Pattern Recognition Letters 100 (2017) 74–82

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Improving multiclass classification using neighborhood search in error correcting output codes

Niloufar Eghbali, Gholam Ali Montazer*

Information Technology Department, School of Engineering, Tarbiat Modares University, P.O. Box 14115-179, Tehran, Iran

ARTICLE INFO

Article history:
Received 13 January 2017
Available online 28 September 2017

Keywords:
Multiclass classification binarization
Sparse error correcting output codes
Neighborhood search

ABSTRACT

Error Correcting Output Code (ECOC) is an effective approach for multiclass classification problems. This method decomposes a multiclass problem to many binary sub-problems and makes a dichotomizer for each sub-problem. It then tries to classify samples by combining outputs of all dichotomizers. One of the main points in ECOC method is to construct an ensemble of independent binary classifiers. Many studies have been conducted to design an optimal ECOC matrix. However, most of these methods aim to construct an ECOC code Matrix without considering the relations between data to design an ensemble of binary classifiers. In this study, a new method is presented based on ECOC which improves the performance of sparse ECOC by considering the neighborhood of samples. The proposed method is evaluated using 16 UCI datasets. The results indicate that our method not only significantly improves the classification accuracy compared to other commonly used ECOC based methods, but it also can result in a lower number of classifiers in comparison with random dense ECOC with the same accuracy.

1. Introduction

The purpose of many real world classification tasks is to distinguish instances of different classes. The complex nature of Multiclass classification causes more errors than binary classification, as it needs to categorize data in more groups, increasing the classification intricacy [16].

Multiclass classification is a major problem in machine learning and pattern recognition. There are two main approaches to solve this problem: the first approach directly employs a multiclass classifier such as a decision tree; however, the second approach converts the multiclass problem to multiple binary sub-problems and combines the results of these problems to predict the class of a new sample (Liu et al. 2016). Most of the existing classification algorithms are designed to solve binary classification or they perform more efficiently in such problems (Bagheri 2015). For this reason, in this study we focus on using binary classifiers to solve multiclass classification problems. Several methods have been developed to cast a multiclass problem to a set of binary classification sub-problems. The most widely used methods are one-versus-one (OVO), one-versus-all (OVA), and Error Correcting Output Codes [17]. In one-versus-one, a binary classifier is constructed for each possible pair of classes. In the one-versus-all, a binary classifier is trained for each class to distinguish the samples of that class from the samples of the remaining classes. Finally, in both of the mentioned methods, the final class prediction is derived from the combination of the prediction of each binary classifier.

ECOC is a general framework for class binarization and has been successfully used in many applications, such as disease diagnosis [7,15], network traffic classification [19], text classification [9], image vision applications [5,6], facial action unit recognition [18] and face recognition [13]. The ECOC matrix encodes the procedure of extension of binary classifiers to multiple class classification. It determines how to generate binary classifiers and how to combine the outputs of the generated classifiers to achieve the final prediction. Diverse and accurate binary classifiers can guarantee the success of ECOC method. Diversity of base classifiers makes the classifier's error independent, thus helping the ECOC correct the errors effectively [5].

According to the literature, many studies have dealt with improving the ECOC performance. Most of these methods try to generate the optimal code matrix as a key component of the ECOC method to construct an efficient ensemble of binary classifiers. Most of the coding designs are pre-defined and do not consider the data relations in constructing the code matrix [10,21,28–31]. Exhaustive search for ECOC code matrix is an NP-hard problem with the number of classes [24]. As a solution, many heuristic methods are proposed to search for the optimal ECOC matrix [3,8,14,25]. Many researchers, however, confirmed the acceptable performance of randomly generated code matrix. They agree that more intricate methods have only marginal effects on testing error [2].

* Corresponding author.
E-mail address: montazer@modares.ac.ir (G.A. Montazer).

https://doi.org/10.1016/j.patrec.2017.09.037
0167-8655/

Table 1
Description of notations.

Notation	Description
N_c	The number of classes in the problem
L	Number of classifiers
M	ECOC Matrix
f_i	ith classifier
y_i	Code word corresponding to class i
x	Test sample code word
c_i	ith class
N	Number of experiments
K	Number of competing methods

The initial ECOC (known as dense ECOC) tries to divide the classes into two sub-groups for each binary classifier. Hence, each binary classifier learns to discriminate the samples of its own sub-groups. An extension of this standard ECOC approach was proposed by Allwein et al. [1] known as sparse ECOC. The sparse ECOC allows ignoring some of the classes. Hence the OVO approach can be formulated as a sparse ECOC and OVA approach can be considered as a dense ECOC design [10].

With regards to sparse methods such as one-versus-one, it is clear that each classifier must assign any pattern to either of two classes. This problem is one of the major drawbacks of this approach, arising from meaningful outputs of binary classifiers which are not trained to identify the class of the instance to be classified [5,21] (Bagheri and Gao, 2015)

If the classes are independent, it is improbable that many classifiers vote the same wrong class, but the probability of such a situation increases if there are similarities among the classes [21].

Some strategies have been proposed to tackle this problem, such as loss-weighted decoding strategy [32], combining local and global learners (Bagheri and Gao, 2015) and undirected cyclic graph based classifier [22]. Most of these methods are designed only for pair-wised approaches such as one vs one approach. However, in this study a general method is proposed to eliminate the adverse effect of irrelevant classifiers in sparse approaches. We propose a novel method based on sparse ECOC to effectively combine binary classifiers which are more effective in classification of each new sample.

The rest of this paper is organized as follows: Section 2 briefly discusses the ECOC framework including the coding and decoding approaches. Section 3 presents a detailed description of a novel method based on ECOC. Section 4 then evaluates the presented model. Finally, Section 5 states the conclusion of the paper.

2. Error correction output codes

In this section, we briefly introduce the ECOC framework. The notation used in this paper is presented in Table 1.

ECOC framework aims to represent each classifier by a binary code. Thus, the basis of this method is to design a binary code for each classifier. For this reason, the ($N_c \times L$) -dimensional matrix (coding matrix) M with values {−1, 1} will be constructed [33]. Each column of M describes a binary classifier known as dichotomizer. Each dichotomizer separates the set of classes into two meta-classes. The classes with the value of +1 in the coding column form a positive meta-class, while the classes with the value of −1 in the coding matrix form a negative meta-class. The purpose of the dichotomizer is to distinguish between the instances of positive and negative meta-classes.

Fig. 1 illustrates an ECOC for a 5-class classification problem. Each column of this matrix is called classifier code which characterizes a classifier. Hence, this matrix consists of 9 classifiers (number of the columns). For instance, the classifier f_5 separates the samples which are the members of the first and third classes as positive samples (white regions) from the samples belonging to the second, fourth and fifth classes as negative samples (black regions). To determine the class of a new sample, the sample is firstly classified by each classifier so that the outputs generate a 9-bit code word. Then, the obtained code word is decoded to one of the classes. Among the decoding strategies, the distance based approaches such as hamming method are the most simple and effective ways to decoding the code word. Finally, the class whose corresponding code has the minimum distance to the obtained code word is considered as a predicted class for the new sample. In the Figure, class4 is assigned to test code word according to minimum Hamming distance.

