World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:9, No:5, 2015

Application of Argumentation for Improving the Classification Accuracy in Inductive Concept Formation

Vadim Vagin, Marina Fomina, Oleg Morosin

Open Science Index, Computer and Information Engineering Vol:9, No:5, 2015 publications.waset.org/10001506.pdf

Abstract—This paper contains the description of argumentation approach for the problem of inductive concept formation. It is proposed to use argumentation, based on defeasible reasoning with justification degrees, to improve the quality of classification models, obtained by generalization algorithms. The experiment's results on both clear and noisy data are also presented.

Keywords—Argumentation, justification degrees, inductive concept formation, noise, generalization.

I. Introduction

At present, more and more attention is paid to the development of Intelligent Decision Support Systems (IDSSs) and expert systems. Application of plausible methods of inference, such as argumentation and induction, can significantly increase the potential (abilities) of these systems. Using the non classical logics was caused, first of all, by the presence of uncertainties, fuzziness, ambiguities and contradictions in data, on the basis of which is required to assess a situation and offer recommendations on possible control actions.

When looking for solutions in the IDSS, it is necessary to use inference methods that allow to find some reasonable, though perhaps not the optimal, solution.

Inductive components in IDSSs and expert systems are intended to improve the decision accuracy, i.e. to increase the number of situations in which an intelligent system is capable to offer a solution (to give a recommendation) as close to the human expert solution as possible. Such decisions can be useful in areas such as economics, medicine, technical diagnostics and so on.

Two problems arise in the inductive component of IDSSs. The first is the problem of constructing generalized descriptions of situation classes that require identical or similar management actions (this is a problem of inductive concept formation). The second problem - a problem of classifying an object or a current situation to one of the possible classes for which acceptable solutions are known (the problem of classifying objects).

In this paper we propose to consider a combination of methods and algorithms for machine learning and argumentation techniques to improve the efficiency of decision-making in the IDSSs.

The structure of the paper is as follows. In Section II we view the inductive concept formation problem. Sections II.*A* to II.*C* are devoted to the formalization of feature-based concept generalization and methods to solve this problem including C4.5 algorithm that builds classifying rules in the form of decision trees. In Section II.*D* foundations of the rough set theory are given. In Sections II.*E* and II.*F* we describe noise models for the generalization problem and influence of noise on the work of generalization algorithms. Section III gives the description of the argumentation theory in inductive concept formation. In sections III.*A* and III.B we present basic definitions of the argumentation theory and introduce justification degrees for setting the quantitative assessment of argument reliability in argumentation systems. Section III.*C* is devoted to formalization of the generalization problem in terms of argumentation. In Section IV.*A* and IV.*B* we describe the methodology of experiments and experiment results for clear data. Section IV.*C* presents experiment result for noisy learning sets. Finally, conclusions contain some questions for future research.

II. Methods of Inductive Concept Formation

There is a number of machine learning algorithms that are able to solve the problem of inductive concept formation on the basis of analyses of real data presented in the form of database tables. Thereby the machine learning algorithms based on a learning set builds classification rules that can be further used to identify a class to which an object belongs, i.e. it needs to analyze properties of object features.

A. Setting up the Problem

Knowledge discovery in databases (DBs) is closely connected with the solution of inductive concept formation problem or the generalization problem.

Let us give the formulation of feature-based concept generalization[1].

Let $O = \{o_1, o_2, ., o_N\}$ be a set of N objects that can be represented in an IDSS. Each object is characterized by K features: $a_1, a_2, ..., a_K$. Denote by $Dom(a_1), Dom(a_2), ..., Dom(a_K)$ the sets of admissible values of features where $Dom(a_k)$ contains all possible values of feature a_k, $1 \leq k \leq K$. Thus, each object $o_i \in O$, where $1 \leq i \leq N$, is represented as a set of features values,

V.N. Vagin - National Research University MPEI, Moscow, Russia. (e-mail: vagin@appmat.ru).

M.V. Fomina - National Research University MPEI, Moscow, Russia (e-mail: m_fonina2000@mail.ru)

O.L. Morosin - National Research University MPEI, Moscow, Russia (e-mail: oleg@morosin.ru).

i.e., $o_i = \{z_i^1, z_i^2, ..., z_i^K\}$, where $z_i^k \in Dom(a_k)$, $1 \leq k \leq K$. Such a description of an object is called a feature description. Quantitative, qualitative, or scaled features can be used as object features [1]. Among a set O of all objects represented in a certain IDSS let's separate a set V of positive objects related to some concept (a class) and W is a set of negative objects not concerned with this concept (a class). We will consider the case where $O = V \cup W$, $V \cap W = \varnothing$. Let a learning set $U = \{x_1, x_2, ..., x_n\}$ be a non-empty subset of objects O such that $U \subseteq V \cup W$. Based on the learning set, it is necessary to build a rule set $\mathbb{R}$ separating positive and negative objects of a learning set.

Thus, the concept was formed if one manages to build a decision rule that for any example from a learning set indicates whether this example belongs to the concept or not. The algorithms that we study form a decision in the form of rules of the type "**IF** <*condition*>, **THEN** <*the desired concept*>".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 that originally did not belong to a learning set.

In practice we deal with raw or noisy data in DBs. The presence of noise in data changes the above setting up of the generalization problem both at the stage of building decision rules and at the stage of the object classification. First of all, the original learning U is replaced by the set U' in which distorted or absent values of attributes occur with a certain probability.

B. Methods for Solving the Generalization Problem

From all methods of solving the generalization problem, we will consider only the methods of decision trees [2], [3] with a combination of forming decision rules[4], [5] on the basis of the theory of rough sets [6].

Decision trees . The decision tree is a tree in which each nonfinite 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 finite node (or a leaf) is reached. In each nonfinite 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 finite node.

Rough sets. A rough set is defined by the assignment of upper and lower boundaries of a certain set called the approximations of this set. Each element of the lower approximation is certainly an element of the set. Each element that does not belong to the upper approximation is certainly not an element of the set. The difference in the upper and lower approximations of a rough set forms the so-called boundary region. The element of the boundary region is probably (but not certainly) an element of the set. Similarly to fuzzy sets, rough sets are mathematical conception for work with fuzziness in data.

