MODELING OF REASONING IN INTELLIGENT DECISION SUPPORT SYSTEMS BY INTEGRATION OF METHODS BASED ON CASE-BASED REASONING AND INDUCTIVE NOTIONS FORMATION

Alexander Eremeev, Marina Fomina

***Abstract**: Modeling of reasoning in intelligent systems on the example of intelligent decision support system of real time by means of integration of methods based on case-based reasoning (accumulated experience) and inductive notion formation in the presence of noisy data are considered.*

Keywords: *intelligent decision support system, real time, plausible reasoning, modeling, case-based, inductive notion formation, noisy data.*

Introduction

Modern research and development concerning perspective intelligent (expert) decision support systems (IDSS), in particular, IDSS of real-time (IDSS RT) [Eremeev and Vagin, 2011] are directly related to the problem of the modeling plausible reasoning (so called "common sense" reasoning) [Vagin et al., 2008]. The presence of such reasoning modeling methods (inductive, abductive, fuzzy inference, 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 of problem situations and aids decision making persons (DMPs) in finding effective managing effects aimed at normalizing the situation.

In this paper, the main attention is given to methods of case–based reasoning and inductive notion formation. The last is applied to situations when a suitable case for a current situation absents in the case library and a corresponding hypothesis must be formed that could constitute a new precedent in the case of its justification.

The methods of case–based reasoning and case retrieval from a system case library (CL) for further usage are considered. The possibility of using different algorithms to retrieve cases is discussed.

Methods and generalization algorithms for searching the regularities are suggested. The problems of dealing with noisy data under searching hidden regularities and choosing control effects in IDSS RT are considered.

Features of an IDSS RT

Now very actual problem in the Artificial Intelligence area is the problem of the construction of intelligent systems, whose typical representative is an IDSS RT oriented to open subject areas and dynamic subject areas [Eremeev and Vagin, 2011].

IDSS RT systems are based on the integration of knowledge representation and knowledge operation models that are capable to adaptation, modification, and learning. Such models are oriented to specific problem areas and respective uncertainty types, what reflects the ability to develop and modify their states.

The generalized structure of an IDSS RT is shown in Fig. 1.

By realizing methods of reasoning modeling in an IDSS RT one should take into consideration the features of these systems:

- the necessity to take a decision under time constrains defined by an actually controlled process;
- the need to consider a time factor in the description of a problem situation in process of finding the solution;
- the impossibility of obtaining all the objective information necessary for decision making and in this connection the usage of subjective expert information;
- the multivariate character of search;
- the necessity of applying methods of plausible reasoning and the active participation of a DMPs in decision making;
- the presence of incomplete, fuzzy and even inconsistent data for description of situations.

The methods of case-based decision search can be used in many IDSS RT units (analyzer, problem solving unit, modeling unit, and prognosis unit) and allow to increase the effectiveness of the DMPs activity in some problematic (irregular) situations.

