INFORMATICA, 2018, Vol. 29, No. 3, 467–486
© 2018 *Vilnius University*
DOI: http://dx.doi.org/10.15388/Informatica.2018.176

The Modified Method of Logical Analysis Used for Solving Classification Problems

Roman KUZMICH[1], Alena STUPINA[2,3]*, Larisa KORPACHEVA[2], Svetlana EZHEMANSKAJA[2], Irina ROUIGA[4]

[1] *Department of Computer Science in Business, School of Business Management and Economics Siberian Federal University, Krasnoyarsk, Russia*

[2] *Department of Economics and Information Technologies for Management School of Business Management and Economics, Siberian Federal University Krasnoyarsk, Russia*

[3] *Department of International Management, Krasnoyarsk State Agrarian University Krasnoyarsk, Russia*

[4] *Department of Economics and Business Process Management School of Business Management and Economics, Siberian Federal University Krasnoyarsk, Russia*

e-mail: romazmich@gmail.com, h677hm@gmail.com, korp_0777@mail.ru, sve-ta_ezh@inbox.ru, irina_rouiga@bk.ru

Received: October 2017; accepted: June 2018

Abstract. The study is dictated by the need to interpret and justify the solutions of classification problems. In this context, a method of logical analysis of data is considered along with its modifications based on the specifically developed algorithmic procedures, the use of which can increase the interpretability and generalization capability of classifiers. The article confirms in an empirical way that the suggested optimization models are suitable for building informative patterns and that the designed algorithmic procedures are efficient when used for the method of logical analysis of data.

Key words: classification, pattern, degree, coverage, information content.

1. Introduction

Working on solutions to today's classification problems is often associated with a need for interpreting and justifying the obtained solutions, apart from ensuring their high accuracy. In particular, the interpretability and justification are key factors in finding the solutions to practical problems that threaten great losses in case of a wrong decision.

The latest survey studies in this field have shown that the most promising algorithms, from an interpretability standpoint, are the logical classification algorithms that formulate a decision rule in the form of a list of final rules (Kotsiantis, 2007). It is worth noting the

*Corresponding author.

scientists who have made the greatest contributions to the development of logical classification algorithms: Yu. Zhuravlyov, K. Rudakov, K. Vorontsov, N. Zagoruyko, P.L. Hammer, T. Bonates, G. Alexe, S. Alexe, Y. Freund, R.E. Schapire.

The most promising research in this field is carried out at the Rutgers University, USA, where they have successfully found solutions to a range of problems, including in medical diagnosis and prediction, by using logical data analysis methods (Alexe *et al.*, 2002; Brauner *et al.*, 2004; Hammer and Bonates, 2005). The acquired results demonstrate the efficiency of the selected approach whose evolution is arguably the foundation of modern decision support systems.

However, at the moment there is a range of challenges associated with the application of the method of logical analysis of data to solving practical classification problems. They include the problem of designing optimization models for building meaningful patterns. When looking into this issue, it is above all necessary to define the criteria and limitations that underpin such optimization models. Another challenge of the reviewed method is about building a classifier that could correctly attribute a new observation, i.e. the observation that was not involved in its creation, to the appropriate class. At this stage of method evolution, the primary task is to increase the interpretability of the classifier and the performance of the classification of new observations, that is, to improve the generalization capability of the classifier.

For the purpose of addressing the foregoing challenges, the article offers modifications to the method of logical analysis of data, which can improve the interpretability and generalization capability of the classifier.

2. Method of Logical Analysis of Data

2.1. *Approach Description*

The study considers the classification problem of the following kind (Kuzmich and Masich, 2014). There is a data set consisting of two disjoint sets Ω^+ and Ω^- of n-dimensional vectors belonging to the positive and the negative class, respectively. The components of the vectors, also called attributes, can be both numeric (nominal) and binary (Stupina *et al.*, 2012). The task is to subsume a certain new observation, also a vector of n variables, under the appropriate class.

The suggested data classification approach is based on the method originating from the theory of combinatorial optimization, which is called *Logical Analysis of Data (LAD)* (Hammer and Bonates, 2005). This method has been usefully employed in solving a range of problems in various fields (Kuzmich and Masich, 2012; Hammer *et al.*, 2004a, 2004b; Herrera and Subasi, 2013). The key idea of the method is to apply a combination of "differentiation" and "integration" actions to a section of the space of original attributes containing the given positive and negative observations. The "differentiation" stage involves defining a family of small subsets sharing characteristic positive and negative features. At the "integration" stage, the unions of these subsets, created in a specific manner, are treated as the approximations of certain areas of the space of attributes consisting of positive and, consequently, negative observations (Kuzmich and Masich, 2014).

The sequence of steps for this method is here (Hammer and Bonates, 2005):

a) To remove redundant variables in the original data set, a subset S is singled out from the set of variables to help to distinguish positive observations from the negative ones. The further steps of the method utilize the projections Ω_s^+ and Ω_s^- of the sets Ω^+ and Ω^- on S.

b) The Ω_s^+ set is covered with a family of similar subsets of a smaller space, each of which significantly overlaps with Ω_s^+, but does not overlap with Ω_s^-; alternatively, a minor overlapping with Ω_s^- is acceptable if it results in a greater overlapping with Ω_s^+. Such subsets are called "positive patterns." In a similar fashion, the Ω_s^- set is covered with "negative patterns."

c) Then it is necessary to identify the subset of positive patterns whose union covers all Ω_s^+ observations and the subset of negative patterns whose union covers all Ω_s^- observations.

d) The fact of whether a certain observation is covered by the union of the two subsets, which are either positive or negative, is then determined using a classifier built on these subsets.

2.2. *Binarization of Attributes*

The studied method is intended for the use against data sets of binary attributes. Since the original data set can include attributes of various types, it is necessary to binarize them.