We consider these methods due to their effectiveness and flexibility and, at the same time, they are not so complex, as neural networks or genetic algorithms. It is especially important when analyzing very large data sets. Classification models obtained by rough sets are sets of decision rules, convenient for using and easy understanding. Classification models in the form of decision trees, in turn, are obvious, and could be simply transformed into a set of decision rules.

C. Induction of Decision Trees

The algorithm *C4.5* as its predecessor *ID3* suggested by J.R.Quinlan [2], [7] refers to an algorithm class 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;
- 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:

1) An attribute for the root edge of a tree is selected, and branches for each possible values of this attribute are formed.
2) 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.
3) 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 [7].

D. Rough Sets

We will consider how the rough set theory can be used to solve notion generalization problem. In Pawlak's works [6], [8] the notion of an information system has been introduced. An *information system* is understood as a pair $S = (U, A)$ where $U = \{x_1, x_2, ..., x_n\}$ is a non-empty finite set of objects named learning set or universe, and $A = \{a_1, a_2, ..., a_K\}$ is a non-empty finite set of *attributes*. Let's introduce a special attribute d that determines the concept (class) to which every object $x \in U$ belongs. A *decision table* (or *decision system*) is an information system of the form $S = (U, A \cup \{d\})$, where $d \notin A$ is a distinguished attribute called *decision* or *decision attribute*, and all $a_k \in A$ are *informative* attributes. Further we will consider the case, when $Dom(d)$ consists of only two possible values(e.g. "yes" or "no") according to whether an object belongs to a concept or not.

Let us introduce the indiscernibility or equivalence relation on the learning set $U : IND(A) \subseteq U \times U$. We will say that if for two objects x and y from U the pair $(x, y) \in IND(A)$

then x and y are indiscernible by values of attributes from A. A set of equivalence classes of the relation $IND(A)$ is denoted by $\{C_1^A, C_2^A, ..., C_m^A\}$. We can approximately define set V using attribute values by constructing of lower and upper approximations of V, designated by $\overline{A}V$ and $\underline{A}V$ respectively.

Since not all informative attributes are equally important, some of them can be excluded from a *decision table* without loss of the information contained in the table. The minimal subset of attributes $B \subseteq A$ which allows to keep the generalized decision for all objects of a learning set, is called *decision-relative reduct* of a table $S = (U, A \cup \{d\})$. Further, instead of a decision-relative reduct we will use the term a reduct [9]. Process of search of a reduct is a very important stage of any method using the rough set approach. For the search of reducts we propose the method, based on the following ideas:

- We consider as the significant attributes those attributes that are contained in the intersection of all reducts of an information system.
- It is necessary to build dynamic reducts [10], i.e. to construct the informative attribute set appearing "sufficiently often" as reducts of the original decision system. The attributes belonging to the "most" of dynamic reducts – approximated reducts – are considered as relevant. The thresholds of values for "sufficiently often" and "most" reducts should be chosen for given data.
- Introduction of the notion of attribute significance allows to express the importance of this attribute in a decision table.

The **G**eneralized **I**terative algorithm based on the **R**ough **S**et approach (GIRS) has been developed by authors [11]. In the given algorithm we combined the discretization of quantitative attributes with the search of significant attributes. Therefore the process of searching a reduct is viewed as a search of attributes belonging to an approximated reduct. An approximated reduct is a generalization of a reduct where generalization is understood as a result of forming significant attributes on the basis of the rough set approach [8], [9]. The error of a reduct approximation shows as exactly an attribute set approximates an informative attribute set. The application of approximated reduct is very useful when processing incomplete and noisy data.

E. Noise Models

Assume that examples in a learning set contain noise, i.e., attribute values may be absent or distorted. The noise arises due to following reasons: incorrect measurement of the input parameter; wrong description of parameter values by an expert; the use of damaged measurement devices; and data lost in transmitting and storing the information [12]. We study two noise models:

1) The noise connected with the absence of attribute values (we cannot receive an attribute value due to different causes). For each attribute A, a domain of admissible values may include the value "not known." Such value corresponding to the situation when the true value of an attribute has been lost, is denoted by N (Not known). Thus, some examples of a learning set U' contain a certain quantity of attributes with the values "not known".
2) The noise connected with the distortion of certain attribute values in a learning set. The true value in this case is replaced by one admissible, but wrong, value (the values are mixed) [12].

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

F. Noise Influence on the Work of Generalization Algorithms

Let a learning set with noise, U', be given; moreover, let the attributes taking both discrete and continuous values be subject to distortions.

Building a system of decision rules with examples having absent values leads to multivariant decisions. Therefore, we try to find the possibility for restoring these absent values. One of possible approach is the nearest neighbour method which was proposed for the classification of the unknown object o on the basis of consideration of several objects with the known classification nearest to it. The decision on the assignment of the object o to one or another class is made by information analysis on whether these nearest neighbours belong to one or another class. We can use even a simple count of votes to do this. The given method is implemented in the algorithm of restoring that was considered in detail in [13], [14].

