Pattern Recognition 46 (2013) 1091–1100

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

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

Error-correcting output codes based ensemble feature extraction

Guoqiang Zhong, Cheng-Lin Liu*

National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences, No. 95 Zhongguancun East Road, Beijing 100190, PR China

ARTICLE INFO

Article history:
Received 21 May 2012
Received in revised form
22 September 2012
Accepted 20 October 2012
Available online 30 October 2012

Keywords:
Feature extraction
Ensemble learning
Error-correcting output codes (ECOC)
Meta learner

ABSTRACT

This paper proposes a novel feature extraction method based on ensemble learning. Using the error-correcting output codes (ECOC) to design binary classifiers (dichotomizers) for separating subsets of classes, the outputs of the dichotomizers are linear or nonlinear features that provide powerful separability in a new space. In this space, the vector quantization based meta classifier can be viewed as an ECOC decoder, where each learned prototype of a class can be seen as a codeword of the class in the new representation space. We conducted extensive experiments on 16 multi-class data sets from the UCI machine learning repository. The results demonstrate the superiority of the proposed method over both existing ECOC approaches and classic feature extraction approaches. In particular, the decoding strategy using a meta classifier is shown to be more computationally efficient than the linear loss-weighted decoding in state-of-the-art ECOC methods.

1. Introduction

Feature extraction is an essential issue in many areas of pattern recognition and machine learning. Informative features extracted from data can benefit the subsequent learning, analysis and recognition. Traditional linear feature extraction methods, such as principal components analysis (PCA) [1] and linear discriminant analysis (LDA) [2], have been widely used for finding a linear subspace of the data. However, they may fail to discover the intrinsic low-dimensional structure when data lie on a nonlinear manifold. Since the publication of two seminal manifold learning algorithms, isometric feature mapping (Isomap) [3] and locally linear embedding (LLE) [4], a plenty of nonlinear manifold learning methods have been developed [5–9]. However, most of them are unsupervised and cannot deal with the out-of-sample problem easily [10]. Moreover, most of the existing feature extraction methods learn new representations directly from the data, without integrating the output information of classifiers. From our observation, probabilistic outputs of a group of classifiers are fairly effective features to indicate to which class a sample belongs. This motivates us to explore an ensemble feature extraction method using a set of basic classifiers.

The learning algorithms that construct a set of classifiers and then predict new data points based on these classifiers are generally called as *ensemble learning* [11]. The notable ensemble learning methods include bagging [12], boosting [13], error-correcting output codes (ECOC) [14], and stacking (stacked generalization) [15], among others. Many works in the literature have proven the advantages of ensemble over single classifiers, both theoretically and experimentally. Meanwhile, ensemble learning methods have been successfully applied to many real-world problems, such as optical character recognition (OCR) [16,17], face recognition [18], speaker recognition [19], remote sensing [20], and multimodal interaction [21]. In this paper, we focus on the ECOC technique, which is a framework for combining binary classifiers (dichotomizers) to address multi-class problems. Based on the ECOC framework, we present an ensemble feature extraction method to learning discriminative representations of the data.

The ECOC framework generally includes two steps: the coding step and the decoding step. The coding strategies include one-versus-all [22], one-versus-one [23], data-driven ECOC [24], discriminant ECOC (DECOC) [25], and ECOC-optimizing node embedding (ECOCONE) [26]. Among them, one-versus-all and one-versus-one are problem-independent ECOC design methods, whilst data-driven ECOC, DECOC and ECOCONE are problem-dependent. The commonly used decoding strategies are Hamming decoding [22] and Euclidean decoding [23]. Some researchers have introduced loss-based function [27] or probabilities [28,29] in decoding. Recently, Escalera et al. [30] proposed two novel ternary ECOC decoding strategies, β-density decoding and loss-weighted decoding, and showed their advantages over the state-of-the-art decoding strategies. To the best of our knowledge, however, there has not been a published work that attempts to integrate feature extraction into the ECOC framework. That is, all the existing methods train dichotomizers in the data space and combine the classifiers using decoding strategies, but not try to explore the intrinsic geometric structure of the data belonging to different classes. On the other hand, learning new features via the

* Corresponding author. Tel.: +86 10 62558820; fax: +86 10 62551993.
E-mail addresses: gqzhong@nlpr.ia.ac.cn (G. Zhong), liucl@nlpr.ia.ac.cn (C.-L. Liu).

0031-3203/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2012.10.015

combination of the basic classifiers can significantly benefit the classification accuracy. In addition, as far as we know, all the existing ECOC methods endow only one codeword for each class. Actually, from the perspective of vector quantization [31], if the data distribution is relatively complex, more than one codewords (or called prototypes) can be helpful to characterize the distribution of the class.

In this paper, we propose a novel ensemble feature extraction method based on the ECOC framework. It takes advantage of the discrimination ability of the dichotomizers for separating subsets of classes designed by ECOC. Accordingly, we call it ECOC based ensemble feature extraction (ECOC-EFE). In ECOC-EFE, the new representation of a datum is actually the probabilistic outputs of the combined dichotomizers, where each element indicates the probability that the datum belongs to the corresponding positive class. Based on the extracted features, we employ a generalized learning vector quantization (GLVQ) classifier [32] as a meta learner for classification in the new feature space. The learned prototypes of a class by GLVQ can be viewed as codewords of the class in the new space, while the classification of the meta learner can be viewed as a decoding step, corresponding to that in the ECOC framework. From the viewpoint of ensemble learning, our method can be considered as a new framework for multi-class learning problems, which includes four steps—ECOC coding, feature extraction, recoding (or meta learning) and decoding (or classification). In experiments on 16 data sets from the UCI machine learning repository using linear and nonlinear binary classifiers, the proposed method is demonstrated superior classification performance compared to state-of-the-art feature extraction and ECOC decoding methods.

In the remainder of this paper, Section 2 gives a brief review of related works on feature extraction and ECOC based ensemble learning; Section 3 introduces the notation used in this paper; Section 4 describes the proposed method in detail; Section 5 presents the experimental results; Section 6 concludes this paper with remarks.

2. Related works

Over the past few decades, many feature extraction methods have been proposed, such as PCA [1], LDA [2], kernel PCA (KPCA) [33] and generalized discriminant analysis (GDA) [34]. From the nature of feature representation, these methods can be classified into two categories: linear and nonlinear. Linear methods, such as PCA and LDA, generally find a linear projection to map the data into a low-dimensional subspace. In contrast, nonlinear methods, such as KPCA, GDA and some manifold learning methods, usually connect the original space with the feature space or low-dimensional manifold via a nonlinear function. Compared to linear methods, nonlinear feature extraction methods can be applied to more complex data, but they mostly suffer from a common problem that they can hardly deliver the exact mapping function except the coordinates of the training data in the new space. Moreover, most of the existing linear and nonlinear feature extraction methods do not explore the class structure of the data adequately to yield sufficiently high classification accuracy.

The ECOC framework is to combine binary classifiers (dichotomizers), such as support vector machines (SVMs) [35] and Adaboost [36], to solve multi-class classification problems. Dietterich and Bakiri [14] presented the basic ECOC framework represented using a coding matrix of *binary* symbols. Each column of the coding matrix represents a binary partition of the whole classes in two subsets $\{-1,+1\}$. Alternatively, each row of the matrix is a codeword assigned to the corresponding class. The one-versus-all [22] strategy is a special case of the binary-symbol-based ECOC. Afterwards, Allwein et al. [27] extended the coding strategy by introducing a third symbol '0', which allows some classes to be neglected by the dichotomizers and leads to the increment of subgroups of classes to be considered in the *ternary* ECOC framework. The one-versus-one (pairwise) classification strategy [23] can be viewed as a special case of the ternary ECOC framework. Most of the ECOC methods specify the coding matrix just in the coding step, i.e., predefine it independently of the problem, such as the above one-versus-all, one-versus-one, and the dense and sparse random coding strategies [27].

Considering the nature of classification problem or the structure of the data can lead to better coding matrix design. The first problem-dependent ECOC design method was proposed by Utschick and Weichselberger [37]. However, their experimental results showed that for many multi-class problems, the best performance was still given by the one-versus-all method. Crammer and Singer [38] have reported improvement in the design of ECOC matrix, but they proved that finding the optimal discrete codes is NP-hard with the number of classes. The discriminant ECOC (DECOC) [25] is a heuristic method for learning the coding matrix by exploring the hierarchical structure of the class space. It generates a binary tree structure for the hierarchical partition by maximizing a discriminative criterion. In addition to its superior classification performance, DECOC leads to a very compact codeword with length $C-1$, where C is the number of classes. Pujol et al. [26] proposed a new approach that improves the initial ECOC matrix in a sub-optimal way. It creates new dichotomizers by minimizing the confusion matrix among classes guided by a validation subset. A length of $2C$ bits for the codeword has been suggested. Recently, Escalera et al. [39] proposed a method to redefine the ECOC matrix without re-training. This re-coding strategy can be applied over any coding design.

