3, we obtain the shortest decision trees (1 to 2 levels), which allows us to classify in one step. All elements of cluster 2 belong to the same class and no additional classification is required. For the elements of cluster 4, the most complex decision tree with the depth of the decision rule to 5 conditions is obtained. However, 80% of the examples from cluster 4 will be assigned to one of the classes in 2 steps.

Note that the decision tree built on the complete learning sample is significantly more complicated: there are 30 final nodes (leaves), and in most cases reaching the final node requires checking 5 conditions (5 steps).

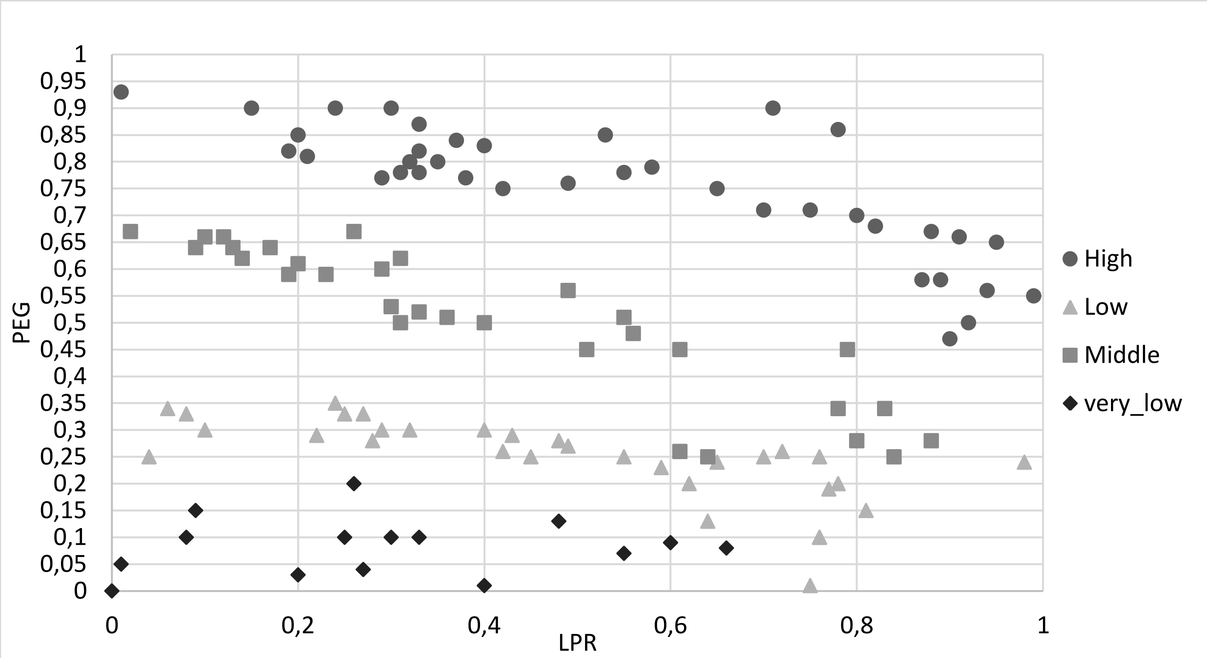

Figure 2. The decisions assigned to the elements of the learning sample.

In the next experiments, various data sets from the UCI Repository of Machine Learning Datasets were used [10].

The results of the experiment on various learning samples are presented in Tables 1 and 2. Table 1 contains the comparative characteristics of the classification models obtained for the C 4.5 algorithm and C 4.5 algorithm with prior clustering. Clustering was performed using the MAXMIN algorithm based on the two most significant attributes. The number of constructed decision trees in this case coincides with the number of clusters. When using the C 4.5 algorithm without clustering, a single decision tree is built. Trees are compared by the number of final nodes (leaf) and by the depth of the search (the number of checks needed to reach the leaf).

If we compare decision trees obtained in two models, we see that using the MAXMIN pre-clustering algorithm significantly reduces the number of checks needed to make a decision (reaching the final node in the decision tree), which will provide an accelerated search for the required precedent in the CB.

Table 2 shows the classification accuracy of test samples for two models.

Table 1. Comparative characteristics of algorithms C 4.5 and C 4.5 with pre- clustering.

	C 4.5		C 4.5 with using MAXMIN		
Data set	Num ber of final nodes	Depth of the search	Number of clusters	Number of clusters, with examples of a single class	Depth of the search
User Knowledge Modeling	10	5	4	2	5
Iris	6	4	3	2	2
Glass	36	10	6	2	2
Transfusion	6	3	5	3	3

Table 2. The accuracy of the classification of test examples (in percent) using two algorithms.

Data set	C 4.5	C 4.5 with using MAXMIN
User Knowledge Modeling	64,69	65,06
Iris	96,67	95
Glass	65,89	64,95
Transfusion	74	76

The proposed method of pre-clustering allows to obtain classification models that are in the form of a set of decision trees. In the results of Table 2, we see that the use of a classifier in the form of several decision trees allowed in some cases to improve the classification accuracy (data sets User Knowledge Modeling, Transfusion). For other data sets, a slight decrease in classification accuracy was observed.

7. Conclusion

The accelerating method of precedents searching is proposed. The method is based on preliminary clustering of precedents and constructing a system of classifiers in the form of a set of decision trees. Merging of clusters necessitates restructuring of decision trees and removing of possible contradictions in the classification rules. The results of the program modelling are presented.

8. References

[1] Eremeev A, Varshavskiy P 2008 Case-based reasoning method for real-time expert diagnostics systems *Intl J Inform. Theories & Applications* **15** (2) pp 119-25

[2] Eremeev A, Varshavsky P 2008 Reasoning by structural analogy taking into account the context for intelligent decision support systems. *Int Book Series Information Science & Computing* Number 3 (Sofia ITHEA) pp 9-16

[3] Wu X, Kumar V et al. 2007 Top 10 algorithms in data mining *J Knowledge and Inform Systems* **14** (1) pp 1-37

[4] Quinlan J 1986 Induction of Decision Trees. *Machine Learning* **1** pp 81-106

[5] Quinlan J 1996 Improved Use of Continuous Attributes in C 4.5. *J of Art. Intelligence Research* **4** pp 77-90

[6] Loohach R, Garg K 2012 Effect of distance functions on simple k-means clustering problem *Int J. of Computer Applications* **49** (6) pp 7-9
[7] Pal N, Bezdek J 1995 On cluster validity for the fuzzy c-means model *IEEE Transactions on Fuzzy Systems* **3** (3) pp 370-9
[8] Vagin V N, Golovina E Ju, Zagoryanskaya A A and Fomina M V 2008 *Exact and Plausible Inference in Intelligent Systems* Ed V N Vagin and D A Pospelov (Moscow FizMatLit) p 716 (in Russian)
[9] Zagoruiko N 2008 Methods of recognition based on the function of rival similarity *Pattern recognition and image analysis* **18** (1) pp 1-6
[10] Merz C and Murphy P 1998 UCI Repository of Machine Learning Datasets '*Information and Computer Science University of California*' [Online] http://archive.ics.uci.edu/ml/

Acknowledgments
This paper was supported by grants from the Russian Foundation for Basic Research №18-01-00201, 17-07-00442a