Expert Systems with Applications 39 (2012) 936–947

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

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

Thinned-ECOC ensemble based on sequential code shrinking

Nima Hatami *

DIEE – Department of Electrical and Electronic Engineering, University of Cagliari, Piazza d'Armi, I-09123 Cagliari, Italy

ARTICLE INFO

Keywords:
Multiple classifier systems (MCS)
Thinning ensemble
Error-correcting output coding (ECOC)
Multi-class classification
Face recognition
Gene expression classification

ABSTRACT

Error-correcting output coding (ECOC) is a strategy to create classifier ensembles which reduces a multi-class problem into some binary sub-problems. A key issue in designing any ECOC classifier refers to defining optimal codematrix having maximum discrimination power and minimum number of columns. This paper proposes a heuristic method for application-dependent design of optimal ECOC matrix based on a thinning algorithm. The main idea of the proposed Thinned-ECOC method is to successively remove some redundant and unnecessary columns of any initial codematrix based on a metric defined for each column. As a result, computational cost of the ensemble is reduced while preserving its accuracy. Proposed method has been validated using the UCI machine learning database and further applied to a couple of real-world pattern recognition problems (the face recognition and gene expression based cancer classification). Experimental results emphasize the robustness of Thinned-ECOC in comparison with existing state-of-the-art code generation methods.

1. Introduction

The ultimate goal in pattern recognition is to achieve the best possible classification performance for the task at hand. A promising approach towards this goal refers to combining classifiers since typically a monolithic classifier is not able to properly handle all complex classification problems. Combining more independent classifiers with acceptable accuracy leads to better performance (Ali & Pazzani, 1995). Therefore, we try to increase the diversity among accurate base classifiers.

There are many techniques in machine learning to generate diverse classifiers such as boosting (Freund & Schapire, 1997), mixture of experts (Jacobs, Jordan, Nowlan, & Hinton, 1991) and ECOC (Dietterich & Bakiri, 1995). Each technique has its own particularity: boosting focuses on changing input samples distribution at each step for each classifier to concentrate on difficult-to-learn points; mixture of experts tries to specialize each local expert in a subset of the input space, each expert being responsible for its own task; ECOC is a technique which manipulates output labels of the classes. ECOC achieved promising results on both synthetic and real datasets (Hatami, Ebrahimpour, & Ghaderi, 2008; Windeatt & Ghaderi, 2003, 2001). In ECOC method, a discrete decomposition matrix (codematrix) is first defined for the multi-class problem at hand. Then this problem is decomposed into a number of binary sub-problems, dichotomies, according to the sequence of 0s and 1s of columns of the codematrix. After training binary classifiers on these dichotomies and testing them on any incoming test sample, a binary output vector is created. The final label is assigned to the class with the smallest distance between this vector and the codewords.

Since the performance of the decomposition stage is highly related to codematrix, the problem of generating optimal code is of great importance. Various methods have been proposed in literature for codematrix generation (Allwein, Shapire, & Singer, 2000; Dietterich & Bakiri, 1995). The algebraic-based BCH codes (Lin & Costello, 2004), dense and sparse random method, pairwise coupling (1vs1) and 1vsA are well-known code generation methods with good results (Allwein et al., 2000; Dietterich & Bakiri, 1995; Peterson & Weldon, 1972; Windeatt & Ghaderi, 2003). Almost all of these methods try to meet two goals: maximizing the distance between any pair of codewords leading to more error-correcting capability and low correlation among matrix columns (dichotomies) leading to increase diversity among binary base classifiers.

All these coding strategies are fixed in the ECOC design step, created regardless of the problem domain or the ensemble accuracy. In fact, very little attention has been paid the coding process of the ECOC classifier. There are some methods proposed in the literature to optimize the coding process but as their results show, many of these approaches do not efficiently tackle the problem of designing optimal problem-dependent ECOCs. Pujol, Radeva, and Vitria (2006) proposed the embedding of discriminant tree structures derived from the problem domain in the ECOC framework. With this method called Discriminant ECOC, a multi-class problem is decomposed into C-1 binary problems. As a result, a compact discrete coding matrix is obtained with a small but fixed number of dichotomizers. In Pujol, Escalera, and Radeva (2008),

* Tel.: +39 070 675 5755; fax: +39 070 675 5782.
E-mail address: nima.hatami@diee.unica.it
URL: http://nimahatami.googlepages.com

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.07.091

proposed a method that improves the performance of any initial codematrix by extending it in a sub-optimal way. They proposed a strategy aimed at creating the new dichotomizers by minimizing the confusion matrix among classes guided by a validation subset. Although there is a significant progress in the coding step of ECOC, the question of how to design the codematrix with high discrimination power balanced against minimum code length is still open.

In this paper, we propose a new method for automatic design of application-dependent optimal codematrix i.e. ECOC with high discrimination power and appropriate code length for the problem at hand. To be more specific, the proposed approach takes advantage of some basic concepts of building the ensemble such as diversity of classifiers and thinning method for designing optimal codematrix. *Thinning ensemble* is a strategy that measures diversity and individual accuracy which are then used in the process of building an accurate ensemble. The main idea of the thinning algorithms is to identify the classifier which is most often incorrect on the ensemble misclassification points and remove it from the ensemble. Inspired by this idea, we have first developed an initial ECOC matrix which is the matrix with large number of columns composed of some known ECOC matrices. Next, the proposed heuristic method called Thinned-ECOC is used to remove some redundant and unnecessary columns successively based on a metric defined for each column. This sequential shrinking of the codematrix continues until the point where any removal of columns negatively affects the ensemble performance. Although reducing the codematrix length results in less complex ensemble, it also boosts or at least preserves the overall accuracy. On the other hand, another open problem is solved: how to automatically determine the number of optimal codematrix for the problem at hand. Thinned-ECOC is successfully applied to two multi-class pattern recognition problems: classification of cancer tissue types based on microarray gene expression data and face recognition based on 2D images.

The remainder of this paper is divided into the following sections: Section 2 provides a brief introduction to the error-correcting output codes and literature review of the problem-dependent code matrices. The thinning strategy for building an ensemble is described in the third section. Section 4 introduces the proposed method for the automatic design of optimal ECOC and illustrates application of the proposed Thinned-ECOC on a synthetic dataset. In Section 5, we validate our proposed method on a number of selected datasets from the UCI machine learning repository, the face recognition problem, the classification of cancer tissues and discuss the results. Section 6 concludes the paper.

2. Error-correcting output coding

2.1. ECOC overview

Given a classification problem with N_c classes, the main idea of ECOC is to create a codeword for each class. Arranging the codewords as rows of a matrix, we define a codematrix M, where $M \in \{-1, 0, +1\}^{N_c \times L}$ and L is the code length. From learning point of view, M specifies N_c classes to train L classifiers (dichotomizers), $f_1 \dots f_L$. A classifier f_l is trained according to the column $M(\cdot, l)$. If $M(N,l) = +1$ then all examples of class N are positive, if $M(N,l) = -1$ then its all examples are negative and, finally, if $M(N,l) = 0$ none of the examples of class N participate in the training of f_l.

Let $\bar{y} = [y_1 \dots y_L],\ y_l \in \{-1, +1\}$ be the output vector of the L classifiers in the ensemble for a given input x. In the decoding step, the class output that maximizes the similarity measure s between $\bar{y}$ and row $M(N,\cdot)$ is selected:

$$Class\ Label = Arg\,Max\, S(\bar{y}, M(N, \cdot)) \tag{1}$$

Concerning the similarity measures, two of the most common techniques are the Hamming decoding distances Eq. (2) where classifier outputs are hard decision and Margin decoding Eq. (3) where the outputs are soft level.

$$S_H(\bar{y}, M(N, \cdot)) = 0.5 \times \sum_{l=1}^{L} 1 + y_l M(N, l) \tag{2}$$

$$S_M(\bar{y}, M(N, \cdot)) = \sum_{l=1}^{L} y_l M(N, l) \tag{3}$$

The ECOC matrix codifies the class labels in order to achieve different partitions of classes, considered by each dichotomizer. The main coding strategies can be divided into problem-indepentent (or fixed) and problem-dependent.

2.2. Problem-independent strategies

Most of the popular ECOC coding strategies up to now are based on pre-designed problem-independent codeword construction, which satisfy the requirement of high separability between rows and columns. These strategies include: *1vsA*, where each classifier is trained to discriminate a given class from the rest of classes using N_c dichotomizers; *random techniques*, which can be divided into the *dense random strategy*, consisting of a binary matrix with high distance between rows with estimated length of 10 $\log_2 N_c$ bits per code, and the *sparse random strategy* based on the ternary symbol and with the estimated optimal length of about 15 $\log_2 N_c$. *1vs1* is one of the most well known coding strategies, with $N_c(N_c - 1)/2$ dichotomizers including all combinations of pairs of classes (Hastie & Tibshirani, 1998). Finally, BCH codes (Lin & Costello, 2004) are based on algebraic techniques from Galois Field theory, and while its implementation is fairly complex, it has some advantages such as generating ECOC codewords separated by a minimum, configurable Hamming distance and good scalability to hundreds or thousands of categories.