One of the simplest binarization methods suggests linking each metric variable to a number of binary variables. A binary variable is assigned 1 if the value of the corresponding metric variable exceeds a certain threshold value, and vice versa. This method is referred to in Rastrigin and Freymanis (1988) as "unitary". Its flaw lies in the fact that it implies having numerous combinations of binary variables that cannot be linked to any points in the original space $(2n - n - 1)$. This flaw makes it difficult to use this method for coding the variable arguments of criterion functions when solving optimization problems, as it will generate a great number of invalid solutions. However, in this case, it does not matter as long as classification is concerned, because the binary variables are obtained by coding the predefined metric variables. The main advantage of this method though is the fact that the distances across the original and binary spaces are equal. It means that points closely spaced in the original space will also stay in proximity of each other in the binarized space. This, in its turn, makes it possible, as early as at the binarization stage, to minimize the number of thresholds by mapping close values of the original variable with the equivalent values within the binary space (provided that the positive and negative subsets of observations remain disjoint) (Hammer and Bonates, 2005).

Also there exists another binarization method referenced in Vorontsov (2010).

An arbitrary attribute $f : X \mapsto D_f$ creates terms verifying that the value of $f(x)$ falls into certain subsets of the D_f set. Some typical structures of this kind are provided in Vorontsov (2010).

- If f is a nominal attribute:

$$\beta(x) = \big[f(x) = d\big], \quad d \in D_f,$$
$$\beta(x) = \big[f(x) \in D'\big], \quad D' \subset D_f.$$

- Iff is an ordinal or quantitative attribute:

$$\beta(x) = \big[f(x) \leqslant d\big], \quad d \in D_f,$$
$$\beta(x) = \big[d \leqslant f(x) \leqslant d'\big], \quad d, d' \in D_f,\ d < d'.$$

For a quantitative attribute $f : X \to R$, it is necessary to only consider those threshold values d that divide the X^ℓ set in different ways. After excluding trivial dissections converting $\beta(x)$ to 0 or 1 across the whole set, the remaining number of such values will not exceed $\ell - 1$. For instance, it is possible to take thresholds of the following kind:

$$d_i = \frac{f^{(i)} + f^{(i+1)}}{2}, \qquad f^{(i)} \neq f^{(i+1)}, \quad i = 1, \dots, \ell - 1, \tag{1}$$

where $f^{(1)} \leqslant \dots \leqslant f^{(\ell)}$ is a sequence of values of the f attribute throughout the observations of the set $f(x_1), \dots, f(x_\ell)$, sorted in ascending order.

Should the resulting terms be later intended for the synthesis of conjunctions, it is recommended to pick the most informative ones right away, to cut down on the iterations of sequential search. With ordinal and quantitative attributes, such problem is solved through the optimal partitioning of the range of attribute values into zones. The process of such partitioning is described below.

Suppose $f : X \to R$ is a quantitative attribute, $d_1, \dots, d_r$ is an ascending sequence of thresholds. Let us define the zones containing the values of the f attribute as terms of the following kind:

$$\varepsilon_0(x) = \big[f(x) < d_1\big],$$
$$\varepsilon_s(x) = \big[d_s \leqslant f(x) < d_{s+1}\big], \quad s = 1, \dots, r - 1,$$
$$\varepsilon_r(x) = \big[d_r \leqslant f(x)\big].$$

For example, a greedy algorithm of zone merging starts with dividing them into “small zones.” The thresholds are calculated according to formula (1) and pass through all the pairs of points x_{i-1}, x_i, of which exactly one belongs to class k.

The initial division comprises alternating zones defined as “only k – only not k”. Later the zones can be consolidated through merging triple points of adjacent zones. It is important to merge specifically triple points, since merging pairs will disrupt the alternation of “k – not k”, resulting in some “small zones” remaining unmerged in the end. The algorithm of merging zones stops when either of the following criteria is satisfied: a specific number r of zones has been reached; or certain original zones ε_{i-1}, ε_i and ε_{i+1} start containing more information than the corresponding merged zone $\varepsilon_{i-1} \vee \varepsilon_i \vee \varepsilon_{i+1}$. The three points to merge are selected so as to achieve the maximum gain in information content after the merger.

2.3. *Building a Support Set*

Representing an excessively large number of attributes in a set can be associated with an enormous computational load. This is the case, for example, in genomics and proteomics, the two most rapidly progressing areas of bioinformatics where the expression for the level of intensity of thousands, if not tens of thousands, genes or proteins is included into the data set, despite the fact that even the smallest subset of these attributes is sufficient to perform an excellent separation of positive and negative observations (Alexe *et al.*, 2002). One of the factors that makes it more difficult to extract an informative subset of attributes is the fact that there is a pronounced difference between the information content of individual attributes and the information content of a set of attributes.

It is necessary to devise some approaches to the identification of a subset of attributes that can help separate, with a high degree of accuracy, the positive and negative observations.

One of such approaches based on the selection of a subset of attributes via building an optimization model in the form of a combinatorial optimization task is provided here.

A set S of attributes is called a support set if the projection Ω_s^+ of the set Ω^+ on S does not intersect with the projection Ω_s^- of the set Ω^- on S. The entire set of attributes is a support set since Ω^+ and Ω^- originally do not intersect. A support set can be called minimal, when the elimination of any remaining variable from it leads to a data set in which some positive and negative observations are identical.

In order to find the minimal support set, one needs to assign to each attribute x_i, $i = 1, \dots, t$ of the binary set a new binary variable y_i, which is equal to 1 if x_i belongs to the support set, and to 0 otherwise. One denotes the binary vector associated with positive observations as $U = (u_1, u_2, \dots, u_t)$ and the one associated with negative observations as $V = (v_1, v_2, \dots, v_t)$. A new variable is then introduced:

$$w_i(U, V) = \begin{cases} 1, & u_i \neq v_i, \\ 0, & u_i = v_i. \end{cases}$$

The separability of the sets Ω_s^+ and Ω_s^- is then conditioned by holding the inequation $\sum w_i(U, V)y_i \geqslant 1$ for any $U \in \Omega_S^+$ and $V \in \Omega_S^-$.

To ensure that the data set is more resistant to any errors occurring during the measurements which produce those data, this condition should be made stricter by replacing 1 in the right side of the inequation with a certain integer d. This means that the positive and negative observations should differ by at least d attributes.

Therefore, the problem of minimizing a support set can be formulated as a conditional pseudo-Boolean optimization problem:

$$\sum_{j=1}^{t} y_j \to \min,$$

$$\sum_{i=1}^{t} w_i(U,V)y_i \geqslant d \quad \text{for any } U \in \Omega_s^+ \text{ and } V \in \Omega_s^-,$$

where $y \in \{0,1\}^t$.