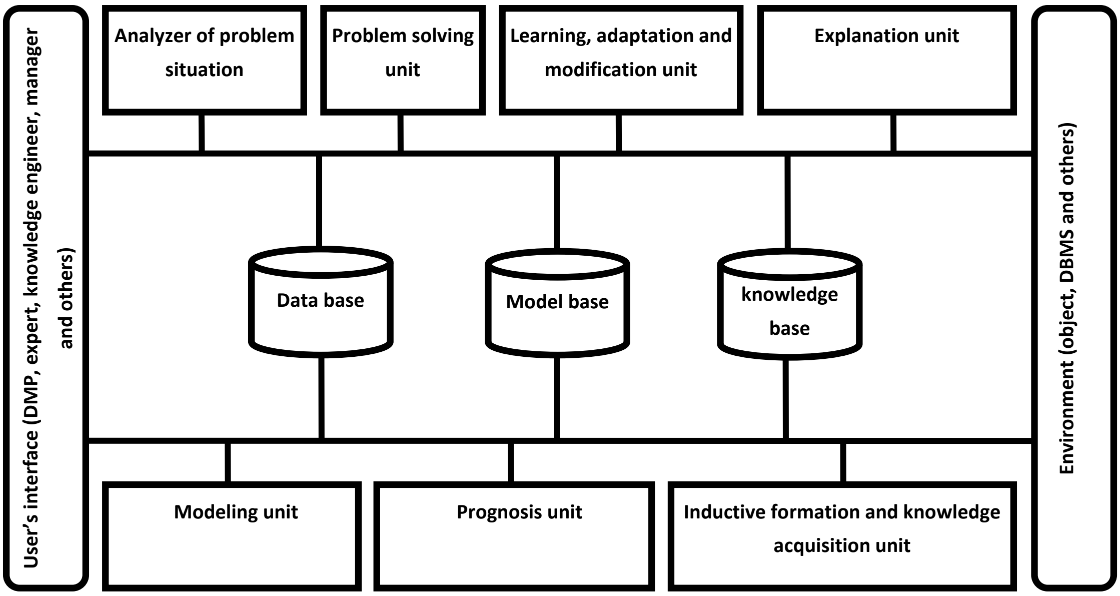

Fig. 1. General architecture of an IDSS RT

Case–Based Reasoning

A case can be defined as a particular situation that has occurred in the past and can serve as an example or justification for subsequent cases of a similar kind.

Case-Based Reasoning (CBR) is an approach that allows solving a new unknown problem using or adapting the solutions of known problems, i.e., using the experience gained in solving similar problems.

By search the solution in IDSS is reasonable to apply the plausible inference methods that allow to find an applicable solution (that is not optimal). One of the approaches is based on the fact that at the first stage of the solution search of a new unknown problem a person (expert or DMP) tries to use the decisions that were taken previously in similar cases and if necessary adapt them to the problem (the current situation).

This approach on the basis of the saved previous experience became a basis for the modeling case-based reasoning methods.

As a rule, case-based inference includes four main stages that form the so-called cycle of case-based reasoning, or CBR cycle [Aamodt and Plaza, 1994]. The CBR cycle is also called the learning cycle by precedents (examples).

The main CBR cycle stages are:

- Retrieval of the most adequate (similar) case (or cases) for the target situation from CL;
- Reusage of the retrieved case in order to try to solve the target problem;
- Revision and adaptation of the solution if it is necessary to match the target problem;
- Saving (memorizing) of a new solution as part of a new precedent.

The main goal of using the case-based tools within IDSS RT consists in giving a ready solution to an decision making person for the current situation on the basis of precedents which already took place in the past in case of control of this or similar objects.

At the first stage of CBR- cycle (under case acquisition) similarity degrees of a current situation with cases from a case library are performed and subsequent case extraction with the goal of solving a new problem situation is produced. For successful implementation of reasoning on the basis of cases, it is necessary to provide correct extraction cases from a case library.

Commonly, a case includes the following components [Alterman, 1989, David, 1991]:

- The problem description (target situation);
- The problem solution (diagnostics of the target situation and recommendation to DMP);
- The result (or prognosis) of solution application.

The result can include the list of actions to be executed, additional comments, and references to other cases. The case can have both positive and negative outcome solution application; also in some cases the choice of the proposed solution can be substantiated and possible alternatives can be given.

The main methods of case presentation can be divided into the following groups:

- Parametric;
- Object-oriented; and
- Special (graphs, trees, logic formulas, etc.).

In most cases simple parametric presentation is enough to present cases, i.e., presentation in the form of the set of parameters with particular values and solutions (diagnosis and recommendations to DMP): $CASE(x_1,\dots, x_n, R)$.

In the given description of a case, select a feature constituent $\{x_1,\dots, x_n\}$, where $x_1,\dots,x_n$ are the parameters of the situation describing the given case.

In the given case each object is characterized by n parameters (attributes): $A_1, A_2, \dots , A_n$. Attributes can accept numerical, logical or symbolic values. Denote by $Dom(A_1), Dom(A_2), \dots , Dom(A_n)$ the sets of admissible values of attributes. For attribute A_k $1 \le k \le n$, $Dom(A_k)=\{x_1, x_2, \ldots x_{q_k}\}$, where q_k is the number of different values of the attribute A_k. Thus, each situation s_i is represented as a set of attributes values, i.e., $s_i = x_{i1}, x_{i2}, \cdots, x_{in}$, where $x_{ik} \in Dom(A_k)$, $1 \le k \le q_k$. Such a description of a situation connected with case, is called a parametric description.