All these codification strategies are defined independently of the data set and satisfy two properties:

- *Row separation*. In order to decrease misclassifications, the codewords should be as far apart from one another as possible. We can still recover the correct label for x even if several classifiers have responded wrongly. A measure of the error-correcting ability of any code is the minimum Hamming distance, H_c, between any pair of codewords. The number of errors that the code is guaranteed to be able to correct is $\left[\frac{H_c - 1}{2}\right]$.
- *Column separation*. It is important that the dichotomies given as the assignments to the ensemble members are as different from each other as possible. This will drive the ensemble towards low correlation between the classification errors (high diversity) which will hopefully increase the ensemble accuracy (Dietterich & Bakiri, 1995).

2.3. Problem-dependent code matrices

All the coding strategies described above are fixed in the ECOC matrix design step, defined without considering the problem characteristics or the classification performance. Recently some researchers (Alpaydin & Mayoraz, 1999; Crammer & Singer, 2002; Escalera, Pujol, & Radeva, 2007; Pujol et al., 2006, 2008; Utschick & Weichselberger, 2001; Zhou, Peng, & Suen, 2008) argue that the selection and the number of dichotomizers must depend on the performance of the ensemble on the problem at hand.

The first approach to design problem-dependent ECOC has been proposed in Alpaydin and Mayoraz (1999) where the backpropagation algorithm is used to drive the codewords for each class. However, this method is only applicable when the base learner is a

multi-layer perceptron. Utschick and Weichselberger (2001) also tried to optimize a maximum-likelihood objective function by means of the expectation maximization (EM) algorithm in order to achieve optimal decomposition of the multi-class problem into two-class problems.

Crammer and Singer (2002) proved that the problem of finding the optimal matrix is computationally intractable since it is (Non-deterministic Polynomial) NP-complete. Furthermore, they introduce the notion of continuous codes and cast the design problem of continuous codes as a constrained optimization problem.

Recently, Zhou et al. (2008) proposed a method called Data-driven ECOC (DECOC) to explore the distribution of data classes and optimize both decomposition process and the number of base learners. The key idea of DECOC is to selectively include some of the binary learners into the predefined initial codematrix based on a confidence score defined for each learner. The confidence score for each column is computed by measuring separability criteria of the corresponding binary problem. This measure is used to determine how likely a learner will be included in the ensemble. The method needs to search the output label space and ensure the validity of each candidate. Therefore, the efficiency of the method on problems with larger number of classes is limited.

The Discriminant ECOC (Pujol et al., 2006) renders each column of the codematrix to the problem of finding the binary partition that divides the whole set of classes so that the discriminability between both sets is maximum. The criterion used for achieving this goal is based on the mutual information between the feature indexes and class labels. Since the problem is defined as a discrete optimization process, the Discriminant ECOC uses the floating search method as a suboptimal search procedure for finding the partition that maximizes the mutual information. The whole ECOC matrix is created with the aid of an intermediate step formulated as a binary tree. Considering all the classes of the problem, a binary tree is built beginning from the root as follows: each node corresponds to the best bi-partition of the set of classes maximizing the quadratic mutual information between the class samples and their labels. The process is recursively applied until sets of single classes corresponding to the tree leaves are obtained. This procedure, ensure decomposition of the multi-class problem into $N_c - 1$ binary subproblems.

Forest ECOC (Escalera et al., 2007) is an extension of Discriminant ECOC. It takes advantage of the tree structure representation of the ECOC method to introduce a multiple-tree structured called "Forest" ECOC. This method is based on embedding different optimal trees in the ECOC approach to obtain the necessary number of classifiers assuring the required classification performance.

ECOC Optimizing Node Embedding (ONE) (Pujol et al., 2008) presents an approach that improves the performance of any initial codematrix by extending it in a sub-optimal way. ECOC-ONE creates the new dichotomizers by minimizing the confusion matrix among classes guided by a validation subset. As a result, overfitting is avoided and relatively small codes with good generalization performance are obtained.

Diversity and accuracy are two important concepts in designing any classifier ensemble, i.e. for building an accurate ensemble, we need diverse classifiers as accurate as possible (Kuncheva, 2004). Unfortunately none of the above mentioned methods considers these two concepts together for creating code matrices. In the following, we propose a heuristic method which produces a compact and Discriminant ECOC based on considering effect of each column of the codematrix on the diversity and accuracy of the whole ensemble.

3. Thinning ensemble

Common intuition suggests that the classifiers in the ensemble should be as accurate as possible and should not make coincident errors. This simple statement explains the importance of accuracy and diversity among members of a multiple classifier system. The methods for building ensembles which rely on inducing accuracy and diversity in an intuitive manner are very successful (Kuncheva, 2004).

For a classification task, error regions in most accurate classifiers highly overlap with each other. Consequently, in the design of any classifier ensemble, there is a trade-off between accuracy and diversity. Thinning the ensemble refers to a general strategy to design an ensemble with high recognition rate and minimum size based on a trade-off between accuracy and diversity of the base classifiers. Thinning strategies aim to improve any given ensemble with large number of base classifiers by removing classifiers that cause misclassifications. In Giacinto and Roli (2001) an ensemble is thinned by attempting to include the most diverse and accurate classifiers. Subsets of similar classifiers (those that make similar errors) are created and the most accurate classifier from each subset is selected. In Latinne, Debeir, and Decaestecker (2001), the McNemar test was used to determine whether to include a decision tree (DT) in an ensemble. This pre-thinning allowed an ensemble to be kept to a smaller size. Banfield, Hall, Bowyer, and Kegelmeyer (2005) introduce two methods for removing classifiers from an initial ensemble based on diversity measures. In *accuracy in diversity (AID) thinning* method, the classifiers that are most often incorrect on examples that are misclassified by many classifiers are removed from the ensemble. Another method called *concurrency thinning* algorithm, is based on the correctness of both the ensemble and the classifier with regard to a thinning set. A classifier is rewarded when a correct decision is made and rewarded even more when a correct decision is made and the ensemble is incorrect. A classifier is penalized in the event that both the ensemble and the classifier are incorrect. The procedure starts with all classifiers and the desired ensemble size is reached by removing one classifier at each step. The original thinning algorithm is shown in Algorithm 1.

Algorithm 1: The concurrency thinning algorithm (Banfield et al., 2005)

For each classifier C_i
  For each sample
    If Ensemble Incorrect and Classifier Incorrect
      $Metric_i = Metric_i - 2$
    If Ensemble Incorrect and Classifier Correct
      $Metric_i = Metric_i + 2$
    If Ensemble Correct and Classifier Correct
      $Metric_i = Metric_i + 1$
Remove C_i with the lowest $Metric_i$

It should be noted that all thinning algorithms will learn to overfit the training set, negatively affecting the generalization accuracy of the ensemble. A potential solution to the overfitting problem is to use a thinning set to determine the optimal ensemble configuration for the thinning algorithms.

4. Thinned-ECOC

We propose to engage the thinning strategy to automatically generate the codematrix for the problem at hand. The introduced method uses the concurrency thinning algorithm to shrink any initial code resulting in a matrix with minimum number of columns and maximum discrimination power, which leads to most efficient and effective ECOC ensemble.

4.1. Thinned-ECOC to design problem-dependent codematrix

This subsection introduces Thinned-ECOC to generate the code matrix by choosing the codewords utilizing the intrinsic information embedded in the training data. The key idea of Thinned-ECOC is to selectively remove some of the columns from the initial code matrix based on a metric defined for each column. This measure is used to determine how likely a column and its corresponding base classifier from the initial ECOC ensemble can be removed. The main steps of the Thinned-ECOC approach given below (Algorithm 2). Note that the process is iterated until the minimum size of desired optimal matrix is reached.

Algorithm 2: The generic algorithm of Thinned-ECOC

Set the initial codematrix M_{init} and minimum size of desired optimal matrix Θ
Train the base classifiers using M_{init} and build an initial ensemble
While $size(M) \geqslant \Theta$ do
  Calculate the $Metric_i$ for each column of M according to Algorithm 1
  Find the column with the lowest *Metric* on the thinning set
  Update M by removing the selected column
  Calculate the accuracy of the new ensemble on the thinning set
  If the accuracy of new ensemble improved
    $M_{opt} = M$

First questions to be addressed in building an ECOC ensemble are as follows: *which of the known code matrices is more suitable for the problem at hand*?, and *is it enough for building an accurate ensemble?* There are some cases where non of the known codes satisfy the classification goals while in other cases the desirable matrix for building accurate ensemble is a mixture of selected columns of some fixed codes. In fact, the main motivation behind the most recent ECOC contributions like ECOC-ONE (Pujol et al., 2008) and ForestECOC (Escalera et al., 2007) is that the fixed code matrices cannot lead to accurate enough ensembles. Therefore, the idea of extending an initial matrix in order to boost their accuracy is explored in some recent studies. For instance, in Zhou et al. (2008) the authors discuss that the final matrix can be considered as the combination of 1vsA, 2vsA, 3vsA and so on.

