CODE DESCRIPTIONS OF CLASSES METHOD FOR PATTERN RECOGNITION WITH MULTIPLE CLASSES

Alexander Dokukin, Vasily Ryazanov, Olga Shut

Abstract: *A new method for solving pattern recognition tasks with multiple classes is proposed that is based on the standard ECOC approach. The main modification involves so-called code description of the classes. Unlike class code of ECOC the code descriptions represent multisets of codes of classes' training objects. Another modification takes advantage of optimization of the initial set of binary subtasks. The method's theoretical substantiation is based on the ideas of algebraic and logical approach to pattern recognition. Its advantage is demonstrated with the model data set.*

Keywords: *pattern recognition, multiple classes, ECOC, correctness, algebraic approach, logical approach, code description of class.*

ACM Classification Keywords*: I.5 Pattern Recognition — I.5.0 General.*

MSC*: 68T10.*

Introduction

A pattern recognition task with multiple classes is considered hereinafter. The problem statement of its standard form from Zhuravlev, [1977a] is used.

Definition 1. *The pattern recognition task Z is defined as follows. Let $\tilde{S}_t(Z) = \{S_1, \ldots, S_m\}$ be a training sample described by real vectors $S_i = (a_{i1}, \ldots, a_{in})$, $i = 1, \ldots, m$. The sample is divided into l classes $K_1, \ldots, K_l$. The classification of the training sample objects is defined by information vectors $\alpha_i = (\alpha_{i1}, \ldots, \alpha_{il})$, where α_{ij} is a value of the predicate "$S_i \in K_j$". It is required to construct an algorithm A for calculating the classification of a new object S.*

If the classes do not overlap the classification of an object can be described by a single number $\alpha_i \in \{1, \ldots, l\}$ that is used further.

A pattern recognition task is called a task with multiple classes when $l > 2$. The case is distinguished by the fact that not all recognition methods are able to solve such tasks directly. Unlike nearest neighbors method or estimates calculating algorithm Zhuravlev, [1977a,b] such methods as support vector machine Cortes et al., [1995] or statistically weighed syndromes Kuznetsov et al., [1996] require additional stages. Firstly, a set of binary subtasks is solved directly and then their results are combined and interpreted in terms of initial set of classes. Some of the multistage approaches are quite obvious. They are one-vs-all Cortes et al., [1995] and one-vs-one Knerr et al., [1990]. Other examples can be found in Rocha et al., [2014]. There are also more general approaches. For example in ECOC (Error Correcting Output Codes) method Dietterich et al., [1995] arbitrary subdivisions of initial set of classes are used. Each class then achieves a binary code as well as each object. The decision on objects classification is made depending on closeness of its code to classes' codes. This method was further generalized in Allwein et al., [2000]. Binary subdivisions in that case consist only of a subset of initial classes and codes became ternary that allows including one-vs-one into general approach.

Thus, three general steps can be distinguished in the approach. Firstly, a set of binary subtasks is constructed. Random subdivisions are often used in that stage in general methods. Secondly, a recognition method is trained for each of the subtasks. Finally, the recognition results of a new object by the set of trained algorithms are interpreted in terms of initial classes.

A novel recognition method is described in the present article in which the first and the third steps are modified. The initial set of subtasks is reduced by solving an optimization task taking into account their recognition quality. The final interpretation is performed based on code descriptions of the classes (CDC), i. e. multisets of codes of its training objects instead of a single class' code.

The next chapter is devoted to the theoretical basis of the method. And the following one to the formal description of the method and experimental results.

Theoretical basis

The two main questions are answered in the present chapter. What requirements are to be satisfied by the reduced set of binary subtasks? And how to process CDCs taking into account importance of its elements and avoiding additional training?

The requirements for the set of subtasks are derived from considerations of correctness. Correctness of the final recognition method is required at least in the case of correctness of its first level algorithms.

Correctness is one of the key qualities considered considered in the algebraic recognition theory developed by academician Yu. I. Zhuravlev in 1970s Zhuravlev, [1977a,b]. It is understood as the ability of the algorithm to recognize given reference sample without errors. A great number of theoretical research of his students is devoted to the problem. Here are some formal definitions.