Commonly, a case includes also the information R i.e. the diagnosis and recommendations to the DMP. Additionally, the description of the results of the solution and additional comments can be present [Eremeev and Varshavskiy, 2008 (1), Eremeev and Varshavskiy, 2008 (2)].

Case-based inference (reasoning) is related first of all with searching the situations in CL analogues to a situation in question, their extraction, assessment and treating.

Methods of case extraction

There are many methods for case extraction and modification. Some methods are based on search cases by similarity: we need to compute measuring the similarity degree between the case and the target situation. For defining a similarity degree, it is necessary to introduce a metric in the parameter space (attributes and properties) to describe cases and the current situation. Then, consequently to chosen metric the distance between the points corresponding to the cases and the point corresponding to the target situation is determined, and a point (a case) that is the nearest to the target situation one should choose.

The approach based on forming inductive descriptions for classes of similar situations is very important. Such approach is related with solving the problem of inductive notion formation or the generalization problem.

Let us give the formulation of feature-based concept generalization. Let S be the set of all situations, represented in a certain IDSS. There is the set V of situations, in which identical or similar decisions were accepted. We can call V class of situations. All situations included in V form the set of positive objects related to some class (concept) and let W be the set of negative objects. We will consider the case where $S = V \cup W$, $V \cap W = \varnothing$. Let K be a non-empty set of objects such that $K = K^+ \cup K^-$, where $K^+ \subset V$ and $K^- \subset W$. We call K a learning sample. Based on the learning sample, it is necessary to build a rule separating positive and negative objects of a learning sample.

Thus, the class description is formed if one manages to build a decision rule which, for any example from a learning sample, indicates whether this example belongs to the class (concept) or not. The algorithms that we use form a decision in the form of rules of the type "IF condition, THEN the class description." The condition is represented in the form of a logical function in which the Boolean variables reflecting the feature values are connected by logical connectives. Further, instead of the notion "feature" we will use the notion "attribute". The decision rule is correct if, in further operation, it successfully recognizes the objects which originally did not belong to the learning sample.

Generalization algorithms

The decision tree T is a tree in which each non-final node accomplishes checking of some condition, and in case a node is finite, it gives out a decision for the element being considered. In order to perform the classification of the given example, it is necessary to start with the root node. Then, we go along the decision tree from the root to the leaves until the final node (or a leaf) is reached. In each non-final node one of the conditions is verified.

Depending on the result of verification, the corresponding branch is chosen for further movement along the tree. The solution is obtained if we reach a final node. Decision tree may be transformed into a set of production rules.

Let us consider two algorithms C4.5 and CART, which are based on a procedure of decision tree building.

The algorithm C4.5 as its predecessor ID3 suggested by J.R.Quinlan [Quinlan, 1986, Quinlan, 1996] refers to an algorithm type building the classifying rules in the form of decision trees. However, C4.5 works better than ID3 and has a number of advantages:

- Numerical (continuous) attributes are introduced;
- Nominal (discrete) values of a single attribute may be grouped to perform more effective checking;
- Subsequent shortening after inductive tree building based on using a test set for increasing a classification accuracy.

The algorithm C 4.5 is based on the following recursive procedure:

An attribute for the root edge of a tree *T* is selected, and branches for each possible values of this attribute are formed.

The tree is used for classification of learning set examples. If all examples of some leaf belong to the same class, then this leaf is marked by a name of this class.

If all leafs are marked by class names, the algorithm ends. Otherwise, an edge is marked by a name of a next attribute, and branches for each of possible values of these attribute are created, go to step 2.

The criterion for choosing a next attribute is the gain ratio based on the concept of entropy [Quinlan, 1996].

In the algorithm CART [Breiman et al., 1984], building a binary decision tree is performed. Each node of such decision tree has two descendants. At each step of building a tree, the rule that shares a set of examples from a learning sample into two subsets is assigned to a current node. In the first subset, examples are entered where a rule is performed, and the second subset includes examples where a rule does not perform. Accordingly for the current node, two descendant nodes are formed and the procedure is recursively repeated until a tree will be obtained. In this tree the examples of a single class are assigned to each final node (tree leaf).

The most difficult problem of the algorithm CART is a selection of best checking rules in tree nodes. To choose the optimal rule, there is used the assessment function of partition quality for a learning set introduced in [Breiman et al., 1984].