The final generated Thinned-ECOC is a subset of the given initial codematrix. Therefore the length of initial matrix must be large enough to achieve suboptimal search region in the column space. For the problems with small number of classes ($N_c < 10$) exhaustive coding can be used as the initial matrix. In the case that the number of classes is relatively high, exhaustive search is computationally unfeasible; for instance, the length of the Exhaustive code matrix for a 20-class problem is as large as $2^{19} - 1$.

The procedure of defining the initial matrix M_{init} as a combination of some known fixed codes considered here provides the possibility for the thinning algorithm to select the columns of final matrix from the columns of known codes which normally perform well in ECOC ensemble. As a result, the performance of the final matrix is expected to be at least as good as the initial components if appropriate thinning metric has been chosen. Fig. 1 shows an instant initial compound matrix for a 4-class problem, which is a combination of the 1vs1, 1vsA and BCH-7 code matrices.

Fig. 2 describes the flow of calculating the metrics and selecting the base learners to be removed. This flow stays at the core of the Thinned-ECOC algorithm. As clearly shown, Thinned-ECOC is a problem-dependent approach for designing codematrix: instead of having a preset matrix, the codematrix is adaptively generated based on the structure of the given training and thinning data.

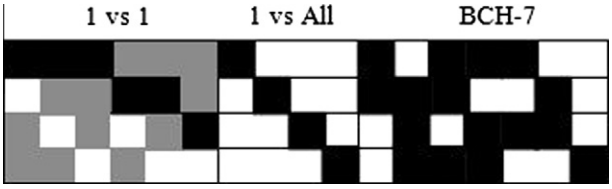

Fig. 1. The initial codematrix is a combination of three popular matrices: 1vs1, 1vsA and BCH-7 code matrices. The black, white and gray boxes represent 1, −1 and 0, respectively.

4.2. Analysis of Thinned-ECOC on a 4-class artificial dataset

To analyze properties of the proposed Thinned-ECOC and compare it with related state-of-the art coding methods, we have designed a 4-class toy problem (see Fig. 3). This synthetic 2D multiclass problem has 300 samples for each of the four classes. Each class is divided into two subsets, 200 samples for training and 100 for testing. 150 samples of each class from training set are used for training binary base classifiers and remaining 50 samples as the thinning data for guiding the thinning process. As shown in Fig. 3, class 1 (red pluses) has two main parts and is more complicated compared to the other classes. The multi-layer perceptron (MLP) with backpropagation learning role is also used for building base classifiers. We randomly set the initial weights of the MLP classifiers. Other parameters, determined experimentally, have been set as follows: number of iterations = 100, number of hidden nodes = 5 and learning rate = 0.05.

As a first experiment, we have investigated the relationship between the thinning metric and the improvement in the overall accuracy of ECOC codematrix. The accuracy of both unthinned initial matrix and final Thinned-ECOC is calculated for 100 independent runs. Fig. 4 depicts the mean metrics as well as the boost in accuracy when comparing accuracy of initial and final Thinned-ECOC. As indicated in Fig. 4, the boost in accuracy of the ECOC matrix is directly related to the average metrics of its columns. Therefore, for any improvement in accuracy of the initial matrix, the average metrics of its selected columns must be increased.

In this experiment, a combination of the 1vsA, 1vs1 and dense random codes is used to build the initial matrix for the thinning process. As indicated in Fig. 5, the initial matrix has 19 columns with relatively low accuracy (65%) while after the thinning process, the Thinned-ECOC is far more compact (7 columns), with higher accuracy (80%). It is worth noting that the thinning process also improves diversity and mean accuracy of the base classifiers by increasing the mean metrics. In fact, the resulted thinned matrix is the combination of 1vsA with three columns of 1vs1 in this case. From another perspective, it can be considered as an extension of 1vsA codematrix.

To investigate the impact of the initial matrix on the final Thinned-ECOC ensemble, two more experiments are considered. The thinning algorithm is run on the toy problem using 50 different random initial codes. The length of M_{init}s is selected so that "Shrinkage ratio" measure varies in a wide range from 1 to 15. This measure shows how big the initial matrix is compared with the final Thinned-ECOC matrix. To determine the approximate length of Thinned-ECOC for the toy problem, an initial matrix with large number of columns is used (15 times in this experiment). The length of M_{init} is then chosen to be smaller step by step (see the resulted graph in Fig. 6).

In the first experiment, the best and worst performances of thinning algorithm are recorded and their differences (BA-WA)

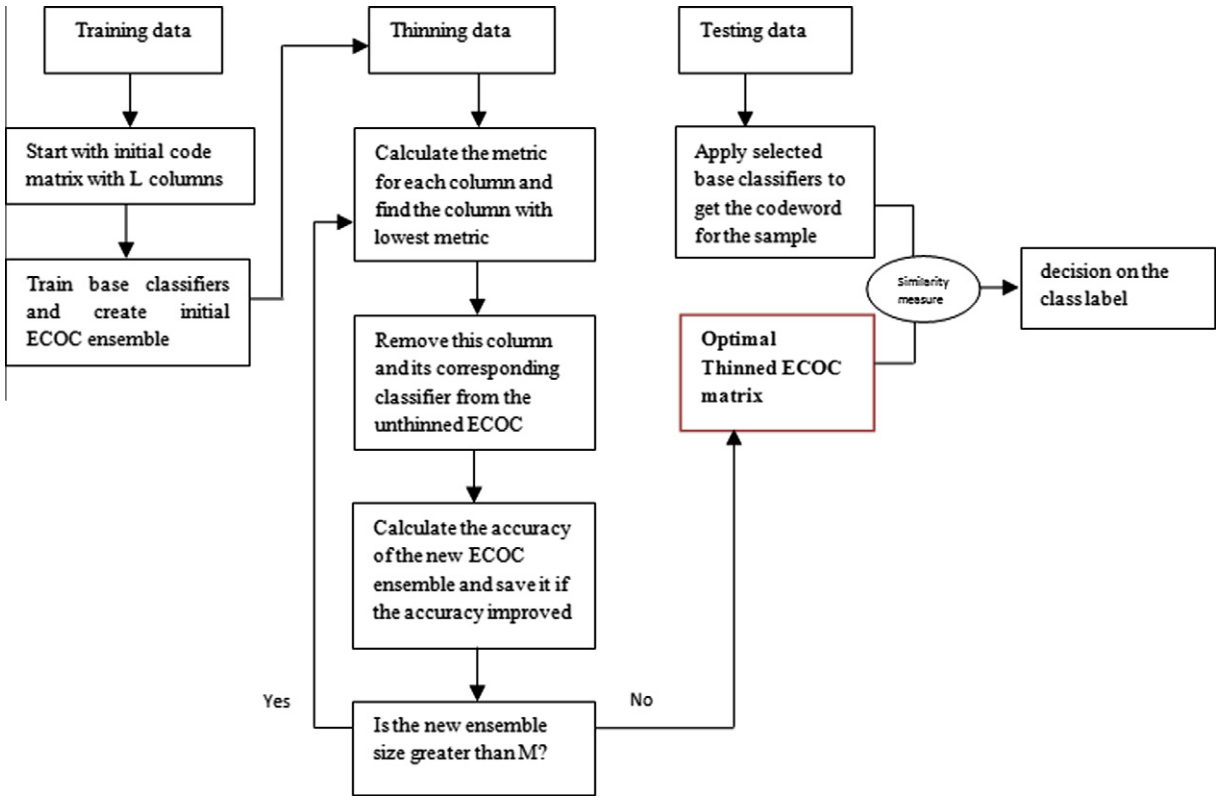

Fig. 2. The flow of training, thinning and testing algorithms for Thinned ECOC.

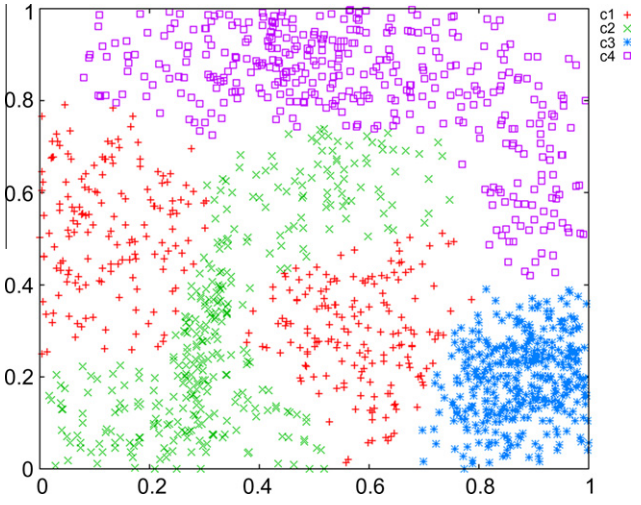

Fig. 3. Distribution of the four classes for the toy problem.