Definition 2. *Let's consider recognition task Z and reference set $\tilde{S}_r(Z) = \{S^1, \ldots, S^q\}$ with known classification $\alpha^t \in \{1, \ldots, l\}$, i. e. the predicate «$S^t \in K_{\alpha^t}$», $t = 1, \ldots, q$ holds. An algorithm A is called correct for the task Z and the reference set $\tilde{S}_r(Z)$ if $A(S^t) = \alpha^t$ for all $t = 1, \ldots, q$. Here $A(S^t) \in \{1, \ldots, l, \Delta\}$ is the algorithm's answer on S^t classification that is its class number or rejection Δ.*

Definition 3. *Let's consider recognition task Z and two disjoint subsets of the set of its classes $K^0 \in \{K_1, ..., K_l\}$, $K^1 \in \{K_1, ..., K_l\}$, $K^0 \cap K^1 = \emptyset$. A binary subtask of the task Z is defined as a recognition task Z' with the following properties: $\tilde{S}_t(Z') = \tilde{S}_t(Z) \cap (K^0 \cup K^1)$, $\tilde{S}_r(Z') = \tilde{S}_r(Z) \cap (K^0 \cup K^1)$, classes correspond to K^0 and K^1. The class K_i is called active in the binary subtask Z' if $K_i \in (K^0 \cup K^1)$. The binary subtask is called full if all initial classes are active in it. The number of active classes in the subtask $r(Z_i)$ is called its rank.*

The main condition of the existence theorem for the correct recognition algorithm Zhuravlev, [1977a,b] is pairwise non-isomorphicity of reference objects, i. e. existence of a training object for each pair of reference ones that distances of the reference objects to it differ in some feature subset: $\forall S^i, S^j \in \tilde{S}_r(Z), \exists S_k \in \tilde{S}_t(Z), p \in \{1, \ldots, n\}$, such as $|a_{kp} - a^i_p| \neq |a_{kp} - a^j_p|$. It and pairwise inequality of classes are the sufficient conditions for the existence of the correct algorithm in the algebraic closure of ECA (estimates calculating algorithms) family Dokukin, [2001]. Two-stage recognition scheme is defined as follows.

Definition 4. *Let's consider recognition task Z and W of its binary subtasks $Z_1, \ldots, Z_W$. An algorithm A_i solving the task Z_i, $i = 1, \ldots, W$ is called a first-stage algorithm. An algorithm A solving the task Z over outputs of the first-stage ones is called a second-stage algorithm.*

At that the vector $\gamma(K_i)$, there $\gamma(K_i)_j = 1$ if $K_i \in K^0_j$; $\gamma(K_i)_j = -1$ if $K_i \in K^1_j$; $\gamma(K_i)_j = 0$ otherwise; is called class K_i code, $i = 1, \ldots, l$, $j = 1, \ldots, W$. Class K_i rank $r(K_i)$ is the number of binary subtask in which it is active $r(K_i) = |\{\gamma(K_i)_j \,|\, \gamma(K_i)_j \neq 0,\ j = 1, \ldots, W\}|$.

Object's code $\gamma(S^t)$ is defined in a similar manner: $\gamma(S^t)_j = 1$ if $K_{A_j(S^t)} \in K^0_j$; $\gamma(S^t)_j = 1$ if $K_{A_j(S^t)} \in K^1_j$; $\gamma(S^t)_j = 0$ otherwise, $t = 1, \ldots, q$, $j = 1, \ldots, W$.

Let's consider recognition task Z with multiple classes and its binary subtasks $Z_1, \ldots, Z_W$. Evidently, if classes' codes are different in the set of subtasks and binary subtasks are full the ECOC algorith is correct. Indeed, reference vector codes are equal to their classes' codes due to correctness of algorithms $A_1, \ldots, A_k$, and since the codes are different there is no collisions possible.

The case of nonfull tasks is more difficult since objects of ignored classes can get arbitrary estimates. If then both full and nonfull subtasks are allowed the disadvantage can be fixed easily by adding to every nonfull subtask a full one in which all active classes of the former one are included into one metaclass and all inactive ones into another. Thus, the most difficult case is the case of nonfull subtasks. Let's describe the sufficient conditions for that case, but first let's change recognition scheme a little.

Let S be a reference object. Its estimate for the class K_j, $j = 1, \dots, l$, is calculated as

$$\Gamma_j(S) = \left| \left\{ t \,|\, K_j \in K_t^d,\ A_t(S) = d,\ d \in \{0,1\},\ t = 1, \dots, W \right\} \right| . \tag{1}$$

I. e. each algorithm A_i by assigning object S into one of the metaclasses increases its estimate for each of containing classes by one. Object S is than assigned to a class with maximum estimate. If there are multiple classes with the maximum estimate the object is rejected.

