Joint Binary Classifier Learning for ECOC-Based Multi-Class Classification

Mingxia Liu, Daoqiang Zhang, Songcan Chen, and Hui Xue

Abstract—Error-correcting output coding (ECOC) is one of the most widely used strategies for dealing with multi-class problems by decomposing the original multi-class problem into a series of binary sub-problems. In traditional ECOC-based methods, binary classifiers corresponding to those sub-problems are usually trained separately without considering the relationships among these classifiers. However, as these classifiers are established on the same training data, there may be some inherent relationships among them. Exploiting such relationships can potentially improve the generalization performances of individual classifiers, and, thus, boost ECOC learning algorithms. In this paper, we explore to mine and utilize such relationship through a joint classifier learning method, by integrating the training of binary classifiers and the learning of the relationship among them into a unified objective function. We also develop an efficient alternating optimization algorithm to solve the objective function. To evaluate the proposed method, we perform a series of experiments on eleven datasets from the UCI machine learning repository as well as two datasets from real-world image recognition tasks. The experimental results demonstrate the efficacy of the proposed method, compared with state-of-the-art methods for ECOC-based multi-class classification.

Index Terms—Multi-class classification, error-correcting output coding (ECOC), (joint) binary classifier learning, relationship.

1 Introduction

Multi-class classification is an important issue in many pattern recognition and machine learning domains [1], [2]. Currently, there are two main lines to solve multi-class learning problems, including "direct multi-class representation" and "(indirect) decomposition design" [3]. The first line aims to design multi-class classifiers directly, such as decision tree [4], neural network [5], and multi-class support vector machines (SVM) [6], etc. In contrast, the second line decomposes the original multi-class problem into multiple binary sub-problems that can be solved by binary classification algorithms and then combine the results of these classifiers for final prediction. As a typical indirect decomposition way to deal with multi-class problems, ECOC [3] has been used widely, and is the very focus of this paper.

In general, there are three major components for ECOC-based multi-class learning methods, i.e., encoding, binary classifier learning, and decoding steps [3]. In the encoding procedure, a coding matrix is usually first determined for multiple classes, where each row in the coding matrix represents a specific class. Then, a group of (supposedly) independent binary classifiers is trained based on a different partition of the original data according to each column of the coding matrix. Finally, a new instance is predicted as a specific class through the decoding procedure based on the outputs of the learned binary classifiers and the coding matrix.

- *M. Liu is with the School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China and is also with the School of Information Science and Technology, Taishan University, Tai'an 271021, China. E-mail: mingxialiu@nuaa.edu.cn.*
- *D. Zhang and S. Chen are with the School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China. E-mail: {dqzhang, s.chen}@nuaa.edu.cn.*
- *H. Xue is with the School of Computer Science and Engineering, Southeast University, Nanjing 210096, China. E-mail: hxue@seu.edu.cn.*

Manuscript received 12 Dec. 2013; revised 28 Jan. 2015; accepted 22 Apr. 2015. Date of publication 12 May 2015; date of current version 10 Oct. 2016.
Recommended for acceptance by M. Belkin.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2015.2430325

In traditional ECOC-based methods, multiple binary classifiers are usually directly taken from many existing classifiers (e.g., SVM, AdaBoost) and trained separately [3], [7], [8], [9], [10], [11], implying that the inherent relationships among these classifiers are ignored. However, as these classifiers usually share the same pool of training data and the same learning algorithm, there may be some relationships among classifiers. For example, Fig. 1 illustrates the pair-wise linear correlation coefficients among weight vectors of SVM binary classifiers using one-versus-all (OVA) encoding on *Vehicle* dataset from UCI machine learning repository [12]. Here, red and yellow in the color bar denote positive correlation coefficients, while blue and green denote negative ones. From Fig. 1, one can see that, although these binary classifiers are trained separately, they are actually highly correlated. Intuitively, exploiting such relationships can improve the learning performances of base classifiers in ECOC-based methods.

In this work, we explore to mine and utilize the inherent relationships among binary classifiers to boost the performances of ECOC-based methods. To be specific, we first propose a joint binary classifier learning (JCL) method, where the training of binary classifiers for ECOC and the learning of the relationships among these classifiers are formulated into a unified objective function. Then, we develop an efficient alternating optimization algorithm, as well as a kernel extension for the proposed JCL model. Experimental results validate our intuition that exploiting the relationships among binary classifiers can boost the performances of ECOC-based multi-class learning methods.

The remainder of the paper is organized as follows. Section 2 introduces some background knowledge and related works of ECOC. In Section 3, we describe the proposed method, the optimization algorithm, and the kernel extension in detail. Experimental results and discussions are given in Sections 4 and 5, respectively. Finally, conclusions are given Section 6.