The above algorithm GIRS has been used to research the effect of noise on forming generalized rules and the classification accuracy of test examples. Entering noise, especially a distortion of attribute values, in a learning set was performed on a decision attribute, what led to appearing inconsistent examples. To restore unknown values, the methods of nearest neighbours and choice of average are used. We have realized the algorithm GIRS including the RECOVERY algorithm. The RECOVERY algorithm is used at the presence of examples containing a noise of the type "absent values" [13]. Below, we present the pseudocode of the GIRS algorithm.

```
Algorithm GIRS
Given: table $S = (U, A \cup \{d\})$
Obtain: decision rules: $\mathbb{R}$
Begin
Obtain $S$
Select noise model (absent values or distorted values ), noise level, one of two noise entering types
If there are "absent values" in $S$ then use RECOVERY algorithm.
    Build A set of equivalence classes $\{C_1^A, C_2^A, ..., C_m^A\}$
    Build a lower approximation of set $V$ and an upper approximation of set $V$
    Find decision-relative reduct of table $S = (U, A \cup \{d\})$
    Find the set of conditions:
        $Pos(V)$ = lower approximation of set $V$
        $BND(V)$ = upper approximation of set $V$  lower approximation of set $V$
        $NEG(V) = \neg$(upper approximation of set $V$)
    **Build** $\mathbb{R}$ – the set of classification rules in form:
        **IF** $Pos(V)$ THEN $V$,
        **IF** $Neg(V)$ THEN $\neg V$,
        **IF** $Bnd(V)$ THEN possible $V$
**Output** decision rules $\mathbb{R}$
**end**
**End** of GIRS
```

Further in section IV we will use the GIRS algorithm in our experiments. It will be used as a basic algorithm for obtaining a decision rule set. Three strategies will be considered:

1) The application of the GIRS algorithm to the whole learning set (basic strategy).
2) Application of GIRS algorithm to two subsets of the learning set (union strategy).
3) Application of the GIRS algorithm together with argumentation (argumentation strategy)

III. Application of the GIRS Algorithm Together with the Argumentation

Now, we examine the possibility of improving the generalization quality due to argumentation methods. The above generalization algorithms build a generalized concept as a set of decision rules $\mathbb{R}$. It is known that the main criterion of the quality for a built generalized concept (i.e. decision rules $\mathbb{R}$) is a successful classification of test set of examples (examples not entering into a learning set U) by the given decision rules.

In works on machine learning, it is emphasized that it is necessary to select a learning set very carefully. However, when working with real data, a learning set often contains noise, and may simply not include all of the examples, relevant for the successful construction of a generalized concept. Data collections used in machine learning usually contain the most representative learning sets, and therefore decision rules, built on these learning sets, usually gives better results than the rules that are built on the real data.

It is proposed to use the argumentation methods for obtaining an improved set $\mathbb{R}$*, that is able to classify test examples with a greater accuracy than the original set $\mathbb{R}$.

Learning sets are used to form the concept as a set of decision rules. The quality of obtained rules depends, primarily, on the representativeness and consistency of a learning set. The basic idea is to divide the learning set of examples U into two subsets U_1 and U_2, such that $U_1 \cup U_2 = U$, $U_1 \cap U_2 = \varnothing$, and to produce separate learning on each of the subsets [15]. The subsets U_1 and U_2 will be less representative than the original learning set U, and classification models, obtained using this subsets, will produce worse results than results obtained for the original set U (see the experiment results for the union strategy in Section IV). It is proposed to use the methods of argumentation for obtaining an improved classification model, combining the results of a separate learning on U_1 and U_2. Here, we will assume that the partition method of a learning set is not deterministic, but the $|U_1| = |U_2| = |U|/2$ if $|U|$ is even, and $|U_1| = |U_2| - 1 = \lfloor |U|/2 \rfloor$ otherwise. Influence of the number and methods of partitioning on the classification quality is a topic for further research. After partitioning, the learning is conducted independently on each of this subsets. Any generalization algorithm that generates classification rules of the form "**IF** <*condition* >, **THEN** <*the desired concept*>" can be used for building separate sets $\mathbb{R}_1$ and $\mathbb{R}_2$. In particular, algorithms *CN2* [4] and *C4.5* [3] can be used. In this paper the algorithm *GIRS* [11] will be used.

Let's build the sets of decision rules $\mathbb{R}_1 = \{R1_1, R1_2, ..., R1_{m1}\}$ and $\mathbb{R}_2 = \{R2_1, R2_2, ..., R2_{m2}\}$, where $R1_i$ is the decision rule obtained on a learning subset U_1, $m1$ is the number of these rules and $R2_j$ is the decision rule obtained on a learning subset U_2, $m2$ is the number of these rules. Our goal is to build (with the help of argumentation) a set $\mathbb{R}$* that combines rules from $\mathbb{R}_1$ and $\mathbb{R}_2$, but it does not generate conflicts at the classification of the examples from a set U. The criterion of success for an obtained rule set $\mathbb{R}$* is the increase of the classification accuracy for test data sets, and the absence of conflicts in the classification of all learning examples. To construct $\mathbb{R}$*, methods of the argumentation theory will be used.

There are several formalizations of the argumentation theory. For instance, there are the abstract argumentation system, proposed by P.M Dung [16], the argumentation system of F. Lin and Y. Shoham [17], the G.A.W. Vreeswijk's system [18], systems based on defeasible reasoning [19] and others.

For the above goals, the most promising is the application of argumentation, based on defeasible reasoning proposed by J. Pollock [19]. We will briefly present the basic concepts of the argumentation theory needed further.

A. The Basic Definitions of the Argumentation Theory

Argumentation is usually considered as the process of building the assumptions with respect to some of the analyzed problem. As a rule, this process involves the detection of conflicts and search of ways to solve them. We will consider the argument as a pair consisting of a set of premises and a conclusion [19]. Such pairs are written so: p/X, where p is a conclusion, and X - a set of premises. For example, the argument $(p \to q)/\{\neg A, B\}$ means that premises $\neg A$, B leads to the conclusion $p \to q$. For arguments with the empty premise set we write only a conclusion. All interrelations between arguments will be represented on an inference graph. It is a graph that shows a way of building new arguments from already existing ones. The inference graph, as well, shows the conflicts between arguments.

Besides arguments, the *inference rules* are given. Defeasible inference rules are understood as rules that are not entirely reliable and exact. In the natural speech, such rules are formulated by expressions such as "as a rule", "usually", "normally", "likely", etc. Defeasible rules can be obtained, for example, by induction or abduction. We will write defeasible rules so: $M| => N$. Arguments, derived using such rule will be called defeasible. On the graph defeasible connections will be displayed by dashed arrows, and defeasible arguments by dual dashed ovals. The notion of a conflict is the basis of the argumentation system. We will consider two types of conflicts – *rebutting* and *undercut* [19].