Statement 1. *The described scheme is equivalent to ECOC.*

Proof. Indeed, let's consider an arbitrary reference object S^t and an arbitrary class K_j. Distance between object's code $\gamma(S^t)$ and class' code $\gamma(K_j)$, $d(S^t, K_j)$, calculated as a number of different positions is equal to a number of subtasks in which the class is inactive or it is active but the algorithm is incorrect. At the same time number of votes achieved by $\gamma(S^t)$ for $\gamma(K_j)$, $v(S^t, K_j)$, is equal to a number of binary subtasks in which the class is active and the algorithm is correct. Thus, $d(S^t, K_j) = W - v(S^t, K_j)$, and minimum of the former is achived simultaniously with the maximum of the latter. The statement is proved. □

Definition 5. *The number $d(K_i, K_j)$ of subtasks in wich both classes K_i and K_j are active but belong to different metaclasses is called distance between the classes.*

$$d(K_i, K_j) = |\{t \in \{1, \dots, W\} \mid \gamma(K_i)_t \neq \gamma(K_j)_t,\ \gamma(K_i)_t \neq 0,\ \gamma(K_j)_t \neq 0\}| . \tag{2}$$

Let's consider recognition task Z and its nonfull binary subtasks $Z_1, \dots, Z_W$ of equal rank $r < l$. Let all first-stage algorithms $A_1, \dots, A_W$ be correct for the corresponding binary subtasks.

Theorem 1. *If for any two classes difference of their ranks is less than distance between them, i. e.*

$$r(K_j) - r(K_i) < d(K_j, K_i),\ \forall i, j = 1, \dots, l,\ i \neq j ,$$

then the second-stage algorithm A is correct.

Proof. Let's consider an arbitrary object $S^t \in K_i$. Since all the first-stage algorithms are correct S^t gets a vote for its class in all the binary subtasks where it is active, i. e. $\Gamma_i(S^t) = r(K_i)$. Let's consider another arbitrary class K_j. S^t gets a vote for K_j in two cases: if K_j is active and K_i is not, and if both classes are active and belong to the same metaclass. Thus, $\Gamma_j(S^t) \leq r(K_j) - d(K_i, K_j)$. Consequently, $\Gamma_i(S^t) - \Gamma_j(S^t) \geq r(K_i) - r(K_j) + d(K_i, K_j)$. By conditions of the theorem $r(K_i) - r(K_j) + d(K_i, K_j) > 0$ so $\Gamma_i(S^t) > \Gamma_j(S^t)$. Q.E.D. □

In case first-stage algorithms are incorrect there are two possibilities. If there exist only few errors the Theorem 1 can be modified to correct those by requiring greater distances between classes. But if errors are numerous the Theorem 1 becomes irrelevant and correctness can be achieved only by considering individual codes of the objects. And nonfull tasks become not very useful too. Indeed, if an algorithm solves some binary subtask incorrectly it nevertheless assigns objects of inactive classes to one of the metaclasses. The reference objects of an inactive class will most probably be assigned to the metaclasses unequally. Thus, by including K_j into the binary subtask the algorithms quality can be decreased a little or even improved.

Considering the initial question the following conditions will be required from the reduced set of subtasks. Firstly, only the full subtasks will be considered. Secondly, the distances between classes' codes will be maximized.

The second question refers to a method for combining first-stage results. It can be made for example by training new algorithm in an objects' codes feature set. But the additional training would require a separate sample and would

complicate an algorithm. On the other hand the combination of a binary training information and the lack of a training stage is typical to the logical approach to recognition and to production expert systems Giarratano et al., [2007] in particular. At that the information can be represented either as description of objects in form of logical formulas and rules, or by precedents as an enumeration of objects and their classes. In the former case resolution methods are used and in the latter one recognition methods, for example the one described in Krasnoproshin et al., [1998]. There are also tasks in which both approaches are used simultaniously, such as medical diagnostics tasks Ablameiko et al., [2011]. At that object resolution method allows using precedent information for logical inference and fuzzy object resolution allows weighing the precedents.

The rest of the chapter is devoted to describing theoretical basis of the method starting with redefinition of the recognition task Z using terms customary to the logical approach Ablameiko et al., [2011]:

Definition 6. *Let X be a subset of objects of arbitrary nature. Let subsets $X_1, \ldots, X_l$ called classes be defined in the set as well as the initial information I_0 of classes $X_1, \ldots, X_l$. It is required to find an algorithm A defined at the whole set X that calculates a result in terms of belonging to a classes $X_1, \ldots, X_l$ for an arbitrary object $x \in X$ using the information I_0.*

Let $S = \{s_1, \ldots, s_n\}$ be a set of all features in universe of discourse of the task Z, where $n < \infty$; D_j be a set of its values $s_j \in S$. Without loss of generality let's say that $D_j = \{0, 1, \ldots, |D_j| - 1\}$ and denote

$$D = \left\{0, 1, \ldots, \max_j \{|D_j| - 1\}\right\} = \{0, 1, \ldots, k-1\}.$$

It is supposed that all features possess values from the set D, where $k \neq 1$.