However, all these ECOC coding strategies, either problem-dependent or problem-independent, suffer from two shortages: they do not use the combined dichotomizers to extract useful features of the data, and they endow only one codeword for each class. Addressing these two problems may lead to significant improvement of classification accuracy.

We noticed that some researchers have tried meta learning for combining binary classifiers for multi-class classification [40–42]. Savický and Fürnkranz [40], Lezoray and Cardot [41] combine pairwise (one-versus-one) classifiers using meta classifier (C4.5) trained with stacking. They observed more or less improved classification performance compared to the other pairwise coupling and fusion algorithms. Shiraishi and Fukumizu [42] combine one-versus-all or one-versus-one binary classifiers using the multinomial logistic regression as the meta learner. These methods were tested on binary classifiers in only one-versus-one or one-versus-all coding strategy, and were not compared to the advanced decoding methods like the recent loss-weighted decoding methods [30]. The results of [30] show that the ECOCONE coding strategy mostly give the best classification performance and the loss-weighted decoding is among the best for combining binary classifiers. Hence, we build our ensemble method on the ECOCONE coding strategy and compare with the loss-weighted decoding.

On the hand of feature extraction, Rueda et al. [43] proposed a method to extract linear subspace features from pairs of classes and combine the two-class decisions by voting and meta learning. This falls in the framework of ECOC but is limited in the sense that it considers only one-versus-one encoding and linear feature extraction.

3. Notation

We use boldface uppercase letters to denote matrices, such as $\mathbf{K}$, and boldface lowercase letters to denote vectors, such as $\mathbf{v}$. The ith row of a matrix $\mathbf{K}$ is denoted as $\mathbf{K}_{i*}$. $\mathbf{K}_{ij}$ denotes the entry at the

ith row and jth column of $\mathbf{K}$. $\mathbf{v}_i$ denotes the ith entry of $\mathbf{v}$. $\mathbf{K}^T$ and $\mathbf{v}^T$ are the transpose of $\mathbf{K}$ and $\mathbf{v}$, respectively. $\mathrm{tr}(\mathbf{K})$ is the trace of matrix $\mathbf{K}$.

For multi-class problems, we denote the given training data as $\mathcal{X}=\{(\mathbf{x}_1,c_1),(\mathbf{x}_2,c_2),\ldots,(\mathbf{x}_{N_{trn}},c_{N_{trn}})\}$, where N_{trn} is the number of training samples, $\mathbf{x}_i \in \Re^D$ is a D-dimensional input vector, and c_i is the class label of $\mathbf{x}_i$. The number of classes is denoted as C, i.e., $c_i \in \{1,\ldots,C\}$. The test data is denoted as $\mathcal{Y}=\{\mathbf{y}_1,\mathbf{y}_2,\ldots,\mathbf{y}_{N_{tst}}\}$, where N_{tst} is the number of test samples. We use the matrices $\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_{N_{trn}}]^T$ and $\mathbf{Y}=[\mathbf{y}_1,\mathbf{y}_2,\ldots,\mathbf{y}_{N_{tst}}]^T$ to denote the training data matrix and test data matrix, respectively.

For ECOC, we denote the coding matrix as $\mathbf{M}$, where $\mathbf{M}_{ij} \in \{-1,0,1\}$. The length of codewords is denoted as p, i.e., $\mathbf{M} \in \{-1,0,1\}^{C\times p}$.

4. ECOC based ensemble feature extraction (ECOC-EFE)

In this section, we describe the details of the proposed method ECOC-EFE. For clarity, we present each step of ECOC-EFE—coding, feature extraction, recoding and decoding, respectively in the following subsections.

4.1. Coding

The coding step of ECOC-EFE is to design an ECOC matrix specifying the dichotomizers to be combined for multi-class classification. For completeness, we analyze the advantages and disadvantages of some existing coding strategies and describe the strategies used in our experiments.

The widely used coding strategies include one-versus-all [22], one-versus-one [23], dense random [27], sparse random [27], DECOC [25], and ECOCONE [26]. Among them, one-versus-all and dense random are binary-symbol-based strategies, while the others are ternary-symbol-based ones. On the other hand, the one-versus-all, one-versus-one, dense random and sparse random are problem-independent coding strategies, while DECOC and ECOCONE are problem-dependent ones.

The one-versus-one strategy can be considered as the most effective with respect to (w.r.t.) the training of the combined dichotomizers. However, it is the one that needs to combine the most number of dichotomizers, which is $O(C^2)$, against $O(C)$ for one-versus-all and DECOC. For large number of classes, the complexity of dischotomizers training and combining is formidable. On the other hand, the one-versus-all strategy, though combines C dichomizers only, needs to use all the training data to learn each dichotomizer. For some dichotomizers (such as the SVM), the training complexity is $O(N_{trn}^2)$ or higher, where N_{trn} is the number of training samples. DECOC pursues a tree structure of the classes, and only involves all the classes in the first column of the ECOC matrix, whose length of codeword is fixed to $C-1$. Thus, DECOC can be considered as an ideal coding strategy for its compactness. ECOCONE is a method that extends incrementally an initial, any type of ECOC matrix. Hence, ECOCONE can be considered as a useful design to capture the discrimination between the subsets of the classes. Based on the experimental results reported in [30] that dense random and sparse random rarely outperform the four coding strategies discussed above, we will not take them into account in our discussion and experiments.

For our ECOC-EFE, we adopt ECOCONE as the coding strategy, which is initialized via a DECOC configuration. Thus, the length of the codewords in our method is generally around C, resulted from the extension of the DECOC initialization. For more algorithmic details of DECOC and ECOCONE, refer to [25,26]. The choice of the coding strategy actually determines the dimensionality of the new feature space of ECOC-EFE since the length of the codeword, i.e., the number of combined dichotomizers, is equal to the dimensionality of the new feature space. We use DECOC to initialize the ECOCONE algorithm such that the dimensionality of the resulting feature space is moderate. Due to the adaptation of ECOCONE, the dimensionality is not necessarily limited to $C-1$, i.e., the dimensionality can be intelligently learned depending on the data structure and separability so as to maximize the discrimination between classes.

After the ECOC matrix is specified, we can train p dichotomizers independently according to the columns of the ECOC matrix, where p is the length of the codeword, and as well, the dimensionality of the new feature space.

4.2. Feature extraction

On training dichotomizers according to the ECOC matrix, the outputs of the dichotomizers on a new input vector can be taken as the features in a new space. In principle, all types of binary classifiers can be used for this purpose. We nevertheless adopt the state-of-the-art SVM [35] as the base classifier and transform its output to probabilistic confidence using the sigmoid function [44–46]. Using the SVM as dichotomizer enables performance comparison with the state-of-the-art multi-class SVM classifier and kernel feature extraction (KPCA and GDA). Transforming the outputs of dichotomizers to probabilities (each value indicates the probability that the input pattern belongs to the corresponding positive class), as a measure of feature normalization, may help improve the classification performance in the new feature space. More specifically, we use the linear SVM dichotomizer for linear feature extraction (ECOC-LFE) and the SVM with radial basis function (RBF) kernel for nonlinear feature extraction (ECOC-NLFE). The RBF kernel has been shown to be among the best choices for nonlinear SVM for classification.