Rebutting is a situation when some arguments rebut the conclusion of other arguments. In other words, the argument $Arg_1 = p_1/X_1$ rebuts the argument $Arg_2 = p_2/X_2$, when the conclusion p_1 rebuts conclusion p_2 and $X_1 \subseteq X_2$. Rebutting is a symmetrical form of attack.

Undercut is a asymmetric form of attack when one argument undercuts a link between premises and the conclusion of other argument.

To set the quantitative assessment of argument reliability in argumentation systems, the mechanism of justification degrees is applied.

B. The Basic Definitions of the Argumentation Theory

First of all it is necessary to define how this degrees could be obtained and the way they are inputted. Here we will use a numerical scale [0, 1] for assigning justification degrees [20]. We assume that 0 means that there are no points for believing the argument is valid (it means that the argument is defeated) and 1 means that the argument is totally undefeated, that there no reasons to doubt it is true. There are two types of justification degrees:

- a justification degree of initial arguments;
- a justification degree of defeasible rules.

The first type of a justification degree is assigned to each initial argument and could be considered as a degree of reliability of the source from which the given argument was derived. For instance, the forecast is that the chance of raining is 70%. So, an argument $Arg_1 : Tomorrow(raining)$ with the justification degree 0.7 could be made. Let's define a function $Jus(A)$ for a justification degree of an argument. The exact mechanisms of obtaining the justification degrees mostly depend on an applied domain. For instance, this may be statistical data or some expert assessment (e.g. the possibility that company's shares will grow is 60%).

The other type of justification degrees is related with defeasible rules of inference. As it was already mentioned above, defeasible rules are often a formalization of expert knowledge of the following type: "If A is true, than as usual B is also true". Such rules of inference can also have some numerical degree of justification. For example, the use of analginum in 85% leads to a decrease in body temperature (formally $R_1 : (\forall x)reception(Analginum, x)| => decreasing\_temperature(x))$. So we need to define function $Jus(Arg)$ that will calculate the justification degree of every argument in an inference graph. We will consider that it is known for every initial argument. The value of this function for an exact argument is influenced by two factors:

1) An inference tree (i.e. justification degrees of arguments, which were used in the inference of that argument);
2) Conflicts with other arguments.

For the sake of convenience, let's consider these two factors separately: $Jus_{anc}(Arg)$ is the inherited justification degree of ancestors and $Jus_{con}(Arg)$ is the value of influence of the conflict the argument has(i.e. how the conflict undercuts the argument justification).

$$Jus_{anc}(Arg) = min(Jus(B_1), ..., Jus(B_m) \cdot Jus(R) \quad (1)$$

In (1) $B_1, ..., B_m$ – arguments that were used in the inference of argument Arg, and $Jus(R)$ is a justification degree of a rule R that was used for inference of the argument Arg. Equation (1) is called the weakest link principle [21]. It is necessary to note, that this is not the only way of calculating justification degree, e.g. in some works, Bayesian approach is applied [22]. Note that it follows from (1) that if to perform calculation of justification degrees recursively, it is possible to search for the minimum among the direct ancestors that were used in the previous step. In the particular case, when the argument has only one ancestor, its Jus_{anc} will be the same as its ancestor degree. Contrary to obtaining Jus_{anc} (when the weakest arguments are considered), calculation of $Jus_{con}(A)$ is based on finding the conflict with the argument, that have the highest degree of justification. Let's consider A_{confl} as a set of arguments that has conflicts with the argument Arg, $m = |A_{confl}|$ and Ac_i is the member of the set A_{confl}.

$$Jus_{con}(Arg) = \begin{cases} \max\limits_{1 \le i \le n} (Jus_{anc}(Ac_i)), \\ \qquad\qquad\qquad \text{if } m{>}0; \\ 0, \qquad\qquad\qquad\qquad \text{otherwise.} \end{cases} \quad (2)$$

In (2) Jus_{anc} is used to avoid cases of mutual conflicts between two arguments. Thus, the justification degree of an argument A can be calculated as follows:

$$Jus(Arg) = \begin{cases} Jus_{anc}(Arg) - Jus_{con}(Arg), \\ \quad \text{if } Jus_{anc}(Arg) > Jus_{con}(Arg); \\ 0, \text{otherwise.} \end{cases} \quad (3)$$

Further we will show how justification degrees could be used to cope with conflicting decision rules.

C. Formalization of the Generalization Problem in Terms of Argumentation

As mentioned above, let decision rule sets $\mathbb{R}_1$ and $\mathbb{R}_2$ be built for learning sets U_1 and U_2. Rules of $\mathbb{R}_1$ and $\mathbb{R}_2$ have the form "**IF** <*condition*>, **THEN** <*the desired concept*>". For example, let's consider MONK's Problems from UCI Repository of Machine Learning Datasets (Information and Computer Science University of California) [23]. Every object in this data set has six informative attributes (further denote them A_1, A_2, A_3, A_4, A_5, A_6), and one decision attribute (further denote it as "$CLASS$"). For example, one of the classification rules for the MONK's problem can be written down as:

$$\text{IF}(A_1 = 1)\&(A_2 = 1) \text{ THEN } CLASS = 1.$$

Such rule can be considered as a defeasible inference rule for argumentation. Further, we will write decision rules, when they are used for argumentation as follows: $X| => Y$, where X are the conditions and the Y is a value of the decision attribute. So, for the above MONK's problem rule can be written so:

$$(A_1 = 1)\&(A_2 = 1)| => (CLASS = 1).$$

In addition, we assume that there are only two possible values ("$CLASS = 1$" and "$CLASS = 0$") of a decision attribute:

belonging the example to the class or not. So, for all rules in $\mathbb{R}_1$ and $\mathbb{R}_2$, the decision attribute $CLASS = 0$ can be replaced with $\neg(CLASS = 1)$. To check, if one example $o_i = z_1^i, z_2^i, ..., z_K^i$ with the decision attribute di produces a conflict on some set of rules $\mathbb{R}$, we do the following:

1) For every value of informative attributes z_j^i of the example $o_i = z_1^i, z_2^i, ..., z_K^i$ and for every attribute a_j from $A = a_1, a_2, ..., a_K$ (where $1 \leq j \leq K$, A is a set of attributes, K is the number of such attributes), we form arguments of the following structure: $Arg_j : (a_j = z_j^i)$.
2) Each argument Arg_j with the justification degree equal to 1 represents initial arguments for an argumentation system.
3) Perform the logical inference with respect to a decision rule set $\mathbb{R}$ and check, whether there are conflicts in the obtained solution.

