Multi-spectral Image Recognition Using Solution Trees Based on Similarity

Vladimir B. Berikov[1,2], Igor A. Pestunov[3], Roman M. Kozinets[2], Sergey A. Rylov[3]

[1] Sobolev Institute of Mathematics, Novosibirsk, Russia, berikov@math.nsc.ru
[2] Novosibirsk State University, Novosibirsk, Russia
[3] Institute of Computational Technologies SB RAS, Novosibirsk, Russia, pestunov@ict.sbras.ru

Abstract. A method for constructing decision trees based on the mutual similarity of objects is proposed. The method allows obtaining complex decision boundaries, which have a clear logical interpretation. The results of the experiments confirm the effectiveness of the method for multispectral image recognition.

Keywords: recognition, multispectral image, decision tree, similarity of observations.

1 Introduction

Classification methods based on logical decision functions presented in the form of decision trees (DT) [1,2] are popular in machine learning. Compared with other approaches, DT has the following advantages:

– give one an opportunity to analyze information of different types (i.e., for quantitative and qualitative characteristics describing objects), in the presence of missed feature values;

– find probabilistic logical rules that reflect cause-and-effect relationships of the phenomenon under study;

– automatically determine the most informative features for each classified object and use them for making a decision;

– in combination with an ensemble approach (e.g., decision forest, boosting on trees [3,4]), DT is able to find sufficiently stable solutions with high generalizing ability.

A recent review of existing methods for DT induction is given in [5]. Despite a large number of known approaches, there is still a need in developing efficient methods with high generalization ability. There are several possible ways to improve quality. The first approach is to find a criterion that will enhance the predictive ability of decisions by optimally combining the accuracy and complexity of the tree for the given data [6]. The second approach involves the development of more sophisticated techniques for representing the tree (for example, using linear decision boundaries in the tree nodes) and applying "deeper" algorithms for searching the optimal tree structure [7].

A "classical" DT is a tree-like graph, in the nodes of which conditions of two possible types are tested. If X is a numerical attribute, then the condition "X(a) < b" is examined, where a is an arbitrary object from the statistical population, X(a) is the value of X for object a, b is some value of the attribute. If X is a categorical attribute, then the condition "X(a)=b" is checked. Depending on the truth or false of the test, the left or the right sub-node is chosen. The leaves (terminal nodes) of the tree are associated with the values (class labels) of the predicted feature. The paths from the root node to leaves represent classification rules. To find an optimal DT, a recursive partition of feature space is performed.

This approach has a significant drawback: the partitioning of feature space occurs strictly parallel to the feature axes (in the case of numerical features), even if the real boundary between classes has a linear shape (Figure 1). To approximate the boundary, it is necessary to use a more complex tree structure (with many additional nodes) that often has a negative influence on the efficiency of decisions.

Some works (e.g., [7]) propose oblique DT (ODT, also called multivariate DT) with more complicated types of statements having the form $\sum \beta_j X_j(a) + \beta_0 < 0$, where the summation is carried out over a subset of numerical features, $\beta_0, \beta_1, \dots$ are real-valued coefficients. The coefficients are estimated by optimizing a given quality functional for the subset of objects in the tree node. A number of algorithms for ODT induction exist:

- Classification and Regression Trees - Linear Combination (CART-LC) [1];
- Simulated Annealing Decision Tree (SADT) [8];
- Linear Machine Decision Trees (LMDT) [9],
- OC1 system [10],

– Based on Support Vector Machine (SVM-ODT) [11], etc.