The objective function of this problem is unimodal, monotonic, pseudo-Boolean function (Antamoshkin and Masich, 2007a, 2007b; Antamoshkin and Semenkin, 1998), i.e. it has a single absolute minimum located in the point $y_0 = (0,0,\dots,0)$ and its output increases as it gets further from the point of minimum (when any of its components changes from 0 to 1). The constraint function is also a unimodal, monotonic, pseudo-Boolean function, besides, it is defined using an algorithm, since its calculation requires iterating through all possible pairs of positive and negative observations.

An alternative approach to selecting the attributes is the specially designed algorithmic procedure, which is based on evaluating the importance of the given attributes and helps to obtain a reduced set (Kuzmich and Masich, 2014).

The importance of any attribute is estimated against the frequency of its inclusion into the patterns involved in the classifier (Brauner *et al.*, 2004). Therefore, the more often the attribute is found in the resulting patterns, the more important it is. Those attributes that cannot be found or are rarely involved in building patterns are considered unimportant.

The algorithmic procedure for generating a reduced set of attributes consists of four stages:

The first stage of the procedure for generating a reduced set of attributes involves conducting a classification of the entire set of attributes in order to determine the importance of each attribute.

The second stage requires a researcher to set an importance threshold as a reference against which it is possible to assess the importance of an individual attribute.

The third stage is about sorting the attributes by their importance and identifying those attributes whose importance value turned out to be beneath the specified threshold.

The fourth stage consists in excluding the attributes singled out at the third stage from consideration. The remaining attributes will combine to the reduced set. In this way, by applying varied importance thresholds, the researcher can obtain different reduced sets of attributes, which can later be used to build patterns.

2.4. *Building Patterns*

The concept of patterns lies at the core of the reviewed approach. A positive pattern is defined as a subcube of a set of Boolean variables B_2^t that intersects with the set Ω_s^+ and does not share elements with the set Ω_s^-. A negative pattern is formed in a similar fashion. A positive a-pattern for $a \in \{0,1\}^t$ is a pattern that contains point a. For every point $a \in \Omega_s^+$, let us find the maximal a-pattern, i.e. the one covering the greatest number of points Ω_s^+ (Kuzmich and Masich, 2014).

The corresponding subcube is defined using y_j variables:

$$y_j = \begin{cases} 1, & \text{if the } i\text{-th attribute is located in the subcube,} \\ 0, & \text{otherwise.} \end{cases}$$

That is, by fixing l variables of the original cube with t dimensions, we obtain a subcube with $(t-l)$ dimensions and 2^{t-l} points.

The condition stipulating that a positive pattern should not contain any points from Ω_s^- demands that for each observation $b \in \Omega_s^-$ the y_j variable is equal to 1 at least for one j, where $b_j \neq a_j$:

$$\sum_{\substack{j=1 \\ b_j \neq a_j}}^{t} y_j \geqslant 1 \quad \text{for any } b \in \Omega_s^-.$$

The limitation can be made stricter to help increase error resistance, in which case the number 1 in the right side of the inequation should be substituted for a positive integer d.

On the other hand, a positive observation $c \in \Omega_s^+$ will only belong to the considered subcube where the y_j variable is equal to 0 for all indices j, where $c_j \neq a_j$. In this manner, the number of positive observations covered by the a-pattern can be calculated using the following formula:

$$\sum_{c \in \Omega_s^+} \prod_{\substack{j=1 \\ c_j \neq a_j}}^{t} (1-y_j).$$

Therefore, the task of building patterns is reduced to a conditional pseudo-Boolean optimization problem with algorithmically defined functions (Bonates *et al.*, 2006; Hammer *et al.*, 2004a, 2004b; Hwang and Choi, 2015):

$$\sum_{c \in \Omega_S^+} \prod_{\substack{j=1 \\ c_j \neq a_j}}^{t} (1-y_j) \to \max, \tag{2}$$

$$\sum_{\substack{j=1 \\ c_j \neq a_j}}^{t} y_j \geqslant d \quad \text{for any } b \in \Omega_s^-, \ \ y \in \{0,1\}^t. \tag{3}$$

The objective function (2) and the constraint function (3) in this problem are both unimodal, monotonic pseudo-Boolean functions.

The task of finding the maximal negative patterns is solved in a similar fashion.

Each identified pattern is characterized by its coverage – the number of captured observations within the corresponding class, and its degree – the number of fixed variables that determine this pattern. According to the above optimization model (2)–(3), the resulting patterns do not cover any observations from the different class (from the training set).

The most valuable are the patterns that demonstrate the greatest coverage. The greater the coverage, the more adequately the pattern reflects the image of the class.

The particular nature of the classification problem described above is in the fact that the database has a large number of unmeasured values (omitted data), whereas the measurements that have been made may be inaccurate or erroneous. It is well known that errors directly depend on measurement accuracy indicating how close the measurement results are to the actual values of the measured entities. The measurement accuracy can be increased or decreased, depending on the allocated resources (cost of measurement tools, spending on the process of measurement, stabilizing the external environment, etc.). It is understood that it must be fit for the task at hand, but not necessarily be of superior quality, because a further increase in accuracy may lead to excessive financial expenditures (Boros *et al.*, 2009).

Sets of quantitative data can have errors in the values of quantitative attributes because of imprecise tools, imperfect measurement methods or human errors. Noise and spikes can lead to observations from different classes "overlapping" with each other and getting in the "areas" of the opposite class. Consequently, the resulting patterns have a higher degree and a much lesser coverage than they would have had without those spikes and errors, while the classifier ends up consisting of a great number of small patterns (with little coverage). This prevents one from building an effective classifier with "well-interpreted" rules involving a small number of attributes and a high degree of classification accuracy.

To make the method more error-resistant, it is recommended to loosen the limitation described in (3). This will reduce the number of calculated patterns and increase their coverage.

The limitation of the optimization model will then look in the following way (Kuzmich and Masich, 2014):

$$\sum_{b\in\Omega_S^-} z_b \leqslant D, \quad \text{where } z_b = \begin{cases} 0, & \text{if } \sum^{t}{}_{\substack{j=1\\ b_j\neq a_j}} y_j \geqslant d, \\ 1, & \text{otherwise}, \end{cases} \tag{4}$$

where D is the number of observations of a different class that are allowed to be covered by the pattern (a non-negative integer).