Let's consider the example Ex_1 that belongs to $CLASS = 1$ with the following attribute values: $A_1 = 2; A_2 = 2; A_3 = 3$. Then the arguments for this example would be $Arg_1 : A_1 = 2; Arg_2 : A_2 = 2; Arg_3 : A_3 = 3$. Further, for convenience we would write down examples in the argumentation form with attributes and their values as follows:

$$Ex_i : (a_1 = z_1^i)\&(a_2 = z_2^i)\&...\&(a_K = z_K^i); CLASS = d^i.$$

We assume that the inference rules are inconsistent, if, at least, for one example from a learning set U, there are defeasible rules relating it to different values of the decision attribute. For example, let's consider two rules from $\mathbb{R}$:

$$R_1 : (A_1 = 2)\&(A_2 = 2)| => (CLASS = 1);$$

$$R_2 : (A_2 = 2)\&(A_3 = 3)| => \neg(CLASS = 1).$$

Let the example $(Ex_1 : (A_1 = 2)\&(A_2 = 2)\&(A_3 = 3); CLASS = 1)$ be in a learning set U. It will generate three arguments: $Arg_1 : A_1 = 2$, $Arg_2 : A_2 = 2$ and $Arg_3 : A_3 = 3$. These arguments will be suited for the conditions of both rules. So, both arguments: $Arg_4 : CLASS = 1$ and $Arg_5 : \neg(CLASS = 1)$ for this example can be inferred. Arguments Arg4 and Arg5 have a conflict of the type "rebutting". Inherited from ancestors, the justification degree of both arguments Arg_4 and Arg_5 are equal to 1, therefore $Jus_{con}(Arg_4) = Jus_{con}(Arg_5) = 1$. Due to (3), justification of both arguments will be 0 and the conflict among them is unsolvable. So, rules R_1 and R_2 should be considered as conflicting for the example Ex1. The inference graph for this example is shown in Fig.1. Further, we'll show, how this conflict could be solved.

Our task is to form a consistent set $\mathbb{R}$* that combines rules from both sets $\mathbb{R}_1$ and $\mathbb{R}_2$. It is proposed to use a mechanism of justification degrees for defeasible rules. The problem is to define justification degrees for all defeasible rules in such a way that all conflicts arising in a learning set becomes solvable. Let us give the learning procedure for searching such justification degrees when the conflicts are becoming solvable.

The learning procedure. Here we assume that $\mathbb{R}_1$ and $\mathbb{R}_2$ are two decision rule set, obtained independently on two learning subsets U_1 and U_2 by certain generalization algorithm. **Procedure:**

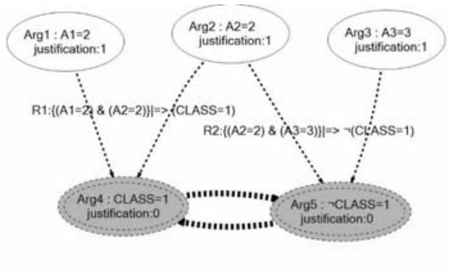


Fig. 1. Example.

1. All decision rules $R1_i \in \mathbb{R}_1$, $1 \leq i \leq |\mathbb{R}_1|$ and $R2_j \in \mathbb{R}_2$, $1 \leq j \leq |\mathbb{R}_2|$ enter into an argumentation subsystem as defeasible rules.

2. For all $R1_i$ and $R2_j$, set up the justification degree equal to 1.

3. For each example $Ex_k : Arg_1\&Arg_2, ..., Arg_n$ from a learning set U, perform the following steps:

3.1. Enter $Arg_1, Arg_2, ..., Arg_n$ as initial arguments with the justification degree equal to 1 in the input of an argumentation system and produce logical inference.

3.2. If a system discovers conflicts, i.e. in an inference graph there are two conflicting arguments $Arg*$ and $Arg**$ go to step 3.3, else go to step 3.7.

3.3. Choose the Arg+ from $Arg*$, $Arg**$ such that its conclusion is the same as the value of decision attribute d and Arg- is another one.

3.4. Obtain two rule sets $\mathbb{R}c$+ and $\mathbb{R}c$-, such that rules from $\mathbb{R}c$+ supports Arg+ and $\mathbb{R}c$- supports Arg-.

3.5. Compute an increased justification degree for all $R_j \in \mathbb{R}c$+, $1 \leq j \leq |\mathbb{R}c+|$ according to the following formula:

$$Jus(R_j) = \begin{cases} (Jus(R_j)(1 + \Delta), & \text{if } Jus(R_j)(1 + \Delta) < 1; \\ 1, & \text{otherwise.} \end{cases}$$

Δ is some factor, whose value is chosen in interval (0,1) empirically depending on the number of decision rules in $\mathbb{R}_1$ and $\mathbb{R}_2$. In the given experiment (see Section IV) the value of parameter Δ was selected equal to 0.05.

3.6. Compute a decreased justification degree for all $R_i \in \mathbb{R}$c-, $1 \leq j \leq |\mathbb{R}\text{c-}|$ according to the following formula:

$$Jus(R_j) = (1 - \Delta)Jus(R_j)$$

. **3.7.** If $k \leq |U|$ then select the next example Ex_{k+1} and go to step 3.1, else step 4.

4. Classify the examples from learning set U taking into account obtained justification degrees. In case there are still unsolvable conflicts on a learning set, go to step 2 and repeat the learning procedure. Otherwise, the learning process is finished.

After the learning process is completed, all decision rules will obtain justification degrees in the interval (0, 1].