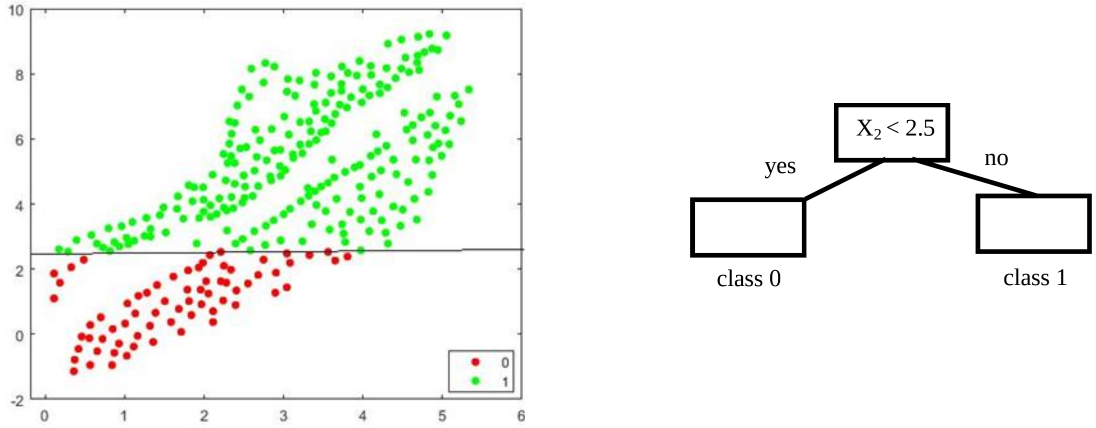


Figure 1. An example of linearly separable classes (labeled 0 and 1) and their partition in accordance with the “classical” decision tree.

Despite the significant improvement of prediction accuracy, this approach also has a number of limitations. First of all, the found linear boundaries are not-easy to interpret in contrast with simple rules of “classical” univariate DT. Another limitation is ODT is applicable only for multivariate data and cannot be used for data described with pairwise similarity matrices.

To overcome the latter difficulty, the work [12] suggests Similarity Forest method in which an ensemble of ODT is built. Each variant of ODT is defined by randomly chosen pair of data points from different classes; the splitting boundary is a hyperplane perpendicular to the segment connecting the pair and crossing its midpoint. In the experiments, the proposed algorithm has demonstrated sufficiently high accuracy in comparison with a number of other methods, especially in the presence of missed feature values, even if the input information has the form of multidimensional data. However, the obtained ensemble decision is hard-to-interpret because it includes a large number of generated trees.

The method proposed in this paper aims to eliminate the above-mentioned drawbacks. We propose to use a more general type of statements regarding the similarity of observations. The similarity can be calculated using various metrics in different feature subspaces. This type of decision tree allows one to get more complex decision boundaries, which at the same time have a clear logical interpretation for the user.

The developed algorithm was experimentally investigated on model data and multispectral satellite images.

2 Similarity-based decision tree (SBDT) in pattern recognition problems

In this work, we consider a pattern recognition problem formulated as follows. Let us denote by $\boldsymbol{\Gamma}$ a general population of objects under consideration, and by point $x = x(a) = (x^{(1)}, \dots, x^{(m)}) \in \boldsymbol{R}^m$ a feature description of object $a \in \boldsymbol{\Gamma}$, where m is feature space F dimensionality. Let Y be a set of class labels. We consider a binary classification problem: $Y = \{-1, +1\}$, although the results can be extended to a multi-class scenario. Denote by $\mathbf{X}$ the set of feature descriptions of objects from $\boldsymbol{\Gamma}$. Let $y^*\colon \mathbf{X} \to Y$ be an objective function with values assigned to the points of the finite set (training or learning sample) $X_{train} \subset \mathbf{X}$. We need to build a decision function $f\colon \mathbf{X} \to Y$ which belongs to a given family; f should approximate y^* and minimize the estimate of misclassification probability for any point $\boldsymbol{x} \in \mathbf{X}$. Let X_{test} be another subset of $\mathbf{X}$ used for evaluating the performance of the decision function, $X_{test} \cap X_{train} = \emptyset$. Denote by $X = X_{train} \cup X_{test}$, and let d be the size of X and l be the size of X_{train}.

We propose a modification of DT in which instead of standard tests, more general statements of the type “object a is more similar to the set A than to the set B in feature subspace F' according to metrics μ” are examined in the internal nodes. Here A, B are subsets of learning sample, typically of small cardinality. In this work, we assume that each set A, B includes exactly one object (its description is called a support point). We also shall assume that $F'=F$ and metrics μ is the Euclidian metrics.

Suppose T is a binary tree with t internal nodes, and $\boldsymbol{A} = \{A^1, \dots A^p\}$, $\boldsymbol{B} = \{B^1, \dots, B^n\}$ are the sets of support points from positive and negative classes respectively. For each internal node v_i of the tree, $i = 1, .., t$, we define the tested statement as follows: "$x \in \mathrm{M}_1^{v_i}$", where $\mathrm{M}_1^{v_i}$ are the points from feature space, which are closer to A^{v_i} than to B^{v_i} (figure 2). Thus, the data is separated linearly (figure 3).