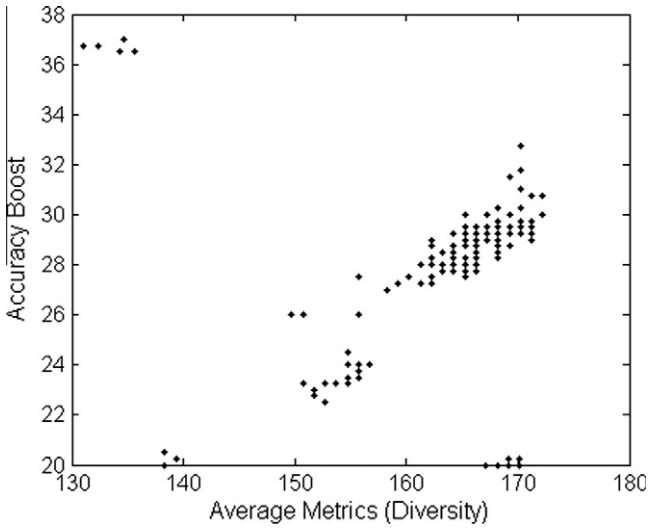

Fig. 4. Average metric against boost in accuracy in the thinning process. The correlation between average metrics and accuracy boost is 0.52.

are calculated for each "shrinkage ratio" point. This can be really helpful in finding the best length for an initial matrix to get the best possible performance. For example, when the length of M_{init} is chosen to be 1.2 times longer than Thinned-ECOC, it is less likely to get best result within the first runs compared to the case when the length of M_{init} is 8 times longer. In the second experiment, the performance of ECOC ensemble before and after the thinning process is recorded. Fig. 6 (right) also shows that longer M_{init} leads to Thinned-ECOC which is nearer to the optimal codematrix up untill the point where it does not help anymore and graph is saturated. Both experiments lead to the conclusion that if the length of an initial matrix is 10 times longer than the final Thinned-ECOC then the achieved result is almost trustable and there is no need to further search.

Fig. 7 illustrates two small examples of using the thinning set for guiding the thinning process. The best and the worst performances of Thinning ECOC are shown in Fig. 7(a) and (b), respectively. The initial codematrix embedded 19 classifiers and the thinning algorithm progressively removes its columns until its length reached to 9 and 7 ($\Theta = 9$ and 7). After getting the maximum value, the accuracy starts to decrease since there are no longer enough classifiers to support an ensemble. In this critical point

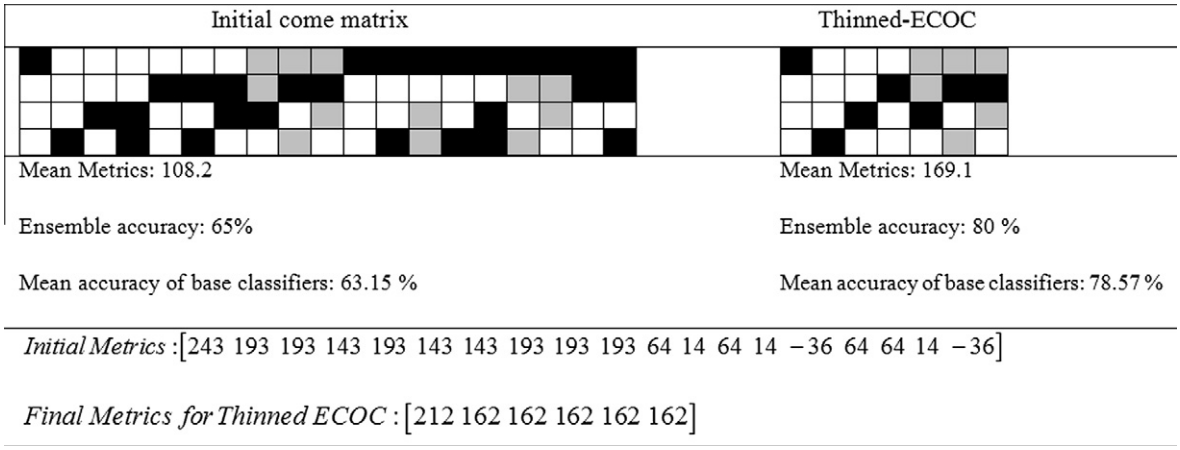

Fig. 5. Comparison of an instance codematrix before and after of the thinning on the toy problem.

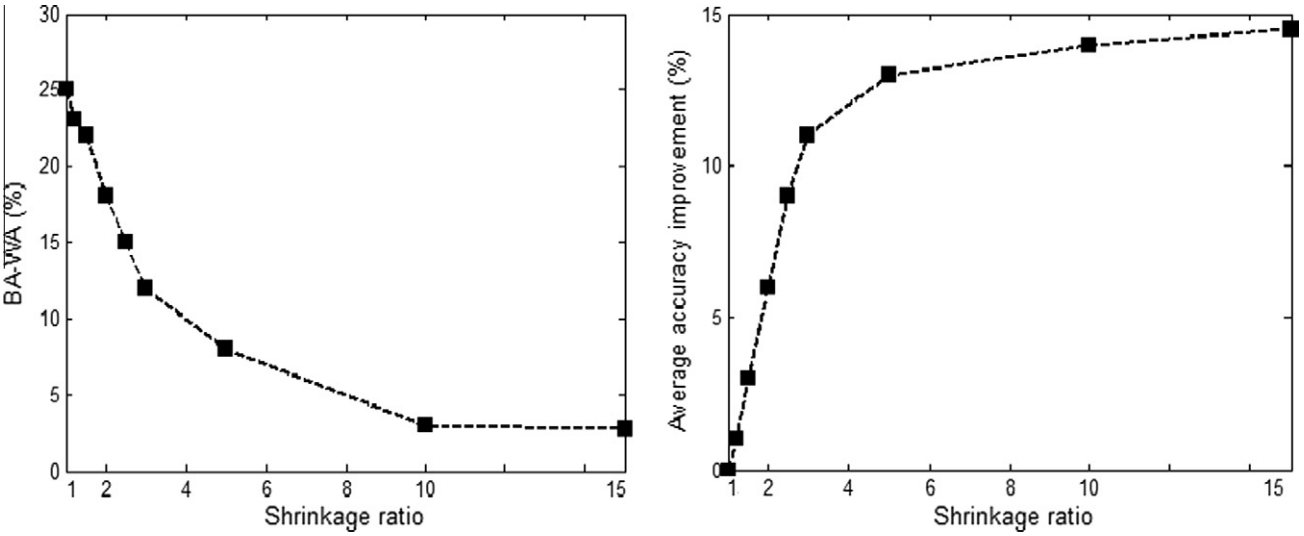

Fig. 6. Impact of length of the initial matrix on the performance of Thinned-ECOC: average differences of the best and worst accuracy (BA-WA) for the final Thinned-ECOC (left) and average accuracy improvement before and after the thinning process (right). *Shrinkage_ratio* = $Length(M_{init})/Length(M_{Thinned})$ where $Length(M_{Thinned})$ is assumed fixed for each problem.

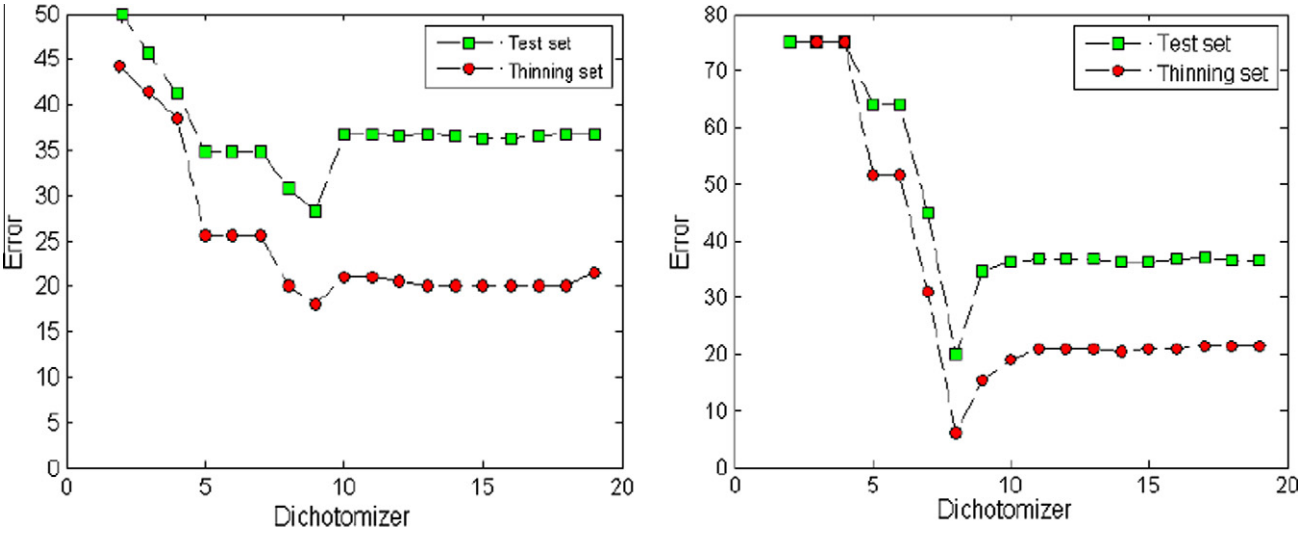

Fig. 7. The worst (right) and the best (left) performances of the Thinned-ECOC algorithm on the toy data.