Before introducing the details of feature extraction, we briefly outline the linear and nonlinear SVMs. The linear SVM solves a quadratic programming problem

$$\min_{\mathbf{w},\mathbf{b},\xi_i} \quad J(\mathbf{w})=\frac{1}{2}\|\mathbf{w}\|^2$$
$$s.t. \quad c_i f(\mathbf{x}_i) \geq 1-\xi_i, \quad \xi_i \geq 0,\ i=1,\ldots,N_{trn}, \tag{1}$$

where $c_i \in \{+1,-1\}$, $\mathbf{w}$ is the weight vector, ξ_i's are the slack variables. The binary discriminant function is

$$f(\mathbf{x})=\mathbf{w}^T\mathbf{x}+b. \tag{2}$$

We solve this problem and obtain

$$\mathbf{w}=\sum_{i=1}^{N_{trn}} \alpha_i c_i \mathbf{x}_i \tag{3}$$

and

$$b=\frac{1}{N_{SV}}\sum_{i=1}^{N_{SV}}(\mathbf{w}^T\mathbf{x}_i-c_i), \tag{4}$$

where α_i's are the non-negative Lagrange multipliers, N_{SV} is the number of support vectors. The dual form of Problem (1) can be written as

$$\max_{\alpha} \quad \sum_{i=1}^{N_{trn}} -\frac{1}{2}\sum_{i,j}\alpha_i\alpha_j c_i c_j \mathbf{x}_i\mathbf{x}_j = \sum_{i=1}^{N_{trn}} -\frac{1}{2}\sum_{i,j}\alpha_i\alpha_j c_i c_j k(\mathbf{x}_i,\mathbf{x}_j)$$
$$\text{s.t.} \quad 0 \leq \alpha_i \leq \lambda,\ i=1,\ldots,N_{trn},$$
$$\sum_{i=1}^{N_{trn}} \alpha_i c_i = 0, \tag{5}$$

where $k(\mathbf{x}_i,\mathbf{x}_j)=\mathbf{x}_i^T\mathbf{x}_j$ is the linear kernel function, λ is a constant number and $\alpha=\{\alpha_1,\ldots,\alpha_{N_{trn}}\}^T$ is the vector of Lagrange multipliers.

Replacing the linear kernel function in Problem (5) with a nonlinear kernel function gives the formulation of the nonlinear SVM. We use the RBF kernel

$$k(\mathbf{x}_i,\mathbf{x}_j)=\exp(-\gamma^{-1}\|\mathbf{x}_i-\mathbf{x}_j\|^2), \tag{6}$$

where γ is the parameter for the kernel function. On solving Problem (5), we can obtain α_i's, b and the discriminant function

$$f(\mathbf{x})=\sum_{i=1}^{N_{SV}}\alpha_i c_i k(\mathbf{x}_i,\mathbf{x})+b. \tag{7}$$

In our experiments, we empirically set γ as half of the average within-class variance, i.e.

$$\gamma=\frac{1}{2N_{trn}}\sum_{l=1}^{C}\sum_{j=1}^{N_{C_l}}\|\mathbf{x}_j-\mathbf{m}_l\|^2, \tag{8}$$

where N_{C_l} is the number of samples in class l and $\mathbf{m}_l$ is the mean vector of data in class l.

To extract new features of the data, we use the trained dichotomizers to classify each training sample $\mathbf{x}_i$ and transform the output discriminant function to approximate posterior probability using the sigmoid function

$$\mathbf{Z}_{i,j}=\sigma(a_j f_j(\mathbf{x}_i)+b_j)=\frac{1}{1+\exp[-(a_j f_j(\mathbf{x}_i)+b_j)]},\quad j=1,\ldots,p. \tag{9}$$

The parameters $\{a_j,b_j\}$, $j=1,\ldots,p$, can be estimated on a validation data set using a regularized cross-entropy criterion [44]. In practice, however, we found that for SVMs, the simple choice of $(a_j=1,b_j=0)$ gives fairly high performance. On testing the training samples in the original space, $\mathbf{Z}=\{\mathbf{z}_1,\ldots,\mathbf{z}_{N_{trn}}\}^T$ is the data matrix of new representation for training meta classifiers in the p-dimensional space. More precisely, for linear feature extraction (ECOC-LFE), we use Eqs. (2) and (9) to calculate p confidence outputs of the linear SVMs as a new representation; while for nonlinear feature extraction (ECOC-NLFE), we use Eqs. (7) and (9) to calculate the new features based on the nonlinear SVMs.

4.3. Recoding (meta learning)

Although many approaches have been developed to learn the ECOC matrix from data, as far as we know, all the existing ECOC methods only define one codeword for each class. As discussed earlier, if the structure of the data is relatively complex, one codeword for each class may not guarantee satisfactory decoding. We address this problem by introducing a meta learning procedure.

We formulate the recoding as a learning vector quantization (LVQ) problem. Specifically, we learn m prototypes for each class in the new feature space of dichotomizers outputs and take the learned prototypes as codewords of that class. We choose the generalized learning vector quantization (GLVQ) [32] as the meta learner, which has demonstrated superiority in nearest-prototype-based classification. The GLVQ algorithm is outlined in the following.

Let ζ_1 be the nearest prototype vector that belongs to the same class of $\mathbf{z}$, ζ_2 be the nearest prototype vector that belongs to a different class from $\mathbf{z}$. The relative distance difference $\varphi(\mathbf{z})$ is defined as

$$\varphi(\mathbf{z})=\frac{d_1-d_2}{d_1+d_2}, \tag{10}$$

where $d_1=\|\mathbf{z}-\zeta_1\|^2$ and $d_2=\|\mathbf{z}-\zeta_2\|^2$ are the squared Euclidean distance of $\mathbf{z}$ from ζ_1 and ζ_2, respectively. The GLVQ learns the prototypes on a labeled data set by minimizing an empirical loss

$$\min E=\sum_{i=1}^{N_{trn}}\sigma(\varphi(\mathbf{z}_i)), \tag{11}$$

where $\sigma(\cdot)$ is the sigmoid function. To minimize E, ζ_1 and ζ_2 are updated by stochastic gradient descent

$$\zeta_1\leftarrow\zeta_1+4\tau\frac{\partial\sigma}{\partial\varphi}\frac{d_2}{(d_1+d_2)^2}(\mathbf{z}-\zeta_1), \tag{12}$$

$$\zeta_2\leftarrow\zeta_2-4\tau\frac{\partial\sigma}{\partial\varphi}\frac{d_2}{(d_1+d_2)^2}(\mathbf{z}-\zeta_2), \tag{13}$$

where τ is the step size and $\partial\sigma/\partial\varphi=\sigma(1-\sigma)$ is the gradient of σ w.r.t. φ. To speed up learning, the modified updating rules were suggested [32]

$$\zeta_1\leftarrow\zeta_1+4\tau\frac{\partial\sigma}{\partial\varphi}\frac{d_2}{(d_1+d_2)}(\mathbf{x}-\zeta_1), \tag{14}$$

$$\zeta_2\leftarrow\zeta_2-4\tau\frac{\partial\sigma}{\partial\varphi}\frac{d_2}{(d_1+d_2)}(\mathbf{x}-\zeta_2). \tag{15}$$

To recode the codewords for each class, we train the GLVQ meta classifier on the new features of the data and take the learned prototypes of each class as new codewords. In our experiments, we empirically set the number of prototypes for each class as $\lceil N_{trn}/(100\times C)\rceil\times 3$, where $\lceil t\rceil$ is the most close integer that is larger than t. We use the k-means clustering algorithm to cluster the training samples of each class and use the cluster centers as initial prototypes of GLVQ.

4.4. Decoding (classification)

In the decoding step, we feed all the test data into the learned dichotomizers and obtain the p-dimensional representation of them. The new feature vectors are then tested using the learned meta classifier. In the case of prototype based meta classifier learned by GLVQ, specifically, the test sample (in the new feature space) is assigned to the class of nearest prototype. This classification rule can be viewed as the extension of distance-based decoding with multiple codewords.

In summary, we show the process of our ECOC-EFE framework in Algorithm 1.

Algorithm 1. Process of ECOC-EFE.

Input:
  $\mathcal{X}=\{(\mathbf{x}_1,c_1),(\mathbf{x}_2,c_2),\ldots,(\mathbf{x}_{N_{trn}},c_{N_{trn}})\}$; $\mathcal{Y}=\{\mathbf{y}_1,\mathbf{y}_2,\ldots,\mathbf{y}_{N_{tst}}\}$;
Output:
  Decoding result;
Steps:
  Coding:
  (1) Learn the ECOC matrix, $\mathbf{M}\in\{-1,0,1\}^{C\times p}$;
  (2) Train the dichotomizers according to $\mathbf{M}$;
  Feature extraction:
  (1) Test each training sample using the learned dichotomizers;
  (2) Get the confidence outputs of each dichotomizer;
  (3) Take the confidence values of each sample as its new features;
  Recoding:
  (1) Train a GLVQ meta classifier based on the new feature representation;
  (2) Take the learned prototypes of each class as new codewords;

Decoding:
(1) Test each test sample using the dichotomizers and obtain the new feature representation;
(2) Decode via the meta classifier.

5. Experiments

To evaluate the classification performance of the proposed ECOC-EFE (including ECOC-LFE and ECOC-NLFE), we conducted extensive experiments on 16 multi-class data sets from the UCI machine learning repository. We compared ECOC-EFE with classic feature extraction methods and start-of-the-art ECOC methods. As below, we present the used data sets, parameter settings, statistical comparison methods, detailed results and discussions, respectively.