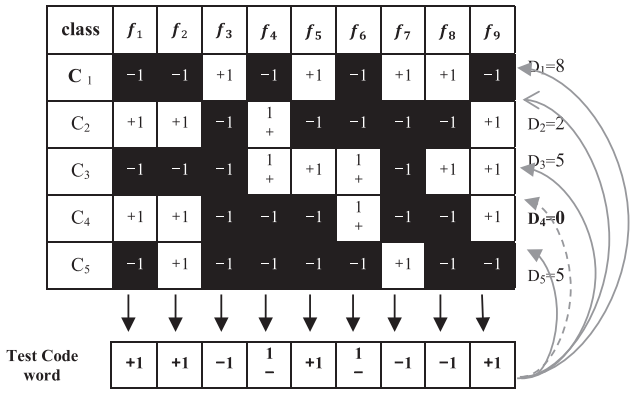


Fig. 1. ECOC design for 4-class problem. The test sample is classified to fourth class according to Hamming distance.

Two processes of coding the code matrix and decoding the output code word play a key role in ECOC Framework to function highly successful. Some of the state-of-the-art strategies in the formerly-mentioned processes are discussed as follows:

2.1. Coding design

Generally, a proper code matrix should fulfill two conditions. First, each row is needed to be completely separable from other rows (implying that the hamming distance between each pair of code words should be a nonzero value). In addition, each column should be uncorrelated with other columns to ensure the independency of outcome classifiers. For this purpose, the hamming distance between a column and each of the other columns should be as large as possible. Moreover, each column and complement of other columns should have the maximum hamming distance from each other.

It is essential to have both discussed properties in ECOC matrix. Row separation property directly affects the error correction ability of this method and the column separation results in diversity of errors, indirectly reducing the overall error. If two columns have the same codes, the corresponding classifiers make the same mistakes simultaneously. Furthermore, if many simultaneous errors occur in the classification process, ECOC will not be able to correct the errors [33].

ECOC coding designs can be categorized into two main subgroups: problem-independent and problem-dependent designs. Problem-independent approaches such as One-versus-one, one-versus-all, sparse random, and dense random, create the ECOC code matrix without including data relation and distribution in design, whereas problem-dependent methods use the data relations and problem conditions to attain the code matrix (Escalera, Pujol and Radeca, 2010). Most of the developed methods have focused on the row and column separation to construct an ideal code matrix [2]. The number of possible distinct columns in k-class problem is $2^{k-1} - 1$, so the search for the optimal code word is

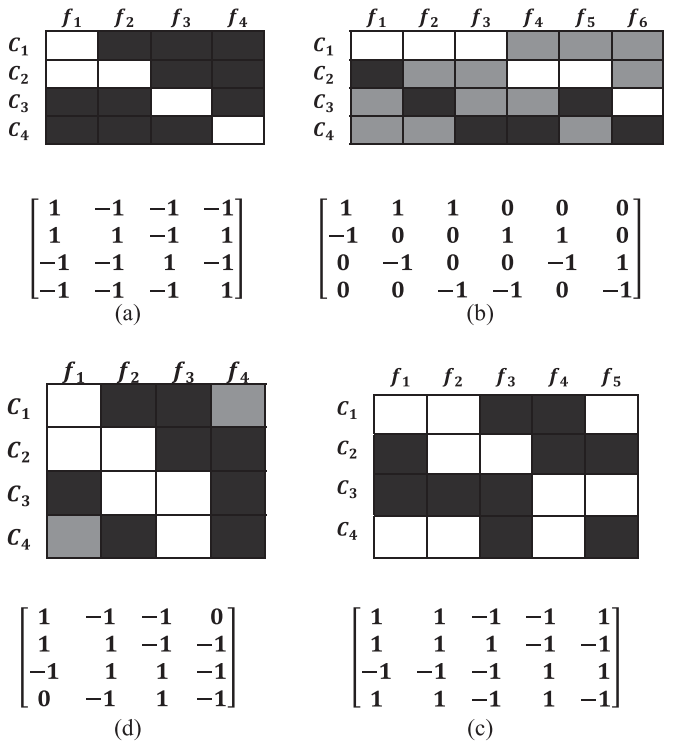


Fig. 2. Coding designs for a 4-class problem: (a) one-versus-all (b) one-versus-one, (c) dense random, (d) sparse random.

a time consuming process. Many researchers, however, have confirmed the acceptable performance of randomly generated code matrix. They agree that more intricate methods have only marginal effects on testing error [2]. Some of the most common problem-independent methods are discussed as follows:

- One-versus-all (OVA): This method trains $L = N_c$ classifiers. Each classifier separates the samples of one class from the rest of classes (Fig. 2(a)).
- One-versus-One (OVO): This approach trains $L = N_c(N_c - 1)/2$ classifiers. Each classifier splits two pair of classes (Fig. 2(b)).
- Random dense ECOC: This method randomly generates lots of matrices $M = \{-1, 1\}^{N_c \times L}$ which each element of the matrices takes to be $+1$ or -1 with the same probability. Among the generated matrices, the one with the maximum row and column separation will be selected. It is suggested to create $L = 10 \log N_c$ binary classifiers in this method (Fig. 2(c)).
- Random sparse ECOC: This method randomly generates lots of matrices $M = \{-1, 0, 1\}^{N_c \times L}$ which each element of the matrices takes the values of $+1$ and -1 with the probability of 0.5 and the value of 0 with the probability of 0.25. The zero value determines that the corresponding class does not participate in the training process. It is suggested to create $L = 15 \log N_c$ binary classifiers in this method [31] (Fig. 2(d)).

2.2. Decoding

Among the decoding strategies, distance-based approaches such as Hamming decoding, inverse Hamming decoding, and Euclidean decoding are the simplest and most effective ways to decode the code word, which have frequently been applied by researchers.

- Hamming decoding: The Hamming distance between two code words is the number of positions at which the corresponding symbols are different. Hamming metric is defined as follows:

$$HD(x, y_i) = \sum_{j=1}^{N_c} \{(1 - \text{sign}(x, y_i(j))\}/2 \quad (1)$$

- Inverse Hamming decoding:

$$IHD(x, y_i) = \max\left(\Delta^{-1} D^T\right) \quad (2)$$

Where, Δ is a matrix which contains all the hamming distances between the code words of M. D is the vector of hamming distances between the test code word (x) and each of the other code words (y_i) [11].

- Euclidean Decoding:

$$ED(x, y_i) = \sqrt{\sum_{j=1}^{n} (x(j) - y_i(j))^2} \quad (3)$$

2.3. Recent approaches to ECOC

Recent approaches attempt to use knowledge of problem domain to achieve the decomposition that increases the generalization performance while keeping the code length short [26].

- The Discriminant ECOC (DECOC) [25]

  This method heuristically searches for the optimal code matrix. It divides classes into two groups in a way that the discriminability between both groups is maximum. Discriminability between groups is achieved by maximizing the mutual information between the feature data and its class label. This method hierarchically partitions the class space using binary tree where each node represents the best bi-partition of the set of classes, maximizing quadratic mutual information the feature data and its class label. This process will continue recursively until single classes corresponding to tree leaves are obtained. As a result, a compact matrix with a high discrimination power will be obtained.