Definition 7. *The object is defined as correspondence*

$$p(s_1, \ldots, s_n) = (D_1^p, \ldots, D_n^p),$$

where $D_j^p \subset D$ is a set of values of feature $s_j \in S$ of object p, and $D_j^p \neq \varnothing$. Objects are called equal if $\forall j \; D_j^p = D_j^q$. If $|D_j^p| = 1$ the feature values is called known. If instead $|D_j^p| > 1$ the object p is considered a set of objects such that s_j enumerates D_j^p and the rest features coincide with corresponding features of p. A set of objects is also called a collection.

Definition 8. *An object is called normalized if all its features are known. A collection is called normalized if all its objects are normalized. Let's denote X^{norm} the set of all normalized objects of universe of discourse of the task Z: $X^{norm} = D^n$. An object for which one feature is known and the rest are undefined is called feature-object:*

$$p^j(s_1, \ldots, s_n) = (D, \ldots, D, \left\{d_j^{p^j}\right\}, D, \ldots, D).$$

Let $V \subset X$, $W \subset X$ be some arbitrary collections. Product of objects p and q is defined by its features $D_j^{pq} = D_j^p \cap D_j^q$. Let's consider the following operations over objects and collections:

1. negation: $\overline{V} = X^{norm} \backslash V$;
2. multiplication: $V \wedge W = \bigcup\limits_{p \in V, q \in W} \{pq\}$, where objects with $D_j^{pq} = \varnothing$ are not included;
3. addition: $V \vee W = V \bigcup W$.

Statement 2. *The set of operation $\{\neg, \wedge, \vee\}$ over collections is full.*

The proof is described in Shut, [2012, 2014].

Thus, an algebra of objects $G =< \rho(X), \{\neg, \wedge, \vee\} >$ is defined where $\rho(X)$ is set of all possible collections. An algebra of normalized objects $G^{norm} =< \rho(X^{norm}), \{\neg, \wedge, \vee\} >$ is defined also, where $\rho(X^{norm})$ is the collection of normalized objects. Let's consider object resolution method for the task Z.

Definition 9. *An object r is called object resolvent for objects p and q, if feature values of r satisfy the condition:*

$$D_j^r = \begin{cases} D_j^p \bigcup D_j^q, j = h \\ D_j^p \bigcap D_j^q, j \neq h \end{cases},$$

where h is index of an arbitrary feature $s_h \in S$. The operation of constructing object resolvent is denoted $r = Or_h(p, q)$.

The appropriateness of object resolution method for constricting new objects based on precedent information is shown in Shut, [2012, 2014]. And the algorithm looks as follows. Let's consider class X_i and determine whether an object x belongs to it. Let's denote $X_i^0 = X^0 \bigcap X_i$ and A_1 the algorithm itself:

Step 1. Set $Y_i = X_i^0$.

Step 2. If $x \in Y_i$ go to step 6, otherwise go to step 3.

Step 3. Get from Y_i an unconsidered triplet (p, q, h), where p and q are objects and h is feature index. If there are no unconsidered triplets go to step 6.

Step 4. Calculate $r = Or_h(p, q)$. If there exists such index j $D_j^r = \varnothing$ go to step 3.

Step 5. If $r \notin Y_i$ set $Y_i := Y_i \bigcup \{r\}$. Go to step 2.

Step 6. Stop.

The algorithm A_1 is applicable for both direct and reverse inference. Direct inference means that if the algorithm has stopped by achieving the object x it implies that the collection $Norm(X_i^0)$ contains it. Thus, $x \in X_i$. Reverse inference states the opposite. If the algorithm has stopped by achieving object o it implies that $Norm(Y_i) = X$, i. e. Y_i potentially contains all objects of X. That is why $x \notin X_i$. If either result is not achieved it means that conclusions about objects x belonging to the class X_i or not can't be achieved with the algorithm.

The next step is to apply the resolution method to the multistage scheme. Let's consider the new task of W binary features where codes of the initial objects formed by the set of first-stage algorithms become objects. The objects are included to the set with their copies so that objects weights can be calculated as corresponding share of the code in class' code description. It allows applying a fuzzy object resolution method to the problem, i. e. object resolution method in case I_0 is described by fuzzy logic functions.

Let E be an arbitrary set. Let's define characteristic function $\mu_E(x)$ which value describes membership function of an element x to the set E: $\mu_E(x) \in [0, 1]$. Let E_1, E_2 be fuzzy subsets of E. Let's consider the following fuzzy logic operations Kofman, [1982]:

1. addition: $\mu_{\overline{E_1}}(x) = 1 - \mu_{E_1}(x)$,
2. intersection: $\mu_{E_1 \bigcap E_2}(x) = \min\{\mu_{E_1}(x), \mu_{E_2}(x)\}$,
3. union: $\mu_{E_1 \bigcup E_2}(x) = \max\{\mu_{E_1}(x), \mu_{E_2}(x)\}$.