The functions (2)–(4) of the created optimization model are defined using an algorithm, i.e. they are calculated over a specific sequence of operations. The optimization problem is solved using optimization algorithms based on looking for boundary points of the permissible region (Antamoshkin and Masich, 2006, 2007a, 2007b). Such algorithms were specially designed for this class of problems and are based on the behaviour of monotonic functions of the optimization model in the space of Boolean variables. The algorithms looking for boundary points are search algorithms, i.e. they do not require defining the functions explicitly, via algebraic expressions. Instead, they calculate the function outcome across a number of points.

According to the model (2, 4), the most preferable patterns are the ones with the maximum coverage. Consequently, the patterns built in this way have a low degree, i.e. they consist of a small number of terms and use only a fraction of attributes. Low-degree patterns correspond to large areas in the space of attributes. This may lead to their covering some observations from a different class (missing in the training set) and the increased number of incorrectly classified observations. This characteristic feature affects the information content of the pattern towards reducing it. Therefore, to increase the information content, the authors suggest using an algorithmic procedure for aggregating patterns. It is applied to each created pattern by driving the degree of the said patterns to a maximum level while at the same time keeping their coverage intact:

$$\sum_{j=1}^{t} y_j \to \max,$$

$$fc(Y) = fc'(Y),$$

where $fc(Y)$ is the value of the objective function (coverage) for the pattern before the aggregation procedure, $fc'(Y)$ is the value of the objective function for the pattern after the aggregation procedure.

This way, the application of the pattern aggregation procedure can increase the information content of the patterns by reducing their coverage by the observation rules from the other class, thus driving up the accuracy of the decisions made by the classifier.

The next stage of this method is dedicated to solving the problem of building an adequate classifier that could classify any incoming observation, i.e. the observation that was not around when the classifier was being built.

2.5. *Building a Classifier*

The result of the previous stage of this method is a family of maximal patterns whose number is limited by the cardinal of the data set $|\Omega^+ \bigcup \Omega^-|$. The classifier consists of a full set of positive and negative patterns.

In order to classify a new observation, let us be guided by the following decision rule (Hammer and Bonates, 2005):

1) If the observation satisfies the conditions of one or more positive patterns and does not satisfy any of the conditions of any negative ones, it is classified as positive.
2) If the observation satisfies the conditions of one or more negative patterns and does not satisfy any of the conditions of any positive ones, it is classified as negative.
3) Choosing the voting algorithm:
   a) Simple voting algorithm. If an observation satisfies the conditions p' of p positive patterns and the conditions q' of q negative patterns, the sign of the observation is determined as $p'/p - q'/q$.
   b) Weighted voting algorithm. If an observation satisfies the conditions p' of p positive patterns and the conditions q' of q negative patterns, the sign of the observation is determined as $\sum_{n=1}^{p'} a_n - \sum_{n=1}^{q'} b_n$, where a and b are weighting factors for the positive and negative patterns respectively. The weight of the n-th positive pattern is calculated according to the formula: $a_n = \frac{H_n}{\sum_{n=1}^{p} H_n}$, where Hn is the information content of the n-th positive pattern calculated using the boosting criterion (6) (Kuzmich and Masich, 2012). The cumulative weight of all positive patterns is equal to 1: $\sum_{n=1}^{p} a_n = 1$. Similarly, it is possible to calculate the information content and the weight of the n-th negative pattern.
4) In case the observation does not meet any conditions of any pattern, either positive or negative, it is assigned to the class that has the lowest price of error.

2.6. *Modifications to the Method of Logical Analysis of Data*

Creating patterns and building a classifier are milestone stages of the method of logical analysis of data. The implementation of these stages is what directly determines the quality of the classification results. For that reason, the design of modifications to the method is associated with developing algorithmic procedures that address these stages.

So, at the pattern-creating stage, the suggested approach to defining the objective function for the optimization model is based on modifying the objective function (2) in order to emphasize the differences between the rules used in the classifier. This approach rests on the premise that the patterns to be voted should be different; otherwise they will serve no purpose for the classification.

According to the objective function (2), each created pattern maximizes its coverage by capturing observations typical for the corresponding class, whereas non-typical observations of the class remain uncovered, and the classifier does not comprise any patterns that take those into account. This way we obtain a set of similar patterns for the class, thus compromising the classification quality. To get a classifier with a higher distinction between the rules that allows allocating significantly different subsets of observations, the authors suggest introducing the following modification to the objective function (2) in order to identify positive patterns:

$$\sum_{c \in \Omega_S^+} K_c \prod_{\substack{j=1 \\ c_j \neq a_j}}^{t} (1 - y_j) \to \max, \tag{5}$$

where K_c is the weight of the positive observation $c \in \Omega_s^+$, which decreases when this observation is covered, effectively lowering its participation priority in building the next pattern in favour of uncovered observations.

The objective function for the optimization model used to identify negative patterns is created in a similar fashion.

To be able to use the optimization model with the objective function (5) for building patterns, it is necessary to specify the initial weights for all observations and the rule for changing the weights of those observations that have participated in creating the current pattern. It is recommended to set the initial weights to 1 for each observation in a training set. Below is the rule for changing the weight of any observation that has already participated in creating the current pattern:

$$K_{i+1} = \max\left[0,\ K_i - \frac{1}{N_{\max}}\right],$$

where K_i, K_{i+1} are the weights of the observation that is being covered during the creation of the current and the next patterns, $N_{\max}$ is a researcher-specified parameter denoting the maximum number of patterns that can cover an observation from the training set in the classifier.

This way, using the optimization model with the objective function (5) to build patterns, one can come up with logical rules that cover significantly different subsets of observations. Later on, those of them that yield a positive outcome of the objective function are selected and aggregated in the classifier.

The next stage of the method is dedicated to solving the problem of building an adequate classifier that could correctly classify any incoming observation, i.e. the observation that did not take part in the creation of the classifier.

In view of a potentially large volume of the data set, a question arises as to the need of reducing the number of patterns, since this quantity in the original classifier is equal to the cardinal of the training data set $|\Omega^+ \bigcup \Omega^-|$. In short, it is necessary to define a classifier consisting of a certain number of patterns in such a way that it would be capable of classifying the same observations that are possible to classify using a complete system of patterns.