The important distinction of the algorithm CART from other algorithms of building the decision trees is the use the mechanism of tree cutting. The cutting procedure is necessary to obtain the tree of an optimal size with a small probability of erroneous classification.

Formed by one of generalization algorithms a decision tree can be used under finding the required cases in a CL. For such searching it is necessary to go along a decision tree from a root up to final nodes (leafs of a tree). Such path from a tree root to a final node (a leaf) corresponds to sequence of checking for attribute values describing a current problem situation. A final node corresponds to one or several cases. If a final node is related with some subset of cases then for choosing the most suitable from them, the method of “nearest neighbours” can be used. Such approach is useful for large CL because the time of decision search is significantly reduced.

Noise models

Assume that examples in a learning sample contain noise, i.e., attribute values may be missed or distorted. Reasons of noise arising are described in [Mookerjee et al., 1995]. Our purpose is to study noise effect on the functioning C 4.5 and CART algorithms.

One of basic parameters of research is a noise level. Let a learning sample K $(|K| = m)$ be represented in the table with m rows and r columns, such table has $N=m\cdot r$ of cells. Each table row corresponds to one example and each column – to certain informative attribute. A noise level is a magnitude p_0, showing that an attribute value in a learning or test set will be distorted. So, among all N cells, $N\cdot p_0$ of cells at the average will be distorted. Modeling a noise includes noise models and ways of their entering as well.

For research, two noise models were chosen: "absent values" and "distorted ones". In the first case for the given noise level with probability p_0, a known attribute value is removed from a table. The second variant of entering a noise is linked with substitution of a known attribute value for another one that may be wrong for the given example. Values for replacement are chosen from domains $Dom(A_k)$, $1 \le k \le r$, where p_0 sets up a probability of such substitution.

At entering a noise of the type "absent values", it is necessary also to select a way of treating absent values. In the paper two ways are considered: omission of such example and restoring absent values on the "nearest neighbors" method [Vagin and Fomina, 2011].

There are several ways of entering a noise in learning sets [Quinlan, 1986]. Let us consider three ways of entering a noise into a table.

Noise is entered evenly in the whole table with the same noise level for all attributes.

Noise of the given level is entered evenly in one or several explicitly indicated attributes. Entering a noise into the single table column, the content of which is the most important attribute (root node), is an extreme case here.

The new way of irregular noise entering in a table was offered. Here a noise level for each column (informative attribute) depends on a probability of passing an accidentally selected example through a tree node marked by this attribute.

We have:

- A sum noise entered into a table corresponds to the given noise level;
- All informative attributes, values of which are checked in nodes of a decision tree, are put on distortions;
- The more "important" an attribute the higher a distortion level of its values.

Principles of noise level account for the third irregular model are proposed. Let the decision tree T have been built on the basis of the learning sample K. Evidently, an accidentally selected example will passes far from through all nodes. Hence, our problem is to efficiently distribute this noise between table columns (attributes) in correspondence with statistical analysis of DBs having a given average noise level p_0.

For each attribute A_k, find a factor of the noise distribution S_k according to a probability of passing some example through the node marked A_k . Clearly, each selected example from K will pass through the root node of a decision tree. Therefore the value 1 is assigned to the factor S_k of the root attribute.

All other tree nodes which are not leaf have one ancestor and some descendants. Let one such node be marked by attribute A_i and have the ancestor marked by A_q. The edge between that nodes is marked by the attribute value x_j where $x_i \in Dom(A_q)$. Let m be the example quantity in K and m_j be the example quantity in K satisfying to the condition: attribute value for A_q is equal to x_j.

Then the factor of noise distribution

$$S_{A_i} = S_{A_q} \frac{m_j}{m}.$$

The value 0 is assigned to all factors for attributes not using in a decision tree. Introduce the norm

$$S = \sum_{i=1}^{r} S_{A_i}$$

Thus, each attribute A_i, will be undergone to influence of a noise where a noise level is

$$d_{A_i} = \frac{S_{A_i}}{S} \cdot p_0 \cdot r$$

Here p_0 is a given noise level, r is an attribute quantity.

It is easy to see that $\left(\sum d_{A_i}\right)/r = p_0$, i. e. the average noise level is the same as the given one.