2 Backgrounds

Currently, there are various methods to decompose the original multi-class problem into several binary sub-problems [3], [13]. The simplest and straightforward way is one-versus-all (OVA) [14], where C binary classifiers (also called as dichotomizers) are generated and each class is compared with all the other ones. Note that C is the class number in this paper. Then, a new instance is predicted as a class with the maximum classification score among all corresponding binary classifiers. The one-versus-one (OVO) strategy compares all pair of classes, where $C(C\text{-}1)/2$ binary sub-problems are generated [15]. The prediction of a new instance is performed by voting of all corresponding binary classifiers. Dietterich and Bakiri [3] proposed a general (binary) ECOC framework, where each class is given an L-length error correcting output coding with each component valued from $\{-1,+1\}$. Accordingly, the coding matrix can be constructed with each row representing a coding for a specific class. After training L binary classifiers with respect to each column of the coding matrix, we can predict a new instance and get an L-length output code. Then, the instance is classified as the "closest" class, measured by Hamming distance between the output code and individual class coding. Allwein et al. [16] extended the general binary ECOC framework to allow the coding matrix to have zero components, called ternary ECOC framework. Then each element of the coding matrix is chosen from $\{-1, 0, +1\}$ where classes with zero values are not considered for that particular dichotomizer. With this technique, classical OVA and OVO encoding strategies can be unified into the common ECOC framework.

In recent years, the encoding and decoding strategies for ECOC (i.e., the construction of coding matrix and the prediction of new instances) have attracted much attention [8], [17], [18], [19], [20], [21], [22]. For the encoding procedure, there are several standard

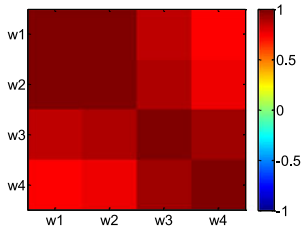


Fig. 1. Pair-wise linear correlation coefficients among weight vectors of SVM binary classifiers using OVA encoding strategy on *Vehicle* dataset.

binary coding designs (e.g., OVA [14], and dense random strategy [8]), and ternary coding designs (e.g. OVO [15], and sparse random strategy [16]). Meanwhile, many problem-dependent coding designs have been proposed to adapt to the learning problem at hand. For example, Pujol et al. [8] developed a discriminant ECOC (DECOC) encoding in favor of class discrimination in the class-set partitions. Escalera et al. [23] proposed an extension of Discriminant ECOC, called Forest ECOC, by taking advantage of the tree structure representation of ECOC methods. Escalera et al. [9] modeled complex problems by splitting the original set of classes into sub-classes, and embedding the binary problems in a sub-class ECOC design. Hatami [24] proposed a thinned-ECOC encoding method. Recently, researchers in [21], [22] proposed to design coding matrix using evolutionary algorithms in the ECOC framework. Meanwhile, much effort has been taken on the decoding strategy. Hastie and Tibshirani [15] presented a Bradley-Terry model-based decoding method, which was extended into ternary coding scheme by Zadrozny [25]. Escalera et al. [26] designed two decoding strategies for ECOC, i.e., linear loss-weighted (LLW) and exponential loss-weighted (ELW) decoding methods. Takashi and Ishii [18] developed a ternary AdaBoost method, as well as a ternary Bradley-Terry model-based decoding method.

On the other hand, comparing to the large amount of the literature for the encoding and the decoding processes, less attention has been paid to the design of binary classifiers for ECOC. Traditionally, binary classifiers are taken directly from existing classifiers, e.g., AdaBoost [8], [18] and SVM [20], [27], [28], which are trained separately without considering the inherent relationship among them. Recently, a few works have been proposed to jointly learn the coding matrix and base classifiers. For instance, Pujol et al. [29] presented an ECOC optimizing node embedding (ECOCone) approach. Zhong et al. [19] developed the Joint ECOC (JECOC) algorithm to learn the coding matrix and binary classifiers jointly from data. However, the main disadvantage of these methods is that they do not consider the underlying relationships among binary classifiers in ECOC-based methods. Intuitively, exploiting such relationships can improve the performances of individual classifiers, and, thus, boost the performances of ECOC-based learning methods. In this work, we explore to mine and utilize such relationships via a joint binary classifier learning method. It is worth noting that in a recent work, Gao and Koller [1] exploited the hierarchical structure in the label space by simultaneously learning the binary classifiers organized in a tree or directed acyclic graph structure, which is different from ECOC that treats the label space as flat.