5.1. Data sets

Following [30], we test the compared methods on 16 multi-class data sets from the UCI machine learning repository. These data sets have various numbers of classes, attributes and samples. The details of them are summarized in Table 1. Particularly, for data sets that include class with less than 10 samples, we perturb the samples by adding standard Gaussian noise and append them to that class until its sample size exceeds 10. For each data set, we rescale all the features to be within $[-1,1]$. For two data sets: Segmentation and Optdigits, some attributes have the same value over all the samples. In this case, we use $1/\sqrt{D}$ to replace the invalid scaled value, where D is the number of attributes. For each data set, the classification results and running times are reported based on average over stratified 10-fold cross-validation.

5.2. Parameter settings

As mentioned earlier, we use ECOCONE as the coding strategy of ECOC-EFE, which is initialized using DECOC. Thus, the dimensionality of the new feature space is around C based on the extension of the initial configuration, where C is the number of classes. Following [30], in the implementation of ECOC-EFE and all the compared ECOC methods, the penalty factor for SVMs is set as $\lambda = 1$. The kernel width for RBF kernel is set to half of the average within-class variance as shown in Eq. (8). We did not attempt to optimize the parameters of SVMs, but it is fair to compare different ECOC methods using the same parameter setting for the dichotomizers. For the GLVQ meta learner, the number of prototypes of each class is empirically set as $\lceil N_{trn}/(100 \times C) \rceil \times 3$, where $\lceil t \rceil$ is the most close integer that is larger than t, and N_{trn} is the number of training samples.

Besides, we use the ECOC library [47] to implement the ECOC algorithms. The parameters for coding and decoding just follow what the library defined. Moreover, we use the OSU-SVM toolbox[1] to train the SVMs.

5.3. Statistical comparison

To statistically compare the classification results, we conduct the Wilcoxon signed-ranks test [48] for the comparison between two methods, whilst the Friedman test [49] and the Nemenyi test [50] for comparing multiple methods, as suggested by [51]. The details of these tests can be found in [51].

[1] http://www.support-vector-machines.org/SVM_soft.html.

Table 1
The UCI data sets (T, training samples; A, attributes; C, classes).

Problem	♯ of T	♯ of A	♯ of C	Problem	♯ of T	♯ of A	♯ of C
Balance	625	4	3	Satimage	6435	36	7
Dermatology	366	34	6	Segmentation	2310	19	7
Ecoli	336	8	8	Shuttle	14,500	9	7
Glass	214	9	7	Thyroid	215	5	3
Iris	150	4	3	Vehicle	846	18	4
Letter	20,000	16	26	Vowel	990	10	11
Optdigits	5620	64	10	Wine	178	13	3
Pendigits	10,992	16	10	Yeast	1484	8	10

5.4. Visualization of features

In this experiment, we show the 2D embedding learned by ECOC-NLFE, PCA and LDA. Fig. 1 plots the results of four data sets: Balance, Iris, Thyroid and Wine, which all have three classes. The training and test data are randomly partitioned with ratio around 9:1. As we can see from Fig. 1, the overlapping between classes in the 2D embeddings obtained by PCA and LDA is generally heavier than that obtained by ECOC-NLFE. As a result, the discrimination between classes yielded by ECOC-NLFE is much better than that yielded by PCA and LDA. The classification results shown in Table 6 confirm this observation.

5.5. Comparison with state-of-the-art ECOC methods

In this experiment, we compare the proposed ECOC-EFE method with some state-of-the-art ECOC methods. The compared coding strategies include one-versus-one (OnevsOne), one-versus-all (OnevsAll), DECOC and ECOCONE, where ECOCONE is initialized by the DECOC method. For all the coding strategies, we use the linear loss-weighted (LLW) decoding method, which was shown superior for multi-class classification [30]. For ECOC-NLFE and the compared ECOC methods, SVMs with RBF kernel are used as dichotomizers. Similarly, for ECOC-LFE and the corresponding ECOC methods, linear SVMs are used as dichotomizers. The classification results obtained by ECOC-NLFE and ECOC based methods are shown in Table 2, while those obtained by ECOC-LFE and the ECOC methods are shown in Table 3.

To evaluate the significance of the performance differences, we conduct the Friedman and the Nemenyi test [51] with a confidence value 0.05 on the results presented in Table 2. The statistical comparison results show that ECOC-NLFE is significantly better than the one-versus-all coding design, and at least comparable with the other ones. We then conduct the same statistical tests on the results presented in Table 3 and observe the tendency as that for ECOC-NLFE. Although the mean rank of ECOC-LFE is a little lower than the one-versus-one method, the performance difference of them is not statistically significant. It is noteworthy that the ECOC-EFE uses much less dischotomizers than the one-versus-one strategy despite their comparable performance.

Table 4 shows the average decoding time of ECOC-NLFE (not including the recoding time) and the compared ECOC methods. It is easy to see, for all the data sets, that ECOC-NLFE is the fastest among the compared methods. For some data sets, such as Dermathology and Ecoli, the decoding of ECOC-NLFE is nearly 50 times faster than that of the one-versus-one method, and nearly 20 times faster than that of the DECOC method. For ECOC-LFE and the compared ECOC methods, we have the same observation that the decoding of ECOC-LFE is dramatically faster than that of the compared ECOC methods. For simplicity, the results are not listed here.

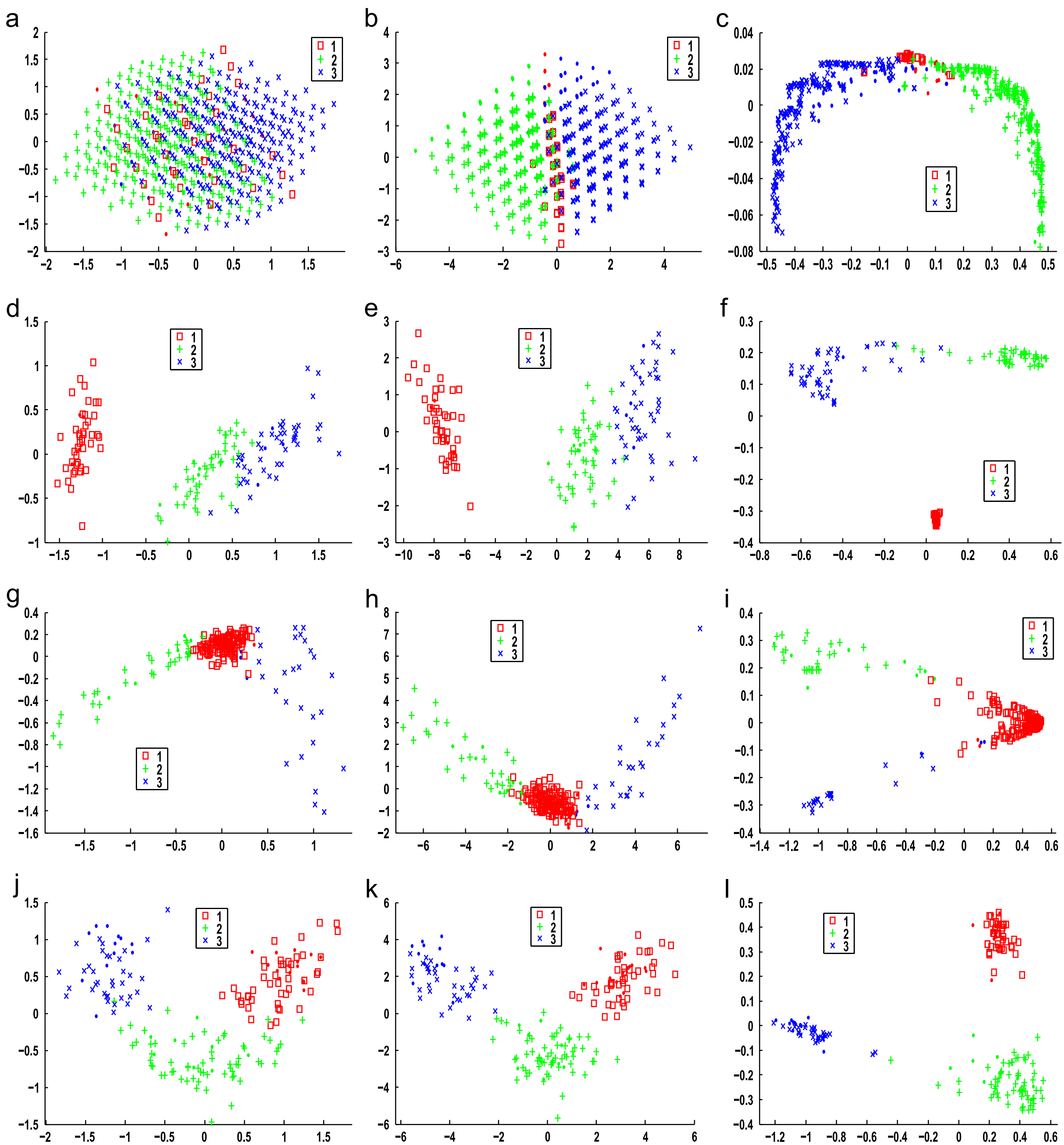

Fig. 1. 2D embedding learned by ECOC-NLFE, PCA and LDA on four UCI data sets: Balance, Iris, Thyroid and Wine (from top to bottom). The training data of different classes are plotted with different symbols and colors, while the test data are shown as points in the same colors with the corresponding class of training data. (a) PCA (b) LDA (c) ECOC-NLFE (d) PCA (e) LDA (f) ECOC-NLFE (g) PCA (h) LDA (i) ECOC-NLFE (j) PCA (k) LDA (l) ECOC-NLFE.