This study offers the following algorithmic procedures for reducing the number of patterns in the original classifier:

- selecting baseline observations for building patterns (Kuzmich and Masich, 2014);
- building a classifier as a composition of informative patterns (Kuzmich and Masich, 2012).

The implementation of the algorithmic procedure of selecting baseline observations for building patterns involves completing a series of consecutive steps. First, based on the observations from the training set, one needs to derive centroids for each class by using the k-means algorithm. According to the k-means clustering algorithm, each observation from the training set has to be put into one of the k-clusters so that each cluster is represented by the centroid of the corresponding observations, whereby the distance from each observation to the centroid of its cluster is shorter that the distance to the centroids of any other cluster. This algorithm makes it possible to pick a range of centroids that most accurately represents the distribution of observations in the training set.

The algorithm comprises the following steps described in Bagirov (2011):

Step 1. Pick k initial centroids $z_1(1), z_2(2), \dots, z_k(l)$. The initial centroids are selected arbitrarily, e.g. the first k observations from the training set.

Step l. At the l-th step of the iteration, distribute the set of observations $X = \{x_1, x_2, \dots, x_m\}$ among k clusters according to the following rule:

$$x \in T_j(l), \quad \text{if } \|x - z_j(l)\| < \|x - z_i(l)\|$$

for every $i = 1, 2, \dots, k$, $i \neq j$, where $T_j(l)$ is the set of observations belonging to the cluster with the centroid $z_j(l)$. In case of equality, the decision is made in arbitrary way.

Step $l + 1$. Based on the results of step l, new centroids of clusters $z_j(l + 1)$, $j = 1, 2, \dots, k$ are derived, on the assumption that the sum of squared distances between all observations belonging to the set $T_j(l)$ and the new centroid of this cluster must be minimal.

The centroid $y_j(l+1)$ ensuring the minimization $J_j = \sum_{x \in T_j(l)} \|x - z_j(l+1)\|^2$, $j = 1, 2, \ldots, k$ is a sample average calculated across the set $T_j(l)$. Therefore, the new cluster centroids are defined as:

$$z_j(l+1) = \frac{1}{N_j} \sum_{x \in T_j(l)} x, \quad j = 1, 2, \ldots, k,$$

where N_j is the number of sample observations included into the set $T_j(l)$. Apparently, the choice of the k-means algorithm is due to the established way of sequential correction of the calculated cluster centroids.

The equation $z_j(l+1) = z_j(l)$, given $j = 1, 2, \ldots, k$, is the condition for the convergence of this algorithm, and upon its achievement the execution of algorithm stops. The resulting sets $T_j(l)$, $j = 1, 2, \ldots, k$ will be the sought-for clusters. If this is not the case, the last step is repeated.

This algorithm is used to partition the observations of the training set of each class into clusters. It produces a separate set of centroids for each class.

Second, one needs to add the resulting sets of centroids to the observations in the training set. Third, the centroids are used as baseline observations for building patterns.

This way, by implementing the heuristic procedure described above, we get a new classifier consisting of a lesser number of patterns. The number of patterns in the classifier will be equal to the cumulative number of centroids obtained for each class. Clearly, the classification accuracy depends on the number of centroids for each class, therefore one needs to conduct multiple experiments with sets of centroids of diverse quantity in order to establish how the classification accuracy depends on the number of centroids for each class.

The procedure of selecting baseline observations for building patterns must be implemented prior to creating the classifier, effectively simplifying its creation due to the significant reduction of the number of patterns to be built, however, this will normally slightly degrade the classification accuracy. To mitigate this shortcoming, another approach can be used to reduce the number of patterns in the original classifier. It is necessary to build a classifier whose number of patterns is equal to the cardinal of the training data set, and to reduce this number of patterns while retaining the high accuracy of classification. This approach can be implemented through the suggested procedure of building a classifier as a composition of informative patterns, which is based on the concept of their information content.

There are several criteria for measuring the information content of a pattern offered in the discipline-specific literature. This study recommends using the boosting criterion, since it adequately assesses the information content of a pattern and is fairly simple to calculate:

$$H(p, n) = \sqrt{p} - \sqrt{n}, \tag{6}$$

where p is the number of observations of own class captured by the created pattern; n is the number of observations from other classes captured by the created pattern.

Initially, the classifier includes all patterns that are built against each observation in the training set. Consequently, as the volume of the training set increases, so does the size of the set of rules for the classifier. Notably, the created patterns are characterized by different information content. The patterns covering a small number of observations are statistically unreliable – they include too many patterns that make more mistakes with independent support data than with a training set. For that reason, it is recommended to only include informative patterns into the classifier, i.e. their information content must exceed a certain information threshold (H_0) specified by the researcher. This will help to reduce the number of patterns in the classifier without compromising the classification accuracy or with only slight changes towards its improvement/deterioration.

The solving of this problem raises the issue of choosing the information threshold. This study addresses this issue through designing the following iterative procedure. The first step of this procedure suggests setting the information threshold to 0 for both positive and negative sets of patterns, thus resulting in the original classifier consisting of the maximum number of patterns possible. At the second step of this procedure, it is necessary to set the information threshold for negative (positive) patterns, which should be equal to the average information content (H_{avg}) across all negative (positive) patterns:

$$H_{avg} = \frac{1}{q}\sum_{i=1}^{q} H_i,$$

where q is the number of negative (positive) patterns in the classifier, H_i is the information content of the i-th negative (positive) pattern calculated using the formula (6).

To get a new classifier consisting of patterns with greater information content, we will remove from the original classifier all negative (positive) patterns whose information content is below the information threshold derived for them. Having calculated the values of the average information content for negative and positive patterns of the current classifier, we will use them to build the next classifier that will consist of patterns whose information content is higher than the values of the average information content for the current classifier. This way we will build each successive classifier, each time utilizing the average information content of the present one. This shortens the number of patterns and increases the average information content for each successive classifier. The procedure should stop as soon as the number of unclassified (uncovered) observations has increased during the classification process, i.e. the patterns included in the current classifier fail to cover certain observations belonging to the test sample. In this case, it is necessary to either get back to the previous classifier and reverse the two information threshold to their previous values, or change the value of only one information threshold for negative (positive) patterns and register how this amendment will affect the number of unclassified observations and the classification results in general.