For any $x \in X$ we define matrix M_x with elements:

$$m(i,j) = \begin{cases} 1, & if\ \ \mu(x, A^i) - \mu(x, B^j) < 0 \\ 0, & otherwize \end{cases} \quad , i=1,\dots,p,\ j=1,\dots,n,$$

where μ is a metric in feature space F.

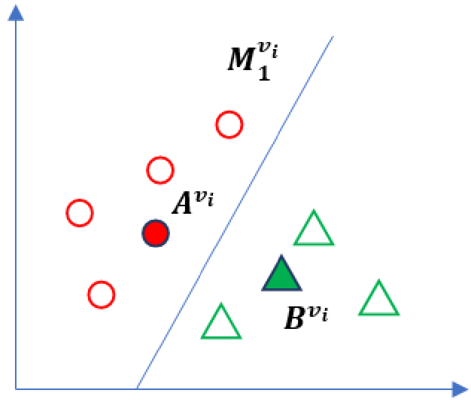


Figure 2. Example of singe separation.

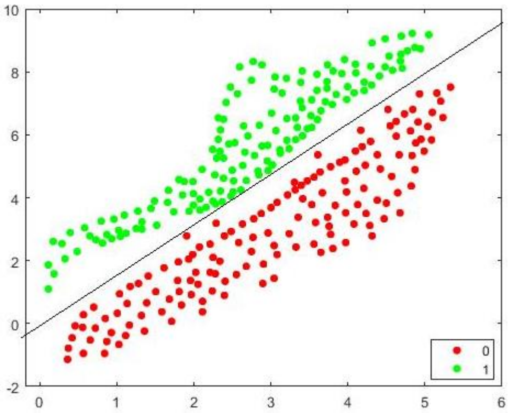


Figure 3. Splitting synthetic data by SBDT into two groups.

Let us transform matrix M_x into a vector $\overrightarrow{M_x}$ of the size pn by the reshaping procedure. Then each point x is described by vector $\overrightarrow{M_x}$ of the size pn. In this way, $X' = \{\bigcup_{x \in X} \overrightarrow{M_x}\}$ is a new feature representation of X.

Consider an example of feature transformation based on data shown in figure 2. As the number of all possible pairs equals one, matrix M_x has only one element. For objects represented by circles we have $M_x = 1$, and for objects represented by triangles, $M_x = 0$.

3 Support points selection

The proposed form of DT splits data points based on their relative position. The support points selection method should consider their informativity for a given sample X. In this work, three methods of support points selection were implemented.

The first approach uses Relief feature selection algorithm introduced in [13]. Let X_+ and X_- be subsamples of X_{train} of +1 and –1 class. After generating vectors $\overrightarrow{M_x}$ for all $x \in X$, applying Relief allows extracting the most informative sets $\boldsymbol{A} \subset X_+$, $\boldsymbol{B} \subset X_-$ reducing feature dimension.

The second way is based on Support vector machine (SVM) [14]. SVM builds a separating hyperplane with maximum distance (margin width) between points of different classes. Data points which are placed on the border of the margin are called support vectors. When SVM is trained on X_{train}, this set is divided into support and non-support vectors. In our approach, support vectors compose the sets of support points $\boldsymbol{A}$ and $\boldsymbol{B}$.

The last method is based on k-means clustering algorithm. We generate S subsamples of the size L from X and apply k-means (k=2) to each subsample to extract cluster's medoids which are considered as support points.

In addition, we used kernel k-means algorithm that uses kernel function instead of a scalar product. Kernel function implicitly transforms initial feature space into another space of larger dimensionality, where the configuration of data points is changed, often resulting in linearly separable form.

4 Construction of similarity-based decision tree

We chose CART algorithm [2] as an add-on method used in SBDT construction. The proposed SBDT algorithm can be represented with the following steps:

- Step 1. Find sets of support points $\boldsymbol{A}$ and $\boldsymbol{B}$ of classes +1, –1.
- Step 2. Compute vector $\overrightarrow{M_x}$ for all $x \in X$.
- Step 3. Build a decision tree in new feature space $\{\bigcup_{x \in X^d} \overrightarrow{M_x}\}$ by CART method.