- ECOC Forest [34]
  This method is the extension of DECOC. It constructs a forest of decision trees which are included in the ECOC framework.
- Subspace approach

  In the study done by Bagheri et al. [3], a third dimension representing the feature subsets was added to the ECOC matrix in order to maximize the diversity among the classifiers. As a result, each dichotomizer uses a subset of features.

- ECOC Optimizing Node Embedding (ONE) [27]

  This method improves the performance of any ECOC coding by iteratively adding dichotomies corresponding to different spatial partitions of subsets of classes. These partitions are obtained by minimizing the confusion matrix among classes guided by a validation subset. As a result, relatively small code words with an acceptable performance will be generated.

- Thinned-ECOC [26]

  This method successively removes some redundant and unnecessary columns of any initial code matrix based on a metric defined for each column.

3. Neighborhood search ECOC (NS-ECOC)

The initially proposed ECOC method, known as Dense ECOC, employs all the classes to train each classifier, that is, the code matrix values can take either $+1$ or -1. However, based on previous discussions, in the sparse approach, some of the classes may not be involved in the training phase. This implies that the zero value can also be assigned to code matrix elements. Thus, One-vs-One approach can be considered as a sparse approach where only the pair of classes is involved in training of each classifier.

In the sparse error correcting output code including n classifiers, c_i corresponds to the code word y_i where $y_i \in \{-1, 0, 1\}^n$, while after classifying the test sample using the ensemble of classifiers, the code word x will be generated where $x \in \{-1, 1\}^n$. Thus, the test code word can never take the zero value in the case of sparse ECOC. In other words, each classifier should decide whether the test sample is a member of the negative group or positive group. Consequently, the sparse method does not allow the classifiers to abstain from voting. Assume that a classifier has not learned the actual class of the test sample. In this situation, the output of the classifier is followed by error. The error correcting nature of the ECOC method which comes from row and column separation can prevent the classification error to a certain extent. In the ECOC method, there is a trade-off between the length of the code words (number of columns) and the ability of correcting the errors. In this study, the neighborhood search helps the classifier to scan the vicinity of the sample before the classification, while employing the classifier only if it is necessary. The essence of using neighborhood search is to minimize the inevitable errors occurring in the classification of the samples, whose actual class is not learnt by the classifier.

The proposed method is presented in two versions. In the first version (NS-ECOC V1), after designing the ECOC code matrix and training the classifiers and before computing outputs of each classifier, k nearest neighbors of the instance to be classified will be computed (k is a parameter of the algorithm). The most frequent class in the neighborhood will then be selected. If the most frequent class in the neighborhood belongs to the class which is not learnt by the classifier (the value of the corresponding bit in the ECOC column being zero), then the output code word does not take the vote of the classifier into account. Instead, the zero value will be assigned to the corresponding bit in the output code word. Otherwise, the test sample will be classified using that classifier and the output will appear in the final code word. In the second version, for each classifier the total number of instances belonging to classes which are not involved in the training phase will be considered. If such instances form the major part of the neighborhood, the output of the classifier will not be considered in the output code word. Instead, the zero value will be assigned to the corresponding bit in the output code word. Fig. 3 shows an example of the aforesaid procedures to classify the test sample in 5-class classification problem with the random ECOC matrix.

Assuming the actual class of the test sample is class3 (star), this sample is classified using three methods: the initial ECOC, NS-ECOC V1 and NS-ECOC V2. In the ECOC method, the new sample will be classified using each base classifier. knowing that the class of the test sample (star) is among the first and second classifiers' training sets, in the best case these classifiers will be able to detect the test sample's group without any mistakes (first classifier output: -1 and the second classifier output: -1). However, the third and the fourth classifiers are associated with error and are not able to detect the test sample's group. Possible output modes and the final predicted class for the test sample are shown in Fig. 4. It is obvious that the ECOC method is not able to discern the correct class of the test sample with certainty.

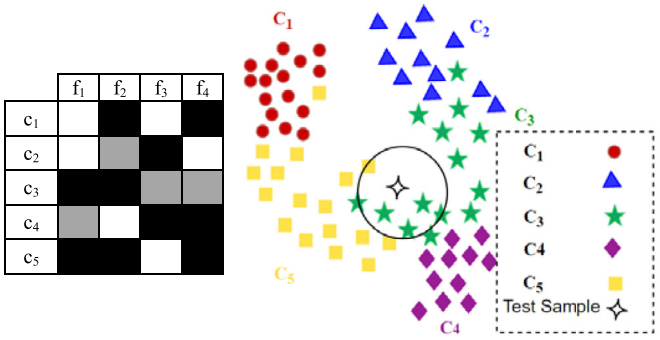
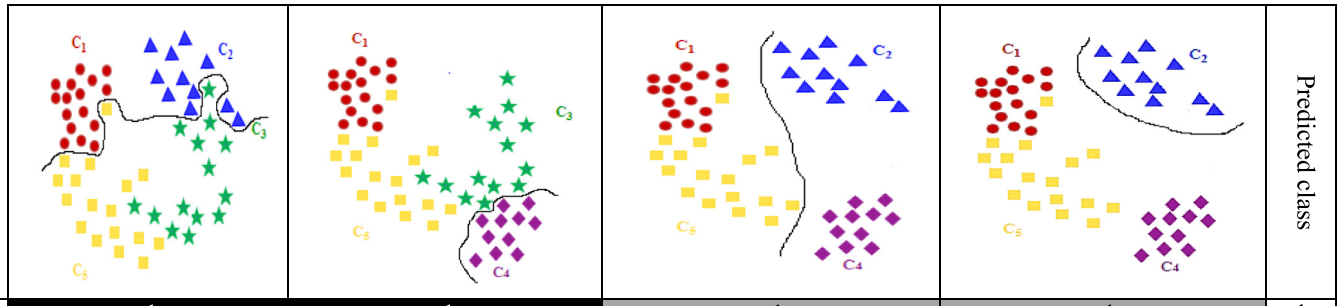


Fig. 3. ECOC code matrix, instances and the neighborhood of the test sample for a 4-class classification problem.

Below, the test sample from the previous example is classified using the first version of proposed NS-ECOC. The first step in ECOC method is to classify the test sample using each dichotomizer, but in the proposed method before computing the output of each dichotomizer, the neighborhood of the test sample is determined and the training samples which are located in that neighborhood are discovered. The most frequent class among discovered samples is employed as a criterion for including or excluding the dichotomizers' vote. As demonstrated in Fig. 3, most of the samples in the neighborhood are from the third class (star). Thus, the zero value in the corresponding bit of the third class in each dichotomizer shows that the dichotomizer's vote should be excluded from the output code word and the zero value should be considered as its output. For this reason, the test sample is classified using the first and second dichotomizers. Thus, the results of the mentioned dichotomizers are involved in the final code word. By contrast, samples of the third class are not involved in the learning process of the fourth and fifth classifiers. Hence, the test sample will not be classified using these two classifiers and the zero value is considered instead of their votes. Following the aforesaid steps, the final code word will be [−1, −1, 0, 0] and the third class will be devoted to the test sample without any uncertainty. The overview of the proposed algorithm is shown in Fig. 5.