Let's consider the learning procedure on the previous example. There are two defeasible rules:

$$R_1 : (A_1 = 2)\&(A_2 = 2)| => (CLASS = 1);$$

$$R_2 : (A_2 = 2)\&(A_3 = 3)| => \neg(CLASS = 1).$$

Let the learning set consist of tree examples:

$$(Ex_1 : (A_1 = 2)\&(A_2 = 2)\&(A_3 = 3); CLASS = 1)$$

$$(Ex_2 : (A_1 = 2)\&(A_2 = 2)\&(A_3 = 1); CLASS = 1)$$

$$(Ex_3 : (A_1 = 1)\&(A_2 = 2)\&(A_3 = 3); \neq (CLASS = 1))$$

Learning on this set will give the following results. Initially $Jus(R_1) = Jus(R_2) = 1$. On the first example Ex1 the system would find an unsolvable conflict as shown in Fig. 1. Due to steps 3.6 and 3.7 we must increase the justification degree of R_1 as it supports the right argument $(Arg4 : CLASS = 1)$ and decrease the justification degree of R_2 as it supports the wrong argument $Arg5 : \neg(CLASS = 1)$. Therefore $Jus(R_1)$ will be left equal to 1 because it is the maximum and $Jus(R_2) = Jus(R_2) - Jus(R_2)\Delta = 0.95$. Examples Ex_2 and Ex_3 don't cause conflicts, so there are no need to change justification degrees. After the learning procedure is completed, the example Ex_1 will have solvable conflict, because the justification degree (3) of $Arg4$ is higher than the justification degree of $Arg5$. The inference graph for this example after the learning procedure is shown in Fig. 2.

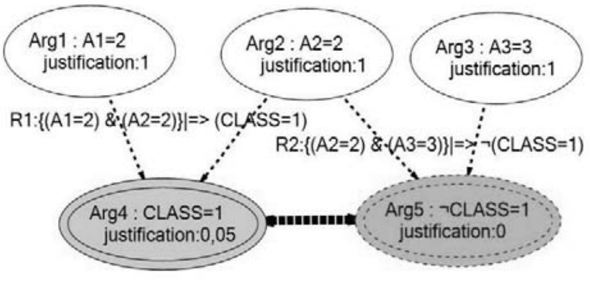


Fig. 2. Justification degrees after learning procedure. $Jus(R_1) = 1$; $Jus(R_2) = 0.95$.

IV. Exmperiments

A. The Methodology of Experiments

To assess the experiment results, three strategies (see Section II.F) were compared.

1) GIRS basic strategy. This strategy is a basic strategy, when decision rules $\mathbb{R}$ are obtained by learning with GIRS algorithm on the full learning set U, and classification of test examples is made on whole rule set $\mathbb{R}$. The dataflow diagram of this method is presented in Fig 3.

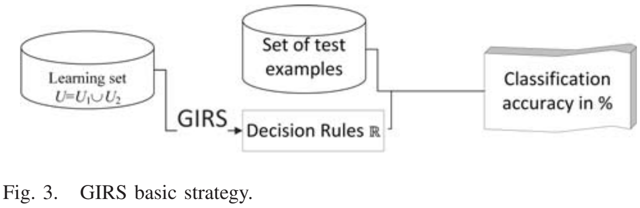


Fig. 3. GIRS basic strategy.

2) GIRS union strategy. In contrast to GIRS basic strategy, the learning set U is divided into two subsets U_1 and U_2 and learning occurs on them independently. So, two independent set of rules $\mathbb{R}_1$ and $\mathbb{R}_2$ are obtained. The resulting set $\mathbb{R}$ is obtained by simple union $\mathbb{R}_1 \cup \mathbb{R}_2$. The classification of test examples is made on this set $\mathbb{R}$. The dataflow diagram of this method is presented in Fig 4.

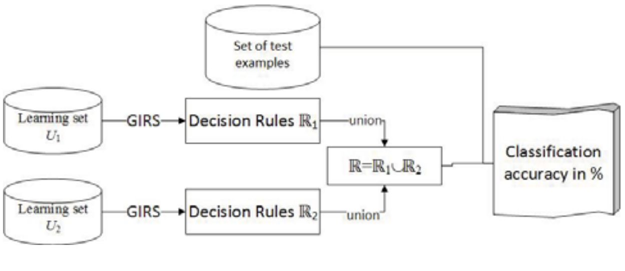


Fig. 4. GIRS union strategy.

3) GIRS argumentation strategy. As in the GIRS union strategy, here we perform GIRS algorithm on two learning subsets U_1 and U_2 and receive two independent rule sets $\mathbb{R}_1$ and $\mathbb{R}_2$. $\mathbb{R}$* is obtained using argumentation approach described in Section III. The classification of test examples is made on this set $\mathbb{R}$*. The dataflow diagram of this method is presented in Fig 5.

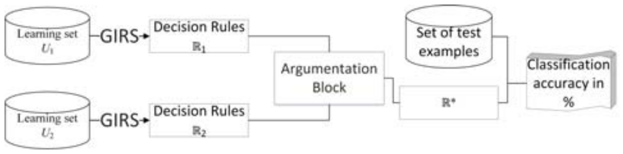


Fig. 5. GIRS argumentation strategy.

B. Experiment Results

Three strategies – GIRS basic strategy, GIRS union strategy and GIRS argumentation strategy – were applied to two problems from UCI Repository [23] – MONK's 2 and MONK's 3 problem.

Let's consider the application of the argumentation strategy for the MONK's 2 problem in detail. The learning set consisting of 169 examples was divided into two disjoint subsets consisting of 84 and 85 examples. On each learning subset, the GIRS algorithm was produced and two sets of decision rules $\mathbb{R}_1$ and $\mathbb{R}_2$ were obtained.