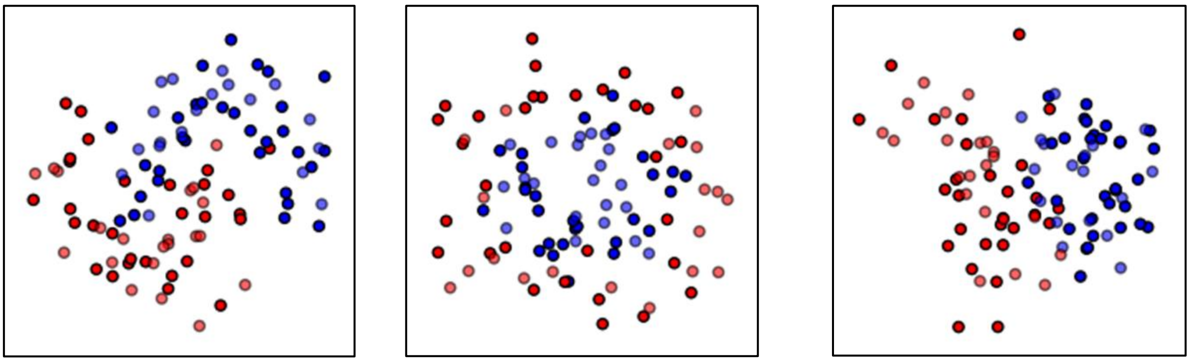
When using SVM or k-means based selection methods, we get linear computational complexity, while using Relief or kernel k-means results in quadratic complexity depending on the sample size.

5 Experimental study on model data

The proposed method was experimentally studied on three two-dimensional model datasets. Each dataset consists of 100 elements belonging to one of two classes. Figure 4 shows the original datasets: «Moons», «Circles» and «Linear». The parameter values for the CART algorithm were selected by default. Support points for the SBDT algorithm were selected with k-means algorithm. Figures 5 and 6 show the classification results obtained by CART and SBDT algorithms respectively. They also indicate classification mistake probability values obtained by the «moving exam» method. These examples demonstrate that the SBDT method provides higher classification accuracy compared to the CART algorithm. These figures also show that the decision boundaries constructed by the CART algorithm are coarser.

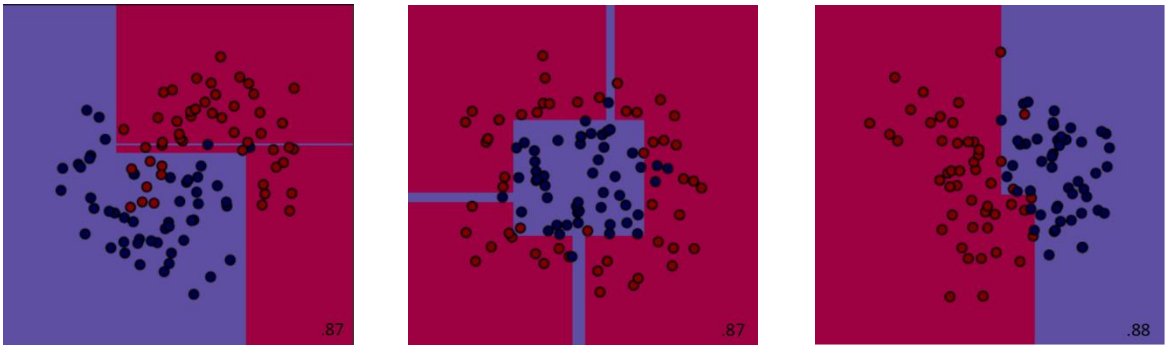
6 Experimental study on satellite data

To compare CART and SBDT algorithms performance on real data, Landsat 8 satellite image of Iskitim city (Novosibirsk region) was used. The RGB composite of this image is introduced in Figure 7a. Figure 7b shows the map of this image made by visual-instrumental methods containing representatives of six classes. It was used to train and to test the classifiers. 10% of the mapped pixels were used for training. As a result, classification accuracy of CART and SBDT algorithms is 88% and 98% respectively. Figure 7c shows SBDT image classification result.