3 The Proposed Joint Binary Classifier Learning Model

In this section, we first introduce the notation in Section 3.1, and then present the proposed JCL model in Section 3.2. The optimization algorithm and the kernel extension for the proposed method are described in Sections 3.3 and 3.4, respectively.

3.1 Notation

Denote C as the number of classes and D as the feature dimension of an instance. Let $\mathbf{P} \in \{-1, 0, +1\}^{C \times L}$ denote the coding matrix under a specific ECOC encoding strategy, where L is the length of the coding for a specific class. Based on each column of $\mathbf{P}$, each of L binary classifiers will be constructed respectively. To be specific, if $\mathbf{P}_{cl} = 1$ (or –1), the data points associated with the c-th class will be regarded as the positive (or negative) class for the l-th binary classifier. Meanwhile, if $\mathbf{P}_{cl} = 0$, data points associated with class c will not be used to construct the l-th classifier.

Denote $\mathbf{x}_i^l \in \mathbf{R}^D$ as the i-th instance in the l-th binary classification problem. For each of L binary classifiers, we want to learn a linear function $f_l(\mathbf{x}_i^l) = \mathbf{w}_l^T \mathbf{x}_i^l + b_l$, where $\mathbf{w}_l$ is the weight vector and b_l is the bias term, respectively.

3.2 Objective Function

Follow the notation in the previous sub-section, and denote $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_L] \in \mathbf{R}^{D \times L}$ where $\mathbf{w}_i \in \mathbf{R}^D (i = 1, \ldots, L)$. Let $\mathbf{b} = [b_1, \ldots, b_L]^T$ be the bias term, and N_l represent the number of instances in the l-th classification problem. Given the training dataset $\mathbf{X}^l = [x_1^l, \ldots, x_{N_l}^l]$ and their corresponding class label $\mathbf{y}^l = [y_1^l, \ldots, y_{N_l}^l]$ for the l-th binary classifier, we propose the following optimization problem to learn these classifiers jointly:

$$\min_{\mathbf{W}} \sum_{l=1}^{L} \sum_{i=1}^{N_l} loss(y_i^l, f_l(\mathbf{x}_i^l)) + \lambda_1 \sum_{l=1}^{L} V(\mathbf{w}_l), \tag{1}$$

where the term $loss(y_i^l, f_l(\mathbf{x}_i^l))$ is a loss function that measures the mismatch between y_i^l and the predicted value $f_l(\mathbf{x}_i^l)$, $V(\mathbf{w}_l)$ is a regularizer that controls the complexity of the weight vector $\mathbf{w}_l$, and λ_1 is a regularization parameter to tune the tradeoff between the empirical loss and the regularization term. In fact, the model defined in Eq. (1) is a very general framework for joint binary classifier learning, as one can utilize various loss functions (e.g., least squares loss and hinge loss) and various regularizers (e.g., l_1-norm regularizer, and l_2-norm regularizer) to adapt to the problems at hand. However, it is not the focus of this work. For simplicity, we adopt the least squares loss function and the l_2-norm regularizer in this paper. Thus, the optimization problem defined in Eq. (1) can be rewritten as follows:

$$\begin{aligned} \min_{\mathbf{W},\mathbf{b},\{\rho_i^l\}} \; & \sum_{l=1}^{L} \sum_{i=1}^{N_l} (\rho_i^l)^2 + \frac{\lambda_1}{2} tr(\mathbf{W}\mathbf{W}^T) \\ s.t \quad & y_i^l(\mathbf{w}_l^T \mathbf{x}_i^l + b_l) = 1 - \rho_i^l, \forall i, l, \end{aligned} \tag{2}$$

where ρ_i^l is a slack variable.

Motivated by the work in Ref. [30], we resort to the column covariance matrix of $\mathbf{W}$ to model the relationships among $\mathbf{w}_l$'s. Accordingly, we get the following objective function:

$$\begin{aligned} \min_{\mathbf{W},\mathbf{b},\mathbf{M},\{\rho_i^l\}} \; & \sum_{l=1}^{L} \sum_{i=1}^{N_l} (\rho_i^l)^2 + \frac{\lambda_1}{2} tr(\mathbf{W}\mathbf{W}^T) + \frac{\lambda_2}{2} tr(\mathbf{W}\mathbf{M}^{-1}\mathbf{W}^T) \\ s.t. \quad & y_i^l(\mathbf{w}_l^T \mathbf{x}_i^l + b_l) = 1 - \rho_i^l, \forall i, l \\ & \mathbf{M} \geq 0, \quad tr(\mathbf{M}) = 1, \end{aligned} \tag{3}$$

where λ_1 and λ_2 are two regularization parameters, and $\mathbf{M}$ is the column covariance matrix of the weight matrix $\mathbf{W}$. It is worth noting that the last term $tr(\mathbf{W}\mathbf{M}^{-1}\mathbf{W}^T)$ is used to model the correlation relation among multiple binary classifiers, where the inverse covariance matrix $\mathbf{M}^{-1}$ plays a role of coupling pairs of weight vectors. The term $tr(\mathbf{M}) = 1$ in the constraint is used to further penalize the complexity of $\mathbf{W}$, and the constraint $\mathbf{M} \geq 0$ is used to restrict $\mathbf{M}$ as positive semi-definite because it denotes the covariance matrix.