Based on the designed algorithmic procedures, the authors suggest the following modifications to the method of logical analysis of data in order to improve the generalization capability of the classifier and make it more interpretable by reducing the number of rules it uses:

- using the objective function (5) and the constraint function (4) to create patterns and build the classifier exclusively on the rules that yield a positive (greater than zero) outcome of the objective function;
- using the algorithmic procedure for selecting baseline observations to create patterns and applying the aggregation procedure to the resulting rules;
- applying the algorithmic procedure of building a classifier as a composition of informative patterns based on the optimization model (2, 4) coupled with the aggregation procedure.

The suggested modifications to the method of logical analysis of data can help improve the quality of the classification of new observations.

3. Obtained Results

The method of logical analysis of data is implemented in a software system that made it possible to solve the following classification problems taken from the UCI Machine Learning Repository: SPAM detection, classification of the results of radar scans of the ionosphere. The problem of complications prediction of the myocardial infarction (MI) is also considered. For the solving of this problem, the staff of the Chair of internal diseases No. 1 of the Krasnoyarsk State Medical Academy collected the information on the course of a disease of 1700 patients with the MI undergoing the treatment in 1989–1995 at the Cardiological center of a Municipal Hospital No. 20 of Krasnoyarsk. Information is obtained from case histories of patients. Each observation (patient) was characterized by a vector of 112 characteristics (Golovenkin *et al.*, 1997). The characteristics are binary (majority) rated and numerical values. There is a considerable number of missed data in this data sample. Among the chosen complications, there exist fibrillation of auricles (FA), fibrillation of ventricles (FV), fluid lungs (FL), cardiorrhesis (CR), and also lethal outcome (LO).

Earlier, the problem of prediction of the MI complications was solved by means of neural networks (Golovenkin *et al.*, 1997). At its solution it was noted that the qualifier yields poor results in a case of essential distinction in number of observations of each class in the initial data sample. Therefore, the following approach to the solution of this problem was offered. The number of patients with some complication (positive observations) is approximately ten times smaller than the number of patients at whom this complication was not observed (the negative observations). The initial data sample (1700 observations) is divided into test data sample and 10 training data samples for every complication. The positive observations in the training data samples remain the same and the negative observations differ. The method is trained on each of training data samples separately but it is tested on the common examining data sample. Finally, the solution on each observation of the examining data sample is made by a majority of votes of all qualifiers received on the basis of 10 training data samples. When using this approach for the solution of our problem, besides classification upgrading, we have an opportunity of classification results comparison of methods of the logical data analysis and neural networks. The number of

Table 1
Structure of data samples of all MI complications.

	FA	FV	FL	CR	LO
Number of positive observations	70	170	159	54	160
Number of negative observations	181	180	173	179	172
Number of observations in the examining data sample	30	50	39	28	50

Table 2
Classification results for the problem of SPAM detection.

Optimization problem	Set of rules	Num. of rules	Coverage of negative observations	Coverage of positive observations	Degree of the rule	Classification accuracy, %
Objective function (2),	neg.	234	49	0	4	98
constraint function (3)	pos.	134	0	29	4	68
Objective function (2),	neg.	234	96	5	5	98
constraint function (4)	pos.	134	5	50	4	79
Objective function (2),	neg.	234	96	4	7	98
constraint function (4) with the application of the augmentation procedure	pos.	134	4	50	5	87
Objective function (5),	neg.	49	69	5	4	96
constraint function (4)	pos.	59	5	31	4	72

patients with complications and without complications of each of 10 selections and the size of test data sample for all considered complications are presented in Table 1.

The rules for each problem were being derived using four optimization models: the "strict" model disallowing the created rules to cover observations from a different class; the modified model allowing the rules to cover a certain limited number of observations from a different class; the modified model with a pattern aggregation procedure; the model for creating patterns covering significantly different subsets of observations from the training set.

Table 2 shows the classification results for one of the aforementioned problems – the SPAM detection. The test was run against 279 negative (non-SPAM) and 181 positive observations (SPAM), with 20% of the set being used in the test. Overall, 20 experiments have been conducted, with their results averaged out.

By applying the pattern aggregation procedure, it is possible to obtain higher-degree patterns with the maximal coverage, which helps to increase the reliability of the decisions made by the classifier. The modification to the method of logical analysis of data involving the application of the objective function (5) allows simplifying the classifier by significantly reducing the number of its patterns.

Let us conduct the check of the procedure for selecting baseline observations for creating patterns. The solution to the problem of classifying the results of a radar scan of the ionosphere requires generating 15 centroids for each class using the k-means clustering algorithm run within the WEKA software. The generated centroids are then added to the original training set, and patterns are built upon them. Ultimately, within the scope of this

Table 3
Accuracy of the solutions to the problem of classifying the results of the ionosphere radar scan.

Set of rules	Coverage of neg. observations in the new/original classifier	Coverage of pos. observations in the new/original classifier	Degree of the rule in the new/original classifier	Number of rules in the new/original classifier	Accuracy of the new classifier, %	Accuracy of the original classifier, %
Neg.	45 / 36	15 / 15	2 / 2	15 / 95	74	68
Pos.	15 / 15	139 / 130	3 / 3	15 / 186	96	98

Table 4
Classification results for the problem of SPAM detection following the change in the value of the information threshold, H_0.

S/n of experiment	Set of rules	Number of rules	Average meaningfulness, H_{avg}	Meaningfulness threshold, H_0	Coverage of negative observations	Coverage of positive observations	Number of uncovered observations	Classification accuracy, %
1	neg.	234	7.84	0	120	10	0	96
	pos.	134	4.49	0	10	57	0	89
2	neg.	132	8.51	7.84	134	10	0	93
	pos.	79	5.49	4.49	10	70	0	85
3	neg.	68	8.85	8.51	141	10	1	87
	pos.	39	6.05	5.49	10	77	1	79
4	neg.	68	8.85	8.51	141	10	0	98
	pos.	79	5.49	4.49	10	70	0	87
5	neg.	34	9.03	8.85	146	10	0	96
	pos.	79	5.49	4.49	10	70	0	89

problem, the test is carried out on just 20% of the set consisting of 240 positive and 141 negative observations. The corresponding classification results are given in Table 3.