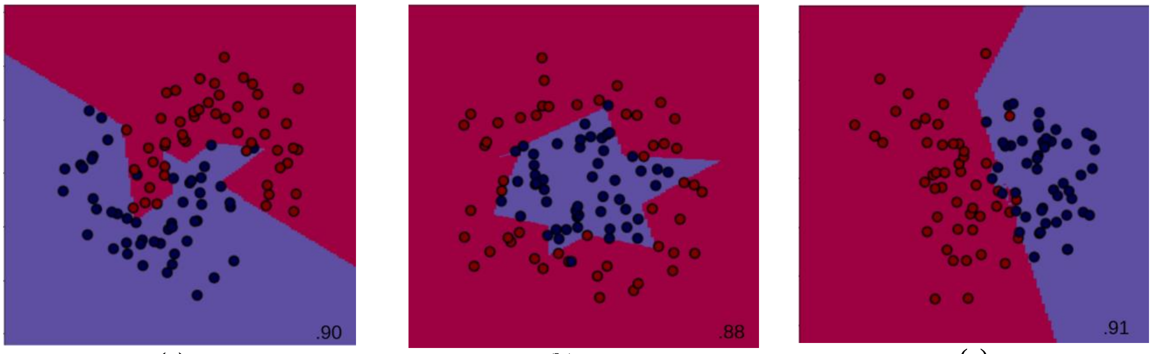
(a) (b) (c)

Figure 4. Initial model data sets: (a) – «Moons»; (b) – «Circles»; (c) – «Linear».



(a) (b) (c)

Figure 5. CART method classification result: (a) – «Moons»; (b) – «Circles»; (c) – «Linear».



(a) (b) (c)

Figure 6. SBDT method classification result: (a) – «Moons»; (b) – «Circles»; (c) – «Linear».

Thus, the proposed SBDT decision tree construction method based on the mutual objects similarity provides higher classification quality compared to the CART method not only on model, but also on real data. The SBDT method allows obtaining more accurate decision boundaries, which have a clear logical interpretation. The results of the experiments confirm the effectiveness of the method for multispectral satellite image classification.