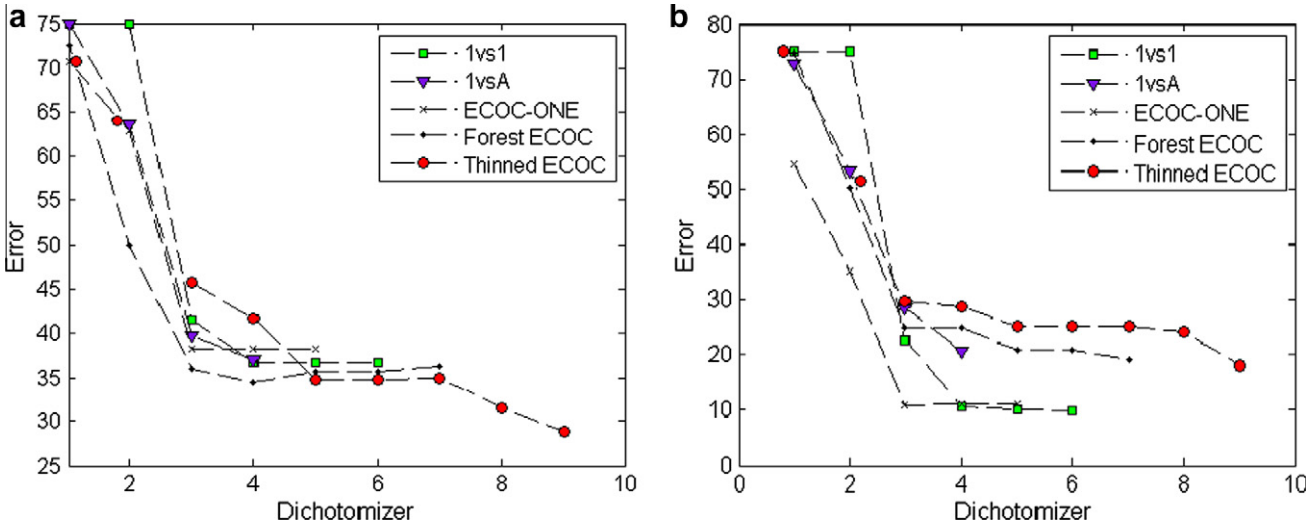

Fig. 8. Testing evaluation (a) and training evaluation (b) for the toy problem.

(where the number of columns reaches to 7 and 9, respectively for best and worst results i.e. when the error rate has its minimum in Fig. 7) the resulted ECOC matrix has its own optimal situation i.e. maximum discrimination power and minimum size. Fig. 7 shows how the diversity and accuracy concepts, embedded together in the metric, can be used to shrink the matrix length while improving accuracy of the overall ECOC classifier. Codematrix resulted from worst thinning process has 9 columns and 28% error rate on test data where as the best performance of the algorithm result in a more compact codematrix (7 columns) and a more accurate ensemble, with 19% error rate. However, as shown in the following, both of these Thinned-ECOCs are more accurate compared to the existing coding methods.

An illustration of the test evolution process for all the selected techniques is shown in Fig. 8(a), in which the error is given as a function of the number of dichotomizers. One can observe a greater error reduction for Thinned-ECOC compared to the rest of methods. The training evolution for the same problem is shown in Fig. 8(b), where the number of dichotomizers and the error rate are shown at the x and y-axis, respectively.

We compare Thinned-ECOC with five state-of-the art coding methods with different sizes: Discriminat ECOC (Pujol et al., 2006), ECOC-ONE (Pujol et al., 2008), Forest ECOC (Escalera et al., 2007), 1vs1 and 1vsA. Table 1 shows the 10-fold cross-validation results for all the considered ECOC. In this table, the number of dichotomizers, mean and variance of the recognition rates are presented. We used two different algorithms for building our base learners: MLP and Fisher Linear Discriminant Analysis (FLDA). Numerical results presented in Table 1 show that our technique outperforms the considered related methods. It should be noted that each base classifier in the different ensemble methods uses identical parameters in order to make a fair comparison.

Table 1
Recognition rates of the ECOC classifiers using different coding strategies and base classifiers for the toy problem. Best results are given in bold.

Code matrix	1vsA	1vs1	Disc. ECOC	Forest ECOC	ECOC ONE	Thinned ECOC
FLDA	50.5	51.0	48.5	50.0	49.5	**54.2**
MLP	59.25	63.1	64.02	64.12	61.7	**74.3**

5. Experimental results

In this section we validate the proposed method using some of the UCI machine learning datasets. Furthermore, we investigate the Thinned-ECOC performance on the following real-world problems: cancer classification and face recognition.

5.1. Validation on the UCI database

To evaluate the proposed Thinned-ECOC, experiments on eleven datasets from the UCI machine learning repository (Murphy & Aha, 1994) are carried out. Commonly used as benchmarks for classification, these datasets (given in Table 2) include both real-world and synthetic problems with various characteristics.

For the datasets with no train/test partitioning, the classification performance assessed by the 10-fold cross-validation provides realistic generalization accuracy for unseen data. Each of the training dataset is split into training and thinning sets with a ratio of 70% and 30%, respectively. We have used different types of classification algorithms: FLDA, MLP and Support Vector Machines (SVM) as base learners in order to show that the proposed algorithm is independent of the particular base classifier. The error backpropagation algorithm was used for the training of the MLP base classifiers and the iterative estimation process was stopped when an average squared error of 0.9 over the training set was obtained, or when the maximum number of iterations is reached (adopted mainly for preventing networks from overtraining). We also varied the number of hidden neurons to experimentally find

Table 2
The main characteristics of the selected UCI datasets.

Problem	# Train	# Test	# Attributes	# Classes
Abalone	4177	–	8	28
Ecoli	336	–	8	8
Dermatology	366	–	34	6
Glass	214	–	9	7
Iris	150	–	4	3
Letter	20,000	–	16	26
Pendigits	7494	3498	16	10
Satimage	4435	2000	36	6
Segment	210	2100	19	7
Vowel	990	–	11	11
Yeast	1484	–	8	10

Table 3
Recognition rate on the selected UCI datasets using FLDA as a base classifier. Best results are given in bold.

Codematrix	1vs1	1vsA	BCH	Dense rand.	Sparse rand.	Disc. ECOC	ECOC ONE	Forest ECOC	Thinned ECOC
Aba.	22.7	2.9	14.0	13.1	15.2	25.8	**28.0**	25.5	25.8
Ecoli	77.9	70.9	70.9	73.0	75.2	79.8	**80.9**	78.1	80.1
Derma.	90.7	86.9	89.1	92.7	**93.4**	90.9	90.2	93	90.0
Glass	75.6	44.5	55.2	44.8	42.6	66.2	66.2	67.0	**77.9**
Iris	95.5	93.2	93.0	95.5	94.1	96.0	**96.9**	95.0	96.7
Letter	69.5	65.9	66.7	68.6	68.9	70.0	69.5	63.9	**70.2**
Pendi.	93.1	40.2	80.3	68.4	70.9	96.1	**97.1**	95.9	94.2
Sat.	**83.9**	81.1	82.9	83.0	82.1	80.0	81.1	79.2	83.7
Seg.	83.19	42.8	80.2	79.1	75.5	84.0	84.3	82.0	**85.86**
Vowel	**71.2**	25.3	33.7	41.3	44.4	52.9	53.9	60.0	60.2
Yeast	52.2	30.5	49.7	47.3	41.7	50.1	49.9	48.7	**54.5**
EUF	385.58	372.33	380.03	379.52	379.39	384.29	384.64	384.13	**385.78**
Rank	2	9	6	7	8	4	3	5	**1**

Table 4
Recognition rate on the selected UCI datasets using MLP as a base classifier. Best results are given in bold.

Codematrix	1vs1	1vsA	BCH	Dense rand.	Sparse rand.	Disc. ECOC	ECOC ONE	Forest ECOC	Thinned ECOC
Aba.	26.6	4.4	17.9	15.5	16.9	25.9	27.3	25.5	**28.0**
Ecoli	80.1	75.0	69.9	74.1	73.0	79.8	80.9	78.1	**81.0**
Derma.	**94.7**	88.9	90.0	93.5	90.8	92.9	93.2	93.0	94.0
Glass	**58.5**	44.4	54.2	50.1	49.5	56.2	56.2	56.0	56.9
Iris	**97.1**	95.5	93.1	94.7	94.5	95.0	95.5	94.8	96.3
Letter	75.5	69.8	73.0	72.2	73.4	74.6	74.9	73.7	**77.2**
Pendi.	96.2	93.3	94.3	96.1	95.7	97.1	97.3	96.9	**97.5**
Sat.	85.0	83.0	84.3	83.3	83.3	86.1	**87.1**	86.1	**87.1**
Seg.	83.1	52.0	78.2	75.7	70.7	84.0	84.3	80.4	**84.8**
Vowel	61.4	55.5	60.0	59.1	60.0	62.9	63.9	62.7	**64.1**
Yeast	54.3	40.8	49.0	49.3	51.5	50.0	49.9	49.7	**55.0**
EUF	385.39	379.07	382.74	382.63	382.46	384.94	385.26	384.55	**385.88**
Rank	2	9	6	7	8	4	3	5	**1**

Table 5
Recognition rate on the selected UCI datasets using SVM as a base classifier. Best results are given in bold.