From the results shown in Tables 3–5, we know that our method, either ECOC-NLFE or ECOC-LFE, is promising for the multi-class classification problems. Compared with the state-of-the-art ECOC methods, it performs significantly better than or at least comparable with them. More importantly, the decoding speed of our method is much faster than that of the other competitive ECOC methods.

From Tables 2 and 3, we find that the decoding results of one-versus-all strategy on the Letter and the Vowel data set are evidently worse than the other compared methods. This is mainly because the linear loss-weighted (LLW) decoding strategy was developed for the ternary coding design methods. Actually, for the one-versus-all coding strategy, straightforward classification without decoding (i.e., classify to the class of maximum dichotomizer output) performs very well. This “no decoding” classification rule should be compared with the LLW decoding, for clarifying the performance of the one-versus-all coding design.

Table 5 shows the classification results of one-versus-all with both linear and nonlinear dichotomizers. We conduct the Wilcoxon signed-ranks test [51] with confidence value 0.05 on the results shown in the second column and the third column of Table 5. The test results show that the performance difference

Table 2
Classification accuracy obtained by ECOC-NLFE and the compared ECOC methods. The best results are highlighted in boldface (the same as below).

Data sets	OnevsOne	OnevsAll	DECOC	ECOCONE	ECOC-NLFE
Balance	0.8855	0.8912	0.8855	0.8855	**0.8931**
Dermathology	0.9731	0.9702	**0.9768**	0.9702	0.9711
Ecoli	0.8647	0.8186	**0.8667**	0.8578	0.8127
Glass	**0.6773**	0.5347	0.6496	0.6280	0.6108
Iris	0.9600	0.9600	0.9533	0.9600	**0.9733**
Letter	0.9377	0.7991	0.8550	0.8585	**0.9483**
OptDigits	**0.9872**	0.9769	0.9849	0.9853	0.9869
Pendigits	**0.9946**	0.9916	0.9926	0.9923	0.9941
Satimage	**0.8874**	0.8714	0.8827	0.8763	0.8837
Segmentation	0.9506	0.9372	0.9281	0.9446	**0.9550**
Shuttle	0.9964	0.9964	0.9968	0.9963	**0.9969**
Thyroid	0.9579	0.9579	0.9626	0.9531	**0.9742**
Vehicle	0.7519	0.7491	0.7534	0.7232	**0.7735**
Vowel	0.7495	0.4414	0.6283	0.6404	**0.7606**
Wine	0.9813	**0.9875**	**0.9875**	**0.9875**	0.9813
Yeast	**0.5956**	0.5470	0.5943	0.5831	0.5545
Mean rank	2.2813	4.0938	2.8750	3.5938	**2.1563**

Table 3
Classification accuracy obtained by ECOC-LFE and the compared ECOC methods.

Data sets	OnevsOne	OnevsAll	DECOC	ECOCONE	ECOC-LFE
Balance	0.8670	**0.9020**	0.8770	0.8703	0.8523
Dermathology	0.9750	0.9673	**0.9779**	0.9588	0.9691
Ecoli	**0.8412**	0.7647	0.8235	0.8186	0.8049
Glass	0.5605	0.4849	0.4921	0.4735	**0.5677**
Iris	**0.9733**	**0.9733**	**0.9733**	**0.9733**	**0.9733**
Letter	0.8491	0.3953	0.4509	0.4684	**0.9204**
OptDigits	**0.9748**	0.9414	0.8487	0.8549	0.9448
Pendigits	**0.9816**	0.8971	0.8013	0.8101	0.9626
Satimage	0.8553	0.7800	0.8083	0.7811	**0.8569**
Segmentation	0.9377	0.9121	0.8091	0.8545	**0.9407**
Shuttle	0.9698	0.9695	0.9189	0.9444	**0.9838**
Thyroid	0.9454	0.9454	0.9264	0.9073	**0.9742**
Vehicle	0.7779	0.7488	0.7711	0.7552	**0.7836**
Vowel	0.5990	0.2768	0.3697	0.3929	**0.6242**
Wine	0.9688	0.9658	**0.9721**	**0.9721**	0.9596
Yeast	**0.5818**	0.4778	0.5239	0.5578	0.5323
Mean rank	**2.0313**	3.7813	3.4063	3.5938	2.1875

Table 4
Comparison of decoding time among OnevsOne, OnevsAll, DECOC, ECOCONE, and ECOC-NLFE.

Data sets	OnevsOne	OnevsAll	DECOC	ECOCONE	ECOC-NLFE
Balance	0.1200	0.1877	0.1149	0.3853	**0.0069**
Dermathology	0.2561	0.2514	0.1168	0.1386	**0.0045**
Ecoli	0.2071	0.2490	0.1007	0.1390	**0.0042**
Glass	0.0954	0.1198	0.0588	0.0923	**0.0032**
Iris	0.0282	0.0376	0.0215	0.0510	**0.0010**
Letter	70.5732	182.5049	78.9832	80.6523	**15.9956**
OptDigits	8.6260	19.0988	8.8707	10.2497	**0.6246**
Pendigits	10.5128	19.5035	6.8941	9.8230	**1.8789**
Satimage	6.1888	16.3594	7.1908	18.4864	**0.8366**
Segmentation	1.4128	2.4341	1.0462	1.3222	**0.0671**
Shuttle	11.3091	14.3191	5.5555	12.3292	**4.9062**
Thyroid	0.0363	0.0547	0.0297	0.1036	**0.0014**
Vehicle	0.3529	0.5869	0.3169	1.0180	**0.0280**
Vowel	0.8855	1.3607	0.4672	0.5278	**0.0211**
Wine	0.0313	0.0470	0.0262	0.1058	**0.0015**
Yeast	1.3556	1.9763	0.7689	1.0972	**0.0541**

between "no decoding" and LLW is not significant. However, we can see that the classification results of "no decoding" is much better than that of LLW on the Letter and the Vowel data set. The statistical tests on the results in the fourth column and the fifth column of Table 5 (nonlinear dichotomizers) show that the performance of "no decoding" is significantly better than that of LLW. Further statistical tests between the results of "no decoding" with nonlinear dichotomizers and those of ECOC-NLFE show that their performance difference is not significant though the average classification accuracy of ECOC-NLFE is higher. This is consistent with Rifkin and Klautau's assertion [52] that the one-versus-all strategy combining well-tuned SVMs as base classifiers can achieve similar classification results with other sophisticated methods. The proposed ECOC-EFE yields higher accuracies than the one-versus-all coding with "no decoding" on most of the data sets, however.

Table 5
Comparison of decoding methods for the one-versus-all strategy with linear and nonlinear dichotomizers.