The main procedure in the second version (NS-ECOC V2) is similar to the first version. The only difference is that in the second version instead of taking the most frequent class as a criterion, the total number of the samples which belong to the classes that are not considered in the training stage is counted. If such samples form most of the neighborhood samples, the dichotomizer will be excluded from the ensemble. Otherwise, the test sample will be classified using that dichotomizer and the output will appear in the final code word. In the following, the formerly mentioned example is solved using this method. The fourth class is not involved in the first dichotomizer. Thus, in the neighborhood, the samples that are the members of the fourth class are counted down. In this case, the number of such samples is 0. Therefore, the sample is classified using the first dichotomizer and the output (-1) is computed. The second dichotomizer is similar to the first. Since the samples of the second class do not form the majority of the samples in the neighborhood, the output of the classifier is computed (-1). In the third and fourth dichotomizers, the third class is not involved in the training phase. Hence, the samples of the third class are counted in the neighborhood. Since the number of such samples is bigger than half of the neighborhood samples, the output of these two dichotomizers is assigned to zero. Finally, the output code word is [−1, −1, 0, 0]. According to Hamming distance the third class should be allocated to the test sample. The overview of the proposed method is shown in Fig. 6.



1	-1	-1	+1	+1	1
2	-1	-1	+1	-1	5
3	-1	-1	-1	+1	2,3,4
4	-1	-1	-1	-1	2

Fig. 4. Decoding test sample of the third class in ECOC method.

Neighborhood Search ECOC Algorithm (first version)

Input: test sample x, training samples, ensemble of binary classifiers, Ecoc Matrix
Output: predicted class of x (C_x)

1. Find K-nearest neighbor in training samples for input sample
2. Choose the most frequent class among the neighborhood samples (C_f)
3. For each binary classifier in the ECOC ensemble:
   - If the corresponding bit of C_f in the classifier code has the zero value, skip this classifier and place the zero value in output code word.
   - else classify the test sample using the classifier and take the output bit (-1 or +1) and place it in output code word
4. Choose the class with the minimum hamming distance to the test code word and assign it to C_x

Fig. 5. The proposed algorithm for aggregation of base classifiers in ECOC (first version).

Neighborhood Search ECOC Algorithm (second version)

Input: test sample x, training samples, ensemble of binary classifiers, Ecoc Matrix
Output: predicted class of x (C_x)

1. Find K-nearest neighbor in training samples for input sample
2. For each binary classifier in the ECOC ensemble:
   Choose the classes which have the zero value in the classifier code (V).
   - If the samples whose real classes are the member of V form the most of the neighborhood, skip this classifier and place the zero value in the output code word.
   - else classify the test sample using the classifier and take the output bit (-1 or +1) and place it in the output code word
3. Choose the class with the minimum hamming distance to the test code word and assign it to C_x.

Fig. 6. The proposed algorithm for aggregation of base classifiers in ECOC (second version).

4. Evaluation

In this section, the two proposed methods are compared with the most common binarization methods such as OVO, OVA, Random Sparse ECOC, random Dense ECOC, DECOC and forest ECOC in order to evaluate the capability of the presented method in solving multiclass classification problems.

4.1. Experimental settings

- Data: the proposed NS-ECOC is evaluated using 16 multiclass benchmark data sets from UCI repository [12]. The summary of these data sets is shown in Table 2.
- Methods: for comparative analysis, four classic ECOC-based methods including OVO, OVA, Random Sparse ECOC, DECOC, ECOC forest and random dense ECOC are used. For the purpose of designing random ECOC, 10,000 random matrices are generated. Then, the matrix with the largest, minimal, pair-wise row distances based on the Hamming measure is chosen. Code word lengths of $15\log_2 N_c$ and $10\log_2 N_c$ is considered respectively for random sparse and random dense strategies. In the proposed methods, the small number of $k = 5$ is set for the neighborhood searching. Classification task is done by means of two learning methods: Multi-layer perceptron having two layers and tan-sigmoid activation function and SVM using RFB kernel. The parameters of learning algorithms are set using preliminary set of experiments over half of the data sets.

10-fold cross-validation method is used for performance evaluation. In order to obtain more reliable results, test and train sets are considered the same across all the methods.

4.2. Experimental results

The average accuracy of the methods using two different learning methods on each dataset is presented in Table 3. As can be seen in the table, both presented methods have improved the classification accuracy in Sparse Random ECOC, OVO, DECOC and forest ECOC. However, the NS-ECOC V2 method performs better than the NS-ECOC V1 method. This means that customized filtering for each binary classifier is more effective than general approaches. In the last row, the average rank according to the method's rank in each dataset is computed as: $R = \frac{1}{J}\sum_j r_j$ where J is the total number of datasets and r_j is the method's rank in jth dataset. Hence, a lower rank corresponds to highest performance. Moreover, further statistical analysis is conducted to have better comparisons between the examined methods. Nonparametric tests do not assume that a certain distribution fits the data. Hence, they are safer than parametric tests such as t-test and ANOVA [4].

The Friedman test is a non-parametric equivalent of the repeated-measures [23] and it is used to compare the competing methods based on their mean rank. In this test, the null hypothesis states that all the algorithms are equivalent and their ranks R_j are equal, as well. The Friedman statistic is distributed according to χ_F^2 with $k-1$ degrees of freedom, when N and k are big enough ($N > 10$ and $k > 5$ where N is the number of experiments and k

Table 2
Summary of used multiclass data sets.

#	Data set	# sample	# feature	# class	# Base classifiers				
					OVO	OVA	Dense random	Random sparse	DECOC
1	Abalone	4177	8	3	3	3	3	6	3
2	Zoo	101	16	7	21	7	28	37	7
3	Wine	178	13	3	3	3	3	6	3
4	Yeast	1484	8	10	45	10	38	56	10
5	Waveform	5000	40	3	3	3	3	6	3
6	Verterbal	310	6	3	3	3	3	6	3
7	Vehicle	846	18	3	3	3	3	6	3
8	Thyroid	215	5	3	3	3	3	6	3
9	Mfeat-Zer	2000	240	10	45	10	38	56	10
10	Mfeat-Mor	2000	6	10	45	10	38	56	10
11	Lymph	148	18	4	6	4	7	16	4
12	Glass	214	9	6	15	6	22	35	6
13	Ecoli	336	7	8	28	8	35	56	8
14	Dermatology	358	34	6	15	6	22	35	6
15	Cmc	1473	9	3	3	3	3	6	3
16	Iris	150	4	3	3	3	3	6	3

is the number of competing methods) [23].

$$\chi_F^2 = \frac{12N}{k(k+1)}\left(\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right) \tag{4}$$

In this study, there are 14 competing methods within 16 experiments. That makes the Friedman statistic χ_F^2 equal to 60 ($\chi_F^2 = 87.40$) when using MLP as learning algorithm and $\chi_F^2 = 101.77$ for SVM. Iman Davenport indicated that χ_F^2 is undesirably conservative. They reformed this statistic and proposed F_F approximation which is distributed according to the F-distribution with $k-1$ and $(k-1)(N-1)$ degrees of freedom [20].