Further, we consider the work of the generalization algorithm in the presence of noise in original data. Our purpose is to assess the classification accuracy of examples in a test sample by increasing a noise level in this sample.

Modeling the algorithms of forming generalized notions in the presence of noise

The above mentioned algorithms C 4.5 and CART have been used to research the effect of a noise on forming generalized rules and on classification accuracy of test examples. It should take into account that using the decision tree for classification of an example with absent values can lead to multivariate decisions. Therefore it is necessary to find a possibility of restoring these absent values. To restore unknown values the methods of nearest neighbours (kNN) and choice of average (MORM) are used [Vagin and Fomina, 2010].

To develop the generalization system, the instrumental environment MS Visual Studio 2008, program language C# has been used. The given environment is a shortened version MS Visual Studio. DBMS MS Access was used to store data sets.

The program IDTUV3 performs the following main functions:

- Loads the original data from DB;
- Enters different variants of noise in learning and test sets;
- Builds the classification model (a decision tree, or binary decision tree) on the basis of the learning sample;
- Forms production rules in accordance with the constructed tree;
- Recognizes (classifies) objects using a classification model;
- Statistics on classification quality is formed.

We present experiment results fulfilled on the following three data groups from the known collection of the test data sets of California University of Informatics and Computer Engineering "UCI Machine Learning Repository" [Merz and Murphy, 1998]:

1. Data of Monk's problems;
2. Repository of data of the StatLog project:

– Australian credit (Austr.credit);

3. Other data sets (from the field of biology and juridical-investigation practice).

We can make the following conclusions. A noise in DBs influences essentially on the classification accuracy and on generalization algorithms as a whole.

The noise entered into a test set has essentially larger influence on the classification accuracy than a noise entered in a learning set (on the average up to 5 – 6% at entering a noise up to 30%).

With increasing a noise level, the irregular way of entering a noise has essentially larger influence on the classification accuracy than the uniform way of entering a noise (on the average up to 3 – 4% at entering a noise up to 30%).

Under growth of a noise level, “distortion model” sometimes is able to increase the classification accuracy.

Table 1. Classification results for examples with noise (“distorted values”) by noise entering to test sample.

Data set	**Method of entering a noise**	**Classification accuracy of “noisy” examples, %**				
		No noise	**5% Noise**	**10% Noise**	**15% Noise**	**20% Noise**
MONKS1	*uniform*	82,3	83,53	82,74	83,06	78,14
	root attribute		81,63	81,12	79,73	76,71
	irregular		83,49	81,89	82,01	76,98
MONKS2	*uniform*	88,54	84,43	82,15	79,35	73,5
	root attribute		83,15	79,36	75,28	65,98
	irregular		82,71	80,82	74,68	68,12
MONKS3	*uniform*	85,44	82,35	83,78	79,2	75,89
	root attribute		82,24	80,46	79,81	70,52
	irregular		82,13	81,59	81,37	71,77
GLASS	*uniform*	70,35	68,93	67,03	62,93	59,15
	root attribute		65,34	63,71	61,26	55,74
	irregular		64,48	64,52	63,68	56,61
AUSTRALIAN CREDIT	*uniform*	83,31	82,73	80,57	73,19	69,41
	root attribute		79,34	75,61	73,33	62,01
	irregular		82,14	77,49	74,07	63,71

The method of “nearest neighbours” gives better classification accuracy in comparison with exclusion from a sample of examples with unknown values (on the average up to 8% under a noise level up to 30%).

The dependence of classification accuracy on a noise level at different variants of entering a noise is close to linear.

From three ways of entering a noise, the most influence on the classification accuracy has entering a noise in the root node.

Conclusion

The question of using the methods off modeling plausible reasoning on the basis of nontraditional logic is discussed. For modeling reasoning on the basis of cases in IDSS RT, the basic ways of representing and extracting the cases from case libraries are considered. For effective extraction of cases from a case base, methods of forming generalized descriptions of situation classes on the basis of decision trees building algorithms are used. The ways of solving the information generalization problem under the noise presence in the original data are researched. The new model of irregular noise insertion in informative attributes of a learning sample is offered. The machine experiments on research of noise influence on the work of generalization algorithms C4.5 and CART are produced. It is shown that the new modal of irregular noise insertion significantly influences on the classification accuracy of test examples and is perspective for further research.