$\mathbb{R}_1$ contains 56 rules:
Rule#1: if$(A_5=2)\&(A_4=1)\&(A_3=1)$ then $CLASS=0$;
Rule#2: if$(A_4=1)\&(A_6=1)$ then $CLASS=0$;
Rule#3: if$(A_5=1)\&(A_3=1)$ then $CLASS=0$;
....
Rule#55 : if$(A_5 = 3)\&(A_4 = 3)\&(A_6 = 1)\&(A_1 = 2)$ then $CLASS=1$;
Rule#56: if$(A_5=4)\&(A_2=2)\&(A_1=2)$ then $CLASS=1$;
$\mathbb{R}_2$ contains 57 rules:
Rule#1 : if$(A_5 = 3)\&(A_2 = 2)\&(A_3 = 2)\&(A_6 = 2)$ then $CLASS=0$;
Rule#2: if$(A_5=4)\&(A_2=2)\&(A_6=2)$ then $CLASS=0$;
Rule#3: if$(A_5=3)\&(A_4=2)\&(A_2=2)$ then $CLASS=0$;

....

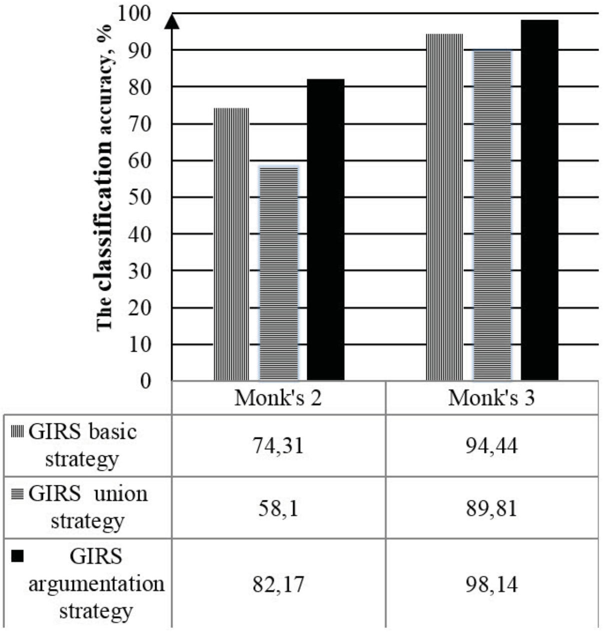
Rule#56 : if$(A_5 = 3)\&(A_4 = 3)\&(A_3 = 1)\&(A6 = 1)$ then $CLASS = 1$;

Rule#57 : if$(A_5 = 3)\&(A_4 = 1)\&(A_2 = 2)\&(A_3 = 2)\&(A_6 = 1)$ then $CLASS = 1$;

For a combined set of rules $\mathbb{R} = \mathbb{R}_1 \cup \mathbb{R}_2$ containing 113 rules, 81 conflicts were discovered (for the full learning set U from 169 examples). The learning procedure described in Section III.C was applied. As a result of learning, all conflicts found in the learning set were solved. In the GIRS union strategy, learning was produced on two subsets, each of which was less representative than the original learning set U. Learning on such subsets gives worse results of the classification accuracy(see the GIRS union strategy in Fig. 6 and Fig. 7). Nevertheless, the application of argumentation allows to build a combined classification model, that gives even better results than results obtained by the GIRS basic strategy. The GIRS argumentation strategy for the classification of the test set from 432 examples gives the following results: 77 examples were classified incorrectly and the classification accuracy was 82.17%.

For comparison, the GIRS basic strategy on the full learning set of 169 examples showed on the test set from 432 examples the classification accuracy assessed to 74.31Thus, the use of argumentation allowed to enhance the classification accuracy of classification on **7.86%** for the given example.

In Fig. 6 the results for three strategies for MONK's 2 and MONK's 3 problem are presented.



	Monk's 2	Monk's 3
▥ GIRS basic strategy	74,31	94,44
▤ GIRS union strategy	58,1	89,81
■ GIRS argumentation strategy	82,17	98,14

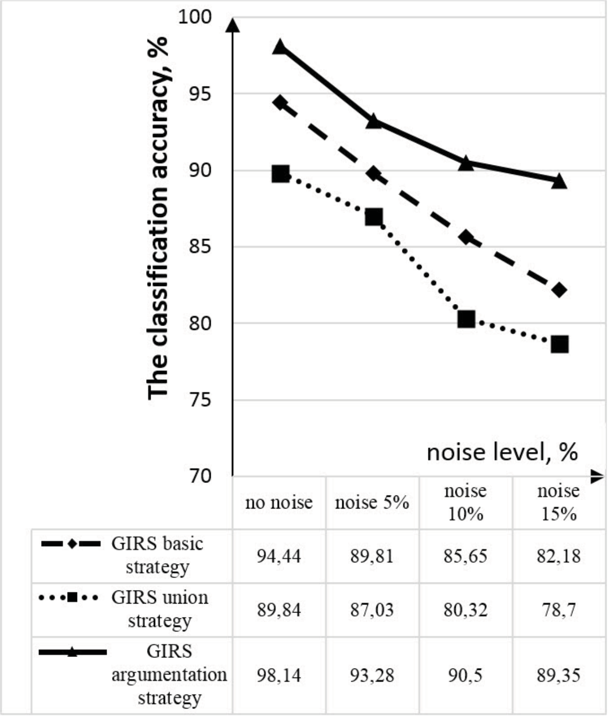
Fig. 6. The classification accuracy of the test data set for MONK's 2 and MONK's 3 problem.

C. The Experiment Results in the Presence of Noise

Adding noise to a learning set increases the number of conflicts for classification rules. Let us try to assess the possibility of applying the argumentation methods to noisy data.

The test methodology remains the same: for noisy learning sets three strategies were applied: GIRS basic strategy, GIRS union strategy and GIRS argumentation strategy.

The noise was introduced in a decision attribute of the learning set for the MONK's 3 problem. Three levels of noise were viewed: 5% (6 examples of 122 learning examples had a wrong value of the decision attribute), 10% (12 examples of 122 learning examples had a wrong value of the decision attribute), 15% (18 examples of 122 learning examples had a wrong value of the decision attribute). The results of the experiments are shown in Fig. 7



	no noise	noise 5%	noise 10%	noise 15%
GIRS basic strategy	94,44	89,81	85,65	82,18
GIRS union strategy	89,84	87,03	80,32	78,7
GIRS argumentation strategy	98,14	93,28	90,5	89,35

Fig. 7. The classification accuracy on MONK's 3 problem with noise.