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2} \tag{5}$$

By applying the F_F equation to compare our proposed methods, the values of $F_F = 10.87$ for MLP and $F_F = 14.37$ for MLP are obtained, which are greater than the critical value of $F(13, 15) = 2.7$ for $\alpha = 0.05$. This suggests that the null hypothesis is rejected. Thus, the rival methods are not equivalent. To proceed with a post-hoc test, the Nemenyi test is employed to compare all the classifiers to each other. The performance of two classifiers is significantly different if the corresponding average ranks differ by at least the critical difference (CD):

$$CD = q_\alpha\sqrt{\frac{k(k+1)}{6N}} \tag{6}$$

In this case, the critical value for 95% of confidence is $CD = 3.2\sqrt{\frac{14(14+1)}{6\times 16}} = 4.7$. The Nemenyi test result is presented in Fig. 7. The mean rank of each method is indicated by star and the horizontal line across the mean rank represents the critical difference.

Statistical tests demonstrate that random sparse with neighborhood search has a superior accuracy than the other rival methods. However, according to Nemenyi test, it can be concluded that there is no significant difference between Sparse Random NS ECOC V1, Sparse Random NS ECOC V2, OVO NS V1, OVO NS V2 and Forest NS ECOC V2 since the corresponding average ranks do not differ by at least the critical difference value. However, considering the number of binary classifiers in each method, use of neighborhood search in OVO ECOC with a lower number of binary classifiers is the better option comparing to Random Sparse ECOC and Forest ECOC.

According to Figs. 7 and8 it can be concluded that use of the both version of neighborhood search ECOC result in superior rank and higher average accuracy in each sparse based ECOC method.

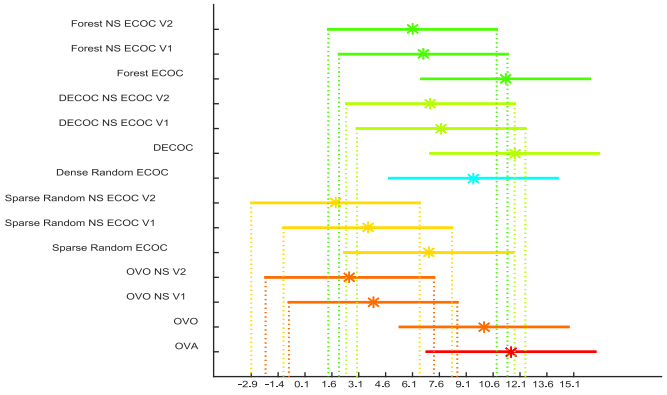


Fig. 7. Comparison result based on Nemenyi test (using MLP).

However, based on statistical tests the first version of the presented method can significantly improve the performance of sparse based ECOC methods.

5. Conclusion

In this paper, a novel method based on ECOC framework has been introduced. The main idea of the proposed method has been to take advantage of both problem-dependent and problem-independent classification approaches. It has designed the ECOC matrix without taking data relations into account. After training base learners and creating ensemble of binary classifiers, it has searched the neighborhood of the new sample to decide whether to include or exclude some of the binary classifier's vote to achieve the final code word. This method obviates the complexities in generating optimal code matrix, and using simple neighborhood search for each binary classifier it tries to reduce possible errors. The experimental results over 16 multiclass datasets have shown the superiority of the proposed method over other commonly-used ECOC based methods in terms of accuracy. We have presented two different versions of the neighborhood search ECOC, with the first version selecting the most frequent class presented in the neighborhood as a criterion, generally for the entire base classifiers involved in the ensemble. However, the second version customizes the criterion for each base classifier according to their coding in the ECOC matrix. Based on satisfactory results, the second version has had a better performance than the first one. The proposed method has had two main advantages: First, finding the neighbors of the new sample creates an overview to select relative classifiers

Table 3
Classification accuracies of different methods.

Method / Data Set	OVA	OVO	OVO NS V1	OVO NS V2	Sparse random ECOC	Sparse random NS ECOC V1	Sparse random NS ECOC V2
	Base classifier: MLP						
Abalone	55.62 ± 11.4	55.05 ± 2.9	56.37 ± 1.7	56.73 ± 0.1	56.16 ± 0.3	**57.85 ± 8.9**	**57.85 ± 0.1**
Zoo	83.05 ± 1.6	91.31 ± 1.6	94.69 ± 2.6	94.69 ± 1.2	93.87 ± 2.7	**95.07 ± 2.7**	**95.07 ± 2.7**
Wine	94.38 ± 0.5	92.35 ± 2.7	94.48 ± 2.7	94.48 ± 2.7	94.34 ± 0.1	**96.62 ± 0.1**	**96.62 ± 0.1**
Yeast	52.05 ± 2.2	51.01 ± 0.5	58.78 ± 0.6	58.59 ± 0.8	57.95 ± 0.1	**58.88 ± 0.6**	58.82 ± 0.4
Waveform	81.76 ± 6.5	82.64 ± 1.4	84.70 ± 0.6	**84.82 ± 0.5**	80.54 ± 0.5	82.82 ± 0.1	82.82 ± 0.1
Verterbal	79.35 ± 0.1	80.14 ± 2.4	82.09 ± 5	**82.90 ± 4.5**	80.44 ± 2.2	**82.90 ± 2.2**	**82.90 ± 2.8**
Vehicle	72.57 ± 2.0	75.53 ± 2.8	78.69 ± 1	78.69 ± 2.8	78.66 ± 4.3	79.78 ± 4.1	**81.02 ± 4.8**
Thyroid	84.4' ± 2.6	88.45 ± 5.5	**95.80 ± 5.2**	**95.80 ± 5.2**	94.86 ± 3.9	95.33 ± 3.9	95.33 ± 3.9
Mfeat-Zer	70.35 ± 7.3	76.45 ± 3.8	76.55 ± 0.9	77.80 ± 0.8	72.60 ± 3.1	75.76 ± 2.8	**82.30 ± 2.8**
Mfeat-Mor	63.7' ± 5.0	74.20 ± 2.0	74.30 ± 2.1	75.75 ± 2.8	75.30 ± 2.2	75.35 ± 2.7	**75.80 ± 2**
Lymph	68.91 ± 4.3	80.40 ± 2.5	81.10 ± 2.8	**83.78 ± 2.8**	81.75 ± 0.9	83.10 ± 2.8	83.75 ± 0.9
Glass	58.41 ± 8.5	62.14 ± 3.3	64.21 ± 3.2	64.21 ± 4.6	61.21 ± 4.6	62.61 ± 10.5	**64.95 ± 11.2**
Ecoli	66.07 ± 1.6	79.46 ± 2.1	84.82 ± 5.4	82.73 ± 6.7	82.44 ± 1.2	84.22 ± 4.6	**84.82 ± 4.6**
Dermatology	89.61 ± 0.7	92.34 ± 2.3	93.71 ± 1.1	95.80 ± 0.3	95.62 ± 0.7	**96.17 ± 0.2**	**96.17 ± 0.2**
Cmc	52.13 ± 1.4	51.40 ± 2.6	52.40 ± 2.4	53.55 ± 2.5	52.13 ± 0.1	53.47 ± 1.1	**53.83 ± 0.1**
Iris	90" ± 0.1	92.54 ± 3.7	**94.33 ± 3.7**	**94.33 ± 3.7**	92.66 ± 0.9	92.66 ± 0.9	92.66 ± 0.9
Mean	72.64 ± 3.4	76.58 ± 2.6	79.18 ± 2.5	79.66 ± 2.6	78.15 ± 1.7	79.53 ± 3.0	**80.29 ± 2.3**
Rank	11.6	10.1	3.9	2.6	7.0	3.6	**1.8**