According to the results (see Table 3), we have achieved a slight change in the classification accuracy for the problem at hand and a 9-fold decrease in the number of rules used by the classifier.

Let us conduct the check of the algorithmic procedure for building a classifier as a composition of informative patterns as applied to the problem of SPAM detection. Only 20% of the set are used for this test. The classification results are given in Table 4. For each experiment presented in Table 4, the researcher only specifies the information threshold. In the first experiment, the information thresholds are set to 0 for each class. In all subsequent experiments, they are equal to the average information content calculated under the previous experiment. Upon the occurrence of uncovered observations, the value of the information content is amended for one class only.

According to the obtained results (see Table 4), it is possible to conclude that the method modification associated with this procedure allows simplifying the classifier, since the number of rules it is comprised of decreases 4-fold with respect to the full set of rules for this problem. This, however, does not compromise the accuracy of the classification or does so to a negligible extent.

Table 5
Comparison of classification algorithms.

Problem	Algorithm measure	1-R	RIP-PER	CART	C4.5	Random forest	Adaboost	LAD
SPAM detection	The number of correctly identified observations, %	82.6	91.3	90.2	90.2	89.1	91.3	92.4
Radar scan of the ionosphere	The number of correctly identified observations, %	78.6	82.8	82.8	81.4	84.2	88.5	90
FA	The number of correctly identified observations, %	58	66	62	70	70	74	76
FV	The number of correctly identified observations, %	87.3	86.7	63.3	83.3	68.3	89	90
FL	The number of correctly identified observations, %	85.7	78.6	85.7	85.7	71.4	89.3	96.4
CR	The number of correctly identified observations, %	69.2	69.2	71.8	76.9	66.7	69.7	79.5
LO	The number of correctly identified observations, %	64	74	74	66	76	74	86

Table 5 provides the comparison of the accuracy of classification results for 6 machine-learning algorithms (1-R, Barsegyan *et al.*, 2004, RIPPER, Vijayarani and Divya, 2011, CART, Shi *et al.*, 2016, C4.5, Vijayarani and Divya, 2011, Random Forest, Provost *et al.*, 2016, Adaboost, Sun *et al.*, 2016) obtained in the WEKA (Weka 3, 2015) data analysis system, with the accuracy of the results obtained using the method of logical analysis of data (LAD) that the authors designed. The data sets for each problem are randomly divided into a training set (80%) and a test set (20%) for SPAM detection and classification of radar scan results of the ionosphere. Twenty experiments have been conducted for each method, with their results averaged out. For the problem of predicting of the MI complications, the sample size used for testing for each complication was determined according to Table 1.

Since the point estimates of the classification accuracy are inessential, Table 6 gives confidence intervals covering the true accuracy values with a confidence probability of 0.95 for all algorithms.

According to the data provided in Tables 5–6, the modified method of logical analysis of data is superior in accuracy to the classification algorithms it has been compared to.

Table 6
Confidence intervals of classification accuracy.

Problem	Algorithm measure	1-R	RIP-PER	CART	C4.5	Random forest	Adaboost	LAD
SPAM detection	The number of correctly identified observations, %	(79.8; 81.4)	(90.8; 91.8)	(89.8; 90.6)	(89.6; 90.8)	(88.6; 89.6)	(90.8; 91.8)	(92; 92.8)
Radar scan of the ionosphere	The number of correctly identified observations, %	(78.1; 79.1)	(82.3; 83.3)	(82.3; 83.3)	(79.7; 82.1)	(83.7; 84.7)	(88; 89)	(89.6; 90.4)
FA	The number of correctly identified observations, %	(57.3; 58.7)	(65.3; 66.7)	(61.3; 62.7)	(69.1; 70.9)	(69.3; 70.7)	(73.3; 74.7)	(75.5; 76.5)
FV	The number of correctly identified observations, %	(86.6; 88)	(86.1; 87.3)	(62.7; 63.9)	(82.5; 84.1)	(67.7; 68.9)	(88.5; 89.5)	(89.5; 90.5)
FL	The number of correctly identified observations, %	(85.1; 86.3)	(78.1; 79.1)	(85.1; 86.1)	(84.9; 86.3)	(69.9; 74.9)	(88.8; 89.8)	(96; 96.8)
CR	The number of correctly identified observations, %	(68.4; 70)	(68.4; 69)	(71; 72.6)	(76; 77.8)	(66; 67.4)	(69; 70.4)	(79; 80)
LO	The number of correctly identified observations, %	(63.3; 64.7)	(73.3; 74.7)	(73.3; 74.7)	(65.1; 66.9)	(75.3; 76.7)	(73.3; 74.7)	(85.5; 86.5)

4. Conclusion

An optimization model has been created for building patterns covering significantly different subsets of observations from the training set. This model helps to improve the generalization capability of the classifier built upon these rules. An algorithmic pattern-aggregation procedure has been designed that leads to an increased information content of the rules, effectively helping to improve the accuracy of the decisions made by the classifier. Algorithmic procedures have been developed to reduce the number of patterns in the original classifier while retaining the high accuracy.

The study offers a modified method of logical analysis of data based on the designed algorithmic procedures, which, when applied, helps to increase the interpretability of the classifier and improve its generalization capability. By finding a solution to practical problems, the authors have empirically verified the applicability of optimization models to the task of building informative patterns and the efficiency of the designed algorithmic procedures in relation to the method of logical analysis of data. The accuracy of the modified method of logical analysis of data has been compared against other classification algorithms on practical problems. It turned out that the method has demonstrated better accuracy when solving the proposed problems.

The acquired results advance the studies in the field of logical algorithms of classification and can provide a framework for designing more enhanced decision support systems working on recognition and prediction. The most important advantage of such systems is going to be the ability to interpret the solutions produced by them and substantiate the recommendations they will give. Experience has proved that often the availability of such opportunities is central to a user's work on recognition and prediction problems.

References

Alexe, G., Alexe, S., Axelrod, D., Boros, E., Hammer, P.L., Reiss, M. (2002). Combinatorial analysis of breast cancer data from image cytometry and gene expression microarrays. *RUTCOR Technical Report*, 3, 1–12.