Data sets	OnevsAll linear		OnevsAll nonlinear	
	No decoding	LLW	No decoding	LLW
Balance	0.8570	**0.9020**	0.8883	**0.8912**
Derma	**0.9693**	0.9673	**0.9750**	0.9702
Ecoli	**0.8225**	0.7647	**0.8647**	0.8186
Glass	**0.5476**	0.4849	**0.6472**	0.5347
Iris	0.9000	**0.9733**	**0.9600**	**0.9600**
Letter	**0.6016**	0.3953	**0.9162**	0.7991
OptDigits	**0.9612**	0.9414	**0.9872**	0.9769
Pendigits	**0.9306**	0.8971	**0.9942**	0.9916
Satimage	**0.8151**	0.7800	**0.8832**	0.8714
Segmentation	**0.9169**	0.9121	**0.9481**	0.9372
Shuttle	0.9063	**0.9695**	**0.9968**	0.9964
Thyroid	0.9026	**0.9454**	**0.9579**	**0.9579**
Vehicle	0.7351	**0.7488**	0.7296	**0.7491**
Vowel	**0.4121**	0.2768	**0.6828**	0.4414
Wine	**0.9750**	0.9658	**0.9875**	**0.9875**
Yeast	**0.5299**	0.4778	**0.5986**	0.5470

Since the experimental results of all the compared methods using either linear or nonlinear dichotomizers lead to similar conclusion, we only show the results using nonlinear dichotomizers in the following parts.

5.6. Comparison with classic feature extraction methods

In this experiment, we compare ECOC-NLFE with some classic feature extraction methods, including PCA, KPCA, LDA and GDA. For fair comparison, the dimensionality of the new feature space is set as $\min\{C-1, D-1\}$, where C is the number of classes and D is the original dimensionality of the data. The classification results are shown in Table 6. We conduct the Friedman and the Nemenyi test [51] with confidence value 0.05 on the results. The statistical tests show that ECOC-NLFE performs significantly better than PCA, KPCA and LDA. Note that the dimensionality of learned features by LDA and GDA is at most $C-1$(if $C \leq D$). However, ECOC-NLFE does not have this limitation since its dimensionality equals the number of dichotomizers, and has the potential of extracting more features and yielding higher classification performance.

5.7. Effect of the meta learner

In this experiment, we compare the classification performance of ECOC-NLFE using different types of meta learner. We first consider different prototype classifiers for meta learning: one prototype per class learned by GLVQ, nearest class mean, multiple prototypes learned by k-means clustering. For k-means clustering and the proposed ECOC-NLFE, the number of prototypes per class is $\lceil (N_{trn}/(100 \times C)) \rceil \times 3$. The classification results are shown in Table 7, where "1-prototype" denotes one prototype per class learned by GLVQ. We conduct the Wilcoxon signed-ranks test

Table 6
Comparison with classic feature extraction methods on the UCI data sets.

Data sets	PCA	KPCA	LDA	GDA	ECOC-NLFE
Balance	0.4700	0.4531	0.8236	0.7707	**0.8931**
Dermathology	0.9188	0.9188	**0.9779**	0.9636	0.9711
Ecoli	0.8284	0.8294	**0.8588**	0.8304	0.8127
Glass	0.6012	0.5716	0.5716	0.6084	**0.6108**
Iris	0.9600	0.9467	**0.9800**	**0.9800**	0.9733
Letter	0.9038	0.9043	0.9349	0.9392	**0.9483**
OptDigits	0.9615	0.9584	0.9143	**0.9893**	0.9869
Pendigits	0.9753	0.9743	0.9654	0.9904	**0.9941**
Satimage	0.8636	0.8685	0.8589	0.8788	**0.8837**
Segmentation	0.9208	0.9203	0.6398	0.9511	**0.9550**
Shuttle	0.9866	0.9888	0.9862	0.9930	**0.9969**
Thyroid	0.9656	0.9626	0.9522	0.9570	**0.9742**
Vehicle	0.5040	0.4944	0.7827	**0.8340**	0.7735
Vowel	0.5717	0.5727	0.5434	0.7343	**0.7606**
Wine	0.9750	0.9750	**0.9875**	0.9658	0.9813
Yeast	0.4999	0.5244	0.5033	0.5405	**0.5545**
Mean rank	3.8125	3.7813	3.3750	2.3438	**1.6875**

Table 7
Classification results of ECOC-NLFE using different meta learners.

Data sets	1-prototype	Class mean	ECOC-NLFE	k-means
Balance	**0.8820**	0.8070	**0.8931**	0.8617
Dermathology	**0.9711**	**0.9711**	0.9711	**0.9750**
Ecoli	**0.8529**	0.8353	**0.8127**	0.7892
Glass	**0.6510**	0.5831	**0.6108**	0.4984
Iris	**0.9533**	**0.9533**	**0.9733**	0.9600
Letter	**0.8190**	0.8097	**0.9483**	0.9448
OptDigits	**0.9816**	0.9801	0.9869	**0.9886**
Pendigits	**0.9872**	0.9871	**0.9941**	0.9934
Satimage	**0.8773**	0.8757	**0.8837**	0.8612
Segmentation	0.9442	**0.9472**	**0.9550**	0.9442
Shuttle	**0.9880**	0.9757	**0.9969**	0.9914
Thyroid	0.9665	**0.9742**	**0.9742**	0.9608
Vehicle	**0.7537**	0.7400	**0.7735**	0.7482
Vowel	0.7212	**0.7313**	**0.7606**	0.7424
Wine	**0.9875**	**0.9875**	**0.9813**	**0.9813**
Yeast	**0.5987**	0.5664	**0.5545**	0.4497

[51] with confidence value 0.05 to compare the results in the second column (one prototype learned by GLVQ) and the third column (one prototype as class mean) of Table 7 and compare the results in the fourth column (multiple prototypes learned by GLVQ) and the fifth column (multiple prototypes by clustering). The statistical tests show that when using learned prototypes, GLVQ performs significantly better than the class mean (one prototype) or *k*-means clustering (multiple prototypes). This indicates that prototype learning by GLVQ is a right choice for meta learning in ECOC-NLFE.

However, the Wilcoxon signed-ranks test [51] with confidence value 0.05 between the second column and the fourth column of Table 7 shows that the GLVQ meta learner with multiple prototypes per class and that with only one prototype per class perform comparably on these multi-class learning problems. This is because for most of the data sets, the data of each class are in single mode. Consequently, GLVQ with only one prototype per class performs fairly well for these problems. However, if the data of each class distribute in multiple modes, GLVQ with only one prototype per class may fail for the multi-class classification problems. To show the performance difference between GLVQ with multiple prototype per class and that with only one prototype per class, we carry out experiments on Letter, Pendigits and Vowel, where the classes are randomly grouped to form a three-class classification problem. Fig. 2 shows the obtained classification results. We conduct the Wilcoxon signed-ranks test [51] with

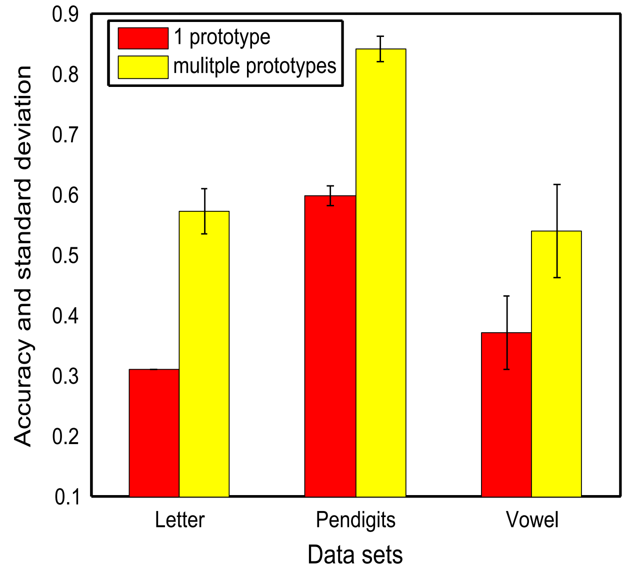

Fig. 2. Comparison of GLVQ meta learner with multiple prototypes per class and that with only one prototype per class. Here, each data set was re-grouped into three classes. For the letter data set, the standard deviation obtained by the GLVQ meta learner with one prototype per class is 0.0003, which is not obvious to see.

confidence value 0.05 on the classification accuracies obtained by 10-fold cross-validation on each data set. The statistical tests show that the GLVQ meta learner with multiple prototypes per class performs significantly better than that with only one prototype per class on all these three data sets. This indicates the necessity of using the GLVQ meta learning with multiple prototypes for each class.