Every collection $V \subset X$ is assigned a characteristic function $\mu_V(p)$ that describes membership function of an object p. It is defined as the corresponding share of the objects. Let N^p denote total number of instances of p in X^0 and N_i^p denotes number of instances of p in X_i^0. Thus, $N^p = \sum_{i=1}^{l} N_i^p$. Let's define $\mu_{X_i}(p)$ as

$$\mu_{X_i}(p) = \frac{N_i^p}{N^p} .$$

so that $\mu_{X_i}(p) \in [0, 1]$.

There are fuzzy logic analogues of the resolution method. One of them is described in Lee, [1972]. Let's describe an algorithm of solving the problem Z by using fuzzy object resolution method.

The algorithm A_1^f:

Step 1. The algorithm A_1 is applied to X. Let Y_i be the set of objects that are considered belonging to X_i, $i = 1, ..., l$ by applying it, i. e. $\forall p \in Y_i$, $\mu_{X_i}(p) > t$, t being a given threshold $t \in [0, 1]$.

Step 2. For every object $p \in X \setminus \left(\bigcup_{i=1}^{l} Y_i \right)$ the steps 3–4 are applied.

Step 3. For each class X_i the value of $\mu_{X_i}(p)$ is calculated.

Step 4. Let $\{\mu_{Y_v}(p)\} = \max\limits_{i}\{\mu_{Y_i}(p)\}$, $w = \max\limits_{v}\{v\}$. If $\mu_{X_w}(p) > t$ the object p is added to Y_w:

$$Y_w = Y_w \bigcup p.$$

Step 5. Stop.

The results of the algorithm A_1^f are interpreted as this. If $p \in Y_i$ when it stops then $p \in X_i$. Thus, the algorithm A_1^f assignes the object p to the class with maximum membership function.

The difference between fuzzy object resolution method and direct comparison of object's and class' codes can be seen in the following example.

Example 1. *Let's consider recognition task with 4 classes and let's define the binary subtasks as every pair of classes against every other (see. Fig. 1).*

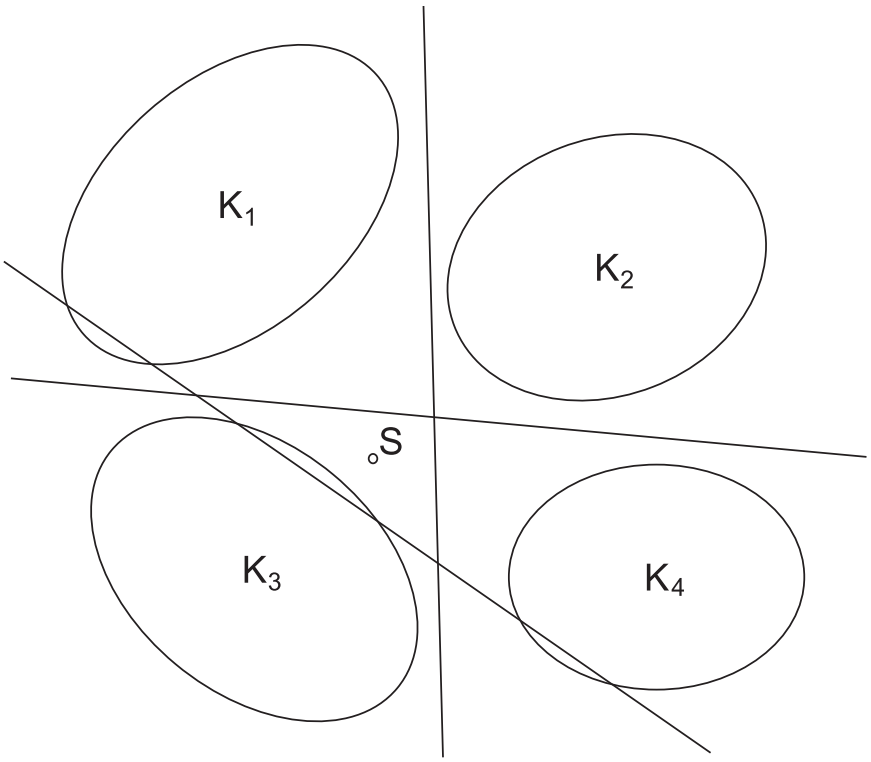

Figure 1: Interpretation of the first-stage results.

There are three of those tasks $\{1, 2\} - \{3, 4\}$, $\{1, 3\} - \{2, 4\}$, $\{1, 4\} - \{2, 3\}$. Consequently the classes are assigned the following codes: $\gamma(K_1) = (1, 1, 1)$, $\gamma(K_2) = (1, 0, 0)$, $\gamma(K_3) = (0, 1, 0)$, $\gamma(K_4) = (0, 0, 1)$. Lines in the Fig. 1 demonstrate trained linear recognition algorithms solving those binary subtasks. The first two tasks are solved correctly but the third one contains a vast number of errors. For example the whole second class is assigned a wrong metaclass and in the other three classes a part of objects (let's say 10 % for clarity) is treated wrong too.