Codematrix	1vs1	1vsA	BCH	Dense rand.	Sparse rand.	Disc. ECOC	ECOC ONE	Forest ECOC	Thinned ECOC
Aba.	25.8	3.1	16.3	14.7	15.7	24.1	25.9	25.5	**27.1**
Ecoli	**81.6**	75.9	70.7	74.1	74.5	79.8	80.9	79.8	80.9
Derma.	93.7	89.1	90.1	92.5	91.4	94.1	**95.0**	93.1	94.4
Glass	58.	49.7	55.2	54.1	54.6	56.2	56.8	55.6	**59.9**
Iris	**96.9**	94.5	94.1	96.5	96.5	96.3	96.3	96.0	**96.9**
Letter	**75.5**	72.1	73.7	74.9	74.4	74.6	75.0	72.9	75.2
Pendi.	**96.0**	91.4	92.3	94.1	93.7	94.2	**96.0**	95.1	**96.0**
Sat.	84.0	82.2	83.1	83.7	82.5	84.0	83.4	83.1	**85.7**
Seg.	84.1	42.8	80.2	79.1	75.5	84.0	84.1	80.4	**85.8**
Vowel	66.4	61.1	63.7	64.1	63.1	62.9	64.2	65.3	**68.4**
Yeast	54.6	37.8	49.7	51.3	51.7	51.6	49.9	50.8	**56.5**
EUF	385.64	378.88	383.02	383.48	383.23	384.79	385.09	385.09	**386.17**
Rank	2	9	8	6	7	5	4	3	**1**

the optimal architecture of the MLPs for each problem. The other parameter values used for training are as follows: learning rate is 0.4 and momentum parameter is 0.6. In the case of SVM serving as base classifiers, linear kernel is used. All other parameters for these three algorithms are chosen according to the standard setting of MATLAB Toolboxes.

The Thinned-ECOC is compared to five popular decomposition methods: 1vsA, 1vs1, BCH, dense and sparse random. For generating dense and sparse random codes, we tested 5000 matrices and selected the matrices that maximize the row and column Hamming distance (Allwein et al., 2000). The decoding process for all mentioned techniques is the Margin decoding as it shows better results compared with Hamming decoding (Ko & Kim, 2005). Tables 3–5 show the recognition rates of different decomposition strategy using the FLDA, MLP and SVM learners on the selected UCI datasets.

As indicated in Tables 3–5, Thinned-ECOC outperforms the other coding algorithms. A statistical analysis is performed to clearly emphasize the ranking of the compared methods. For this purpose, we used the expected utility approach (Golden & Assad, 1984) based on the following function: $\gamma - \beta(1 - \bar{b}t)^{-\bar{c}}$, where $\gamma = 500$, $\beta = 100$ and $t = 0.005$. $\bar{b}$ and $\bar{c}$ are calculated using the following expressions: $\bar{b} = \frac{s^2}{\bar{x}}$ and $\bar{c} = \left(\frac{\bar{x}}{s}\right)^2$, where $\bar{x} = \frac{1}{M}\sum_{i=1}^{M} x_i$, $s^2 = \frac{1}{M}\sum_{i=1}^{M}(x_i - \bar{x})^2$ and x_i, $i = 1,\ldots,M$ (M is the number of datasets considered) is the percentage deviation of an obtained accuracy from the best possible solution (100%). The last two lines of Tables 3–5 present the statistical analysis results: the expected utility function (denoted by EUF) values and the

Table 6
Codematrix length of different methods for the selected UCI datasets.

Codematrix	1vs1	1vsA	BCH	Dense rand.	Sparse rand.	Disc. ECOC	ECOC ONE	Forest ECOC	Thinned ECOC
Aba.	378	28	511	48	72	27	58	70	61
Ecoli	28	8	31	30	45	7	18	25	16
Derma.	15	6	15	25	38	5	9	13	9
Glass	21	7	31	28	42	6	9	15	9
Iris	3	3	7	15	23	2	4	7	3
Letter	325	26	255	47	70	25	40	68	41
Pendi.	45	10	31	33	49	9	16	26	18
Sat.	15	6	15	25	38	5	10	22	8
Seg.	21	7	31	28	42	6	13	18	13
Vowel	55	11	63	34	51	10	22	40	21
Yeast	45	10	31	33	49	9	20	33	22

corresponding rank for each method. The Thinned-ECOC obtains first rank for all base learners while 1vs1 is ranked second based on a near EUF value.

In addition, Table 6 compares the codematrix length of the considered methods for each UCI dataset. It can be observed that the proposed Thinned-ECOC leads to reasonably short codematrices. Although 1vsA and Discriminant ECOC result in more compact matrices, it was statistically proven to be the less accurate methods (as shown in Tables 3–5). This result further highlights Thinned-ECOC classifiers as the most accurate with a reasonably short possible length.

Fig. 9 shows an example on how the proposed thinning algorithm shrinks the initial matrix from 78 to 22 on the Segment data while its accuracy is increased from 77% to 88%. It is worth noting that, unlike the other coding methods, in Thinned-ECOC length and structure of the codematrix is not fixed and may differ from a run to another.

5.2. Gene expression-based cancer classification

The bioinformatics problem of classifying different tumor types is of great importance in cancer diagnosis and drug discovery. Accurate prediction of different tumor types has great value in providing better treatment and toxicity minimization to the patients. However, most cancer classification studies are clinical-based and have limited diagnostic ability. Cancer classification using gene expression data is known to contain the keys for addressing the fundamental problems related to cancer diagnosis and drug discovery. The recent advent of DNA microarray technique has made simultaneous monitoring of thousands of gene expressions possible. With this abundance of gene expression data, researchers have started to explore the possibilities of cancer classification using gene expression data. The issue of "high dimension low sample size" referred to as HDLSS problem in statistics (Marron & Todd, 2002; Raudy & Jain, 1991) has to be tackled in this context. A considerable number of methods have been proposed in recent years with promising results. However, there are still a lot of issues which need to be addressed and understood.

To evaluate the effectiveness of the proposed approach, we carried out experiments on the two multi-class datasets of gene expression profiles. Two expression datasets popularly used in research literature are the NCI (Ross et al., 2000; Scherf et al., 2000) and Lymphoma (Alizadeh et al., 2000). The details of these data sets are summarized in Table 7. Note that the number of tissue samples per class is generally small (e.g. <10 for NCI data) and unevenly distributed (e.g. from 46 to 2 in Lymphoma data). This aspect together with the large number of classes (e.g. 9 for Lymphoma data) makes the classification task very complex.

As shown in Table 7, these datasets contain expression levels of thousands of genes originally, i.e., the dimension of the data is very high. Therefore, we first applied a popular dimension reduction method, i.e. Principal Component Analysis (PCA) (Fukunaga, 1990).

The classification performance is assessed using "Leave-One-Out Cross Validation" (LOOCV). For presentation clarity, we give the number of LOOCV errors in Tables 8 and 9. Then, we compare Thinned-ECOC with five common decomposition methods: 1vsA, 1vs1, BCH, dense and sparse random. Tables 8 and 9 show the error rates of different decomposition strategy using the SVM, MLP and decision tree (DT) techniques, on the NCI and Lymphoma data sets,

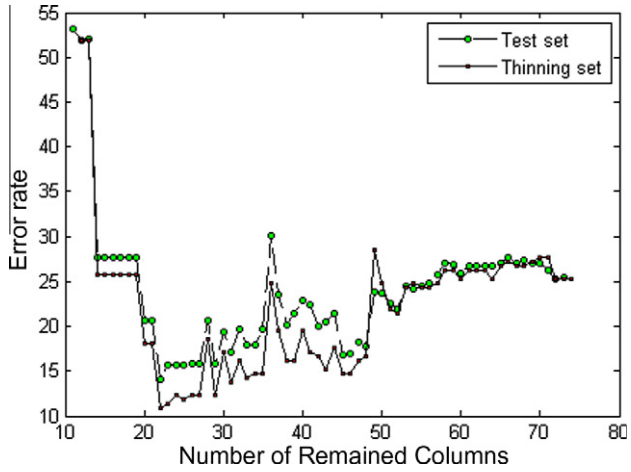

Fig. 9. Thinned set versus test set error rate on Segment data from the UCI repository.

Table 7
Multi-class gene expression data sets for different tissue types.

Class	NCI data set		Lymphoma data set	
	Class name	# of samples	Class name	# of samples
1	NSCLC	9	Diffuse large B cell Lympho.	46
2	Renal	9	Chronic Lympho. leukemia	11
3	Breast	8	Activated blood B	10
4	Melanoma	8	Follicular Lymphoma	9
5	Colon	7	Resting/activated T	6
6	Leukemia	6	Transformed cell lines	6
7	Ovarian	6	Resting blood B	4
8	CNS	5	Germinal center B	2
9	Prostate	2	Lymph node/tonsil	2
Total number of samples		60		96
Dimension		9703		4026

Table 8
Error rate on the NCI dataset. Best results are given in bold.