In addition, due to the use of a certain ECOC encoding strategy, there may be different number of instances in different binary

classifiers. To avoid the data imbalance problem where one binary classifier with many instances dominates the empirical loss, we use weights that are inversely proportional to the number of instances for training these binary classifiers [30], [31], and reformulate the problem in Eq. (3) as follows:

$$\begin{aligned} \underset{\mathbf{W},\mathbf{b},\mathbf{M},\{\rho_i^l\}}{min} \; & \sum_{l=1}^{L}\frac{1}{N_l}\sum_{i=1}^{N_l}(\rho_i^l)^2+\frac{\lambda_1}{2}tr\big(\mathbf{W}\mathbf{W}^T\big)+\frac{\lambda_2}{2}tr\big(\mathbf{W}\mathbf{M}^{-1}\mathbf{W}^T\big) \\ s.t. \quad & y_i^l\big(\mathbf{w}_l^T\mathbf{x}_i^l+b_l\big)=1-\rho_i^l,\forall i,l \\ & \mathbf{M}\geq 0, \quad tr(\mathbf{M})=1. \end{aligned} \tag{4}$$

which is called JCL model in this paper.

It is worth noting that the proposed model is different from the support vector machine (SVM) [32], although they use the same loss function. The main difference is that traditional SVM binary classifiers for ECOC are trained separately ignoring the relationships among base classifiers, while the proposed method aims to exploit such relationships among SVM-like classifiers through a joint learning scheme to improve the performances of individual classifiers. Also, the proposed JCL model is different from standard classifier ensembles. A key difference is that each binary classifier in the proposed model solves a different two-class problem whereas in standard classifier ensembles all binary classifiers solve the same (possibly multi-class) problem [3].

3.3 Alternating Optimization Algorithm

It is easy to find that the proposed model in Eq. (4) is jointly convex with respect to $\mathbf{W}$, $\mathbf{b}$ and $\mathbf{M}$. Following the work in [30], we adopt an alternating optimization algorithm to solve the problem in Eq. (4). The first step is to optimize $\mathbf{W}$ and $\mathbf{b}$ given a fixed $\mathbf{M}$, and the second step is to optimize $\mathbf{M}$ when $\mathbf{W}$ and $\mathbf{b}$ are fixed.

3.3.1 *Optimize* $\mathbf{W}$ *and* $\mathbf{b}$ *when* $\mathbf{M}$ *is fixed*

Given a fixed $\mathbf{M}$, the optimization problem defined in Eq. (4) can be expressed in the following form:

$$\begin{aligned} \underset{\mathbf{W},\mathbf{b},\{\rho_i^l\}}{min} \; & \sum_{l=1}^{L}\frac{1}{N_l}\sum_{i=1}^{N_l}\big(\rho_i^l\big)^2+\frac{\lambda_1}{2}tr\big(\mathbf{W}\mathbf{W}^T\big)+\frac{\lambda_2}{2}tr\big(\mathbf{W}\mathbf{M}^{-1}\mathbf{W}^T\big) \\ s.t. \quad & y_i^l\big(\mathbf{w}_l^T x_i^l+b_l\big)=1-\rho_i^l,\forall i,l. \end{aligned} \tag{5}$$

The Lagrangian of problem defined in Eq. (5) can be written as follows:

$$\begin{aligned} G=&\sum_{l=1}^{L}\frac{1}{N_l}\sum_{i=1}^{N_l}\big(\rho_i^l\big)^2+\frac{\lambda_1}{2}tr\big(\mathbf{W}\mathbf{W}^T\big)+\frac{\lambda_2}{2}tr\big(\mathbf{W}\mathbf{M}^{-1}\mathbf{W}^T\big) \\ &-\sum_{l=1}^{L}\sum_{i=1}^{N_l}\alpha_i^l[y_i^l\big(\mathbf{w}_l^T\mathbf{x}_i^l+b_l\big)-1+\rho_i^l]. \end{aligned} \tag{6}$$