Method / Data Set	Dense Random ECOC	DECOC	DECOC NS ECOC V1	DECOC NS ECOC V2	Forest ECOC	Forest NS ECOC V1	Forest NS ECOC V2
	Base Classifier: MLP						
Abalone	54.82 ± 2.8	53.57 ± 3.7	55.48 ± 2.2	55.48 ± 2.2	54.19 ± 4.3	55.46 ± 0.7	55.22 ± 3.7
Zoo	90.05 ± 2.9	88.10 ± 2.6	91.10 ± 2.6	91.13 ± 2.06	93.05 ± 2.7	94.07 ± 1.3	94.07 ± 1.3
Wine	91.62 ± 2.5	91.16 ± 2.4	93.46 ± 2.4	93.46 ± 2.4	92.56 ± 0.7	94.19 ± 0.7	94.19 ± 0.7
Yeast	55.76 ± 2.1	54.18 ± 2.3	55.71 ± 1.7	55.98 ± 2'	55.08 ± 0.1	58.45 ± 0.4	58.24 ± 0.7
Waveform	82.94 ± 2.1	82.22 ± 0.1	81.00 ± 0.8	81.92 ± 0.6	82.08 ± 0.2	83.68 ± 0.5	84.80 ± 0.05
Verterbal	79.35 ± 2.7	76.22 ± 3.3	81.90 ± 3.3	81.90 ± 5'	75.38 ± 7.7	81.87 ± 5'	81.80 ± 5
Vehicle	78.14 ± 4.3	73.23 ± 3'	78.43 ± 2'	78.43 ± 2.5	77.66 ± 3.3	78.18 ± 5.5	78.43 ± 4.6
Thyroid	89.62 ± 3.3	94.02 ± 3.3	94.40 ± 0.7	94.40 ± 2.6	93.34 ± 2.6	95.34 ± 2.6	95.34 ± 2.6
Mfeat-Zer	81.35 ± 2.3	79.32 ± 2.6	81.85 ± 2.6	81.85 ± 2.4	79.51 ± 2.3	81.85 ± 2.1	81.85 ± 2.1
Mfeat-Mor	75.70 ± 3.1	70.80 ± 3.4	75.55 ± 3.1	75.55 ± 4.7	73.34 ± 3.4	75.55 ± 3.2	75.64 ± 4.5
Lymph	77.02 ± 1.9	76.62 ± 7.6	79.94 ± 8.5	80.62 ± 7.6	78.40 ± 2.8	81.75 ± 0.9	81.08 ± 2.8
Glass	59.81 ± 1.3	59.34 ± 8.5	63.34 ± 5.9	63.41 ± 7.2	60.14 ± 7.2	60.74 ± 17.5	63.55 ± 6.6
Ecoli	82.44 ± 2.1	65.04 ± 3.3	75.77 ± 1.2	76.66 ± 0.6	67.94 ± 0.9	75.59 ± 11.7	75.00 ± 13.4
Dermatology	90.62 ± 1.5	87.43 ± 5'	89.97 ± 6.1	89.34 ± 5.7	88.16 ± 3'	93.16 ± 1.1	93.16 ± 0.1
Cmc	51.78 ± 1.5	49.39 ± 2.8	51.46 ± 4.3	51.79 ± 3.1	49.73 ± 0.4	51.18 ± 0.4	51.79 ± 0.1
Iris	90.66 ± 1.8	92.66 ± 0.9	92.66 ± 0.9	92.66 ± 0.9	89.33 ± 0.9	91.33 ± 0.9	91.33 ± 0.9
Mean	76.98 ± 2.3	74.58 ± 3.4	77.62 ± 3.0	77.78 ± 3.2	75.61 ± 2.6	78.27 ± 3.4	78.46 ± 3.0
Rank	9.5	11.8	7.7	7.1	11.3	6.7	6.1

Method / Data Set	OVA	OVO	OVO NS V1	OVO NS V2	Sparse Random ECOC	Sparse Random NS ECOC V1	Sparse Random NS ECOC V2
	Base Classifier: SVM						
Abalone	55.11 ± 0.67	55.16 ± 2.15	55.69 ± 2.19	55.76 ± 1.5	56.24 ± 2.2	56.76 ± 1.9	57.14 ± 2.15
Zoo	72.36 ± 13.9	81.36 ± 13.1	84.27 ± 11.4	84.36 ± 12.5	85.27 ± 11.2	86.16 ± 12.5	86.36 ± 13
Wine	77.34 ± 8.4	86.23 ± 7.8	88.23 ± 7.8	88.29 ± 7.8	87.71 ± 13.1	88.23 ± 12.6	88.79 ± 8.4
Yeast	50.29 ± 4.9	55.52 ± 4	56.80 ± 3.6	58.60 ± 3.2	51.95 ± 3.5	58.08 ± 2.9	59.68 ± 4.4
Waveform	68.68 ± 2.3	82.02 ± 1.01	82.22 ± 1.1	82.22 ± 1.2	82.16 ± 1.1	82.26 ± 1	82.32 ± 1.2
Verterbal	79.96 ± 5.1	82.45 ± 7.6	82.90 ± 7.6	83.54 ± 6.7	80.54 ± 6.7	83.54 ± 6.7	82.90 ± 7.6
Vehicle	70.05 ± 3.4	70.22 ± 3.8	71.58 ± 4.8	72.52 ± 4.4	70.69 ± 4.1	72.56 ± 4.4	72.81 ± 4.3
Thyroid	93.32 ± 3.8	93.52 ± 2.9	95.43 ± 3.8	96.27 ± 3.6	94.27 ± 3.6	96.27 ± 3.6	96.27 ± 3.8
Mfeat-Zer	79.30 ± 1.1	93.20 ± 3.2	98.05 ± 3.5	97.65 ± 3.1	85.30 ± 1.1	89.20 ± 4.2	89.15 ± 5.2
Mfeat-Mor	69.70 ± 5.0	71.67 ± 3.1	72.30 ± 2.8	72.85 ± 4.2	72.12 ± 3.7	72.20 ± 3	73.35 ± 3
Lymph	62.28 ± 11.1	63.61 ± 11.1	65.85 ± 11.1	65.85 ± 11.1	64.28 ± 11.1	66.28 ± 11.1	66.28 ± 11.1
Glass	70.13 ± 6.8	69.85 ± 10.8	70.67 ± 10.9	70.67 ± 10	70.12 ± 8.3	71.10 ± 8.9	71.81 ± 11.2
Ecoli	78.34 ± 6.2	83.71 ± 8.4	85.60 ± 9.3	85.11 ± 6.7	83.01 ± 7.1	85.30 ± 6.9	86.93 ± 8.9
Dermatology	72.08 ± 8.4	96.25 ± 7.4	96.53 ± 7.5	99.53 ± 8.9	99.07 ± 8~	99.80 ± 8.6	99.63 ± 8.6
Cmc	45.98 ± 4.9	44.96 ± 9.1	52.73 ± 7.8	52.26 ± 3.8	46.71 ± 3.9	52.47 ± 4.3	52.47 ± 6.8
Iris	94.00 ± 6.4	96.00 ± 5.6	96.00 ± 5.6	96.00 ± 5.6	96.00 ± 5.6	96.00 ± 5.6	96.00 ± 5.6
Mean	71.18 ± 5.7	76.60 ± 6.3	78.42 ± 6.2	78.84 ± 5.8	76.59 ± 5.8	78.51 ± 6.1	**78.86 ± 6.5**
Rank	12	8.3	3.5	3.0	6.9	2.4	**1.6**

(continued on next page.)

