Heuristic Ternary Error-Correcting Output Codes Via Weight Optimization and Layered Clustering-Based Approach

Xiao-Lei Zhang, *Member, IEEE*

***Abstract*—One important classifier ensemble for multiclass classification problems is error-correcting output codes (ECOCs). It bridges multiclass problems and binary-class classifiers by decomposing multiclass problems to a serial binary-class problems. In this paper, we present a heuristic ternary code, named weight optimization and layered clustering-based ECOC (WOLC-ECOC). It starts with an arbitrary valid ECOC and iterates the following two steps until the training risk converges. The first step, named layered clustering-based ECOC (LC-ECOC), constructs multiple strong classifiers on the most confusing binary-class problem. The second step adds the new classifiers to ECOC by a novel optimized weighted (OW) decoding algorithm, where the optimization problem of the decoding is solved by the cutting plane algorithm. Technically, LC-ECOC makes the heuristic training process not blocked by some difficult binary-class problem. OW decoding guarantees the nonincrease of the training risk for ensuring a small code length. Results on 14 UCI datasets and a music genre classification problem demonstrate the effectiveness of WOLC-ECOC.**

***Index Terms*—Ensemble learning, error-correcting output code (ECOC), multiclass classification, multiple classifier system.**

I. Introduction

OVER the last decades, classifier ensembles [1]–[10], such as bagging [11], boosting [12] and their variations, have demonstrated their effectiveness on many learning problems [13]–[15]. Their success relies on a good selection of base learners and a strong diversity among the base learners, where the word "diversity" means that when the base learners make predictions on an identical pattern, they are different from each other in terms of errors. As summarized in [1]–[4] there are generally four groups of classifier ensembles: 1) manipulating training examples; 2) manipulating input features; 3) manipulating training parameters; and 4) manipulating output targets.

One method of manipulating output targets is error-correcting output codes (ECOCs) [16] which is motivated from information theory for correcting bits caused by noisy communication channels. The key idea of ECOC is summarized as follows. Given a multiclass problem, ECOC assigns each class a unique codeword. All codewords form an ECOC coding matrix, where each row of the coding matrix is a codeword and each column defines a bipartition of the classes. Training dichotomizers (i.e., binary-class classifiers) on different bipartitions of the classes gets an ECOC ensemble. ECOC has two merits: 1) it bridges multiclass problems and dichotomizers and 2) it may correct errors by proper codeword designs. ECOC consists of two parts—coding and decoding. Coding assigns each class a unique codeword. Decoding predicts a test pattern by matching the predicted codeword with its most similar codeword in the coding matrix.

The coding techniques can be categorized to two classes. The first class is problem-independent codings [16]–[18] which use coding matrices that have strong error-correcting abilities in the view of channel coding. The second class is problem-dependent codings [19]–[37] which aim to solve given multiclass problems without considering the error-correcting ability of coding matrices much. This class attracted much attention in recent years, such as discriminant ECOC (DECOC) [26] ECOC-optimizing node embedding (ECOC-ONE) [27], [38] subclass-ECOC [28] manipulations of features [39], [40] and manipulations of the parameters of base dichotomizers [34]. The decoding methods are various distance metrics, including hamming distance (HD), Euclidean distance (ED), probabilistic [41] loss-based (LB) [42] and loss-weighted (LW) [43] decodings.

In this paper, we propose a heuristic ternary ECOC, named weight optimization and layered clustering-based ECOC (WOLC-ECOC). As shown in Fig. 1, it begins with an arbitrary valid ECOC ensemble and iteratively adds new dichotomizers to the ensemble in a greedy training manner by the following two steps until the training risk converges, where the word "valid" means that each codeword is unique. The first step trains a dichotomizer that discriminates the most confusing pair of classes by a new layered clustering-based ECOC (LC-ECOC) approach. The second step adds the dichotomizer to ECOC by a new optimized weighted (OW) decoding algorithm. The left side of the dotted line of Fig. 1 summarizes the contributions of this paper, while the right side was proposed in [38] and [27].

1) A novel LC-ECOC coding method is proposed. The key idea of LC-ECOC is to construct multiple strong dichotomizers on a single pair of classes by first clustering the pair to small nonoverlapped regions multiple

Manuscript received June 22, 2012; revised February 18, 2013 and March 17, 2014; accepted May 5, 2014. Date of publication June 2, 2014; date of current version January 13, 2015. This paper was recommended by Associate Editor F. Karray.

The author is with the Tsinghua National Laboratory for Information Science and Technology, Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: huoshan6@126.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2014.2325603

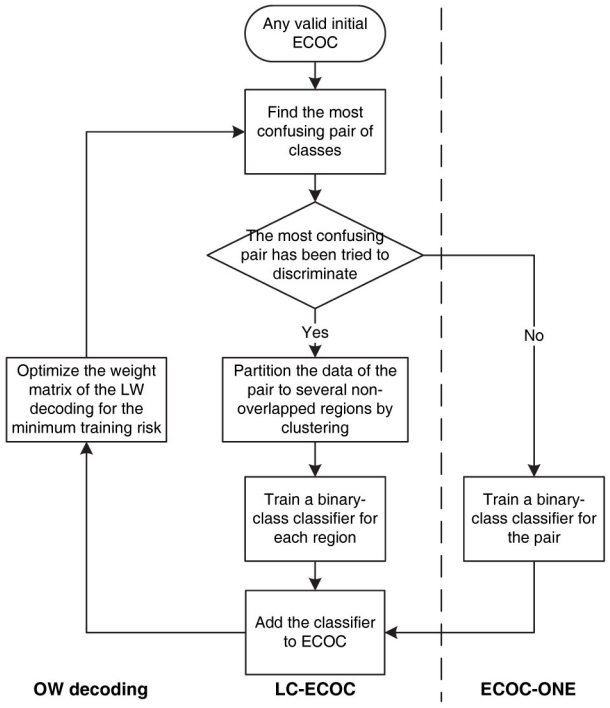

Fig. 1. System overview of WOLC-ECOC. The left side of the dotted line is our contribution. The right side of the dotted line is ECOC-ONE [27], [38].

times and then training a classifier for each region in each time of clustering, where all classifiers in each time of clustering group to a strong dichotomizer. It is motivated from the weakness of ECOC-ONE [27], [38] in which the heuristic training process might be blocked by some difficult binary-class problems; although subclass-ECOC [28] has shown its advantage on the most confusing problems by embedding a tree into each problem, it is difficult to control the growth of the tree.

2) A novel cutting-plane algorithm (CPA)-based OW decoding method is proposed. Like LW decoding [43] OW decoding is also a nonbiased decoding for ternary codes, but OW decoding improves LW decoding by optimizing the empirical weight matrix of the LW decoding for the minimum training risk. We solve the optimization problem via CPA [44]–[47]. The CPA-based OW decoding has linear time and storage complexities.
3) A novel WOLC-ECOC classifier system is proposed. As shown in Fig. 1, WOLC-ECOC iterates LC-ECOC (and also ECOC-ONE) and OW decoding until the training risk converges. The iteration integrates the merits of the aforementioned two items together: 1) LC-ECOC ensures that the greedy training will not be blocked by some difficult binary-class problems and 2) OW decoding guarantees the nonincrease of the training risk whenever adding a new dichotomizer to ECOC, so that the heuristic training can be easily controlled via the training risk, which makes a small code length available.

WOLC-ECOC inherits the advantages of ECOC-ONE [27] subclass-ECOC [28] and LW decoding [43] and meanwhile overcomes their drawbacks.

4) A brief literature survey of ECOC is conducted.

The experimental comparison with 15 coding–decoding methods on 14 UCI benchmark datasets with two kinds of base classifiers shows that WOLC-ECOC outperforms comparison methods when the discrete AdaBoost is used as the base classifier, outperforms 12 comparison methods when the Gaussian radial-basis-function (RBF) kernel-based SVM is used as the base classifier, and meanwhile maintains a small code length in both scenarios.

The rest of the paper is organized as follows. In Section II, we conduct a brief literature survey on ECOC. In Section III, we present the LC-ECOC coding method. In Section IV, we present the CPA-based OW decoding method. In Section V, we present WOLC-ECOC. In Section VI, we report the experiment results and further apply WOLC-ECOC to a real-world problem—music genre classification. Finally, we conclude this paper in Section VII.

We first introduce some notations here. Bold small letters, e.g., $\mathbf{w}$, indicate column vectors. Bold capital letters, e.g., $\mathbf{M}$ and $\mathbf{W}$, indicate matrices. Letters in calligraphic fonts, e.g., $\mathcal{W}$, indicate sets, where $\mathbb{R}^d$ denotes a d-dimensional real space. $\mathbf{0}$ ($\mathbf{1}$) is a column vector with all entries being 1 (0).

II. Brief Literature Survey

ECOC originally views "machine learning as a kind of communication problem in which the identity of the correct output class for a new example is being transmitted over a channel. The channel consists of the input features, the training examples, and the learning algorithm." [16]. Given a P class classification problem with a set of labeled examples $\{(\boldsymbol{\rho}_i, y_i)\}_{i=1}^n$ where $\boldsymbol{\rho}_i \in \mathbb{R}^d$ and $y_i \in \{1, 2, \ldots, P\}$ is the label of $\boldsymbol{\rho}_i$, ECOC aims to solve the problem by for example Q dichotomizers. The relation between the classes and the dichotomizers can be expressed by a binary coding matrix $\mathbf{M} \in \{-1, 1\}^{P\times Q}$ or a ternary coding matrix $\mathbf{M} \in \{-1, 0, 1\}^{P\times Q}$, where the p-th row of $\mathbf{M}$ expresses the codeword of class p, denoted as $\mathbf{c}_p$, and the q-th column expresses the q-th dichotomizers, denoted as h_q.

A. Survey on the Coding Phase

Two common output codes are the one-versus-all (1versusALL) and one-versus-one (1versus1) matrices [48]. Because they have no error-correcting ability, later on, channel codes with large HDs between the codewords were tried, which is known as problem-independent codings [16]. However, unlike channel codes in communication, the "channels" in ECOC are influenced by the bipartitions of classes: if the classes are partitioned improperly, the "noise" (errors) of the channels may be rather high. Furthermore, because there are only $2^{P-1}-1$ possible bipartitions in any binary codes, the code length is limited when P is small [34]. Finally, the error-correcting ability of ECOC is severely limited. Until now, to our knowledge, few evident proofs showed the error-correcting ability [49] and in most cases, 1versusALL and 1versus1 are still prevalent [50]. Although Tapia *et al.* [17], [18] declared improved performance with low-density parity-check codes and special bipartitions, we do not know how much the codes contribute to the improvement compared to the bipartitions.

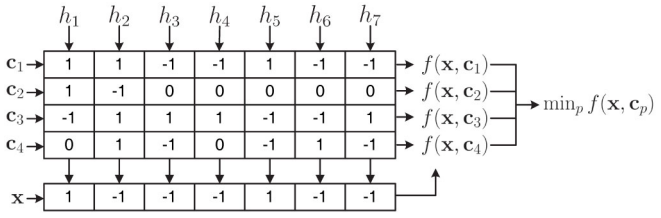

Fig. 2. Coding matrix $\mathbf{M}$ of a ternary ECOC [43]. In the coding phase, if the entry of $\mathbf{M}$, denoted as $m_{p,q}$, equals to 1, the dichotomizer h_q takes class p as part of the positive superclass. If $m_{p,q} = -1$, h_q takes class p as part of the negative superclass. If $m_{p,q} = 0$, h_q does not take class p into training [42]. In the decoding phase, taking a test example $\boldsymbol{\rho}$ into $h_1, \ldots, h_Q$ successively gets a test codeword of $\boldsymbol{\rho}$, denoted as $\mathbf{x} = [x_1, \ldots, x_Q]^T$. Given a decoding strategy $f(\mathbf{x}, \mathbf{c}_p)$, the prediction of $\boldsymbol{\rho}$ can be formulated as a minimization problem $\min_{\mathbf{c}_p \in \mathcal{M}} f(\mathbf{x}, \mathbf{c}_p)$, where $\mathcal{M} = \{\mathbf{c}_p\}_{p=1}^{Q}$ is the set of codewords.