After calculating the gradient of G with respect to $\mathbf{W}$, b_l and ρ_i^l, and setting them to 0, we get the following equations:

$$\begin{aligned} &\frac{\partial G}{\partial \mathbf{W}}=\mathbf{W}\big(\lambda_1\mathbf{I}_L+\lambda_2\mathbf{M}^{-1}\big)-\sum_{l=1}^{L}\sum_{i=1}^{N_l}\alpha_i^l y_i^l\mathbf{x}_i^l\big(\mathbf{e}_i^l\big)^T=0 \\ &\Rightarrow \mathbf{W}=\sum_{l=1}^{L}\sum_{i=1}^{N_l}\alpha_i^l y_i^l\mathbf{x}_i^l\big(\mathbf{e}_i^l\big)^T(\lambda_1\mathbf{I}_L+\lambda_2\mathbf{M}^{-1})^{-1} \end{aligned} \tag{7}$$

$$\frac{\partial G}{\partial b_l}=-\sum_{i=1}^{N_l}\alpha_i^l y_i^l=0\Rightarrow\sum_{i=1}^{N_l}\alpha_i^l y_i^l=0 \tag{8}$$

$$\frac{\partial G}{\partial \rho_i^l}=\frac{2}{N_l}\rho_i^l-\alpha_i^l=0\Rightarrow\rho_i^l=\frac{N_l}{2}\alpha_i^l, \tag{9}$$

where $\mathbf{e}_l$ is the l-th column vector of $L\times L$ identity matrix $\mathbf{I}_L$. Combining Eqs. (7), (8), (9) into the constraint in Eq. (5), we obtain the following linear system:

$$\begin{pmatrix}\mathbf{K}+\frac{1}{2}\Lambda & \mathbf{Q}_{12}\\ \mathbf{Q}_{21} & 0_{L\times L}\end{pmatrix}\begin{pmatrix}\alpha\\ \mathbf{b}\end{pmatrix}=\begin{pmatrix}1_{N\times 1}\\ 0_{L\times 1}\end{pmatrix}, \tag{10}$$

where the matrix Λ is diagonal with elements value N_l if the corresponding instance belongs to the l-th binary sub-problem, and $\alpha=[\alpha_1^1,\ldots\alpha_{N_1}^1,\ldots,\alpha_1^L,\ldots,\alpha_{N_L}^L]^T$. Note that $\mathbf{K}$ is a kernel matrix on all instances for all binary classifiers, with element defined as follows:

$$k(\mathbf{x}_{i1}^{l1},\mathbf{x}_{i2}^{l2})=[\mathbf{e}_{l1}^T(\lambda_1\mathbf{I}_L+\lambda_2\mathbf{M}^{-1})^{-1}\mathbf{e}_{l2}][y_{i1}^{l1}y_{i2}^{l2}][(\mathbf{x}_{i1}^{l1})^T\mathbf{x}_{i2}^{l2}]. \tag{11}$$

Given the label vector $\mathbf{y}^l$ of training data in the l-th classification problem, $\mathbf{Q}_{12}$ and $\mathbf{Q}_{21}$ are expressed in the following form:

$$\mathbf{Q}_{12}=\mathbf{Q}_{21}^T=\begin{bmatrix}\mathbf{y}^1 & 0 & 0\\ 0 & \ddots & 0\\ 0 & 0 & \mathbf{y}^L\end{bmatrix}. \tag{12}$$

If the total number of instances for all binary sub-problems is very large, solving the linear system directly in Eq. (10) usually requires much computational cost. In such cases, we can use another optimization method to solve the proposed objective function. It is easy to know that the dual form of Eq. (5) can be formulated as

$$\begin{aligned} \underset{\alpha}{min} \quad & \frac{1}{2}\alpha^T\Big(\mathbf{K}+\frac{1}{2}\Lambda\Big)\alpha-\sum_{l=1}^{L}\sum_{i=1}^{N_i}\alpha_i^l \\ s.t. \quad & \sum_{i=1}^{N}\alpha_i^l y_i^l=0,\forall l, \end{aligned} \tag{13}$$

which can be solved efficiently by the sequential minimal optimization (SMO) algorithm [32].

3.3.2 *Optimize* $\mathbf{M}$ *When* $\mathbf{W}$ *and* $\mathbf{b}$ *Are Fixed*

If $\mathbf{W}$ and $\mathbf{b}$ are fixed, the problem in Eq. (4) can be reformulated as the following:

$$\begin{aligned} \underset{\mathbf{M}}{min} \quad & tr\big(\mathbf{W}\mathbf{M}^{-1}\mathbf{W}^T\big) \\ s.t. \quad & \mathbf{M}\geq 0, tr(\mathbf{M})=1. \end{aligned} \tag{14}$$

Then we can get an analytical solution for Eq. (14) [30], which is:

$$\mathbf{M}=\frac{(\mathbf{W}^T\mathbf{W})^{\frac{1}{2}}}{tr((\mathbf{W}^T\mathbf{W})^{\frac{1}{2}})}. \tag{15}$$

The previous two steps are performed alternatively, until the optimization procedure converges or the maximal iteration number is reached.