V. Conclusion

The paper was devoted to the possibilities of applying the argumentation theory to enhance effectiveness of methods of inductive concept formation. The method of combining multiple sets of inductive formation rules in a conflict-free set of rules with the help of argumentation and justification degrees was proposed.

Application of argumentation methods for the generalization problem allowed to enhance the classification accuracy for test problems. Furthermore, it was analyzed the influence of noise on the classification accuracy. The use of argumentation for noisy data as well significantly improved classification results. However, for real data and situations that can occur in IDSS, learning sets used for generalization can contain noise. The presence of noise and non-representative examples in learning sets influence the quality of generalization models. Methods of argumentation allow us to decrease this influence when there are conflicts in data (in the examples), that can be resolved properly by using the calculated justification degrees. Nevertheless, several problems remain unresolved and are subject for further analysis.

Firstly, the question about effective portioning of an original learning set is left unsolvable. In this study, the learning set was divided into two equal parts randomly. It is necessary to check the influence of different ways of partitioning, as well as the number of such partitions.

Secondly, the question about the classification quality depends on the choice of the parameter $\Delta that changes a justification degree of rules.$

In general, the problem of enhancing the effectiveness of methods of inductive concept formation by means of argumentation has been successfully executed.

Acknowledgment

This paper was prepared under the financial support of the Government Task No. 2.737.2014/K, and RFBR Grant No. 14-07-00862, 15-01-05567.

References

[1] N. Vagin, E. Golovina, A. Zagoryanskaya and M. Fomina. Exact and Plausible Inference in Intelligent Systems./V. Vagin and D. Pospelov, Eds., Moscow: FizMatLit, 2008, p. 714 (in Russian).

[2] Quinlan J. R. Induction of Decision Trees. // *Machine Learning*, vol. 1, 1986, pp. 81-106.

[3] Quinlan J.R., C4.5: Programs for Machine Learning. San Francisco: Morgan Kaufmann Publishers Inc., 1993, p. 302.

[4] P. Clark and R. Boswell., Rule Induction with CN2: Some Recent Improvements. // *Machine Learning - Proceedings of the Fifth European Conference (ESWL-91)*, Berlin, Springer-Verlag, 1991, pp. 151-163.

[5] P.Clark, T.Niblett, The CN2 Induction Algorithm, // *Machine Learning* , vol. 3, 1989, pp. 261-283.

[6] Z. Pawlak, Rough sets and intelligent data analysis. // *Information Sciences*, vol. 147, no. 1, 2002, pp. 1-12.

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

[8] J. Komorowski, Z. Pawlak, L. Polkowski and A. Skowron, Rough Sets: A Tutorial, Springer-Verlag, Singapore,1999, pp. 3-98.

[9] S. Nguyen and H. Nguyen, Some efficient algorithms for rough set methods // *Proc. of Information Processing and Management of Uncertainty on Knowledge-Based Systems (IPMU-96)*, Spain, vol. III, 1996, pp. 1451-1456.

[10] Bazan J. A comparison of dynamic and non-dynamic rough set methods for extracting laws from decision tables / Rough Sets in Knowledge Discovery 1: Methodology and Applications // Polkowski L., Skowron A. (eds.), Physica-Verlag, 1998.

[11] Vagin, M. Fomina and A. Kulikov, The Problem of Object Recognition in the Presence of Noise in Original Data // *10th Scandinavian Conference on Artificial Intelligence SCAI*, 2008, pp. 60-67.

[12] Mookerjee, M. Mannino and R. Gilson, Improving the Performance Stability of Inductive Expert Systems under Input Noise, // *Information Systems Research*, vol. 6, no. 4, 1995, pp. 328-356.

[13] V. Vagin and M. Fomina, Problem of Knowledge Discovery in Noisy Databases // *International Journal of Machine Learning and Cybernetics*, vol. 2, no. 3, 2011, pp. 135-145.

[14] V. Vagin and M. Fomina, Methods and Algorithms of Information Generalization in Noisy Databases // *Advances in Soft Computing. 9th Mexican Intern. Conference on AI*, MICAI, Pachuca, 2010, pp. 44-55.

[15] Santiago Ontanon and Enric Plaza, Multiagent Inductive Learning: an Argumentation-based Approach // *Proc. ICML-2010, 27th International Conference on Machine Learning*, 2010, pp. 839-846.

[16] A. Bondarenko, P. Dung, R. Kowalski, F. Toni and A. Bondarenko, An Abstract Argumentation-Theoretic Framework for Defeasible Reasoning. // *Artificial Intelligence*, vol. 93, no. 1-2, 1997, pp. 63-101.

[17] 15. F. Lin and Y. Shoham, Argument Systems. A Uniform Basis for Nonmonotonic Reasoning, // Principles of Knowledge Representation and Reasoning, San Mateo, CA, Morgan Kaufmann, 1989, pp 245-255.

[18] G. Vreeswijk, Abstract Argumentation Systems // *Artificial Intelligence*, vol. 90, 1997, pp. 225-279.

[19] J. Pollock, Oscar – a General Purpose Defeasible Reasoner. // *Journal of Nonclassical Logics*, vol. 6, 1996, pp. 89-113.

[20] G. Betz, On Degrees of Justification. // Erkenntnis, vol. 2, 2012, pp. 237-272.

[21] Pollock J.L. Defeasible reasoning with variable degrees of justification, // *Artificial Intelligence*, vol. 133, 2001, pp. 233-282.

[22] R. Haenni, J. Kohlas and N. Lehmann, Probabilistic Argumentation Systems. // *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, vol. 5: Algorithms for Uncertainty and Defeasible Reasoning, Dordrecht, Kluwer, 1999, pp. 221-287.

[23] C. Merz and M. P., UCI Repository of Machine Learning Datasets. // Information and Computer Science University of California, 1998. (Online). Available: http://archive.ics.uci.edu/ml/. (Accessed 10.03.2014).