Now let's consider the object S. It's code is $\gamma(S) = (0, 1, 1)$ and it is equally distant from the three nearest class codes: $\gamma(K_1) = (1, 1, 1)$, $\gamma(K_3) = (0, 1, 0)$, $\gamma(K_4) = (0, 0, 1)$. Thus, the object's class is impossible to tell though it is likely to be class 3.

However, the object resolution method will assign S to the third class. Indeed, first class precedents get codes $(1,1,1)$ (90%) and $(1,1,0)$ (10%), second class ones are $(1,0,1)$, third class ones are $(0,1,0)$ (90%), $(0,1,1)$ (10%), and fourth class ones are $(0,0,1)$ (90%), $(0,0,0)$ (10%).

The algorithm has its drawbacks. For example if a code is not presented in the training set it will not be recognized. To compensate for the said effect a modification of it is proposed. An exact description is given in the following chapter but the idea is to use a monotonically decreasing function in the neighborhood of each code to produce estimates for the missing ones.

Implementation

This chapter is devoted to the formal description of the proposed Code Description of Classes Method as well as the set of experiments demonstrating its quality. The method is based on the results of the previous chapter as well as other researches of the ECOC approach. In Berger, [1999] the probability of mixing of results of different classes was analyzed in case of random binary subtasks. It was shown that the algorithm is inclined to err if the codes of corresponding classes are close. In Dietterich et al., [1995] it is also stated that good performance of the algorithm requires both separability of metaclasses and codes. We will maximize distances between class codes that covers the mentioned conditions as well as conditions of the Theorem 1 in case of full subtasks.

The initial set of binary subtasks is generated randomly. After that an optimization task is solved to maximize distances between initial codes. The modification of the code set is made by weighing its components. Let $\|\alpha_{ij}\|_{l\times W}$ be a code matrix where l is the number of initial classes and W is the number of the binary subtasks. Then the task is described by the formula

$$\sum_{j=1}^{W} |\alpha_{\nu j} - \alpha_{\mu j}| x_j \geq y;\ \forall \nu,\ \mu;\ \nu > \mu;\ \nu,\ \mu = 1,\ldots,l\,,$$

$$\sum_{j=1}^{W} x_j = W\,,$$

$$y \to \max\,.$$

It should be noted that the optimal weights are often zeroed that allows reducing number of subtasks. The second role of the weights is to modify distance function by considering their importance

$$d(S^t, K_j) = d(\gamma(S^t), \gamma_j) = \sum_{j=1}^{W} |\alpha_{ij} - \beta_j| x_j\,, \tag{3}$$

where β is code of the object S^t, $\gamma(S^t) = \beta_j$.

After the subtasks are generated and first-stage algorithms are trained, code class descriptions are formed. They are multisets of codes of training objects calculated by the same first-stage algorithms. Let class K_j be described by a set of pairs $\{\gamma_{ji}, \nu_{ji}\}$, $i = 1,\ldots,W_j$, where $\gamma_{ji} = \gamma(S)$, $S \in K_j \cap \tilde{S}_t(Z)$ are codes of objects of class K_j; ν_{ji} is the share of code γ_{ji} in the description of class K_j, $\nu_{ji} = \frac{|\{S|S\in K_j \cap \tilde{S}_t(Z), \gamma(S)=\gamma_{ji}\}|}{|K_j \cap \tilde{S}_t(Z)|}$, W_j is the number of different codes in the description of class K_j. The estimate of an arbitrary object S for the class K_j is then calculated by formula

$$\Gamma_j(S) = \sum_{i=1}^{W_j} \nu_{ji} \frac{1}{(1 + d(\gamma(S), \gamma_{ji}))^2}\,,$$

where $d(\gamma_1, \gamma_2)$ is either Hamming distance between codes γ_1 and γ_2, or the distance defined in (3).

For the experimental purposes the two modifications were tested against simple ECOC separately and in combination. Thus, four methods were involved in tests. The tests were performed with a model data set of 12 classes. At that

the random sets of binary subtasks of given cardinality were generated and the same ones were used with each of the methods. It should be mentioned that binary subtasks were solved with SVM implementation from scikit-learn package Pedregosa et al., [2011].

The model task is specially designed to complicate separation of the classes. 20 normally distributed samples were generated on the flat with centers ordered in five columns and four rows. After that some pairs or triplets of the said samples were joined into total of 12 classes (see Fig. 2).