After learning the optimal solutions of $\mathbf{W}$, $\mathbf{b}$ and $\mathbf{M}$, the prediction for test instances can be performed. Given a test instance $\mathbf{z}$ of the l-th sub-problem, its label can be predicted through the following:

$$f_l(\mathbf{z})=sign\Big(\sum_{i=1}^{N_l}\alpha_i^l y_i^l k\big(\mathbf{x}_i^l,\mathbf{z}\big)+b_l\Big). \tag{16}$$

3.4 Kernel Extension

In the above sub-section, we only consider the linear case of the proposed JCL method. Now we will provide a non-linear extension

TABLE 1
UCI DataSets Used in the Experiments

Dataset	#class	#instance	#feature
Balance	3	625	4
Cmc	3	1,473	9
Tae	3	151	5
Wine	3	178	13
Thyroid	3	215	5
Vehicle	4	846	18
Dermatology	6	366	33
Glass	6	214	10
Zoo	7	101	17
Ecoli	8	336	8
Vowel	11	990	10

of our JCL model. The optimization problem for the kernel extension is the same as the problem defined in Eq. (4), with the only difference being that the data point $\mathbf{x}_i^l$ is mapped to $\Phi(\mathbf{x}_i^l)$ in some reproducing kernel Hilbert space where $\Phi(\cdot)$ denotes the feature mapping. At the same time, the corresponding kernel function $k(\cdot,\cdot)$ satisfies $k(x_1,x_2)=\Phi(x_1)^T\Phi(x_2)$.

Similar to the linear version, we can also use an alternating method to solve the optimization problem. In the first step, we use the non-linear kernel for data points from different base classifiers $k_{NL}(\mathbf{x}_{i1}^{l1},\mathbf{x}_{i2}^{l2})=[\mathbf{e}_{l1}^T(\lambda_1\mathbf{I}_L+\lambda_2\mathbf{M}^{-1})^{-1}\mathbf{e}_{l2}][y_{i1}^{l1}y_{i2}^{l2}]k(\mathbf{x}_{i1}^{l1},\mathbf{x}_{i2}^{l2})$. The rest is the same as the linear case. In the second step, the change is required in the calculation of $\mathbf{W}^T\mathbf{W}$ as $\mathbf{W}$ is in the form of Eq. (7). In this way, we have $\mathbf{W}^T\mathbf{W}=\sum_{p,q}\sum_{i,j}\alpha_p^q\alpha_i^j y_p^q y_i^j\ k_{NL}(\mathbf{x}_p^q,\mathbf{x}_i^j)$, $(\lambda_1\mathbf{M}+\lambda_2\mathbf{I}_L)^{-1}\mathbf{M}\mathbf{e}_p(\mathbf{e}_i)^T\mathbf{M}(\lambda_1\mathbf{M}+\lambda_2\mathbf{I}_L)^{-1}$. We omit the details here since it is similar to the calculation of above liner kernel version.

4 Experiments

To evaluate the efficacy of the proposed method, we perform a series of experiments on eleven multi-class UCI datasets [12] and two datasets from real-world image recognition tasks. We first introduce the datasets and the experimental setup in Sections 4.1 and 4.2, respectively. Then, the experimental results and analysis are given in Sections 4.3 and 4.4.

4.1 Datasets

First, we evaluate our proposed method on eleven datasets from UCI machine learning repository [12] by performing classification experiments. The characteristics of these UCI datasets are shown in Table 1.

Furthermore, we also evaluate the proposed method on two multi-class image recognition datasets, including texture images and brain images.

Texture. The texture image dataset used in this paper is a collection of texture images from the well-known Outex datasets [33]. Ten texture classes are included in our experiments, including *wood*, *crack*, *brick*, *carpet*, *fur*, *knit*, *pebble*, *upholstery*, *water*, and *glass*. We randomly choose 40 images for each class and obtain 400 samples for ten classes. To obtain the low-level features, we adopt the non-subsampled contourlet transform [34] with a 4-level decomposition structure, and then compute the mean and the variance of each coefficient matrix in the 4-th level as features. Therefore, we obtain 34 features for an original texture image.