Feat. size	Base learner	1vs1	1vsA	BCH	Dense rand.	Sparse rand.	Disc. ECOC	ONE	Forest ECOC	Thinned ECOC
30	SVM	8.0	14.7	10.3	8.7	11.0	9.4	8.6	8.9	**7.1**
	MLP	7.3	15.8	11.6	7.2	11.1	7.3	7.0	7.9	**6.2**
	DT	9.1	17.3	13.4	8.6	12.2	8.0	7.9	8.3	**7.7**
60	SVM	5.2	10.1	7.8	6.5	8.1	**5.0**	6.7	7.1	6.3
	MLP	**5.3**	11.0	6.7	5.4	6.5	5.5	5.5	5.9	**5.3**
	DT	6.4	12.2	8.7	6.7	6.2	6.4	**6.0**	6.5	**6.0**

Table 9
Error rate on the Lymphoma dataset. Best results are given in bold.

Feat. size	Base learner	1vs1	1vsA	BCH	Dense rand.	Sparse rand.	Disc. ECOC	ONE	Forest ECOC	Thinned ECOC
30	SVM	8.2	14.1	10.0	8.5	10.0	7.9	7.5	7.9	**7.1**
	MLP	8.7	15.7	10.8	9.1	10.1	8.7	8.5	8.3	**7.5**
	DT	8.9	14.9	10.5	9.3	10.7	8.5	**7.9**	8.9	**7.9**
60	SVM	**6.0**	10.3	7.3	**6.0**	7.1	6.3	6.1	6.5	**5.9**
	MLP	7.7	11.1	7.9	7.2	8.0	7.7	7.3	7.9	**6.7**
	DT	7.3	11.9	7.9	**6.9**	7.5	7.9	7.5	7.5	**6.9**

respectively. The decision tree classifiers are developed using C4.5 algorithm (Quinlan, 1993), which is an extension of ID3 (Quinlan, 1986). At each node of the tree, C4.5 chooses one attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. The criterion is the normalized information gain that results from choosing an attribute for splitting the data. The attribute with the highest normalized information gain is chosen to make the decision. The C4.5 algorithm improves computing efficiency, deals with continuous values, handles attributes with missing values and avoids over fitting. All experiments have been carried out using two different numbers of input feature size, 30 and 60. As shown in Tables 8 and 9, proposed Thinned-ECOC outperforms the other code generation algorithms for this problem. For the NCI dataset, Thinned-ECOC is able to obtain clearly better results for each codematrix considered than any other related method (see Table 8). The results for the Lymphoma dataset (see Table 9) also emphasize the proposed method as the most accurate one. For the latter dataset, the Dense Random code is also able to obtain competitive results when the feature space size is 60 and SVM or DT are used as base classifers.

5.3. Face recognition problem

Machine recognition of faces from still and video images has numerous commercial and law enforcement applications. These application areas range from access control, information security and smart cards to surveillance and biometrics. In this subsection, the performance of Thinned-ECOC on the face recognition problem is investigated. The Yale face database is engaged in experiments. This database contains 165 gray scale images of 15 individuals (11 images for each individual). The images demonstrate variations in lighting condition, facial expression (normal, happy, sad, sleepy, surprised and wink) and accessories. Samples of the Yale database are shown in Fig. 11. We used first 6 samples of each individual for training and the remaining 5 samples for testing our classifiers.

The proposed face recognition model consists of two processing stages (see Fig. 10): (i) representation and (ii) recognition and reliability measure. In the representation stage, any input retinal image is transformed into a low dimension vector with an appropriate representation in the MLP input. The recognition stage is of vital importance and relies on an ECOC ensemble with MLP as a base learner.

5.3.1. Representation stage

In the first stage of our face recognition model, we use PCA (Fukunaga, 1990) to avoid a high dimensional and redundant input space, and optimally design and train the binary classifiers. The resulting low-dimensional representation is used for face processing. PCA is the simplest and most efficient method for coding faces (Turk & Pentland, 1991); however, other methods such as linear Discriminant Analysis, LDA, (Duda, Hart, & Stork, 2000) and independent-component analysis, ICA, (Bartlett, Movellan, & Sejnowski, 2002) have revealed good results. For the current model, it is only important to have a low dimensional code to ease generalization to new faces. For the proposed model, we consider the first 50 eigenvectors with the largest 50 eigenvalues.

5.3.2. Recognition stage and reliability measure

We used the MLP as a base learner in an ECOC ensemble for the recognition stage of the model. Each base learner has 50 input nodes for PCA components, 25 neurons in its hidden layer and 1 output node for binary labels. They have been trained with 200 iterations and $\eta = 0.03$. Here we extend this idea and define the *Robustness Rate* (*RR*) of a decision for the face recognition model as follows:

$$RR = \frac{H_d(cw_2, \bar{y}) - H_d(cw_1, \bar{y})}{H_d(cw_2, cw_1)} \times 100 \tag{4}$$

where cw_1 and cw_2 are the closest and second closest rows of the codematrix to the output vector $\bar{y}$ given by ECOC classifier for each test sample and H_d is the Hamming distance between two codewords. A robustness threshold can be set on *RR* so that testing samples with *RR* smaller than the threshold can be rejected. The threshold can be adjusted based on trade-off between recognition rate and error rate. For example, for applications with low tolerance on errors such as those in information security, the threshold is set higher and the error rate can be reduced at the cost of more rejections. A common choice of the threshold is around 0.1 so that if the difference between the top two Hamming distances is less than 10% of the length of codeword then the testing sample is rejected.

Finally, the reliability of the face recognition model is defined as follows:

$$Reliability = \frac{RecognitionRate}{RecognitionRate + ErrorRate} \tag{5}$$

The performance of an ECOC ensemble, with different coding strategies on the Yale face dataset is depicted in Table 10. The results emphasize that the proposed Thinned-ECOC has better performance and reliability in comparison with other coding strategies. For the fixed code matrices, there is a trade-off between accuracy and their length. The short matrix of 1vsA has lowest accuracy while the 1vs1 is the most accurate with number of

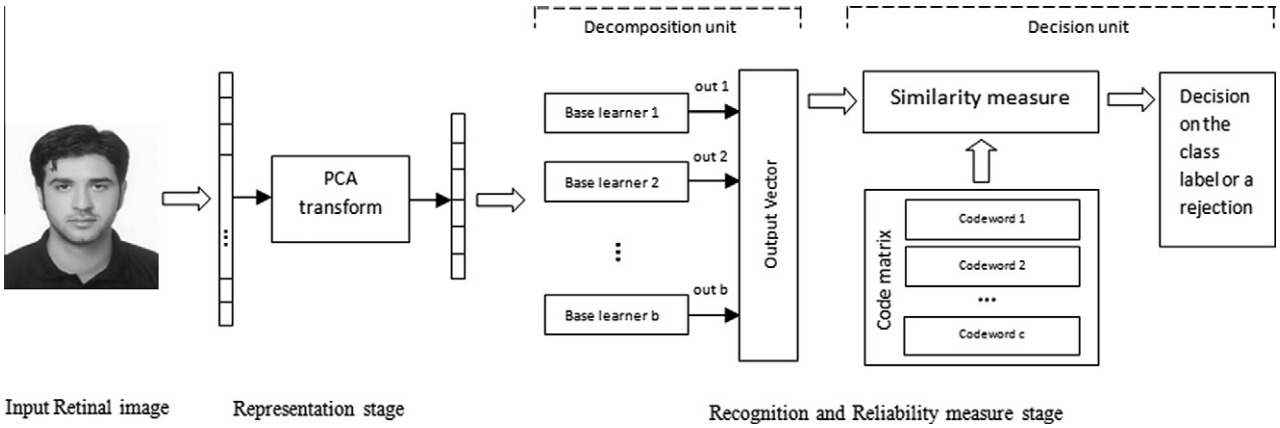

Fig. 10. The proposed model consists of two main stages: face representation and recognition.