Table 3
(continued)

Method / Data Set	Base Classifier: SVM						
	Dense Random ECOC	DECO NS ECOC V1	DECOC NS ECOC V1	DECOC NS ECOC V2	Forest ECOC	Forest NS ECOC V1	Forest NS ECOC V2
Abalone	56.17 ± 1.6	53.78 ± 2.7	54.78 ± 2.6	54.78 ± 2.6	52.44 ± 2.02	54.49 ± 1.9	54.44 ± 1.9
Zoo	85.36 ± 11.9	80.36 ± 10.2	82.36 ± 11.6	71.45 ± 18.6	78.36 ± 14.3	79.36 ± 13.1	78.36 ± 13.1
Wine	79.34 ± 8.4	87.79 ± 12.6	**88.79 ± 13.1**	86.56 ± 13.4	84.45 ± 9.8	84.34 ± 12.1	84.90 ± 12.1
Yeast	58.63 ± 3.0	49.54 ± 8.4	49.13 ± 10.0	49.60 ± 9.2	54.99 ± 6.9	56.60 ± 5.3	57.68 ± 4.5
Waveform	50.66 ± 2.3	68.28 ± 1.3	67.08 ± 1.4	68.48 ± 1.5	68.28 ± 1.3	67.08 ± 1.4	68.48 ± 1.5
Verterbal	80.96 ± 5.1	80.61 ± 5.0	79.35 ± 5.9	80.64 ± 5.2	80.29 ± 4.7	79.03 ± 5.5	80.32 ± 4.9
Vehicle	72.57 ± 3.7	71.69 ± 3.2	71.74 ± 3.7	71.15 ± 4.0	71.69 ± 3.2	71.74 ± 3.7	71.15 ± 4.0
Thyroid	95.32 ± 3.5	94.80 ± 3.4	94.87 ± 5.1	95.32 ± 4.4	95.27 ± 2.9	95.80 ± 3.4	94.87 ± 5.1
Mfeat-Zer	79.20 ± 4.2	64.45 ± 4.8	72.45 ± 7	72.40 ± 7.1	70.90 ± 3.9	82.35 ± 3.6	82.75 ± 3.9
Mfeat-Mor	73.70 ± 2.1	72.80 ± 2.4	69.65 ± 5.1	71.85 ± 3.0	72.80 ± 2.4	72.15 ± 3.3	73.45 ± 4.3
Lymph	64.28 ± 11.1	**65.61 ± 10.4**	63.61 ± 10.2	64.95 ± 10.0	63.28 ± 11.1	64.95 ± 11.8	64.95 ± 11.8
Glass	68.26 ± 6.9	64.41 ± 10.1	64.93 ± 10.9	64.93 ± 9.9	64.02 ± 7.2	69.71 ± 8.3	66.88 ± 7.6
Ecoli	85.40 ± 6.3	82.61 ± 6.9	84.21 ± 6.8	84.21 ± 6.7	84.41 ± 5.6	85.11 ± 7.4	86.59 ± 6.1
Dermatology	87.45 ± 10.0	75.55 ± 8.2	78.45 ± 7.4	77.63 ± 7.7	77.19 ± 8.3	80.63 ± 7.3	79.27 ± 7.6
Cmc	46.36 ± 4.0	37.15 ± 6.9	38.28 ± 6.9	38.15 ± 6.9	42.98 ± 7.2	43.64 ± 6.9	43.98 ± 7.1
Iris	**96.00 ± 6.4**	94.33 ± 6.3	95.33 ± 6.3	95.33 ± 6.3	94.33 ± 6.3	95.33 ± 6.3	95.33 ± 6.3
Mean	72.22 ± 5.6	71.73 ± 6.4	74.23 ± 7.1	74.276 ± 7.2	72.546 ± 6.0	75.13 ± 6.3	75.89 ± 6.3
Rank	9.0	12.5	7.6	7.5	12.1	7.75	6.5

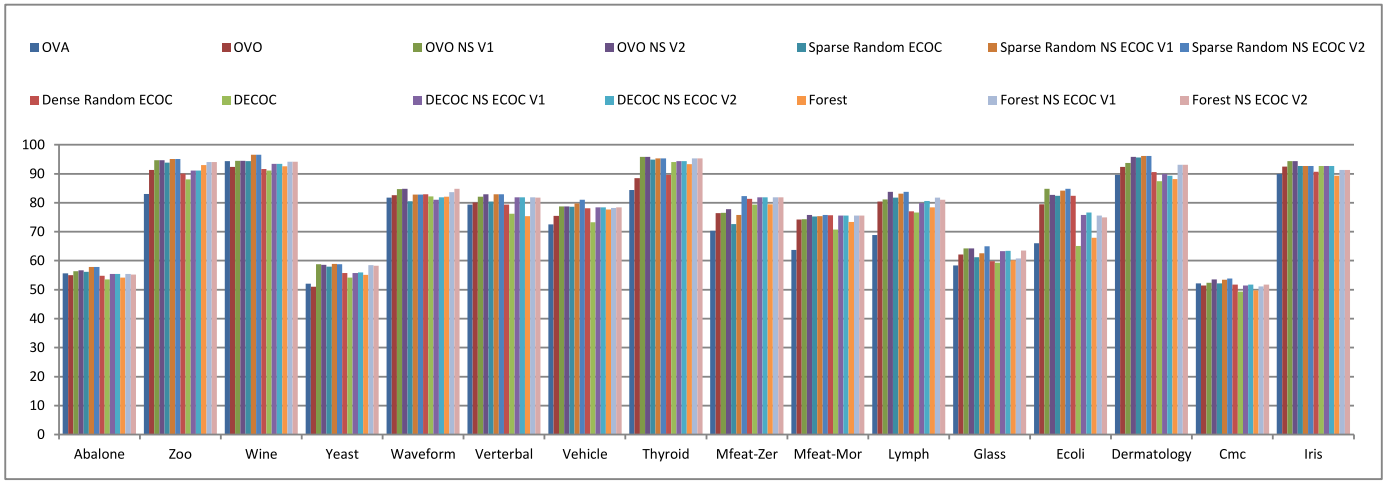


Fig. 8. Average accuracy of sparse based methods on different datasets (using MLP).

for classifying new samples. Consequently, it reduces misclassification in error-prone cases by eliminating unrelated classifiers. Moreover, this method can decrease the size of the code words and produce similar results in comparison with methods without the inclusion of neighborhood search. Therefore, the proposed method is an effective method in dealing with multiclass classification problems. Current line of research in this study is centered on employing the entire feature space rather than choosing the best performing sub-space in each base classifier. This step along with the more problem-dependent criterion for selecting competent classifiers can be a valuable direction for future studies to investigate the possibility of increasing the discriminative power of ECOC.