Alzheimer's Disease (AD). The data used in the preparation of this paper were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (www.adni.loni.usc.edu). There are three kinds of subjects, including (1) Alzheimer's disease (AD) subjects, (2) Mild Cognitive Impairment (MCI) subjects, and (3) Healthy Control (HC) subjects. In this paper, only ADNI subjects with the Magnetic Resonance Imaging (MRI) baseline data are included. This yields 830 subjects, including 198 AD patients, 403 MCI patients, and 229 healthy controls. Image pre-processing (i.e., de-noising, registration, and segmentation) is performed for all MR images, with details described in [35]. After the pre-processing stage, for each of the 93 Region of Interest (ROI) regions in the labeled MR images, we compute the volume of gray matter tissue in that ROI region as a feature. For each subject, we totally obtain 93 features from the MRI images.

4.2 Experimental Setup

In the experiments, we apply the proposed JCL method to several state-of-the-art ECOC encoding designs, including OVO [15], OVA [14], DECOC [8], and Forest ECOC (Forest) [23]. The parameters of the coding strategies are the predefined or the default values given by the authors. At the same time, two state-of-the-art decoding methods, i.e., Hamming distance (HD) [3] and linear loss-weight (LLW) [26], are used in the experiments. For all the encoding and the decoding computations, we resort to the ECOC library toolbox [36] as a platform. We first compare the proposed JCL method with traditional independent base classifier learning methods (e.g., SVM), and the corresponding experimental results are shown in Section 4.3.

Furthermore, we compare the proposed method with two joint learning algorithms for ECOC (i.e., ECOCone [29] and JECOC [19]), with results given in Section 4.4. It is worth noting that both ECOCone and JECOC only jointly learn encoding matrices and individual binary classifier without considering the relationship among binary base classifiers, which is significantly different from JCL.

In the experiments, we adopt a five-fold cross-validation strategy to compute the mean and the variance of classification accuracy. The regularization parameters C for SVM are selected from $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\}$ through five-fold cross-validation on the training data. The parameters λ_1 and λ_2 for the proposed JCL model are chosen from $\{10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$ through five-fold cross-validation on the training data. Note that both linear kernel and Radial Basis Function (RBF) kernel are adopted for SVM and JCL methods. The bandwidth for RBF kernel is selected by five-fold cross-validation from $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\}\times\sigma$, where σ is the average distance between training instances.

4.3 Comparison with Independent Base Classifier Learning Methods

We first compare our proposed JCL method with independent base classifier learning method (i.e., SVM) for ECOC-based multi-class classification. In Table 2, we report the performances of JCL and SVM with linear kernel using four state-of-the-art encoding and the LLW decoding strategies. Here, the best results are in boldface, and the entries in the bracket are the variances of the accuracies among five-fold cross validation. We also carry out the paired t-test with significance level 95 percent to compare different methods. In addition, we further show the win/tie/lose counts (with significance level 95 percent) of JCL over the other methods in the rows named "W/T/L".

From Table 2, one can observe that, compared with SVM, the proposed JCL method achieves the overall best average accuracies, no matter what encoding strategies are used. Specifically, JCL consistently and significantly outperforms SVM on both *Texture* and *AD* datasets, and achieves better performances than SVM in most cases on UCI datasets. For example, JCL significantly outperforms SVM on 11 out of 13 datasets using OVA encoding strategy. This indicates our proposed JCL method, which exploits the inherent relationships among base classifiers, can help improve the classification performances of ECOC-based multi-class learning methods.

TABLE 2
Results Based on LLW Decoding Strategy Using Linear Kernel (%)