Therefore, ECOC is more properly viewed as a bridge between powerful dichotomizers and multiclass problems without considering the error-correcting ability much, which results in the following three types of problem-dependent codings.

The first type learns ECOC in a single objective. Because finding an optimal binary coding matrix in a single objective is NP-complete, researchers relaxed the binary coding matrix to a continuous one and reformulated the problem to a regularized optimization problem. Typical methods include multiclass-SVM [51] and several large margin related works [19]–[24] However, it is worthy noting that multiclass-SVM does not perform better than 1versusALL and 1versus1, and even suffers from longer training time [48]. Motivated from multiclass-SVM [51], Zhong *et al.* [25] further took base dichotomizers into optimization. Because the objective is too complicated, it has to be solved approximately via the nonconvex constrained concave-convex procedure (CCCP) [52], [53]. Moreover, the continuous coding matrix has to be normalized after each CCCP iteration, making the convergence of the objective unguaranteed. Summarizing the aforementioned, it might be difficult and time consuming to learn a problem-dependent coding matrix in a single objective.

The second type uses ternary codes.

1) Allwein *et al.* [42] extended binary coding to ternary coding, i.e., $\mathbf{M} \in \{-1, 0, 1\}^{P \times Q}$, see Fig. 2 for an example. The entry $M(p, q) = 0$ indicates that the q-th dichotomizer does not take the p-th class into training. This method greatly enlarges the number of all possible bipartitions and makes each binary-class problem easily solved.
2) Pujol *et al.* [26] proposed DECOC which embeds a binary decision tree into the ternary code and takes the bipartition that maximizes the mutual information as a new node of the tree whenever adding a new node to the tree. Yang and Tsang [54] further proposed to find the most discriminative bipartition in terms of maximum separating margin. These methods need at most $P - 1$ dichotomizers.
3) To overcome the weakness of decision tree that the nodes of a tree cannot rectify misclassified examples made by their father nodes, Pujol *et al.* [27], [38] proposed ECOC-ONE which iteratively adds dichotomizers that discriminate the most confusing pairs.
4) To overcome the weakness of ECOC-ONE that the training process may be blocked by some stubborn binary problems, Escalera *et al.* [28] further proposed subclass-ECOC, which splits the most confusing class to several subsets (called subclasses) by a decision tree. Because it is also hard to decide when to stop splitting, Escalera *et al.* [28] used three hyperparameters to control the splitting process, and Bouzas *et al.* [29] tried to find the optimal hyperparameters by searching the hyperparameter spaces.

The third type focuses on improving the diversity between base dichotomizers.

1) The following methods improve the diversity by manipulating output codes. Kuncheva and Whitaker [30], [31] and Escalera *et al.* [32] designed new decoding metrics between codewords. Escalera *et al.* [33] suggested to selectively replace some 0 positions of an original ternary ECOC codes with 1 or -1 according to the accuracies of the base learners at the corresponding classes, which enlarges the distance between the codewords. Escalera *et al.* [35] combined multiple different DECOC trees. Hatami [55] tried to delete the columns of a coding matrix that have weak diversities.
2) Other types of diversity were seldom explored: Only Prior and Windeatt [34] manipulated different parameter settings of multilayer perceptrons. Bagheri *et al.* [39], [40] trained different base dichotomizers with different feature subsets. Our LC-ECOC—a method of manipulating training examples—was partially motivated from this fact.

There are also many other ECOC coding designs and applications, such as the evolution computing-based methods [36], [37] probability ECOC [56] structured outputs of ECOC [57] online ECOC [58], [59] and reject rule-based ECOC which rejects to use extremely confusing examples [60], [61].

B. Survey on the Decoding Phase

The representative decoding methods are HD, ED, probabilistic [41], LB [42], and LW decodings [43]. Here, we focus on reviewing LW decoding since it has a compact theory and performs better than other decoding methods in practice.

In [43], and in previous works [33] and [38], Escalera *et al.* argued that a good decoding strategy should make all codewords have the same decoding dynamic range and zero decoding dynamic range bias. Based on the argument, they proposed the LW decoding for ternary ECOCs, which is the first decoding strategy of ternary ECOCs that satisfies the aforementioned two goals. The LW decoding introduces a predefined weight matrix $\mathbf{W} = [\mathbf{w}_1^T, \ldots, \mathbf{w}_P^T]^T = \begin{bmatrix} w_{1,1} \ldots w_{1,Q} \\ \vdots \quad \ddots \quad \vdots \\ w_{P,1} \ldots w_{P,Q} \end{bmatrix} \in \mathcal{W}$ that has the same size as $\mathbf{M}$ and satisfies the following two constraints:

$$w_{p,q} \begin{cases} = 0 & , \text{ if } m_{p,q} = 0 \\ \in [0, 1] & , \text{ if } m_{p,q} \neq 0 \end{cases} \quad \forall p = 1, \ldots, P, \forall q = 1, \ldots, Q \tag{1}$$

$$\sum_{q=1}^{Q} w_{p,q} = 1, \quad \forall p = 1, \ldots, P \tag{2}$$

where $m_{p,q}$ is an element of $\mathbf{M}$ and $\mathcal{W}$ is the set of all feasible weight matrices (i.e., $\mathbf{W} \in \mathcal{W}$). When $m_{p,q} \neq 0$, $w_{p,q}$ is assigned empirically according to the training accuracy of the q-th base dichotomizer on the p-th class.

The prediction function of the LW decoding is given by

$$\min_{\mathbf{c}_p \in \mathcal{M}} f_{LW}(\mathbf{x}, \mathbf{c}_p) = \min_{\mathbf{c}_p \in \mathcal{M}} \sum_{q=1}^{Q} w_{p,q} \ell(x_q c_{p,q}) \tag{3}$$

where $\ell(\cdot)$ is a user defined loss function, such as the linear loss function $\ell(\theta) = -\theta$.

III. LC-ECOC

In this section, we first review the layered clustering-based approach for classifier ensembles [3] and then propose a new LC-ECOC.

A. Layered Clustering-Based Approach

The layered clustering-based approach [3] is an ensemble learning method that manipulates training examples for enlarging diversity. Specifically, it first splits training examples to several nonoverlapping regions by clustering, where the classification problem in each region is further solved by a classifier. The classifiers in all regions group to a super-classifier. Then, it repeats the above procedure several times. Each independent repeat forms a layer of super-classifier. All layers of super-classifiers vote for a test example.

This method contains two complementary properties. First, the clustering-based approach can identify overlapping patterns that are hard to differentiate, so that the classifier in each layer may achieve a high accuracy. But the clustering-based approach do not include any mechanism to incorporate diversity. Second, the layered approach uses the mechanism of bagging to achieve diversities between the super-classifiers. This layered structure, as proved in [1, p. 2] (an article appeared before [3]) will improve the discriminability of a classifier ensemble on a given binary-class problem.

B. LC-ECOC

Motivated by ECOC-ONE [27] and subclass-ECOC [28] the proposed LC-ECOC also uses the greedy training strategy, a strategy that iteratively adds new dichotomizers that intend to solve the most difficult binary-class problems of previous iterations. The difference between them lies on how they deal with the "stubborn" binary-class problems, where "stubborn" means that a binary-class problem has been tried to solve by a dichotomizer, but it appears to be the most difficult problem again. When such a situation happens, ECOC-ONE has to stop training, subclass-ECOC employs a decision-tree to further split the problem, and our LC-ECOC trains one layer of clustering-based dichotomizer [3] on the problem. Because different layers of clustering-based dichotomizers are different in terms of errors, LC-ECOC will not be blocked by the stubborn problems.

	$h_1^{(s)}$	$h_2^{(s)}$	$h_3^{(s)}$	$h_4^{(c)}$	$h_5^{(c)}$
$\mathbf{c}_1$	1	0	1	1	0
$\mathbf{c}_2$	-1	1	-1	-1	1
$\mathbf{c}_3$	-1	-1	0	0	-1

Initial ECOC

Fig. 3. Example of LC-ECOC for a three-class classification problem. $h^{(s)}$ indicates a simple dichotomizer. $h^{(c)}$ indicates a clustering-based dichotomizer.

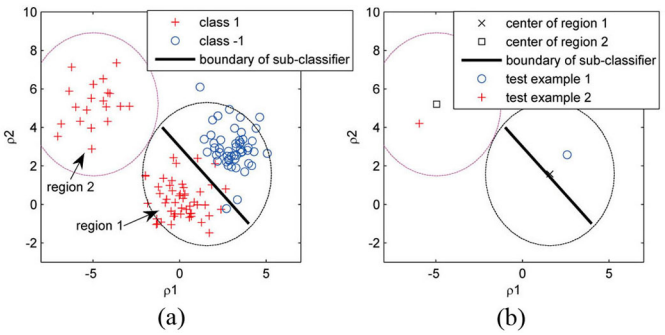

Fig. 4. Example of the heterogeneous clustering-based dichotomizer. (a) Training. (b) Prediction.

Fig. 3 gives an example of LC-ECOC for a three-class classification problem. It is initialized with a compact code $\mathbf{M}$. At the first iteration, it finds the most difficult binary-class problem, supposing to be $\mathbf{m} = [1, -1, 0]^T$. Because $\mathbf{m}$ is not a column of $\mathbf{M}$, LC-ECOC trains a simple base dichotomizer $h_3^{(s)}$ to discriminate classes 1 and 2. At the second iteration, when observing the fact that the most difficult problem $[1, -1, 0]^T$ has already appeared as the third column of $\mathbf{M}$. it trains one layer of clustering-based dichotomizer $h_4^{(c)}$, so as to $h_5^{(c)}$.

We adopt the heterogeneous clustering-based approach [3], [62] to train each complicated clustering-based dichotomizer (Algorithm 1). Specifically, in the training process, the heterogeneous clustering-based approach splits the space of a pair of classes to N_c regions ($N_c > 1$) without considering the class attributes. For each region, if the region contains examples from both classes, it trains a simple base dichotomizer on the region; otherwise, it remembers the class attribute of the region. In the prediction process, a test example is first assigned to its host region, a region whose center has the minimum ED from the example over all regions. Then, if the region owns a base dichotomizer, the approach predicts the test example by the base dichotomizer; otherwise, it assigns the class attribute of the region to the test example.

Fig. 4 gives an example of the training and prediction of a heterogeneous clustering-based dichotomizer. In the training process [Fig. 4(a)], it first finds the most confusing region by splitting the training examples to two regions by k-means. Because region 1 consists of two classes, it trains a simple dichotomizer to discriminate the two classes in the region. Because region 2 consists of only class 1, it simply remembers the class attribute. In the prediction process [Fig. 4(b)], because example 1 falls into region 1, it classifies example 1 to class -1 by the simple dichotomizer in region 1. Because example 2 falls into region 2 and because region 2 belongs to class 1, it classifies example 2 to class 1.