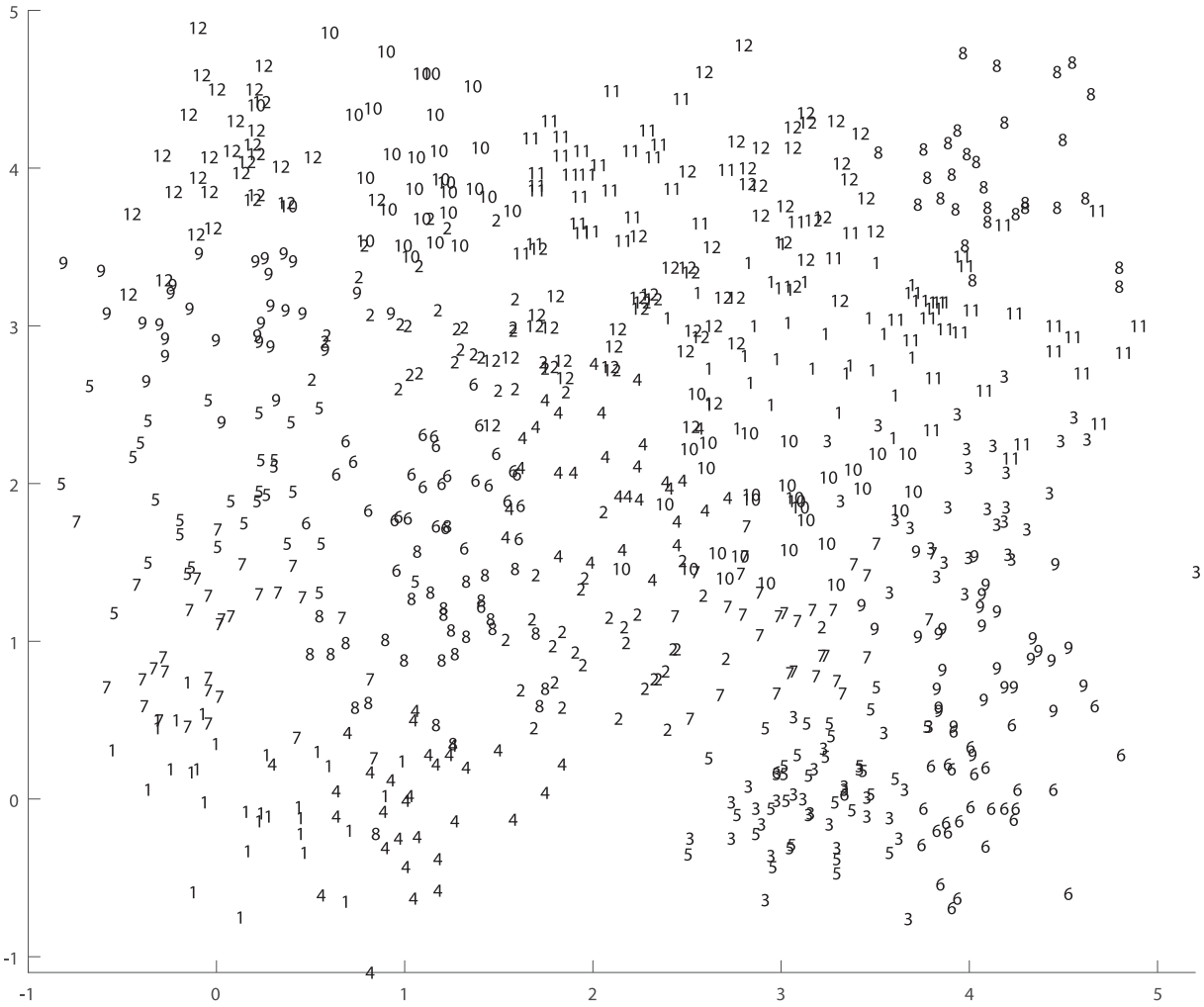

Figure 2: The model sample of 12 classes.

The results are show in Table 1. Though the experimental set is small it demonstrates one particular advantage of the CDC method. Its result in that task is better than that of ECOC regardless the number of initial subtasks. Even though separate application of its two modifications can provide better results, the same cannot be told about them. Indeed, optimization tend to provide better results with a greater number of subtasks while the code descriptions work better with less of them.

Table 1: The experimental results.

No. of subtasks	ECOC	Optimization	Code descriptions	CDC
20	68.9	66.7	70.8	70.0
40	69.9	68.4	71.6	70.9
60	69.5	71.1	71.2	72.5
80	69.6	71.3	70.0	71.1

As a conclusion it can be stated that the article gives only a first idea of the method. Its implementation is yet far from ideal. For example the form of monotonically decreasing function used in the algorithm is pure heuristic with no theorerical or experimental basis to support the choice. Although it seem to work in some cases.