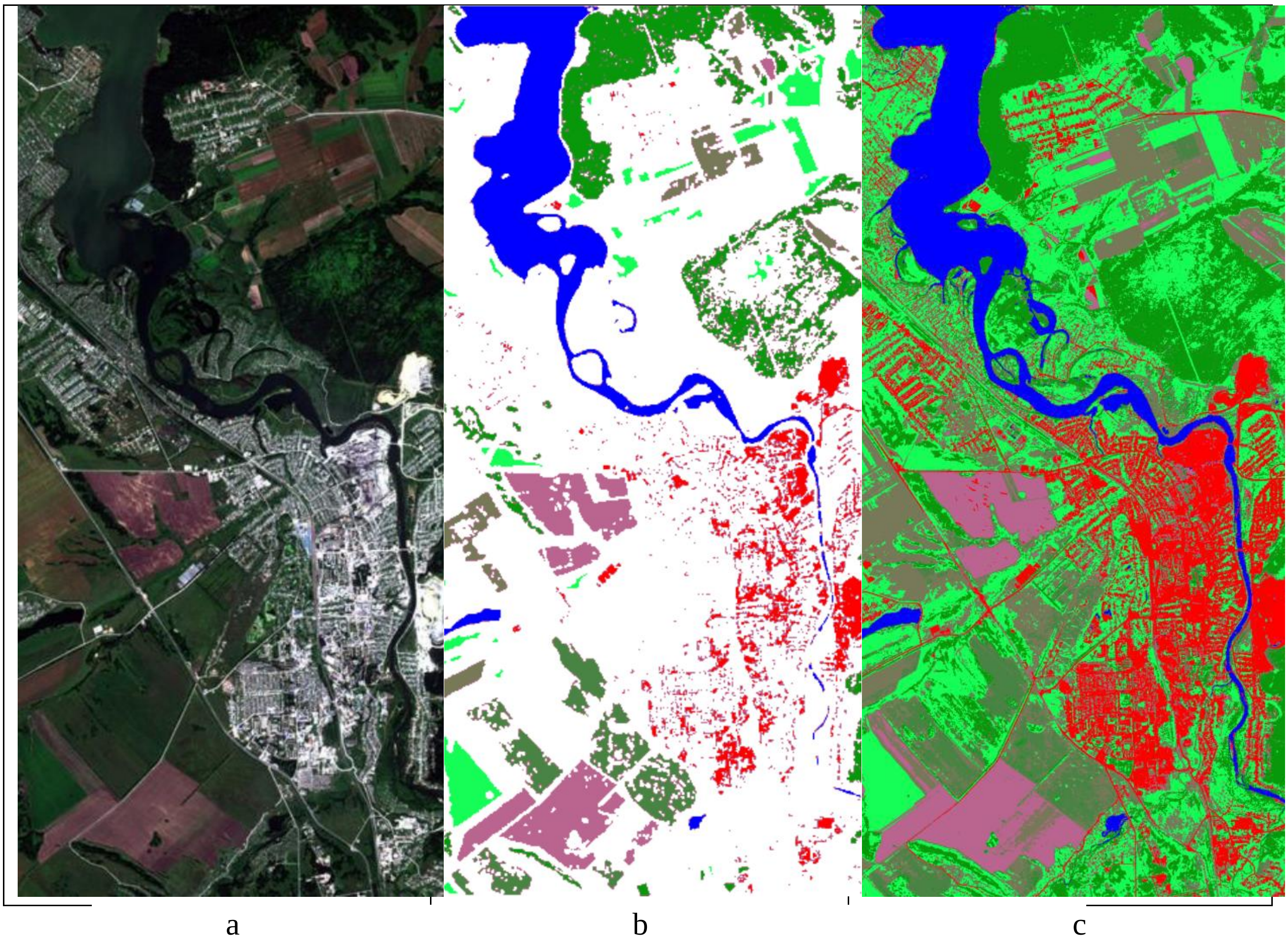


Figure 7. Landsat 8 satellite image classification results. a – RGB composite image, b – image map, c – image classification result with SBDT.

Acknowledgements. This work was supported by the RFBR (Grant No. 18-04-0063a) and the Interdisciplinary Integration Research program of the SB RAS for 2018-2020 (project No. 37).

References

[1] *Lbov G.S.* Logical Function in the Problems of Empirical Prediction Handbook of statistics. 1982. Vol. 2. Amsterdam: North-Holland Publishing Company. P. 479-492.

[2] *Breiman L., Friedman J.H., Olshen R.A., Stone C.J.* Classification and Regression Trees. New York: Routledge. 1984. 368 p.

[3] *Breiman L.* Random forests // Machine learning. 2001. Vol. 45(1). P. 5-32.

[4] *Schapire R.E.* The boosting approach to machine learning: An overview // Nonlinear estimation and classification. Springer. 2003. New York: Springer. P. 149-171.

[5] *Kotsiantis S.B.* Decision trees: a recent overview // Artificial Intelligence Review. 2013. Vol. 39(4). P. 261-283. doi: https://doi.org/10.1007/s10462-011-9272-4.

[6] *Berikov V.B., Lbov G.S.* Choice of optimal complexity of the class of logical decision functions in pattern recognition problems // Doklady Mathematics. 2007. Vol. 76. N. 3. P. 969-971.

[7] *Lbov G.S., Berikov V.B.* Recursive Method of Formation of the Recognition Decision Rule in the Class of Logical Functions // Pattern Recognition and Image Analysis. 1993. Vol. 3(4). P. 428-431.

[8] Bucy R.S., Diesposti R.S. Decision tree design by simulated annealing // ESAIM: Mathematical Modelling and Numerical Analysis. 1993. Vol. 27. N. 5. P. 515-534.

[9] Utgoff P.E., Brodley C.E. An incremental method for finding multivariate splits for decision trees // Proc. Seventh Int. Conf. on Machine Learning. 1990. P. 58-65.

[10] Murthy S.K., Kasif S., Salzberg S. A system for induction of oblique decision trees // Journal of artificial intelligence research. 1994. Vol. 2. P. 1-32.

[11] Menkovski V., Christou I.T., Efremidis S. Oblique decision trees using embedded support vector machines in classifier ensembles // Proc. 7th IEEE Int. Conf. on Cybernetic Intelligent Systems. 2008. P. 1-6.

[12] Sathe S., Aggarwal C.C. Similarity forests // Proc. 23rd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining. 2017. P. 395-403.

[13] *Kira K., Rendell L.* A Practical Approach to Feature Selection // Proc. Ninth Int. Conf. on Machine Learning. 1992. P. 249-256.

[14] *Cortes C., Vapnik V.* Support-vector networks // Machine learning. 1995. Vol. 20(3). P. 273-297.