Algorithm 1 LC-ECOC

```
/* Training */
repeat
    Find the most confusing pair of classes
    if the pair has not been tried to solve by ECOC then
        Train a simple dichotomizer for the pair
    else
        /* Training a clustering-based dichotomizer */
        Partition the space of the pair to N_c regions by clustering
        for i = 1, ..., N_c do
            if the examples in the i-th region are from both classes then
                Train a dichotomizer on the region
            else
                Remember the class attribute of the region
            end if
        end for
    end if
    Add the new dichotomizers to the ECOC ensemble
until the training risk converges

/* Prediction */
for q = 1, ..., Q do
    if the dichotomizer h_q is a simple one then
        Predict the example by h_q
    else
        Assign the test example to its host region
        if the region owns a dichotomizer then
            Predict the example by the dichotomizer of the region
        else
            Assign the class attribute of the region to the example
        end if
    end if
end for
Decode the predicted codeword of the example
```

Note that the clustering algorithms that have high accuracies, such as spectral clustering [63] agglomerative clustering [64] maximum margin clustering [15] or clustering ensemble [65] are not suitable for this job. The more "weak" and unstable the clustering algorithm is, the more suitable it seems to be. Hence, *k*-means clustering is adopted.

IV. CPA-Based OW Decoding for ECOC

In this section, we first propose the OW decoding, and then employ CPA to accelerate the decoding algorithm.

A. OW Decoding

OW decoding optimizes the weight matrix of the LW decoding [43] for the minimal training risk, which is formulated as a linear programming problem that can be solved in time $\mathcal{O}(n \log n)$.

The weight matrix is optimized as follows. Given a training example $\boldsymbol{\rho}_i$ with its predicted codeword from the dichotomizers, denoted as $\mathbf{x}_i$, and ground truth label y_i, if $\boldsymbol{\rho}_i$ is classified correctly, according to (3), the following criterion is satisfied:

$$\sum_{q=1}^{Q} w_{y_i,q}\ell(x_{i,q}c_{y_i,q}) \le \sum_{q=1}^{Q} w_{p,q}\ell(x_{i,q}c_{p,q}) \quad \forall p = 1, \ldots, P \tag{4}$$

where $\ell(\theta)$ can be defined as $\ell(\theta) = -\theta$. Letting $\mathbf{u}_{i,p} = [\ell(x_{i,1}c_{p,1}), \ldots, \ell(x_{i,Q}c_{p,Q})]^T$ can rewrite (4) as

$$\mathbf{w}_{y_i}^T\mathbf{u}_{i,y_i} - \mathbf{w}_p^T\mathbf{u}_{i,p} \le 0, \quad \forall p = 1, \ldots, P \tag{5}$$

where any $\mathbf{u}_{i,p}$ should be normalized to $\mathbf{u}_{i,p}/\max_{i,p,q}|u_{i,p,q}|$, so as to prevent unexpected numerical problems. If $\boldsymbol{\rho}_i$ is misclassified, it will cause a training loss ξ_i. One possible measurement of ξ_i is the hinge loss

$$\xi_i = \max_{p=1,\ldots,P}\left(0, \mathbf{w}_{y_i}^T\mathbf{u}_{i,y_i} - \mathbf{w}_p^T\mathbf{u}_{i,p}\right). \tag{6}$$

Minimizing the training risk is to minimize the sum of the training loss of all examples, which is formulated as the following convex linear programming problem:

$$\min_{\mathbf{W}\in\mathcal{W}} \mathcal{J}(\mathbf{W}) \triangleq \min_{\mathbf{W}\in\mathcal{W}} \sum_{i=1}^{n} \max_{p=1,\ldots,P}\left(0, \mathbf{w}_{y_i}^T\mathbf{u}_{i,y_i} - \mathbf{w}_p^T\mathbf{u}_{i,p}\right) \tag{7}$$

which can be rewritten as the following constrained optimization problem:

$$\begin{aligned} \min_{\mathbf{W}\in\mathcal{W},\xi_i\ge 0} \quad & \sum_{i=1}^{n}\xi_i \\ \text{subject to} \quad & \mathbf{w}_p^T\mathbf{u}_{i,p} - \mathbf{w}_{y_i}^T\mathbf{u}_{i,y_i} \ge -\xi_i \\ & \forall i = 1, \ldots, n, \quad \forall p = 1, \ldots, P. \end{aligned} \tag{8}$$

Note that the definition of ξ_i in (6) is important to the difficulty of the optimization. If it is defined as the training error, i.e., $\xi_i \in \{0, 1\}$, problem (8) will be an integer matrix optimization problem with an NP-complete complexity. Usually, we use some convex surrogate function, such as hinge loss, to relax ξ_i to a continuous value. As will be shown in Section V, this relaxation enforces us to pick the most confusing pair of classes according to the training risk matrix but not the confusion matrix of classification errors.

B. CPA-Based OW Decoding

Because problem (8) has $\mathcal{O}(n)$ parameters and $\mathcal{O}(n)$ constraints, solving problem (8) is still inefficient for large-scale problems. Here, we employ the well-known CPA [44]–[47], [66] to further lower its time complexity to $\mathcal{O}(n)$.

CPA is an efficient optimization tool for those convex optimization problems with large amounts of constraints. Its time and storage complexities are irrelevant to the number of constraints. In CPA terminology, a problem with a full constraint set is called a master problem [47] while a problem with only a constraint subset from the full set is called a reduced problem,

or a cutting-plane subproblem. Generally, CPA begins with a reduced problem that has only an empty working constraint set, and then iterates the following two steps: 1) solving the reduced problem with the working constraint set and 2) adding the most violated constraint of the current solution point from the full set to the working constraint set, so as to form a new reduced problem. If the new voilated constraint violates the solution of the previous reduced problem by no more than ϵ, CPA is stopped, where ϵ is a user defined solution precision. It has been proven that the number of iterations is upper bounded by $\mathcal{O}(1/\epsilon)$ [46] which is irrelevant to n.

For our problem, we first reformulate problem (8) to the following equivalent optimization problem:

$$\begin{aligned} \min_{\mathbf{W}\in\mathcal{W},\xi\geq 0} \quad & \xi \\ \text{subject to} \quad & \sum_{i=1}^{n}\sum_{p=1}^{P} g_{i,p}\left(\mathbf{w}_p^T\mathbf{u}_{i,p}-\mathbf{w}_{y_i}^T\mathbf{u}_{i,y_i}\right)\geq -\xi \\ & \forall \mathbf{G}\in\mathcal{Z}^n \end{aligned} \tag{9}$$

where $\mathbf{g}_i = [g_{i,1},\ldots,g_{i,P}]^T$, $\mathbf{G} = \left[\mathbf{g}_1,\ldots,\mathbf{g}_n\right] = \begin{bmatrix} g_{1,1}\cdots g_{n,1} \\ \vdots \ddots \vdots \\ g_{1,P}\cdots g_{n,P}\end{bmatrix}$, and the set $\mathcal{Z}=\{\mathbf{z}_p\}_{p=1}^{P}$ with $\mathbf{z}_p$ defined as

$$z_{p,k} = \begin{cases} 1\ , & \text{if } k=p \\ 0\ , & \text{otherwise}\end{cases}, \quad k=1,\ldots,P. \tag{10}$$

Problem (8) and (9) are equivalent in the following theorem.

Theorem 1: Any solution $\mathbf{W}$ of problem (9) is also a solution of problem (8), and vice versa, with $\xi=\frac{1}{n}\sum_{i=1}^{n}\xi_i$.

Proof: See Appendix VII-A. ■

Comparing problem (9) to (8), we can see that although problem (9) has only one slack variable, the number of its constraints is as high as P^n. Fortunately, problem (9) can be solved approximately by CPA. The CPA-based OW decoding algorithm is described in Algorithm 2. The derivation, which is omitted here, is similar to the well-known SVM$^{\text{perf}}$ toolbox [45], [66], [67].

Because problem (11) has very few constraints, the time complexity of Algorithm 2 is $\mathcal{O}(n)$, which is consumed on calculating $\sum_{i=1}^{n} g_{i,p}\mathbf{u}_{i,p}$ in (11). Besides the linear time complexity, the CPA-based OW decoding has another important merit: its storage complexity is irrelevant to the implementation method of the linear programming toolbox, since the linear programming problem (11) has only $\mathcal{O}(1)$ parameters and $\mathcal{O}(1)$ constraints. We take the standard linear programming toolbox in MATLAB as an example: if we rewrite both (8) and (11) to the standard form "$\min_{\mathbf{x}}\mathbf{f}^{\mathbf{T}}\mathbf{x}$ subject to $\mathbf{Ax}\leq\mathbf{b}$," matrix $\mathbf{A}$ in (8) is $(PQ+n)\times n$ in size, while $\mathbf{A}$ in (11) is only $(PQ+1)\times|\Omega|$ in size where $|\Omega|$ denotes the size of the working constraint set and is a small integer that is irrelevant to n. As a result, the original OW decoding cannot handle middle scale datasets in the MATLAB environment, while the CPA-based OW decoding is not limited by the scale of the dataset.

V. WOLC-ECOC

The framework of WOLC-ECOC is presented in Fig. 1. The training procedure of WOLC-ECOC is detailed in Algorithm 3 and described as follows.

Algorithm 2 CPA-Based OW Decoding

Input: Dataset $\mathcal{U}=\left\{\{\mathbf{u}_{i,p}\}_{p=1}^{P},y_i\right\}_{i=1}^{n}$.
Output: Optimal weight matrix $\mathbf{W}$.
Initialization: Arbitrary initial weight matrix $\mathbf{W}$ ($\mathbf{W}\in\mathcal{W}$), empty initial working constraint set $\mathbf{\Omega}=\{\}$, the size of working constraint set $|\Omega|\leftarrow 0$.

1: **repeat**
2: $|\Omega|\leftarrow|\Omega|+1$
3: Calculate the most violated constraint $\mathbf{G}_{|\Omega|}$

$$g_{i,p}^{|\Omega|}\leftarrow\begin{cases}1, & \text{if } p=\arg\max_p\left(\mathbf{w}_{y_i}^T\mathbf{u}_{i,y_i}-\mathbf{w}_p^T\mathbf{u}_p\right)\\ 0, & \text{otherwise}\end{cases}$$

4: Add the most violated constraint $\mathbf{G}_{|\Omega|}$ to $\mathbf{\Omega}$

$$\mathbf{\Omega}\leftarrow\mathbf{\Omega}\cup\mathbf{G}_{|\Omega|}$$

5: Solve the reduced problem

$$\begin{aligned}\min_{\mathbf{W}\in\mathcal{W},\xi\geq 0}\quad & \xi \\ \text{subject to}\quad & \sum_{i=1}^{n}\sum_{p=1}^{P} g_{i,p}\left(\mathbf{w}_p^T\mathbf{u}_{i,p}-\mathbf{w}_{y_i}^T\mathbf{u}_{i,y_i}\right)\geq-\xi, \\ & \forall\mathbf{G}\in\mathbf{\Omega}\end{aligned} \tag{11}$$

6: **until** $\mathbf{\Omega}$ is unchanged

WOLC-ECOC starts with any valid ECOC $\{\mathbf{M},\mathcal{C}\}$ with $\mathcal{C}=\{h_1,\ldots,h_Q\}$, such as 1versusALL, 1versus1, or compact code (i.e., $Q<P$), and then iterates the following two steps.