Antamoshkin, A.N., Masich, I.S. (2006). Heuristic search algorithms for monotonic pseudo-Boolean function conditional optimization. *Problems of Mechanical Engineering and Automation*, 5(1), 55–61.

Antamoshkin, A.N., Masich, I.S. (2007a). Identification of pseudo-Boolean function properties. *Problems of Mechanical Engineering and Automation*, 2, 66–69.

Antamoshkin, A.N., Masich, I.S. (2007b). Pseudo-Boolean optimization in case of unconnected feasible sets. *Models and Algorithms for Global Optimization. Series: Springer optimization and Its applications*, 4(16), 111–122.

Antamoshkin, A., Semenkin, E. (1998). Local search efficiency when optimizing unimodal pseudoboolean functions. *Informatica*, 9(3), 279–296.

Bagirov, A.M. (2011). Fast modified global *k*-means algorithm for incremental cluster construction. *Pattern Recognition*, 44, 866–876.

Barsegyan, A.A., Kupriyanov, M.S., Stepanenko, V.V., Kholod, I.I. (2004). *Method and Models of Data Analysis: OLAP and Data Mining*. BHV-Peterburg, Saint Petersburg (in Russian).

Bonates, T., Hammer, P.L., Kogan, A. (2006). Maximum patterns in datasets. *RUTCOR Research Report*, 9, 1–18.

Boros, E., Hammer, P.L., Kogan, A., Crama, Y., Ibaraki, T., Makino, K. (2009). Logical analysis of data: classification with justification. *RUTCOR Technical Report*, 5, 1–34.

Brauner, M.W., Brauner, D., Hammer, P.L., Lozina, I., Valeyre, D. (2004). Logical analysis of computer tomography data to differentiate entities of idiopathic interstitial pneumonias. *RUTCOR Research Report*, 30, 1–17.

Golovenkin, S.E., Gorban, A.N., Schulman, B.A. et al. (1997). *Complications of Myocardial Infarction: Database for Approbation of Recognition and Forecast Systems*. Computing Center of Siberian Branch of Russian Academy of Sciences, Krasnoyarsk (in Russian).

Hammer, P.L., Bonates, T. (2005). Logical analysis of data: from combinatorial optimization to medical applications. *RUTCOR Research Report*, 10, 1–27.

Hammer, P.L., Kogan, A., Lejeune, M. (2004a). Modeling country risk ratings using partial orders. *RUTCOR Research Report*, 24, 1–30.

Hammer, P.L., Kogan, A., Simeone, B., Szedmak, S. (2004b). Pareto-optimal patterns in logical analysis of data. *Discrete Applied Mathematics*, 144, 79–102.

Herrera, J.F.A., Subasi, M.M. (2013). Logical analysis of multi-class data. *RUTCOR Technical Report*, 5, 1–24.

Hwang, H.K., Choi, J.Y. (2015). Pattern generation for multi-class LAD using iterative genetic algorithm with flexible chromosomes and multiple populations. *Expert Systems with Applications: An International Journal*, 42(2), 833–843.

Kotsiantis, S.B. (2007). Supervised machine leaning: a review of classification techniques. *Informatica*, 31, 249–268.

Kuzmich, R., Masich, I. (2012). Building a classification model as a composition of informative patterns. *Management Systems and Information Technologies*, 2 (48), 18–22 (in Russian).

Kuzmich, R., Masich, I. (2014). Modification to an objective function for building patterns aimed at increasing the distinction between the rules of the classification model. *Management Systems and Information Technologies*, 2 (56), 14–18 (in Russian).

Provost, F., Hibert, C., Malet, J.-P. (2016). Automatic classification of endogenous landslide seismicity using the Random Forest supervised classifier. *Geophysical Research Abstracts*, 18, 23–35.

Rastrigin, L., Freymanis, E. (1988). Solving problems of multiple-scale optimization using random-search methods. *Problems of Random Search*, 11, 9–25 (in Russian).

Shi, K.-Q., Zhou, Y.-Y., Yan, H.-D., Li, H., Wu, F.-L., Xie, Y.-Y., Braddock, M., Lin, X.-Y., Zheng, M.-H. (2016). Classification and regression tree analysis of acute-on-chronic hepatitis B liver failure: Seeing the forest for the trees. *Journal of Viral Hepatitis*, 24(2), 132–140.

Stupina, A., Ezhemanskaja, S., Kuzmich, R., Vaingauz, A., Korpacheva, L., Fyodorova, A. (2012). Multiple-attribute decision making method based on qualitative information. *Modern Problems of Science and Education*, 5, 1–8 (in Russian).

Sun, B., Chen, S., Wang, J., Chen, H. (2016). A robust multi-class AdaBoost algorithm for mislabeled noisy data. *Knowledge-Based Systems*, 102, 87–102.

Vijayarani, S., Divya, M. (2011). An efficient algorithm for generating classification rules. *International Journal of Computer Science and Technology*, 2(4), 512–515.

Vorontsov, K. (2010). *Lectures on logical algorithms of classification*. Access mode:
http://www.machinelearning.ru/wiki/images/3/3e/Voron-ML-Logic.pdf (in Russian).

Weka 3. (2015). *Data Mining with Open Source Machine Learning Software in Java*. Access mode: http://www.cs.waikato.ac.nz/˜ml/weka/index.html.

R. Kuzmich is a candidate of technical sciences, an associate professor of Siberian Federal University (Krasnoyarsk, Russia). His research interests are optimization techniques, modelling, control systems.

A. Stupina is a doctor of technical sciences, a professor of Siberian Federal University (Krasnoyarsk, Russia). Her research interests are n-version programming, modelling, control systems.

L. Korpacheva is a candidate of technical sciences, an associate professor of Siberian Federal University (Krasnoyarsk, Russia). Her research interests are modelling, system analysis.

S. Ezhemanskaja is a candidate of technical sciences, an associate professor of Siberian Federal University (Krasnoyarsk, Russia). Her research interests are modelling, system analysis.

I. Rouiga is a candidate of economical sciences, an associate professor of Siberian Federal University (Krasnoyarsk, Russia). Her research interests are economic-mathematical modelling, investment and innovation policy at the regional level.