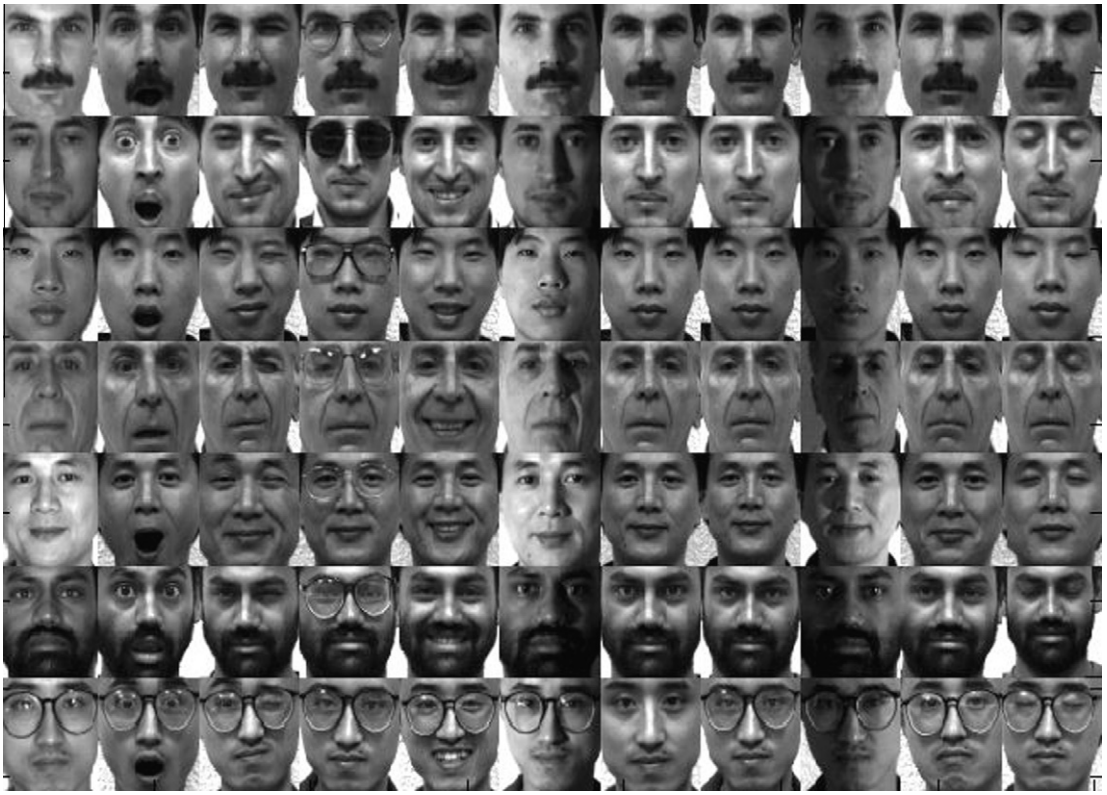

Fig. 11. Samples of face images from the Yale dataset, with variation in facial expression and illumination.

Table 10
Reliability and recognition rate of ECOC ensemble with different coding strategies on the Yale face dataset. Best results are given in bold.

Codematrix	1vs1	1vsA	Dense rand.	Sparse rand.	BCH	Disc. ECOC	ECOC ONE	Forest ECOC	Thinned ECOC
Length	105	15	39	59	63	**14**	23	38	28
Rec. rate	**72.15**	66.99	69.68	70.93	68.5	67.4	70.5	69.1	**74.67**
Rel.	**89.2**	85.1	88.1	88.5	86.0	84.1	88.8	86.5	**90.24**

columns as much as 105. However current results indicate that Thinned-ECOC does not follow this rule since it leads to the most accurate ECOC with relatively small number of base learners.

6. Conclusions and future work

Thinned-ECOC is introduced as a heuristic method for application-dependent design of error-correcting output codes. The proposed method takes advantage of the thinning algorithm for building ensemble in the problem of designing optimal codematrix. A metric has been defined for the thinning process and used to shrink any initial codematrix by removing some redundant and unnecessary columns. As a result, a compact matrix with high discrimination power is obtained leading to an ECOC ensemble with high accuracy and lower number of base classifiers. The proposed Thinned-ECOC algorithm is validated using the UCI machine

learning database and applied successfully to two real-world problems: the face recognition and the classification of cancer tissue types. Statistical analysis results confirm Thinned-ECOC as the most accurate ECOC ensemble.

Problems with huge number of classes such as web page classification and text categorization are of significant importance in today's real-world applications. These classification tasks are too difficult to handle using existing code generation methods since they result in huge size matrices hard to manage. Future work focuses on the application of the proposed method to these challenging tasks exploring one of the major advantages of Thinned-ECOC i.e. building accurate ECOC classifier with smaller number of base classifiers.

Acknowledgments

The author thanks Camelia Chira, Giuliano Armano, Fabio Roli and Terry Windeatt for valuable suggestions that helped improve this paper. This work has been partially supported by Iranian Telecommunication Research Centre.

References

Ali, K. M., & Pazzani, M. J. (1995). *On the link between error correlation and error reduction in decision tree ensembles*. Technical Report ICS-UCI (pp. 95–138).

Alizadeh, A. A. et al. (2000). Distinct types of diffuse large Bcell lymphoma identified by gene expression profiling. *Nature, 403*, 503–511.

Allwein, E. L., Shapire, R. E., & Singer, Y. (2000). Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research, 1*, 113–141.

Alpaydin, E., & Mayoraz, E. (1999). Learning error-correcting output codes from data. In *International conference on artificial neural networks (ICANN99)* (Vol. 2, pp. 743–748).

Banfield, R. E., Hall, L. O., Bowyer, K. W., & Kegelmeyer, W. P. (2005). Ensemble diversity measures and their application to thinning. *Information Fusion, 6*, 49–62.

Bartlett, M., Movellan, J., & Sejnowski, T. (2002). Face recognition by independent component analysis. *IEEE Transactions on Neural Networks, 13*(6), 1450–1464.

Crammer, K., & Singer, Y. (2002). On the learnability and design of output codes for multiclass problems. *Machine Learning, 47*(2), 201–233.

Dietterich, T. G., & Bakiri, G. (1995). Solving multiclass learning problems via error correcting output codes. *Journal of Artificial Intelligence Research, 2*, 263–286.

Duda, R. O., Hart, P. E., & Stork, D. G. (2000). *Pattern classiffication*. Hoboken, NJ: Wiley-Interscience.

Escalera, S., Pujol, O., & Radeva, P. (2007). Boosted landmarks of contextual descriptors and forest-ECOC: A novel framework to detect and classify objects in cluttered scenes. *Pattern Recognition Letters, 28*, 1759–1768.

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalisation of on-line learning and an application to boosting. *Journal of Computer and System Science, 55*(1), 119–139.

Fukunaga, K. (1990). *Introduction to statistical pattern recognition* (2nd ed.). Boston: Academic Press.

Giacinto, G., & Roli, F. (2001). An approach to automatic design of multiple classifier systems. *Pattern Recognition Letters, 22*, 25–33.

Golden, B. L., & Assad, A. A. (1984). A decision-theoretic framework for comparing heuristics. *European Journal of Operational Research, 18*, 167–171.

Hastie, T., & Tibshirani, R. (1998). Classification by pairwise grouping. *The Annals of Statistics, 26*(5), 451–471.

Hatami, N., Ebrahimpour, R., Ghaderi, R. (2008). ECOC-based training of neural networks for face recognition. In *3rd IEEE international conference on cybernetics and intelligent systems (CIS)* (pp. 450–454).

Jacobs, R. A., Jordan, M. I., Nowlan, S. E., & Hinton, G. E. (1991). Adaptive mixture of experts. *Neural Computation, 3*, 79–87.

Ko, J., & Kim, E. (2005). On ECOC as binary ensemble classifiers. *LNAI, 3587*, 110.

Kuncheva, L. I. (2004). *Combining pattern classifiers: Methods and algorithms*. Wiley Interscience.

Latinne, P., Debeir, O., & Decaestecker, C. (2001). Limiting the number of trees in random forests. In *2nd Internatinal workshop on multiple classifier systems* (pp. 178–187).

Lin, S., & Costello, D. J. (2004). *Error control coding* (2nd ed.). Prentice-Hall, Inc..

Marron, J. S., & Todd, M. (2002). *Distance weighted discrimination*. Technical Report. School of Operations Research and Industrial Engineering. Cornell University.

Murphy, P. M., & Aha, D. W. (1994). *UCI repository of machine learning databases*. Irvine: Dept. of Information and Computer Science, Univ. of California.

Peterson, W. W., & Weldon, J. R. (1972). *Error-correcting codes*. Cambridge, MA: MIT Press.

Pujol, O., Escalera, S., & Radeva, P. (2008). An incremental node embedding technique for error correcting output codes. *Pattern Recognition, 41*, 713–725.

Pujol, O., Radeva, P., & Vitria, J. (2006). Discriminant ECOC: A heuristic method for application dependent design of error correcting output codes. *IEEE Transactions on PAMI, 28*(6), 1001–1007.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning, 1*, 81–106.

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. Morgan Kauffman.

Raudy, S. J., & Jain, A. K. (1991). Small sample size effects in statistical pattern recognition: Recommendations for practitioners. *IEEE Transactions on PAMI, 13*(3), 252–264.

Ross, D. T., Scherf, U., et al. (2000). Systematic variation in gene expression patterns in human cancer cell lines. *Nature Genetics, 24*(3), 227–234.

Scherf, U., Ross, D. T., et al. (2000). A cDNA microarray gene expression database for the molecular pharmacology of cancer. *Nature Genetics, 24*(3), 236–244.

Turk, M., & Pentland, A. (1991). Face recognition using eigenfaces. In *IEEE conference computer vision and pattern recognition* (pp. 586–591).

Utschick, W., & Weichselberger, W. (2001). Stochastic organization of output codes in multiclass learning problems. *Neural Computation, 13*(5), 1065–1102.

Windeatt, T., & Ghaderi, R. (2001). Binary labeling and decision level fusion. *Information Fusion, 2*, 103–112.

Windeatt, T., & Ghaderi, R. (2003). Coding and decoding strategies for multi-class learning problems. *Information Fusion, 4*, 11–21.

Zhou, J., Peng, H., & Suen, C. Y. (2008). Data-driven decomposition for multi-class classification. *Pattern Recognition, 41*, 67–76.