1) The first step optimizes the weight matrix $\mathbf{W}$ of the OW decoding and obtains the minimal training risk $\mathcal{J}_o$ by the weightoptimization function which is described in Section IV.
2) The second step first finds the top s most confusing pairs of classes, denoted as $\{\mathbf{m}_k\}_{k=1}^{s}$, and then adds all s dichotomizers $\{h'_k\}_{k=1}^{s}$ that discriminate $\{\mathbf{m}_k\}_{k=1}^{s}$, respectively to $\mathcal{C}$. For training h'_k, as presented in LC-ECOC (Algorithm 1), two situations should be considered: if $\mathbf{m}_k$ does not equal to any column of $\mathbf{M}$, we train a new simple dichotomizer $h_k'^{(s)}$ as usual by the simplelearning function; otherwise, we train a complicated clustering-based dichotomizer $h_k'^{(c)}$ by the *ClusteringBasedLearning* function in Section III.

The loop stops when the maximum iteration number T is reached or the following inequality is satisfied for continuous Z iterations:

$$\frac{\mathcal{J}'_o-\mathcal{J}_o}{\mathcal{J}_o}\leq\eta \tag{12}$$

where $\mathcal{J}_o$ and $\mathcal{J}'_o$ are the training risks of the current and previous iterations respectively, and η is a user defined solution precision. Finally, the ECOC ensemble $\{\mathbf{M}_o,\mathcal{C}_o,\mathbf{W}_o\}$ that achieves the minimum risk is returned. Here, we have to note that although OW decoding can reach its global minimum solution at each WOLC-ECOC iteration, the overall heuristic training process only reaches a local minimum solution.

Algorithm 3 WOLC-ECOC

Input: Dataset $\mathcal{D} = \{\boldsymbol{\rho}_i, y_i\}_{i=1}^n$, the number of the most confusing pairs per iteration s, maximum iteration number T, solution precision η, parameter for the termination condition Z.

Output: ECOC coding matrix $\mathbf{M}_o$ and the corresponding classifier ensemble $\mathcal{C}_o$, optimal weight matrix $\mathbf{W}_o$.

Initialization: initial ternary ECOC coding matrix $\mathbf{M} \in \{-1, 0, 1\}^{P\times Q}$ and the classifier ensemble $\mathcal{C} = \{h_1, \ldots, h_Q\}$ that is learned from $\mathbf{M}$ and $\mathcal{D}$, $\mathcal{J}_o' \leftarrow inf$, $z \leftarrow 0$, $t \leftarrow 0$.

repeat
  for $i = 1, \ldots, n$ **do**
    Predict $\boldsymbol{\rho}_i$ by the LC-ECOC prediction process
    Calculate $\{\mathbf{u}_{i,p}\}_{p=1}^P$ defined in (4)
  end for
  /* Optimize weight matrix */
  $\{\mathbf{W}, \mathcal{J}_o\} \leftarrow$*WeightOptimization*$(\mathcal{U}, \mathbf{M})$, where $\mathcal{U} = \left\{\{\mathbf{u}_{i,p}\}_{p=1}^P, y_i\right\}_{i=1}^n$
  if $\mathcal{J}_o = 0$ **then**
    $\mathbf{M}_o \leftarrow \mathbf{M}$, $\mathcal{C}_o \leftarrow \mathcal{C}$, $\mathbf{W}_o \leftarrow \mathbf{W}$
    return
  end if
  /* Get the most confusing pairs */
  Find s pairs of classes that have the highest training risks $\{\mathbf{m}_k\}_{k=1}^s$. Get their corresponding training risks $\{\epsilon_k\}_{k=1}^s$
  /* Learn the base dichotomizers from $\{\mathbf{m}_k\}_{k=1}^s$ */
  for $k = 1, \ldots, s$ **do**
    if $\epsilon_k \neq 0$ **then**
      if $\mathbf{m}_k$ does not equal to any column of $\mathbf{M}$ **then**
        $h_k' \leftarrow$ *SimpleLearning*$(\mathcal{D}, \mathbf{m}_k)$
      else
        $h_k' \leftarrow$ *ClusteringBasedLearning*$(\mathcal{D}, \mathbf{m}_k)$
      end if
      $\mathbf{M} \leftarrow [\mathbf{M}, \mathbf{m}_k]$, $\mathcal{C} \leftarrow \mathcal{C} \cup h_k'$
    end if
  end for
  /* Control the termination criterion */
  if $(\mathcal{J}_o' - \mathcal{J}_o)/\mathcal{J}_o \leq \eta$ **then**
    $z \leftarrow z + 1$
  else
    $z \leftarrow 0$
    $\mathbf{M}_o \leftarrow \mathbf{M}$, $\mathcal{C}_o \leftarrow \mathcal{C}$, $\mathbf{W}_o \leftarrow \mathbf{W}$
  end if
  $t \leftarrow t + 1$, $\mathcal{J}_o' \leftarrow \mathcal{J}_o$
until $z = Z$ or $t = T$

WOLC-ECOC has two merits when compared to its components. First, the monotonic decrease of the training risk of WOLC-ECOC is guaranteed, see Appendix VII-B for the proof. Second, a small ECOC code length is ensured, since discriminating the most difficult binary-class problem at each iteration make ECOC obtain the maximum performance gain.

In Algorithm 3, we have considered the following three issues for the robustness and efficiency of WOLC-ECOC.

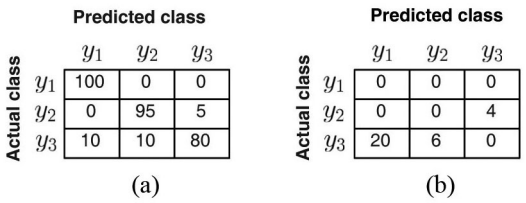

Fig. 5. Comparison of the confusion matrix and the training risk matrix of a three-class classification problem. (a) Confusion matrix. (b) Training risk matrix.

TABLE I
DESCRIPTIONS OF THE DATASETS. "n" IS THE DATASET SIZE, "d" IS THE DIMENSION, "P" IS THE NUMBER OF THE CLASSES

ID	Data	n	d	P
1	Dermathology	366	34	6
2	Iris	150	4	3
3	Ecoli	336	7	8
4	Wine	178	13	3
5	Glass	214	9	7
6	Thyroid	215	5	3
7	Vowel	990	10	11
8	Balance	625	4	3
9	Yeast	1484	8	10
10	Satimage	6435	36	7
11	Pendigits	10992	16	10
12	Segmentation	2310	19	7
13	OptDigts	5620	64	10
14	Vehicle	846	18	4

First, how to balance the discriminability and the code length? Multiple layers of clustering-based dichotomizers might trigger a significant performance improvement with a risk of overfitting, while one or two layers might not improve the performance. To solve the problem, the following termination criterion is used: if the training risk does not decrease in a rate of η [in (12)] for Z continuous iterations, we stop the training procedure. Usually, setting Z to an arrange of 3–5 is enough.

Second, how to make the performance robust? Sometimes, the most confusing pair is too stubborn to overcome. To prevent this unwanted situation, we discriminate the top s most confusing pairs of classes, denoted as $\{\mathbf{m}_k\}_{k=1}^s$, instead of a single most confusing pair.

Third, how to define the most confusing pair of classes? ECOC-ONE [27] selects the most confusing pair of classes by the confusion matrix $\boldsymbol{\epsilon}$ which is defined as

$$\epsilon_{i,j} = \sum_{k:\boldsymbol{\rho}_k \in \text{ class } i} e_{i,j}(\boldsymbol{\rho}_k) \tag{13}$$

where function $e_{i,j}(\cdot)$ is defined as

$$e_{i,j}(\boldsymbol{\rho}) = \begin{cases} 1, & \text{if } \boldsymbol{\rho} \in \text{class } i \text{ but is misclassified to } j \\ 0, & \text{otherwise.} \end{cases}$$

However, because OW decoding relaxes the loss function from classification error $\{0, 1\}$ to a convex continuous surrogate function (6) with a range of $[0, +\infty)$, Algorithm 3 minimizes the training risk $J(\mathbf{W})$ instead of classification error. That is to say, for each iteration, Algorithm 3 picks a pair of

TABLE II
ACCURACY COMPARISON (%) OF THE ECOC CODING–DECODING METHODS ON THE UCI DATASETS. THE BASE LEARNER IS THE DISCRETE ADABOOST. IN EACH GRID, THE FIRST LINE IS THE ACCURACY AND THE SECOND LINE IS THE STANDARD DEVIATION. THE ROW "RANK" IS THE AVERAGE RANK OVER ALL 14 DATASETS

Coding	1vs1			1vsALL			Random			ECOC-ONE			DECOC			WOLC-ECOC
Decoding	HD	LB	LW	HD	LB	LW	HD	LB	LW	HD	LB	LW	HD	LB	LW	OW
Dermathology	91.11	91.11	**92.18**	87.51	87.51	89.44	81.06	80.86	82.47	89.10	89.23	91.86	70.35	71.19	73.16	91.56
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(2.29)	(3.91)	(2.96)	(0.43)	(0.49)	(0.00)	(2.11)	(1.87)	(1.84)	(0.22)
Iris	94.64	94.64	94.64	**96.73**	**96.73**	96.03	96.03	95.96	95.89	95.34	95.34	95.62	96.03	96.03	96.03	96.03
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.46)	(0.61)	(0.44)	(0.00)	(0.00)	(0.36)	(0.00)	(0.00)	(0.00)	(0.00)
Ecoli	85.00	85.00	84.75	81.27	81.27	79.99	76.30	76.63	77.52	80.17	80.16	78.84	75.16	72.36	78.47	**87.40**
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(2.35)	(1.33)	(1.36)	(1.18)	(1.00)	(0.83)	(4.19)	(4.26)	(2.37)	(0.82)
Wine	**94.31**	**94.31**	**94.31**	91.44	91.44	91.44	93.27	93.00	93.20	92.05	91.70	91.87	**93.87**	93.58	**93.93**	93.69
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.88)	(0.92)	(0.62)	(0.55)	(0.63)	(0.68)	(0.56)	(0.24)	(0.69)	(0.00)
Glass	67.78	67.78	67.38	57.12	57.12	**68.15**	61.81	63.14	63.63	60.45	60.85	65.00	58.21	57.25	63.48	67.28
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(1.49)	(3.14)	(2.50)	(1.93)	(2.63)	(2.20)	(3.98)	(4.35)	(2.55)	(0.66)
Thyroid	93.45	93.45	93.45	93.95	93.95	93.95	**94.57**	94.14	94.16	93.95	93.95	93.95	93.78	93.81	93.93	**95.45**
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.92)	(0.93)	(0.87)	(0.00)	(0.00)	(0.00)	(0.60)	(1.05)	(0.72)	(0.00)
Vowel	58.74	58.74	58.74	39.80	39.80	45.97	39.58	37.92	40.99	42.60	42.10	46.50	43.24	45.80	45.28	**60.61**
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(2.60)	(1.67)	(1.95)	(1.65)	(1.11)	(1.49)	(2.74)	(1.99)	(2.44)	(0.82)
Balance	86.40	86.40	86.56	87.52	87.52	87.67	86.75	86.74	**87.55**	77.49	77.49	77.81	76.70	76.70	76.70	**88.97**
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(1.35)	(1.96)	(1.53)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.40)
Yeast	53.93	53.93	53.99	39.24	39.24	54.06	45.48	43.82	45.50	44.96	43.61	50.53	45.51	46.94	50.53	**56.28**
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.96)	(1.99)	(1.51)	(1.10)	(0.93)	(0.81)	(1.65)	(2.15)	(0.99)	(0.18)
Satimage	86.84	86.84	**86.92**	82.36	82.36	82.29	84.70	84.47	85.01	83.26	83.25	83.26	77.69	79.08	84.83	85.74
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.55)	(0.90)	(0.34)	(0.39)	(0.24)	(0.21)	(2.77)	(3.47)	(0.61)	(0.11)
Pendigits	97.16	97.16	**97.24**	84.88	84.88	86.25	76.46	76.05	77.65	86.13	86.08	87.13	78.37	77.87	78.84	96.70
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(1.03)	(0.90)	(1.18)	(0.24)	(0.44)	(0.19)	(1.00)	(1.00)	(1.28)	(0.15)
Segmentation	95.18	95.18	95.31	90.03	90.03	93.06	91.48	91.28	92.43	92.46	92.46	94.20	93.37	93.37	93.37	**95.60**
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(1.02)	(1.02)	(0.69)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.18)
OptDigts	95.03	95.03	95.28	83.27	83.27	84.09	71.66	72.80	74.69	85.80	85.80	86.03	75.27	75.27	75.27	**95.67**
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(1.17)	(2.06)	(0.94)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.13)
Vehicle	73.40	73.40	73.52	65.12	65.12	72.33	70.39	70.21	73.07	68.16	67.72	72.35	70.88	71.29	**74.28**	**75.41**
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(1.00)	(1.30)	(0.69)	(0.81)	(0.27)	(0.32)	(1.31)	(1.02)	(1.04)	(0.13)
Rank	3.93	4.29	3.64	9.86	10.07	6.43	8.79	9.50	6.86	8.50	9.07	6.86	8.93	9.14	6.86	2.14