References

[1] E.L. Allwein, R.E. Schapire, Y. Singer, Reducing multiclass to binary: a unifying approach for margin classifiers, J. Mach. Learn. Res. 1 (2001) 113–141.

[2] M.A. Bagheri, Q Gao, S. Escalera, Three-dimensional design of error correcting output codes, in: Proc. International Conference on Machine Learning and Data Mining, 2012.

[3] M.A. Bagheri, Q. Gao, S. Escalera, A genetic-based subspace analysis method for improving error-correcting output coding, Pattern Recognit. 46 (10) (2013) 2830–2839.

[4] M.A. Bagheri, Q. Gao, S. Escalera, Combining local and global learners in the pairwise multiclass classification, Pattern Anal. Appl. 18 (4) (2015) 845–860.

[5] M.A. Bagheri, G.A. Montazer, S. Escalera, Error correcting output codes for multiclass classification: application to two image vision problems, in: Artificial Intelligence and Signal Processing (AISP), 2012 16th CSI International Symposium on, IEEE, 2012, pp. 508–513.

[6] M.A. Bagheri, G.A. Montazer, E. Kabir, A subspace approach to error correcting output codes, Pattern Recognit. Lett. 34 (2) (2013) 176–184.

[7] X. Bai, S.I. Niwas, W. Lin, B.F. Ju, C.K. Kwoh, L. Wang, C.C. Sng, M.C. Aquino, P.T. Chew, Learning ECOC code matrix for multiclass classification with application to glaucoma diagnosis, J. Med. Syst. 40 (4) (2016) 1–10.

[8] M.Á. Bautista, S. Escalera, X. Baró, O. Pujol, On the design of an ECOC-compliant genetic algorithm, Pattern Recognit. 47 (2) (2014) 865–884.

[9] A. Berger, Error-correcting output coding for text classification, IJCAI-99: Workshop on machine learning for information filtering, 1999.

[10] S. Escalera, O. Pujol, P. Radeva, On the decoding process in ternary error-correcting output codes, IEEE Trans. Pattern Anal. Mach. Intelligence 32 (1) (2010) 120–134.

[11] S. Escalera, O. Pujol, P. Radeva, Error-correcting output codes library, J. Mach. Learn. Res. 11 (2010) 661–664.

[12] Frank, A. Asuncion, UCI Machine Learning Repository, University of California, Irvine, School of Information and Computer Sciences, 2010.

[13] N. Hatami, R. Ebrahimpour, R. Ghaderi, ECOC-based training of neural networks for face recognition, in: 2008 IEEE Conference on Cybernetics and Intelligent Systems, IEEE, 2008, pp. 450–454.

[14] K.H. Liu, Z.H. Zeng, V.T.Y. Ng, A Hierarchical ensemble of ECOC for cancer classification based on multi-class microarray data, Inf. Sci. 349 (2016) 102–118.

[15] M. Liu, D. Zhang, S. Chen, H. Xue, Joint binary classifier learning for ecoc-based multi-class classification, IEEE Trans. Pattern Anal. Mach. Intell. 38 (11) (2016) 2335–2341.

[16] A.C. Lorena, A.CPLF. De Carvalho, JoãoMP Gama, A review on the combination of binary classifiers in multiclass problems, Artif. Intell. Rev. 30 (1-4) (2008) 19–37.

[17] A. Rocha, S.K. Goldenstein, Multiclass from binary: expanding one-versus-all, one-versus-one and ECOC-based approaches, IEEE Trans. Neural Netw. Learn. Syst. 25 (2) (2014) 289–302.
[18] R.S. Smith, T. Windeatt, Facial action unit recognition using multi-class classification, Neurocomputing 150 (2015) 440–448.
[19] Y. Zhao, X. Xie, M. Jiang, Hierarchical real-time network traffic classification based on ECOC, TELKOMNIKA Indonesian J. Electric. Eng. 12 (2) (2014) 1551–1560.
[20] R.L. Iman, J.M. Davenport, Approximations of the critical region of the fbietkan statistic, Commun. Stat. Theory Method. 9 (6) (1980) 571–595.
[21] N. Garcia-Pedrajas, D. Ortiz-Boyer, Improving multiclass pattern recognition by the combination of two strategies, IEEE Trans. Pattern Anal. Mach. Intell. 28 (6) (2006) 1001–1006.
[22] I. Mendialdua, G. Echegaray, I. Rodriguez, E. Lazkano, B. Sierra, Undirected cyclic graph based multiclass pair-wise classifier: classifier number reduction maintaining accuracy, Neurocomputing 171 (2016) 1576–1590.
[23] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[24] K. Crammer, Y. Singer, On the learnability and design of output codes for multiclass problems, Mach. Learn. 47 (2) (2002) 201–233.
[25] O. Pujol, P. Radeva, J. Vitria, Discriminant ECOC: a heuristic method for application dependent design of error correcting output codes, IEEE Trans. Pattern Anal. Mach. Intell. 28 (6) (2006) 1007–1012.
[26] N. Hatami, Thinned-ECOC ensemble based on sequential code shrinking, Expert Syst. Appl. 39 (1) (2012) 936–947.
[27] O. Pujol, S. Escalera, P. Radeva, An incremental node embedding technique for error correcting output codes, Pattern Recognit. 41 (2) (2008) 713–725.
[28] R. Rifkin, A. Klautau, In defense of one-vs-all classification, The Journal of Machine Learning Research 5 (2004) 101–141.
[29] N. J. Nilsson. Learning Machines. McGraw-Hill, 1965.
[30] E. Allwein, R. Schapire, Y. Singer, Reducing multiclass to binary: A unifying approach for margin classifiers, J. Mach. Learn. Res. 1 (2002) 113–141.
[31] S. Escalera, P. Pujol, P. Radeva, Separability of ternary codes for sparse designs of errorcorrecting output codes, Pattern Recognition Letters 30 (2009) 285–297.
[32] S. Escalera, P. Pujol, P. Radeva, On the decoding process in ternary error-correcting output codes, IEEE Trans. Pattern Anal. Mach. Intelligence 32 (1) (2010) 120–134.
[33] T.G. Dietterich, G. Bakiri, Solving multiclass learning problems via error-correcting output codes, J. Artifi. Intelligence Res. 2 (1995) 263–286.
[34] S. Escalera, P. Pujol, P. Radeva, Boosted landmarks of contextual descriptors and ForestECOC: A novel framework to detect and classify objects in clutter scenes, Pattern Recognition Letters 28 (13) (2007) 1759–1768.