Acknowledgements

The paper is supported by RFBR, grant No. 15-51-04028, and BRFBR, grant No. F15PM-037.

Bibliography

Zhuravlev Yu. I. Correct algebras other sets of incorrect (heuristical) algorithms I (in Russian)// Cybernetics. — 1977. — No. 4. — Pp. 14–21.

Zhuravlev Yu. I. Correct algebras other sets of incorrect (heuristical) algorithms II (in Russian)// Cybernetics. — 1977. — No. 6. — Pp. 21–27.

Cortes C., Vapnik V. Support-vector networks // Machine Learning. — 1995. — Vol. 20, No. 3. — Pp. 273–297.

Kuznetsov V. A., Senko O. V., et al. Recognition of fuzzy systems by method of statistically weighed syndromes and its using for immunological and hematological norm and chronic pathology // Chemical Physics. — 1996. — Vol. 15, No. 1. — Pp. 81–100.

Knerr S., Personnaz L., Dreyfus G. Single-layer learning revisited: A stepwise procedure for building and training neural network // Neurocomputing: Algorithms, Architectures and Applications, NATO ASI. — 1990. — Vol. 68. — Pp. 41–50

Rocha A., Goldenstein S. K. Multiclass from binary: Expanding one-versus-all, one-versus-one and ECOC-based approaches // IEEE Trans. on Neural Network. — 2014. — Vol. 25, No. 2. — Pp. 289–302.

Dietterich T. G., Bakiri G. Solving multiclass learning problems via error-correcting output codes // Journal of Artificial Intelligence Research. — 1995. — No. 2. — Pp. 263–286.

Allwein E., Shapire R., Singer Y. Reducing multi-class to binary: A unifying approach for margin classifiers // Journal of Machine Learning Research (JMLR). — 2000. — Vol. 1, No. 1. — Pp. 113–141.

Dokukin A. A. The construction of a recognition algorithm in the algebraic closure // Computational Mathematics and Mathematical Physics. — 2001. — Vol. 41, No. 12. — Pp. 1811–1815.

Giarratano J., Riley G. Expert Systems: Principles and Programming. — 4th edition. — Course Technology Inc, 2004. — 842 pages.

Krasnoproshin V. V., Obraztsov V. A. Trained recognition as choice problem (in Russian) // Digital image processing. — Minsk: ITK, 1998. — Pp. 80–94.

Ablameiko S. V., Krasnoproshin V. V., Obraztsov ï£¡. ï£¡. Models and technologies of pattern recognition with application to data mining (in Russian) // BSU Herald. Series 1, Physics, Mathematics, Informatics. — 2011. — No. 3. — Pp. 62–72.

Shut O. V. Pattern recognition method based on logical and precedent models (in Russian) // Informatics. — 2012. — No. 3. — ï£¡. 35–50.

Shut O. V. Synthesis of pattern recognition algorithms in discrete spaces of finite dimension (in Russian) // BSU Herald. Series 1, Physics, Mathematics, Informatics. — 2014. — No. 1. — Pp. 56–62.

Koffman A. Introduction to the Theory of Fuzzy Subsets. — Academic Pr, 1975. — 432 pages.

Lee R. C. T. Fuzzy Logic and the Resolution Principle // Journal of the ACM. — 1972. — Vol. 19, No. 1. — Pp. 109–119.

Berger A. Error-correcting output coding for text classification // In Proceedings of IJCAI: Workshop on machine learning for information filtering. — 1999.

Pedregosa F., Varoquaux G., Gramfort A, Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., Vanderplas J., Passos A., Cournapeau D., Brucher M., Perrot M., Duchesnay E. Scikit-learn: Machine Learning in Python // Journal of Machine Learning Research. — 2011. — Vol. 12. — Pp. 2825–2830.

Authors' Information

Alexander Dokukin *— Dorodnicyn Computing Centre, Federal Research Center "'Computer Science and Control"' of Russian Academy of Sciences, researcher, 40 Vavilova St., Moscow, 119333, Russian Federation; e-mail: dalex@ccas.ru.*
Major Fields of Scientific Research: Algebraic Approach to Pattern Recognition.

Vasily Ryazanov *— Moscow Institute of Physics and Technology, post-graduate, 9 Institutskiy per., Dolgoprudny, Moscow Region, 141700, Russian Federation; e-mail: vasyarv@mail.ru.*
Major Fields of Scientific Research: Machine Learning.

Olga Shut *— Belarusian State University, lecturer, 4 Nezavisimosti avenue, Minsk, 220030, Republic of Belarus; e-mail: olgashut@tut.by.*
Major Fields of Scientific Research: Pattern Recognition, Artificial Intelligence.