classes that has the highest training risk but not the one that has the highest classification error. Correspondingly, the training risk matrix $\boldsymbol{\epsilon}$ is defined as

$$\epsilon_{i,j} = \sum_{k:\boldsymbol{\rho}_k \in \text{ class } i} \left(\mathbf{w}_i^T \mathbf{u}_{k,i} - \mathbf{w}_j^T \mathbf{u}_{k,j}\right) \delta\left(\min_{p=1,\ldots,P;p\neq j} \mathbf{w}_p^T \mathbf{u}_{k,p} - \mathbf{w}_j^T \mathbf{u}_{k,j}\right) \quad (14)$$

where $\delta(\cdot)$ is the indicator function

$$\delta(a) = \begin{cases} 1, & \text{if } a > 0, \\ 0, & \text{otherwise.} \end{cases}$$

An example comparison between the confusion matrix and the training risk matrix is shown in Fig. 5. From Fig. 5(a), we observe that: 1) each class consists of 100 examples; 2) the candidate confusing pairs of classes are $\mathbf{m}^{1,2} = [1, -1, 0]^T$, $\mathbf{m}^{1,3} = [1, 0, -1]^T$, and $\mathbf{m}^{2,3} = [0, 1, -1]^T$ with the numbers of misclassified examples being $\epsilon_{1,2} = 0 + 0 = 0$, $\epsilon_{1,3} = 10 + 0 = 10$, and $\epsilon_{2,3} = 10 + 5 = 15$ respectively; and 3) the most confusing pair is selected as $\mathbf{m}^{2,3}$.

But from Fig. 5(b), we observe that: 1) the training risk pairs are $\epsilon_{1,2} = 0 + 0 = 0$, $\epsilon_{1,3} = 0 + 20 = 20$, and $\epsilon_{2,3} = 4 + 6 = 10$, respectively and 2) the highest training risk pair is $\mathbf{m}^{1,3}$. Comparing Fig. 5(a) with (b), we can see that different optimization objectives might give a binary-class problem different training priorities.

VI. EXPERIMENT ANALYSIS

In this section, we first compare WOLC-ECOC with 15 coding–decoding pairs on 14 UCI benchmark datasets with two kinds of base dichotomizer—AdaBoost and SVM, then study the convergence behavior of WOLC-ECOC, and finally apply WOLC-ECOC to a music genre classification problem.

A. Experiment Settings

We used 14 multiclass datasets in the UCI machine learning repository database.[1] The properties of the datasets are listed in Table I. All datasets were normalized into the range of [0, 1] in dimension [68].

For the proposed WOLC-ECOC, the number of the most confusing pairs per iteration s was set to 3. The termination condition Z was set to 3. The solution precision η was set to 0.01. The initial ECOC was 1versusALL. The maximum iteration number T was set to $3P$ where P is the number of classes.

To show the effectiveness of WOLC-ECOC, we compared it with five state-of-the-art ECOC coding designs, including 1versus1, 1versusALL, random ECOC[42], DECOC [26] and ECOC-ONE using 1versusALL as its initialization [38]. Each of the comparison coding methods combined three decoding methods, including HD, LB [42], and LW [43] decodings. We

[1] http://archive.ics.uci.edu/ml/

TABLE III
ACCURACY (%) COMPARISON OF THE ECOC CODING–DECODING METHODS ON THE UCI DATASETS. THE BASE LEARNER IS THE GAUSSIAN RBF KERNEL-BASED SVM. IN EACH GRID, THE FIRST LINE IS THE ACCURACY AND THE SECOND LINE IS THE STANDARD DEVIATION. THE ROW "RANK" IS THE AVERAGE RANK OVER ALL 14 DATASETS

Coding	1vs1			1vsALL			Random			ECOC-ONE			DECOC			WOLC-ECOC
Decoding	HD	LB	LW	HD	LB	LW	HD	LB	LW	HD	LB	LW	HD	LB	LW	OW
Dermathology	**96.93**	**96.76**	**96.88**	94.87	94.63	95.82	94.82	95.30	95.94	94.73	94.74	95.59	94.76	95.16	95.40	95.17
	(0.59)	(0.35)	(0.51)	(0.46)	(0.53)	(0.84)	(1.33)	(1.00)	(0.49)	(2.88)	(2.51)	(0.55)	(0.71)	(0.78)	(0.86)	(0.55)
Iris	**96.80**	96.66	96.51	95.41	95.00	**96.67**	**97.52**	**97.30**	**96.91**	96.32	96.19	**96.87**	**96.97**	**96.86**	96.75	96.69
	(0.96)	(0.63)	(0.60)	(1.54)	(0.72)	(1.08)	(0.76)	(0.38)	(0.67)	(0.47)	(1.33)	(0.76)	(0.57)	(0.79)	(0.79)	(0.37)
Ecoli	85.07	**85.17**	84.81	80.52	80.72	82.75	80.93	81.09	82.40	81.59	81.66	83.28	74.59	74.39	82.70	83.49
	(0.81)	(0.60)	(0.75)	(1.03)	(0.79)	(0.98)	(2.15)	(2.12)	(1.13)	(1.13)	(0.73)	(0.67)	(5.18)	(6.04)	(1.40)	(0.25)
Wine	96.05	96.16	96.33	**96.65**	96.15	**96.60**	**97.37**	**96.93**	**97.04**	**97.16**	96.64	**96.70**	96.38	**96.77**	**96.60**	95.85
	(1.20)	(0.79)	(0.85)	(0.87)	(0.78)	(0.89)	(0.76)	(0.93)	(0.58)	(0.81)	(0.63)	(0.70)	(0.96)	(0.67)	(0.99)	(0.80)
Glass	**62.95**	**63.84**	**64.01**	52.98	52.03	61.27	61.00	**62.09**	**61.57**	56.59	56.60	**63.06**	58.10	56.75	59.91	**63.18**
	(1.79)	(2.01)	(3.16)	(2.61)	(1.99)	(1.37)	(2.24)	(2.49)	(2.03)	(2.03)	(2.22)	(2.54)	(5.02)	(4.08)	(2.76)	(1.80)
Thyroid	**96.20**	**96.14**	**96.22**	95.21	**95.45**	**95.93**	**96.21**	**96.23**	**95.77**	94.99	94.64	**95.69**	94.50	94.51	93.67	95.63
	(0.75)	(0.60)	(0.81)	(1.03)	(0.96)	(0.62)	(0.93)	(0.69)	(0.67)	(0.83)	(1.05)	(0.63)	(1.37)	(1.27)	(1.02)	(0.56)
Vowel	67.11	67.81	67.87	34.88	34.67	36.96	31.07	33.17	34.37	37.56	37.55	39.87	43.40	41.04	41.87	**70.87**
	(1.40)	(1.19)	(1.84)	(0.78)	(1.58)	(1.36)	(2.89)	(1.88)	(2.05)	(2.43)	(1.69)	(1.47)	(3.33)	(2.54)	(1.62)	(1.20)
Balance	90.12	88.89	89.48	90.25	**90.34**	90.28	89.74	89.78	89.66	88.19	87.71	87.35	88.76	88.95	88.74	**91.29**
	(1.07)	(1.41)	(0.99)	(0.88)	(1.28)	(0.93)	(0.70)	(0.94)	(0.98)	(0.49)	(0.75)	(1.55)	(0.91)	(0.65)	(0.61)	(0.92)
Yeast	**58.98**	**58.95**	**59.35**	38.17	38.41	54.73	51.90	50.97	53.19	43.00	43.71	54.98	51.97	51.97	55.14	55.27
	(1.10)	(0.56)	(0.63)	(1.38)	(1.44)	(0.62)	(1.02)	(2.22)	(1.35)	(2.26)	(2.24)	(1.02)	(2.35)	(3.11)	(1.33)	(0.57)
Satimage	85.73	**85.74**	**85.81**	80.07	79.98	81.05	81.95	81.43	82.07	81.20	81.27	81.49	74.95	75.86	82.48	**86.10**
	(0.19)	(0.21)	(0.20)	(0.18)	(0.30)	(0.27)	(0.45)	(0.90)	(0.58)	(0.62)	(0.69)	(0.81)	(3.47)	(3.20)	(0.68)	(0.27)
Pendigits	**99.01**	**99.01**	**98.97**	91.79	91.69	92.29	85.19	85.20	86.05	92.53	92.60	93.24	88.23	88.43	88.97	98.25
	(0.06)	(0.06)	(0.06)	(0.19)	(0.15)	(0.17)	(1.16)	(0.76)	(0.74)	(0.23)	(0.16)	(0.31)	(1.50)	(1.15)	(0.83)	(0.16)
Segmentation	**94.86**	**95.14**	**95.06**	85.20	85.16	89.45	86.41	86.93	87.60	89.21	89.23	91.79	87.03	86.93	86.67	**95.12**
	(0.45)	(0.47)	(0.42)	(0.98)	(0.76)	(0.70)	(1.54)	(1.72)	(1.16)	(0.78)	(0.66)	(0.58)	(0.89)	(1.08)	(1.08)	(0.30)
OptDigts	**97.80**	**97.74**	**97.79**	92.99	92.88	94.39	88.42	88.18	89.35	94.66	94.74	94.79	89.33	89.33	89.23	97.58
	(0.09)	(0.12)	(0.07)	(0.13)	(0.14)	(0.15)	(0.71)	(1.37)	(0.81)	(0.16)	(0.13)	(0.18)	(0.23)	(0.31)	(0.17)	(0.11)
Vehicle	79.62	79.66	80.01	69.02	68.53	75.49	75.16	76.28	77.77	71.34	72.12	76.85	74.48	75.12	76.99	**82.51**
	(0.93)	(0.83)	(0.64)	(0.70)	(1.43)	(0.82)	(0.60)	(1.62)	(1.32)	(1.03)	(1.06)	(1.50)	(0.83)	(1.74)	(1.33)	(0.34)
Rank	2.07	2.79	2.43	10.64	10.86	6.14	8.29	6.93	6.07	8.36	8.79	5.07	9.71	9.21	7.29	4.57

followed the ECOC library [69][2] for the implementations of the referenced methods.