The paper has been written with the financial support of RFBR (No. 14-01-00427, 12-01-00589a).

Bibliography

[Aamodt and Plaza, 1994] Aamodt A. and Plaza E., Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches, AI Communications, IOS Press, 1994, vol. 7, pp. 39–59.

[Alterman, 1989] Alterman R., Panel Discussion on Case Representation in: Proc. of the 2nd Workshop on Case-Based Reasoning, Pensacola Beach, Fl, US: 1989.

[Breiman et al., 1984] Breiman L., Friedman J. H., Olshen R. A., Stone C. T. «Classification and Regression Trees».— Wadsworth, Belmont, California, 1984.

[David, 1991] David B.S., Principles for Case Representation in a Case-Based Aiding System for Lesson Planning. In: Proc. of the Workshop on Case-Based Reasoning, Madison Hotel, Washington, May 8–10, 1991.

[Eremeev and Vagin, 2011] Eremeev Alexander P. and Vagin Vadim N. Common Sense Reasoning in Diagnostic Systems. In: Practice and Challenges from Current to Future, Chiang Jao (Ed.), ISBN: 978-953-307-326-2, InTech, 2011, pp. 99-120. Available from: http://www.intechopen.com/articles/show/title/common-sense-reasoning-in-diagnostic-systems

[Eremeev and Varshavskiy, 2008 (1)] Eremeev A., Varshavskiy P. Case-based reasoning method for real-time expert diagnostics systems. In: International Journal “Information Theories & Applications”, 2008, Volume 15, Number 2, pp. 119-125.

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

[Merz and Murphy, 1998] C.J.Merz, P.M.Murphy. UCI Repository of Machine Learning Datasets, (1998) Information and Computer Science University of California, Irvine, CA 92697-3425 http://archive.ics.uci.edu/ml/

[Mookerjee et al., 1995] V. Mookerjee, M. Mannino, R. Gilson: Improving the Performance Stability of Inductive Expert Systems under Input Noise. In: Information Systems Research 6(4), 1995, pp. 328-356

[Quinlan, 1986] Quinlan J. R. The effect of noise on concept learning. In Machine Learning Vol. II (Michalski R. S., Carbonell J. G. and Mitchell T. M., eds.) Chapter 6. Palo Alto, CA: Tioga, 1986

[Quinlan, 1986] Quinlan J.R.: Induction of Decision Trees. In: Machine Learning 1, 1986, pp. 81-106

[Quinlan, 1996] Quinlan J.R.: Improved Use of Continuous Attributes in C 4.5. In: Journal of Artifical Intelligence Research 4, 1996, pp. 77-90

[Vagin and Fomina, 2010] V. Vagin, M. Fomina. Methods and Algorithms of Information Generalization in Noisy Databases. In: Advances in Soft Computing. 9th Mexican Intern. Conference on AI, MICAI 2010, Pachuca, Mexico, November 8-13, 2010, Proceedings, Part II. / G. Sidorov, A.H. Aguirre, C.A.R. Garcia (Eds). Springer Verlag Berlin, 2010, pp. 44-55

[Vagin and Fomina, 2011] V.Vagin , M. Fomina. Problem of Knowledge Discovery in Noisy Databases. In: International Journal of Machine Learning and Cybernetics. Vol. 2, Number 3, 2011, pp. 135-145

[Vagin et al., 2008] Vagin, V.N., Golovina, B.Yu., Zagoryanskaya A.A., Fomina M.V. Exact and Plausible Reasoning in Intelligent Systems./Eds. Vagin, V.N. and Pospelov, D.A., Moscow; FIZMALIT, 2008 (in russian).

Authors' Information

Alexander Eremeev – Head of the Applied Mathematics Department, Professor, National Research University „Moscow Power Engineering Institute“, 111250 Krasnokazarmennaja str., 14, Moscow, Russia; e-mail: eremeev@appmat.ru

Major Fields of Scientific Research: Intelligent (expert) decision support systems

Marina Fomina – Computer Science Department, Docent, National Research University „Moscow Power Engineering Institute“, 111250 Krasnokazarmennaja str., 14 , Moscow, Russia; e-mail: m_fomina2000@mail.ru

Major Fields of Scientific Research: Inductive notion formation, Knowledge discovery in Databases.