Besides the GLVQ, many classifiers can also be applied as meta learner, such as the one nearest neighbor (1-NN) classifier and SVMs combined with one versus one strategy or one versus all strategy. To evaluate the performance of different meta learners, we compare ECOC-NLFE with GLVQ as meta learner against that with one nearest neighbor meta learner (1-NN meta), that with combined SVMs based on the one-versus-all strategy (with no decoding as introduced in Section 5.5, SVM meta1vA), that with combined SVMs based on the one-versus-one strategy (using majority voting for the decoding, SVM meta1v1), 1-NN classifier in the original space (1-NN original), and GLVQ in the original space (GLVQ original). Both the SVMs based meta learners are implemented using linear kernels, which perform sufficiently well in the new feature space spanned by the dichotomizers outputs. The classification results are shown in Table 8. We conduct the Wilcoxon signed-ranks test [51] with confidence value 0.05 to compare the results in the second column (1-NN meta) and third column (1-NN original), the results in the fourth column (GLVQ original) and the seventh column (ECOC-NLFE), the results in the second column (1-NN meta) and the seventh column (ECOC-NLFE), the results in the fifth column (SVM meta1vA) and the seventh column (ECOC-NLFE), and the results in the sixth column (SVM meta1v1) and the seventh column (ECOC-NLFE), respectively. The statistical tests show that ECOC-NLFE with 1-NN meta learner performs significantly better than the 1-NN classifier in the original space, and meanwhile, ECOC-NLFE with GLVQ meta learner performs significantly better than GLVQ in the original space. The performance of ECOC-NLFE with 1-NN meta learner and that with GLVQ meta learner are comparable, but the GLVQ meta learner is far more efficient since it stores only a few prototypes per class, while the 1-NN classifier needs to store all the training samples as prototypes. Moreover, statistical tests show that the performance of ECOC-NLFE with combination of

Table 8
Comparison of ECOC-NLFE with 1-NN meta learner, SVM meta learner and the ones in the original space.

Data sets	1-NN meta	1-NN original	GLVQ original	SVM meta1vA	SVM meta1v1	ECOC-NLFE
Balance	0.8988	0.6549	0.7517	**0.9131**	0.8976	0.8931
Dermathology	0.9702	0.9606	0.9739	**0.9750**	**0.9750**	0.9711
Ecoli	0.8186	0.8069	0.8529	**0.8627**	0.8608	0.8127
Glass	0.5821	0.6280	0.6031	0.6333	**0.6759**	0.6108
Iris	0.9533	0.9533	0.9600	0.9533	0.9600	**0.9733**
Letter	**0.9708**	0.9601	0.9039	0.9144	0.9421	0.9483
OptDigits	0.9870	0.9840	0.9779	0.9864	**0.9878**	0.9869
Pendigits	**0.9946**	0.9933	0.9789	0.9939	0.9945	0.9941
Satimage	0.8697	0.8777	0.8785	0.8815	**0.8870**	0.8837
Segmentation	**0.9654**	0.9719	0.9286	0.9481	0.9554	0.9550
Shuttle	0.9984	**0.9986**	0.9865	0.9970	0.9973	0.9969
Thyroid	0.9647	0.9599	0.9445	0.9674	0.9665	**0.9742**
Vehicle	0.7476	0.6901	0.6926	0.7635	0.7656	**0.7735**
Vowel	**0.7646**	0.7141	0.5727	0.7071	0.7424	0.7606
Wine	0.9688	0.9408	0.9563	0.9750	**0.9813**	**0.9813**
Yeast	0.5059	0.4881	0.5208	0.5974	**0.6001**	0.5545

Table 9
Classification accuracy obtained by variants of stacking-FE and ECOC-NLFE on the UCI data sets.

Data sets	S-OnevsOne	S-OnevsAll	S-DECOC	S-ECOCONE	ECOC-NLFE
Balance	**0.9178**	0.9165	0.9009	0.8883	0.8931
Derma	0.9750	**0.9807**	0.9682	0.9779	0.9711
Ecoli	**0.8480**	0.8108	0.8333	0.8206	0.8127
Glass	0.5663	0.4572	0.4490	0.5079	**0.6108**
Iris	0.9533	0.9600	0.9667	0.9467	**0.9733**
Letter	0.9423	**0.9485**	0.9469	0.9466	0.9483
OptDigits	0.9847	**0.9881**	0.9855	0.9867	0.9860
Pendigits	0.9932	**0.9948**	0.9932	0.9933	0.9941
Satimage	0.8774	0.8783	0.8712	0.8767	**0.8837**
Segmentation	**0.9571**	0.9558	0.9476	0.9541	0.9550
Shuttle	0.8951	0.8652	0.8920	0.8923	**0.9969**
Thyroid	0.9733	0.9685	0.9685	0.9647	**0.9742**
Vehicle	0.7565	0.7503	0.7311	0.7332	**0.7735**
Vowel	**0.7626**	0.7313	0.7313	0.7162	0.7606
Wine	**0.9813**	0.9750	0.9750	0.9750	**0.9813**
Yeast	**0.5600**	0.5238	0.5410	0.5277	0.5545
Mean rank	2.4375	2.8750	3.8438	3.7500	**2.0938**

SVMs based meta learners and that with GLVQ meta learner are comparable. In contrast to combination of SVMs based meta learners, the GLVQ meta learner has better interpretation because the learned prototypes can be seen as codewords. However, it is hard to consider the support vectors learned by the base classifiers, SVMs, as codewords. All these results indicate the effectiveness of GLVQ as the meta learner of ECOC-NLFE.

5.8. Comparison with stacking based feature extraction

Our method trains the meta learner by 'reusing' the transformed data from the training samples for training the dichotomizers. This raises a question of possible overfitting. On the other hand, the stacking strategy, that generates training data for meta learning by cross-validation, has been demonstrated effective in combining multiple classifiers, including multi-class classification by combining dichotomizers. We hence compare the performance of ECOC-NLFE with stacking based feature extraction (Stacking-FE).

For Stacking-FE, we use 10-fold cross-validation on the training data to extract new features for meta learning. To guarantee the same length of codewords for different partitions of data by ECOCONE, we learn the ECOC matrix using all the training data before the cross-validation. Except the difference of meta learner training data generation (reuse versus stacking), the settings of dichotomizers and meta learner (GLVQ) are the same for ECOC-NLFE and Stacking-FE. We implement Stacking-FE with four ECOC coding strategies: one-versus-one (S-OnevsOne), one-versus-all (S-OnevsAll), DECOC (S-DECOC) and ECOCONE (S-ECOCONE). The classification results are shown in Table 9.

We conduct the Friedman and the Nemenyi test [51] with confidence value 0.05 on the results in Table 9. The statistical test shows that ECOC-NLFE performs significantly better than S-DECOC and S-ECOCONE, and meanwhile, it is at least comparable with S-OnevsOne and S-OnevsAll. This indicates that there is no evident overfitting caused by ECOC-NLFE (reusing).

The inferior performance of Stacking-FE compared to ECOC-NLFE can be explained as follows. First, although ECOCONE can learn the ECOC matrix by extending an initial configuration, it cannot be used in the cross-validation procedure of Stacking-FE, since the learned codewords may have different lengths and render the corresponding partitions of data having different dimensionalities. To overcome this problem, we learn the ECOC matrix of Stacking-FE on all the training data. This loses the optimality of cross-validation for Stacking-FE. In contrast, ECOC-NLFE is more flexible, which learns the new features of the data directly from the ECOC matrix and the trained dichotomizers. Second, when the sizes of the data in different classes are dramatically unbalanced, such as the Shuttle data set, Stacking-FE may fail to learn effective new features of the data. Generally speaking, this problem can be overcome by carefully tuning the dichotomizers, but it is a time consuming. On the contrary, the ECOC-NLFE turns out to be less sensitive to data imbalance, as it generates the meta learner training data directly from all the training samples.

6. Conclusion

In this paper, we propose an ECOC based ensemble feature extraction (ECOC-EFE) method to take advantage of the coding matrix learning ability and discrimination ability of the dichotomizers. Both linear features and nonlinear features can be extracted easily. Using the probabilistic outputs of the dichotomizers as new features of the data, we use a meta classifier to perform multi-class classification. Specifically, we use the generalized learning vector quantization (GLVQ) for meta learning. The learned prototype of a class can be viewed as a codeword of that class. Extensive experiments on 16 data sets from the UCI machine learning repository demonstrated the effectiveness, efficiency, robustness and flexibility of our method. Particularly, the ECOC-EFE performs superiorly or comparably with the state-of-the-art ECOC methods and feature extraction methods. In the future, we would like to exploit the fast ECOC-EFE algorithm and its application to large-scale problems using new models of dichotomizers [53].

Acknowledgments

This work is supported by the National Natural Science Foundation of China (NSFC) Grant 60825301 and the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant XDA06030300). The authors would like to thank Sergio Escalera for the private communication and suggestions to this work.

References

[1] I. Jolliffe, Principal Component Analysis, Springer-Verlag, New York, 1986.
[2] R.A. Fisher, The use of multiple measurements in taxonomic problems, Annals of Eugenics 7 (1936) 179–188.