To demonstrate how a base classifiers affects the performance, we used two popular base classifiers—discrete AdaBoost [70] and Gaussian RBF kernel-based SVM [66].[3] AdaBoost uses 40 *decision stump* weak learners. The parameters of SVM were searched in grid: parameter C was searched through $\{2^{12}, 2^{13}, \ldots, 2^{18}\}$, and the kernel width σ of the RBF kernel was searched through $\{0.25\gamma, 0.5\gamma, \gamma, 2\gamma, 4\gamma\}$, where γ is the average ED between the training examples.

For each dataset, we ran each pair of the coding–decoding methods 10 times and recorded the average experimental results. For each single run, we applied a stratified sampling and 10-fold cross-validation, and tested for confidence interval at 95% with the two-tailed t-test. Therefore, we conducted 100 independent runs on each dataset for each pair of coding–decoding methods.

B. Effectiveness

Tables II and III list the classification accuracies of all coding–decoding methods with respect to AdaBoost and SVM, respectively. From Table II, we can see clearly that WOLC-ECOC is the most effective one. But from Table III, we observe that WOLC-ECOC is less effective than the 1versus1 coding but more effective than other coding methods.

The reason why the WOLC-ECOC with AdaBoost performs better than the WOLC-ECOC with SVM may be explained from information theory. It is well known in information theory that the error-correcting ability of any coding method is upper-bounded by the Shannon limit which is irrelevant to the coding method. That is to say, it is possible that the performance of a strong coding method in a noisy channel is worse than the performance of a weak coding method in a clean channel.

The channel of an ECOC problem, as presented in Section II, is determined by the features, base learner and coding method.

1) The more suitable the bipartitions of the classes are and the stronger the base learner is, the cleaner the channel will be. Because 1versus1 bipartitions data according to their natural distributions, its channel has minimum noise in most datasets. Similarly, AdaBoost introduces more noise to the channel than SVM. We can image that the Shannon limits of different coding methods with AdaBoost as the base learner tend to be more similar than those with SVM as the base learner.
2) On the other side, the more diverse the dichotomizers are and the larger the minimum distance between the codewords is, the stronger the error-correcting ability of the codes will be, where the term "diverse" is also named independent in some papers [39], [40].

[2]http://sourceforge.net/projects/ecoclib/
[3]http://svmlight.joachims.org/svm$_{\text{p}}$erf.html

TABLE IV
CODE LENGTH COMPARISON OF THE ECOC CODING–DECODING METHODS ON THE UCI DATASETS

Coding	1vs1	1vsALL	Random	ECOC-ONE						DECOC	WOLC-ECOC	
Decoding	–	–	–	HD		LB		LW		–	OW	
Base classifier	–	–	–	Ada	SVM	Ada	SVM	Ada	SVM	–	Ada	SVM
Dermathology	15.00 (0.00)	6.00 (0.00)	10.00 (0.00)	7.09 (0.08)	7.26 (0.37)	7.11 (0.09)	7.30 (0.28)	7.50 (0.00)	7.74 (0.42)	5.00 (0.00)	9.09 (1.27)	6.00 (0.00)
Iris	3.00 (0.00)	3.00 (0.00)	10.00 (0.00)	4.50 (0.00)	4.93 (0.81)	4.50 (0.00)	4.99 (0.55)	6.31 (0.26)	6.68 (0.39)	2 .00 (0.00)	7.00 (0.00)	5.93 (0.78)
Ecoli	28.00 (0.00)	8.00 (0.00)	10.00 (0.00)	9.48 (0.13)	9.05 (0.09)	9.46 (0.13)	9.10 (0.13)	9.65 (0.20)	9.29 (0.20)	7.00 (0.00)	14.75 (2.01)	15.24 (4.16)
Wine	3.00 (0.00)	3.00 (0.00)	10.00 (0.00)	7.00 (0.00)	7.00 (0.45)	7.00 (0.00)	7.36 (0.36)	7.00 (0.00)	7.36 (0.38)	2.00 (0.00)	3.00 (0.00)	3.00 (0.00)
Glass	15.00 (0.00)	6.00 (0.00)	10.00 (0.00)	7.35 (0.13)	7.40 (0.15)	7.23 (0.11)	7.39 (0.18)	7.93 (0.41)	7.61 (0.32)	5.00 (0.00)	9.44 (0.37)	12.50 (1.08)
Thyroid	3.00 (0.00)	3.00 (0.00)	10.00 (0.00)	6.63 (0.00)	6.33 (0.32)	6.63 (0.00)	6.45 (0.74)	6.63 (0.00)	6.18 (0.56)	2.00 (0.00)	3.00 (0.00)	3.35 (0.26)
Vowel	55.00 (0.00)	11.00 (0.00)	10.00 (0.00)	12.10 (0.11)	12.20 (0.13)	12.05 (0.06)	12.23 (0.15)	12.10 (0.11)	12.05 (0.06)	10.00 (0.00)	26.64 (0.58)	24.25 (2.59)
Balance	3.00 (0.00)	3.00 (0.00)	10.00 (0.00)	8.00 (0.00)	7.93 (0.16)	8.00 (0.00)	7.69 (0.39)	8.00 (0.00)	7.71 (0.45)	2.00 (0.00)	15.16 (1.96)	13.60 (3.29)
Yeast	45.00 (0.00)	10.00 (0.00)	10.00 (0.00)	11.20 (0.15)	11.13 (0.10)	11.14 (0.12)	11.11 (0.09)	12.73 (0.29)	11.19 (0.24)	9.00 (0.00)	13.30 (0.63)	16.45 (2.55)
Satimage	15.00 (0.00)	6.00 (0.00)	10.00 (0.00)	7.09 (0.10)	7.06 (0.07)	7.04 (0.08)	7.10 (0.13)	7.00 (0.00)	7.60 (0.32)	5.00 (0.00)	10.70 (2.52)	16.78 (5.18)
Pendigits	45.00 (0.00)	10.00 (0.00)	10.00 (0.00)	11.43 (0.17)	11.10 (0.11)	11.38 (0.18)	11.13 (0.17)	11.04 (0.06)	11.09 (0.12)	9.00 (0.00)	24.06 (4.43)	22.74 (6.23)
Segmentation	21.00 (0.00)	7.00 (0.00)	10.00 (0.00)	8.00 (0.00)	8.00 (0.00)	8.00 (0.00)	8.00 (0.00)	8.25 (0.00)	8.08 (0.09)	6.00 (0.00)	13.18 (2.05)	14.71 (3.30)
OptDigts	45.00 (0.00)	10.00 (0.00)	10.00 (0.00)	11.00 (0.00)	11.00 (0.00)	11.00 (0.00)	11.00 (0.00)	11.00 (0.00)	11.08 (0.12)	9.00 (0.00)	22.45 (5.62)	24.14 (1.93)
Vehicle	6.00 (0.00)	4.00 (0.00)	10.00 (0.00)	5.05 (0.07)	5.09 (0.12)	5.02 (0.05)	5.00 (0.00)	5.43 (0.43)	5.68 (0.46)	2.00 (0.00)	10.96 (0.68)	13.56 (4.60)

When the Shannon limits are similar, the performance is determined by the error-correcting ability of the coding methods, which explains the advantage of WOLC-ECOC in Table II; otherwise, the performance is determined by the Shannon limits, which explains the inferior of the WOLC-ECOC to 1versus1 coding in Table III.

Note that WOLC-ECOC was initialized by 1versusALL in all experiments. If it is initialized by other coding methods that are better than 1versusALL, it may achieve better performance.

C. Efficiency

The efficiency of an ECOC method is influenced by its code length. The shorter the code length is, the more efficient the ECOC method will be.

Table IV lists the code lengths of all comparison methods. From the table, we can see that WOLC-ECOC has a much shorter code length than 1versus1, though it has a slightly longer code length than the other codings. Generally, it is worthy sacrificing some efficiency for much better performance.

D. Study of the Convergence Behavior

In this subsection, we verify the convergence behavior of WOLC-ECOC empirically. For simplicity, we only give two examples on the dermathology and vehicle datasets, which are shown in Figs. 6 and 7, respectively. The training risk (i.e., objective value) in both figures is calculated by (7), and the accuracy is defined as the ratio of the number of correctly classified training/test examples over the total number.

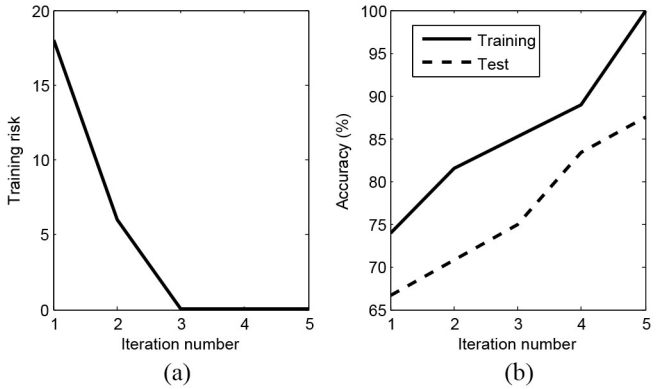

Fig. 6. Convergence behavior of WOLC-ECOC on the dermathology dataset with discrete AdaBoost as the base learner. (a) Convergence behavior of the training risk (objective value). (b) Curves of the training and test accuracies.

From the figures, we observe that the training risks decrease rigorously with respect to the numbers of training iterations.

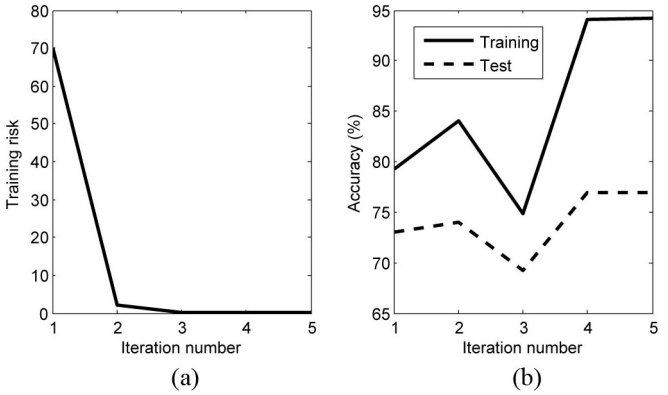

Fig. 7. Convergence behavior of WOLC-ECOC on the vehicle dataset with discrete AdaBoost as the base learner. (a) Convergence behavior of the training risk (objective value). (b) Curves of the training and test accuracies.

TABLE V
ACCURACY (%) COMPARISON OF THE ECOC CODING–DECODING METHODS ON THE DORTMUND MUSIC DATASET WITH THREE KINDS OF FEATURES. IN EACH GRID, THE FIRST LINE IS THE ACCURACY AND THE SECOND LINE REPRESENTS ITS CORRESPONDING DECODING METHOD

Coding	1vs1	1vsALL	DECOC	ECOC-ONE	WOLC-ECOC
MMFCC	43.15	47.33	45.34	49.00	**50.49**
	LW	LW	LW	LW	OW
MOSC	44.41	47.89	46.76	50.15	**52.78**
	LW	LW	LW	LW	OW
MNASE	45.75	50.85	46.42	50.93	**52.86**
	LW	LW	LW	LW	OW

TABLE VI
CODE LENGTH COMPARISON OF THE ECOC CODING–DECODING METHODS ON THE DORTMUND DATASET WITH THREE KINDS OF FEATURES

Coding	1vs1	1vsALL	DECOC	ECOC-ONE	WOLC-ECOC
MMFCC	45.00	9.00	8.00	16.62	27.64
MOSC	45.00	9.00	8.00	14.24	22.78
MNASE	45.00	9.00	8.00	14.75	24.23

We also observe that the training and test accuracies increase in general along with the decrease of the objective values.