[3] J.B. Tenenbaum, V.d. Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323.
[4] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.
[5] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Computation 15 (6) (2003) 1373–1396.
[6] Z. Zhang, H. Zha, Principal manifolds and nonlinear dimensionality reduction via tangent space alignment, SIAM Journal on Scientific Computing 26 (1) (2004) 313–338.
[7] K.Q. Weinberger, L.K. Saul, Unsupervised learning of image manifolds by semidefinite programming, International Journal of Computer Vision 70 (1) (2006) 77–90.
[8] T. Lin, H. Zha, Riemannian manifold learning, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (5) (2008) 796–809.
[9] G. Zhong, W.-J. Li, D.-Y. Yeung, X. Hou, C.-L. Liu, Gaussian process latent random field, in: AAAI, 2010, pp. 679–684.
[10] Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N.L. Roux, M. Ouimet, Out-of-sample extensions for LLE, Isomap, MDS, eigenmaps, and spectral clustering, in: NIPS, 2003.
[11] T.G. Dietterich, Ensemble methods in machine learning, in: Multiple Classifier Systems, 2000, pp. 1–15.
[12] L. Breiman, Bagging Predictors, Machine Learning 24 (2) (1996) 123–140.
[13] R.E. Schapire, The strength of weak learnability, Machine Learning 5 (1990) 197–227.
[14] T.G. Dietterich, G. Bakiri, Solving multiclass learning problems via error-correcting output codes, Journal of Artificial Intelligence Research (JAIR) 2 (1995) 263–286.
[15] D.H. Wolpert, Stacked generalization, Neural Networks 5 (1992) 214–259.
[16] L. Lam, C.Y. Suen, Optimal combinations of pattern classifiers, Pattern Recognition Letters 16 (9) (1995) 945–954.
[17] Y. Freund, R.E. Schapire, Experiments with a new boosting algorithm, in: ICML, 1996, pp. 148–156.
[18] T. Windeatt, G. Ardeshir, Boosted ECOC ensembles for face recognition, in: ICVIE, 2003, pp. 165–168.
[19] H. Altincay, M. Demirekler, An information theoretic framework for weight estimation in the combination of probabilistic classifiers for speaker identification, Speech Communication 30 (2000) 255–272.
[20] G.J. Briem, J.A. Benediktsson, J.R. Sveinsson, Multiple classifiers applied to multisource remote sensing data, IEEE Transactions on Geoscience and Remote Sensing 40 (10) (2002) 2291–2299.
[21] L. Wu, S.L. Oviatt, P.R. Cohen, From members to teams to committee robust approach to gestural and multimodal recognition, IEEE Transactions on Neural Networks 13 (4) (2002) 972–982.
[22] N. Nilsson, Learning Machines, McGraw-Hill, 1965.
[23] T. Hastie, R. Tibshirani, Classification by pairwise coupling, Annals of Statistics 26 (2) (1998) 451–471.
[24] J. Zhou, H. Peng, C.Y. Suen, Data-driven decomposition for multi-class classification, Pattern Recognition 40 (1) (2008) 67–76.
[25] O. Pujol, P. Radeva, J. Vitrià, Discriminant ECOC: a heuristic method for application dependent design of error correcting output codes, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (6) (2006) 1007–1012.
[26] S. Escalera, O. Pujol, ECOC-ONE: a novel coding and decoding strategy, in: ICPR, 2006, pp. 578–581.
[27] E.L. Allwein, R.E. Schapire, Y. Singer, Reducing multiclass to binary: a unifying approach for margin classifiers, Journal of Machine Learning Research 1 (2000) 113–141.
[28] A. Passerini, M. Pontil, P. Frasconi, New results on error correcting output codes of kernel machines, IEEE Transactions on Neural Networks (2004) 45–54.
[29] O. Dekel, Y. Singer, Multiclass learning by probabilistic embeddings, in: NIPS, 2002, pp. 945–952.
[30] S. Escalera, O. Pujol, P. Radeva, On the decoding process in ternary error-correcting output codes, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (1) (2010) 120–134.
[31] A. Gersho, R.M. Gray, Vector Quantization and Signal Compression, Kluwer Academic Publishers, Norwell, MA, 1991.
[32] A. Sato, K. Yamada, Generalized learning vector quantization, in: NIPS, 1995, pp. 423–429.
[33] B. Schölkopf, A.J. Smola, K.-R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Computation 10 (5) (1998) 1299–1319.
[34] G. Baudat, F. Anouar, Generalized discriminant analysis using a kernel approach, Neural Computation 12 (10) (2000) 2385–2404.
[35] V. Vapnik, The Nature of Statistical Learning Theory, Springer, 1995.
[36] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences 55 (1) (1997) 119–139.
[37] W. Utschick, W. Weichselberger, Stochastic organization of output codes in multiclass learning problems, Neural Computation 13 (5) (2001) 1065–1102.
[38] K. Crammer, Y. Singer, On the learnability and design of output codes for multiclass problems, Machine Learning 47 (2–3) (2002) 201–233.
[39] S. Escalera, O. Pujol, P. Radeva, Re-coding ECOCs without re-training, Pattern Recognition Letters 31 (7) (2010) 555–562.
[40] P. Savický, J. Fürnkranz, Combining pairwise classifiers with stacking, in: IDA, 2003, pp. 219–229.
[41] O. Lezoray, H. Cardot, Comparing combination rules of pairwise neural networks classifiers, Neural Processing Letters 27 (1) (2008) 43–56.
[42] Y. Shiraishi, K. Fukumizu, Statistical approaches to combining binary classifiers for multi-class classification, Neurocomputing 74 (5) (2011) 680–688.
[43] L. Rueda, B.J. Oommen, C. Henríquez, Multi-class pairwise linear dimensionality reduction using heteroscedastic schemes, Pattern Recognition 43 (7) (2010) 2456–2465.
[44] J.C. Platt, Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, in: Advances in Large Margin Classifiers, MIT Press, 1999, pp. 61–74.
[45] H.-T. Lin, C.-J. Lin, R.C. Weng, A note on platt's probabilistic outputs for support vector machines, Machine Learning 68 (3) (2007) 267–276.
[46] C.-L. Liu, Classifier combination based on confidence transformation, Pattern Recognition 38 (1) (2005) 11–28.
[47] S. Escalera, O. Pujol, P. Radeva, Error-correcting output codes library, Journal of Machine Learning Research 11 (2010) 661–664.
[48] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics 1 (1945) 80–83.
[49] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, Journal of the American Statistical Association 32 (1937) 675–701.
[50] P.B. Nemenyi, Distribution-Free Multiple Comparisons, Ph.D. Thesis, Princeton University, 1963.
[51] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.
[52] R.M. Rifkin, A. Klautau, In defense of one-vs-all classification, Journal of Machine Learning Research 5 (2004) 101–141.
[53] S. Shalev-Shwartz, Y. Singer, N. Srebro, Pegasos: primal estimated sub-gradient solver for SVM, in: ICML, 2007, pp. 807–814.

Guoqiang Zhong received the BS degree in Mathematics and Applied Mathematics from Hebei Normal University, Shijiazhuang, China, in 2004 and the MS degree in Operational Research and Cybernetics from Beijing University of Technology, Beijing, China, in 2007. He got his PhD degree in Pattern Recognition and Intelligent System from the Institute of Automation of Chinese Academy of Sciences, Beijing, China, in 2011. Currently, he is a postdoctoral fellow at the École de technologie supérieure, University of Quebec, Montreal, Canada. His research interests include pattern recognition and machine learning.

Cheng-Lin Liu is a Professor at the National Laboratory of Pattern Recognition (NLPR), Institute of Automation of Chinese Academy of Sciences, Beijing, China, and is now the deputy director of the laboratory. He received the B.S. degree in electronic engineering from Wuhan University, Wuhan, China, the M.E. degree in electronic engineering from Beijing Polytechnic University, Beijing, China, the Ph.D. degree in pattern recognition and intelligent control from the Chinese Academy of Sciences, Beijing, China, in 1989, 1992 and 1995, respectively. He was a postdoctoral fellow at Korea Advanced Institute of Science and Technology (KAIST) and later at Tokyo University of Agriculture and Technology from March 1996 to March 1999. From 1999 to 2004, he was a research staff member and later a senior researcher at the Central Research Laboratory, Hitachi, Ltd., Tokyo, Japan. His research interests include pattern recognition, image processing, neural networks, machine learning, and especially the applications to character recognition and document analysis. He has published over 140 technical papers at prestigious international journals and conferences, and co-authored a book on character recognition.