E. Application to Music Genre Classification

The fast development of multimedia technologies enable people to enjoy a large amount of music, which calls for developing tools to classify music effectively and efficiently. The SVM-based 1versus1 and 1versusALL classifier ensembles are popular for the music classification problems [71]. The purpose of this subsection is to show the advantage of the WOLC-ECOC over the aforementioned two coding methods on this problem.

The music genre dataset is the Dortmund dataset [72].[4] It consists of 1886 recordings of music pieces of 10 s duration, which are classified to nine types of music. Each music piece is a 44.1 kHz, 16-bits, stereo MP3 file. Here, we converted each file to a mono audio file and extracted three kinds of acoustic features from the file as in [73] which were the modulation spectral analysis of the Mel-frequency Cepstral coefficients (MMFCC), octave-based spectral contrast (MOSC), and normalized audio spectral envelope (MNASE). As a result, each file was formulated as an example with three kinds of features. The parameters settings of the ECOC methods and SVM were as same as those in Section VI-A.

[4]http://www-ai.cs.uni-dortmund.de/audio.html

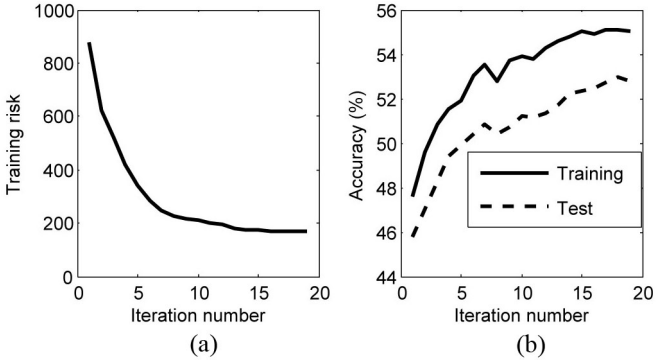

Fig. 8. Convergence behavior of WOLC-ECOC on the Dortmund music genre dataset with MNASE as the acoustic feature.

Tables V and VI list the accuracy and code length comparisons of the ECOCs with the three acoustic features. From Table V, it is clear that WOLC-ECOC is the most powerful one. From Table VI, we observe that the code length of WOLC-ECOC is much shorter than 1versus1, though the code length of WOLC-ECOC is slightly longer than the other three methods.

Fig. 8 gives an example of the convergence behavior of the training risk of WOLC-ECOC with MNASE as the feature. From Fig. 8(a), we observe that the training risk decreases rigorously with respect to the number of iterations.

VII. CONCLUSION

In this paper, we have proposed a heuristic ternary WOLC-ECOC. First, we have proposed LC-ECOC, a greedy training method that iteratively constructs multiple strong dichotomizers to discriminate the most confusing binary-class problem. Then, we have proposed the CPA-based OW decoding. OW decoding improves LW decoding by optimizing the weight matrix of the latter for the minimum training risk. The optimization problem is further solved by CPA, which makes the OW decoding have linear time and storage complexities. At last, we have proposed WOLC-ECOC, which iteratively executes LC-ECOC and the CPA-based OW decoding until the training risk converges. WOLC-ECOC not only inherits all merits of LC-ECOC and the CPA-based OW decoding but also ensures the decrease of the training risk.

We have conducted an extensive experimental comparison with 15 state-of-the-art ECOC coding–decoding pairs on 14 UCI datasets with the discrete AdaBoost and well-tuned RBF kernel-based SVM as two base learners. Experimental results have shown that: 1) when AdaBoost is used as the base learner, WOLC-ECOC outperforms all 15 coding–decoding pairs; 2) when SVM is used as the base learner, WOLC-ECOC is weaker than the traditional 1versus1 coding method but better than other coding methods; and 3) the code length of WOLC-ECOC is much shorter than that of 1versus1. We have

explained the experimental phenomena in the view of information theory. Moreover, we have applied WOLC-ECOC to a music genre classification problem. Experiment results have shown that WOLC-ECOC outperforms all referenced coding methods including 1versus1.

APPENDIX

A. Proof of Theorem 1

The proof is similar with the proof of [45, Theorem 1]. The key point is to prove that the training loss of problem (9) and the training loss of problem (8) are equivalent

$$\sum_{i=1}^{n}\xi_i = \sum_{i=1}^{n}\max_{p=1,\ldots,P}\left(0, \mathbf{w}_{y_i}^T\mathbf{u}_{i,y_i} - \mathbf{w}_p^T\mathbf{u}_{i,p}\right) = \sum_{i=1}^{n}\max_{\forall \mathbf{g}_i\in\mathcal{Z}}\left(\sum_{p=1}^{P} g_{i,p}\left(\mathbf{w}_{y_i}^T\mathbf{u}_{i,y_i} - \mathbf{w}_p^T\mathbf{u}_{i,p}\right)\right) \quad (15)$$

where set $\mathcal{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_P\}$ with $\mathbf{z}_p$ defined as

$$z_{p,k} = \begin{cases} 1\ , & \text{if } k = p \\ 0\ , & \text{otherwise} \end{cases}, \quad k = 1, \ldots, P. \quad (16)$$

Equation (15) can be reformulated as

$$\sum_{i=1}^{n}\xi_i = \max_{\forall \mathbf{G}\in\mathcal{Z}^n}\left(\sum_{i=1}^{n}\sum_{p=1}^{P} g_{i,p}\left(\mathbf{w}_{y_i}^T\mathbf{u}_{i,y_i} - \mathbf{w}_p^T\mathbf{u}_{i,p}\right)\right) = \xi \quad (17)$$

where $\mathbf{G}$ is defined as $\mathbf{G} = [\mathbf{g}_1, \ldots, \mathbf{g}_n] = \begin{bmatrix} g_{1,1}\cdots g_{n,1} \\ \vdots \ddots \vdots \\ g_{1,P}\cdots g_{n,P} \end{bmatrix}$. Theorem 1 is proved.

B. Proof of the Monotonic Nonincrease of the Training Risk of WOLC-ECOC

Given the coding matrix $\mathbf{M}^{(t)}$, WOLC-ECOC classifier ensemble $\mathcal{C}^{(t)}$, minimum training risk $\mathcal{J}_o^{(t)}$, and optimal weight matrix $\mathbf{W}_o^{(t)}$ of the t-th iteration, where $\mathcal{C}^{(t)} = \{h_1, h_2, \ldots, h_q\}$ with q denoting the code length of the t-th iteration, and

$$\mathcal{J}_o^{(t)} = \min_{\mathbf{W}^{(t)}\in\mathcal{W}^{(t)}} \mathcal{J}^{(t)}\left(\mathbf{W}^{(t)}\right) \quad (18)$$

$$\mathbf{W}_o^{(t)} = \arg\min_{\mathbf{W}^{(t)}\in\mathcal{W}^{(t)}} \mathcal{J}^{(t)}\left(\mathbf{W}^{(t)}\right) \quad (19)$$

with the training risk function $\mathcal{J}^{(t)}\left(\mathbf{W}^{(t)}\right)$ defined in (7). Suppose we get a new dichotomizer h_{q+1} at the $(t+1)$-th iteration, we can obtain $\mathbf{M}^{(t+1)}$, $\mathcal{C}^{(t+1)}$, $\mathcal{J}_o^{(t+1)}$, and $\mathbf{W}_o^{(t+1)}$ in the same way as we did in the t-th iteration, where $\mathcal{C}^{(t+1)} = \mathcal{C}^{(t)} \cup h_{q+1}$ and $\mathbf{M}^{(t+1)} = [\mathbf{M}^{(t)}, \mathbf{m}]$ with $\mathbf{m}$ denoted as the most difficult binary-class problem (in a vector form). We have the following theorem.

Theorem 2: The nonincrease of the training risk of WOLC-ECOC is guaranteed by the OW decoding

$$\mathcal{J}_o^{(0)} \geq \mathcal{J}_o^{(1)} \geq, \ldots, \geq \mathcal{J}_o^{(t)} \geq \mathcal{J}_o^{(t+1)} \geq, \ldots$$

Proof: We extend the optimal weight matrix $\mathbf{W}_o^{(t)}$ to another equivalent form $\mathbf{W}^{(t+1)'} = \left[\mathbf{W}_o^{(t)}, \mathbf{0}_{P\times 1}\right]$. It is easy to know that $\mathbf{W}^{(t+1)'} \in \mathcal{W}^{(t+1)}$. Because, $\mathbf{W}^{(t+1)'}$ yields an objective value that is equivalent to $\mathcal{J}_o^{(t)}$, and also because $\mathbf{W}^{(t+1)'}$ is a point in $\mathcal{W}^{(t+1)}$ and problem (7) is a convex optimization problem with $\mathcal{J}_o^{(t+1)}$ as its minimum value, the inequality $\mathcal{J}_o^{(t)} \geq \mathcal{J}_o^{(t+1)}$ holds. Theorem 2 is proved. ■

ACKNOWLEDGMENT

The author would like to thank the editors and the anonymous referees for their valuable advice, and would also like to thank the researchers who opened the codes of their excellent works.

REFERENCES

[1] T. Dietterich, "Ensemble methods in machine learning," in *Proc. MCS*, Cagliari, Italy, 2000, pp. 1–15.
[2] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits Syst. Mag.*, vol. 6, no. 3, pp. 21–45, Sep. 2006.
[3] A. Rahman and B. Verma, "Novel layered clustering-based approach for generating ensemble of classifiers," *IEEE Trans. Neural Netw.*, vol. 22, no. 5, pp. 781–792, May 2011.
[4] M. Re and G. Valentini, *Ensemble Methods: A Review*. London, U.K.: Chapman & Hall, 2011.
[5] P. Yang *et al.*, "Sample subset optimization techniques for imbalanced and ensemble learning problems in bioinformatics applications," *IEEE Trans. Cybern.*, vol. 44, no. 3, pp. 445–455, Mar. 2014.
[6] M. Pérez-Ortiz, P. A. Gutiérrez, and C. Hervás-Martínez, "Projection-based ensemble learning for ordinal regression," *IEEE Trans. Cybern.*, vol. 44, no. 5, pp. 681–694, May 2014.
[7] D. Vazquez *et al.*, "Occlusion handling via random subspace classifiers for human detection," *IEEE Trans. Cybern.*, vol. 44, no. 3, pp. 342–354, Mar. 2014.
[8] Y. Han, K. Yang, Y. Ma, and G. Liu, "Localized multiple kernel learning via sample-wise alternating optimization," *IEEE Trans. Cybern.*, vol. 44, no. 1, pp. 137–148, Jan. 2014.
[9] K. Leung, F. Cheong, and C. Cheong, "Generating compact classifier systems using a simple artificial immune system," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 5, pp. 1344–1356, Oct. 2007.
[10] Y. Xu, X. Cao, and H. Qiao, "An efficient tree classifier ensemble-based approach for pedestrian detection," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 1, pp. 107–117, Feb. 2011.
[11] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.
[12] R. E. Schapire, "The strength of weak learnability," *Mach. Learn.*, vol. 5, no. 2, pp. 197–227, 1990.
[13] G. B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 513–529, Apr. 2012.
[14] S. Wang and X. Yao, "Multiclass imbalance problems: Analysis and potential solutions," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 4, pp. 1119–1130, Aug. 2012.
[15] X. L. Zhang and J. Wu, "Linearithmic time sparse and convex maximum margin clustering," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 6, pp. 1–24, Dec. 2012.
[16] T. G. Dietterich and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes," *J. Aritif. Intell. Res.*, vol. 2, no. 1, pp. 263–286, 1995.
[17] E. Tapia, J. González, A. Hütermann, and J. García, "Beyond boosting: Recursive ECOC learning machines," in *Proc. MCS*, Cagliari, Italy, 2004, pp. 62–71.
[18] E. Tapia, P. Bulacio, and L. Angelone, "Recursive ECOC classification," *Pattern Recognit. Lett.*, vol. 31, no. 3, pp. 210–215, 2010.
[19] G. Fung and O. Mangasarian, "Multicategory proximal support vector machine classifiers," *Mach. Learn.*, vol. 59, no. 1, pp. 77–97, 2005.
[20] J. Weston and C. Watkins, "Support vector machines for multi-class pattern recognition," in *Proc. 7th Eur. Symp. Artif. Neural Netw.*, Apr. 1999, pp. 219–224.
[21] Y. Guermeur, "Combining discriminant models with new multi-class SVMs," *Pattern Anal. Appl.*, vol. 5, no. 2, pp. 168–179, 2002.
[22] L. Yoonkyung, L. Yi, and W. Grace, "Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data," *J. Amer. Statist. Assoc.*, vol. 99, no. 465, pp. 67–81, 2004.

[23] P. Chen, K. Y. Lee, T. J. Lee, Y. J. Lee, and S. Y. Huang, "Multiclass support vector classification via coding and regression," *Neurocomput.*, vol. 73, nos. 7–9, pp. 1501–1512, 2010.
[24] S. Ghorai, A. Mukherjee, and P. K. Dutta, "Discriminant analysis for fast multiclass data classification through regularized kernel function approximation," *IEEE Trans. Neural Netw.*, vol. 21, no. 6, pp. 1020–1029, Jun. 2010.
[25] G. Zhong, K. Huang, and C. Liu, "Learning ECOC and dichotomizers jointly from data," in *Proc. ICONIP*, Sydney, NSW, Australia, 2010, pp. 494–502.
[26] O. Pujol *et al*., "Discriminant ECOC: A heuristic method for application dependent design of error correcting output codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 6, pp. 1007–1012, Jun. 2006.
[27] O. Pujol, S. Escalera, and P. Radeva, "An incremental node embedding technique for error correcting output codes," *Pattern Recognit.*, vol. 41, no. 2, pp. 713–725, 2008.
[28] S. Escalera, D. M. J. Tax, O. Pujol, P. Radeva, and R. P. W. Duin, "Subclass problem-dependent design for error-correcting output codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 6, pp. 1041–1054, Jun. 2008.
[29] D. Bouzas, N. Arvanitopoulos, and A. Tefas, "Optimizing linear discriminant error correcting output codes using particle swarm optimization," in *Proc. ICANN*, Espoo, Finland, 2011, pp. 79–86.
[30] L. I. Kuncheva and C. J. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," *Mach. Learn.*, vol. 51, no. 2, pp. 181–207, 2003.
[31] L. I. Kuncheva, "Using diversity measures for generating error-correcting output codes in classifier ensembles," *Pattern Recognit. Lett.*, vol. 26, no. 1, pp. 83–90, 2005.
[32] S. Escalera, O. Pujol, and P. Radeva, "Separability of ternary codes for sparse designs of error-correcting output codes," *Pattern Recognit. Lett.*, vol. 30, no. 3, pp. 285–297, 2009.
[33] S. Escalera, O. Pujol, and P. Radeva, "Recoding error-correcting output codes," in *Proc. MCS*, 2009, pp. 11–21.
[34] M. Prior and T. Windeatt, "Over-fitting in ensembles of neural network classifiers within ECOC frameworks," in *Proc. MCS*, Seaside, CA, USA, 2005, pp. 286–295.
[35] S. Escalera, O. Pujol, and P. Radeva, "Boosted landmarks of contextual descriptors and forest-ECOC: A novel framework to detect and classify objects in cluttered scenes," *Pattern Recognit. Lett.*, vol. 28, no. 13, pp. 1759–1768, 2007.
[36] M. Bautista *et al*., "Compact design of ECOC for multi-class object categorization," in *Proc. 5th CVCRD*, 2010, pp. 54–57.
[37] M. Bautista *et al*., "Compact evolutive design of error-correcting output codes," in *Proc. ECML*, 2010, pp. 119–128.
[38] S. Escalera, O. Pujol, and P. Radeva, "ECOC-ONE: A novel coding and decoding strategy," in *Proc. 18th ICPR*, Hong Kong, 2006, pp. 578–581.
[39] M. A. Bagheri, G. Montazer, and E. Kabir, "A subspace approach to error correcting output codes," *Pattern Recognit. Lett.*, vol. 34, no. 2, pp. 176–184, 2012.
[40] M. A. Bagheri, Q. Gao, and S. Escalera, "Rough set subspace error-correcting output codes," in *Proc. 12th ICDM*, 2012, pp. 822–827.
[41] A. Passerini, M. Pontil, and P. Frasconi, "New results on error correcting output codes of kernel machines," *IEEE Trans. Neural Netw.*, vol. 15, no. 1, pp. 45–54, Jan. 2004.
[42] E. L. Allwein, R. E. Schapire, and Y. Singer, "Reducing multiclass to binary: A unifying approach for margin classifiers," *J. Mach. Learn. Res.*, vol. 1, pp. 113–141, Dec. 2001.
[43] S. Escalera, O. Pujol, and P. Radeva, "On the decoding process in ternary error-correcting output codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 120–134, Jan. 2010.
[44] J. E. Kelley, "The cutting-plane method for solving convex programs," *J. Soc. Ind. Appl. Math.*, vol. 8, no. 4, pp. 703–712, 1960.
[45] T. Joachims, "Training linear SVMs in linear time," in *Proc. 12th ACM Int. Conf. KDD*, Philadelphia, PA, USA, 2006, pp. 226–235.
[46] C. H. Teo, A. Smola, S. V. N. Vishwanathan, and Q. V. Le, "A scalable modular convex solver for regularized risk minimization," in *Proc. 13th ACM SIGKDD Int. Conf. KDD*, San Jose, CA, USA, 2007, pp. 727–736.
[47] V. Franc and S. Sonnenburg, "Optimized cutting plane algorithm for support vector machines," in *Proc. 25th ICML*, New York, NY, USA, 2008, pp. 320–327.
[48] C. W. Hsu and C. J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 415–425, Mar. 2002.
[49] F. Masulli and G. Valentini, "Effectiveness of error correcting output codes in multiclass learning problems," in *Proc. MCS*, Cagliari, Italy, 2000, pp. 107–116.
[50] R. Rifkin and A. Klautau, "In defense of one-vs-all classification," *J. Mach. Learn. Res.*, vol. 5, pp. 101–141, Jan. 2004.
[51] K. Crammer and Y. Singer, "On the learnability and design of output codes for multiclass problems," *Mach. Learn.*, vol. 47, no. 2, pp. 201–233, 2002.
[52] A. J. Smola, S. V. N. Vishwanathan, and T. Hofmann, "Kernel methods for missing variables," in *Proc. 10th Int. Workshop Artif. Intell. Statist.*, 2005, pp. 325–332.
[53] A. L. Yuille and A. Rangarajan, "The concave-convex procedure," *Neural Comput.*, vol. 15, no. 4, pp. 915–936, 2003.
[54] J.-B. Yang and I. W. Tsang, "Hierarchical maximum margin learning for multi-class classification," in *Proc. 27th Conf. UAI*, 2011, pp. 753–760.
[55] N. Hatami, "Thinned-ECOC ensemble based on sequential code shrinking," *Expert Syst. Appl.*, vol. 39, no. 1, pp. 936–947, 2011.
[56] J. D. Zhou, X. D. Wang, and H. Song, "Research on the unbiased probability estimation of error-correcting output coding," *Pattern Recognit.*, vol. 44, no. 7, pp. 1552–1565, 2011.
[57] T. Kajdanowicz, M. Wozniak, and P. Kazienko, "Multiple classifier method for structured output prediction based on error correcting output codes," in *Proc. ACIIDS*, Daegu, Korea, 2011, pp. 333–342.
[58] S. Escalera, D. Masip, E. Puertas, P. Radeva, and O. Pujol, "Adding classes online in error correcting output codes framework," in *Proc. 20th ICPR*, Istanbul, Turkey, 2010, pp. 2945–2948.
[59] S. Escalera, D. Masip, E. Puertas, P. Radeva, and O. Pujol, "Online error correcting output codes," *Pattern Recognit. Lett.*, vol. 32, no. 3, pp. 458–467, 2010.
[60] C. Marrocco, P. Simeone, and F. Tortorella, "Embedding reject option in ECOC through LDPC codes," in *Proc. MCS*, Prague, Czech Republic, 2007, pp. 333–343.
[61] P. Simeone, C. Marrocco, and F. Tortorella, "Design of reject rules for ECOC classification systems," *Pattern Recognit.*, vol. 45, no. 2, pp. 863–875, 2012.
[62] B. Verma and A. Rahman, "Cluster oriented ensemble classifier: Impact of multi-cluster characterisation on ensemble classifier learning," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 4, pp. 605–618, Apr. 2011.
[63] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2002.
[64] X. T. Yuan, B. G. Hu, and R. He, "Agglomerative mean-shift clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 2, pp. 209–219, Feb. 2012.
[65] P. Li, J. Bu, C. Chen, Z. He, and D. Cai, "Relational multimanifold coclustering," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1871–1881, May 2013.
[66] T. Joachims, T. Finley, and C. N. J. Yu, "Cutting-plane training of structural SVMs," *Mach. Learn.*, vol. 77, no. 1, pp. 27–59, 2009.
[67] T. Joachims and C. N. J. Yu, "Sparse kernel SVMs via cutting-plane training," *Mach. Learn.*, vol. 76, no. 2, pp. 179–193, 2009.
[68] C. W. Hsu, C. C. Chang, and C. J. Lin. (2003). *A Practical Guide to Support Vector Classification* [Online]. Available: http://www.csie.ntu.edu.tw/˜cjlin/papers/guide/guide.pdf
[69] S. Escalera, O. Pujol, and P. Radeva, "Error-correcting ouput codes library," *J. Mach. Learn. Res.*, vol. 11, pp. 661–664, Feb. 2010.
[70] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Proc. 13th ICML*, 1996, pp. 148–156.
[71] T. Li and M. Ogihara, "Toward intelligent music information retrieval," *IEEE Trans. Multimedia*, vol. 8, no. 3, pp. 564–574, Jun. 2006.
[72] H. Homburg, I. Mierswa, B. Möller, K. Morik, and M. Wurst, "A benchmark dataset for audio classification and clustering," in *Proc. ISMIR*, 2005, pp. 528–531.
[73] C. H. Lee, J. L. Shih, K. M. Yu, and H. S. Lin, "Automatic music genre classification based on modulation spectral analysis of spectral and cepstral features," *IEEE Trans. Multimedia*, vol. 11, no. 4, pp. 670–682, Jun. 2009.

Xiao-Lei Zhang (S'08–M'12) received the Ph.D. degree in information and communication engineering from Tsinghua University, Beijing, China, in 2012.

From 2013 to 2014, he was a Visiting Scholar with the Department of Computer Science and Engineering, Ohio State University, Columbus, OH, USA. His current research interests include machine learning, computational audition, statistical signal processing, and bioinformatics.