Dataset	OVO		OVA		DECOC		Forest	
	SVM	JCL	SVM	JCL	SVM	JCL	SVM	JCL
Balance	87.69(0.03)	**88.41(0.04)**	**91.67(0.02)**	88.02(0.04)	**88.04(0.03)**	**88.04(0.03)**	87.87(0.29)	**89.22(0.23)**
Cmc	52.89(0.03)	**53.43(0.04)**	48.67(0.05)	**52.47(0.18)**	50.24(0.08)	**50.38(0.07)**	**50.24(0.79)**	49.83(0.24)
Tae	**54.64(0.41)**	51.64(0.21)	48.00(0.46)	**54.64(0.47)**	50.04(0.57)	**53.26(0.72)**	50.04(0.57)	**51.72(0.79)**
Wine	98.82(0.02)	**99.52(0.01)**	97.75(0.15)	**97.76(0.05)**	98.46(0.02)	**98.93(0.05)**	98.93(0.02)	**99.26(0.02)**
Thyroid	**91.16(0.06)**	88.83(0.49)	88.37(0.14)	**89.31(0.07)**	89.30(0.02)	**90.70(0.22)**	**88.83(0.04)**	86.05(0.36)
Vehicle	78.25(0.08)	**80.22(0.02)**	75.37(0.04)	**78.58(0.01)**	**77.39(0.03)**	76.98(0.02)	77.09(0.03)	**77.39(0.12)**
Dermatology	96.78(0.04)	**97.74(0.03)**	96.09(0.04)	**98.30(0.01)**	96.69(0.03)	**97.50(0.01)**	95.50(0.03)	**96.77(0.10)**
Glass	65.22(0.07)	**67.35(0.40)**	61.61(0.25)	**65.74(0.48)**	63.48(0.48)	**64.85(0.17)**	62.50(0.48)	**64.48(0.14)**
Zoo	92.22(0.72)	**93.63(0.39)**	93.22(0.40)	**94.32(0.49)**	87.06(1.09)	**88.51(0.57)**	**93.63(1.09)**	91.67(1.02)
Ecoli	**87.66(0.16)**	85.89(0.12)	77.64(0.41)	**85.77(0.14)**	84.57(0.07)	**85.98(0.29)**	84.57(0.07)	**85.08(0.02)**
Vowel	47.84(0.16)	**49.49(0.10)**	42.12(0.64)	**47.07(0.12)**	55.25(0.13)	**56.36(0.06)**	55.25(0.14)	**56.34(0.08)**
Texture	82.25(0.06)	**84.36(0.04)**	73.00(0.04)	**77.50(0.03)**	71.25(0.14)	**72.56(0.13)**	71.52(0.21)	**74.85(0.16)**
AD	55.61(0.30)	**60.97(0.12)**	53.42(0.25)	**59.76(0.18)**	53.30(0.26)	**54.57(0.02)**	52.25(0.15)	**55.56(0.08)**
Average	76.23(0.16)	**77.04(0.15)**	72.84(0.22)	**76.10(0.17)**	74.24(0.23)	**75.28(0.18)**	74.48(0.30)	**75.25(0.26)**
W/T/L	8/2/3	-	11/1/1	-	8/5/0	-	8/3/2	-

TABLE 3
Comparison with Joint Learning Methods (%)

Dataset	ECOCone	$SVM_{ECOCone}$	$JCL_{ECOCone}$	JECOC	SVM_{JECOC}	JCL_{JECOC}
Balance	**89.62(0.03)**	87.06(0.02)	88.74(0.04)	**91.68(0.02)**	87.69(0.03)	88.41(0.05)
Cmc	48.89(0.05)	45.07(0.04)	**49.67(0. 09)**	52.27(0.05)	52.89(0.03)	**55.22(0.05)**
Tae	40.75(0.98)	46.62(0.43)	**47.48(0.23)**	52.00(0.35)	52.64(0.41)	**53.44(0.43)**
Wine	93.75(0.10)	96.58(0.14)	**98.52(0.03)**	96.58(0.06)	98.22(0.03)	**99.68(0.01)**
Thyroid	**92.55(0.06)**	86.04(0.10)	88.40(0.10)	**95.34(0.09)**	91.16(0.06)	93.14(0.15)
Vehicle	65.45(0.11)	75.38(0.09)	**75.95(0.01)**	78.72(0.03)	78.34(0.08)	**80.45(0.02)**
Dermatology	64.93(2.46)	95.81(0.05)	**96.19(0.87)**	96.06(0.07)	95.78(0.03)	**96.83(0.03)**
Glass	63.15(1.77)	62.74(0.47)	**65.00(0.20)**	64.22(0.12)	65.22(0.07)	**68.75(0.23)**
Zoo	78.54(2.92)	**88.21(1.19)**	87.50(0.32)	94.44(0.21)	94.22(0.10)	**95.83(0.07)**
Ecoli	65.85(2.37)	80.89(0.56)	**81.93(1.08)**	87.39(0.12)	**87.66(0.16)**	87.41(0.02)
Vowel	32.52(0.52)	49.29(0.44)	**49.91(0.40)**	67.98(0.21)	64.84(0.15)	**78.65(0.10)**
Average	66.90(1.03)	73.97(0.32)	**75.39(0.31)**	79.69(0.12)	78.95(0.10)	**81.91(0.11)**
W/T/L	8/2/1	8/3/0	-	7/2/2	8/3/0	-

In the online supplementary material, we also report the experimental results using RBF kernel based on LLW decoding method, with the similar trend as using linear kernel in Table 2. In addition, we have performed experiments using another decoding strategy (i.e., HD) and also obtained significant performance improvements, with results reported in the online supplementary material. Moreover, the comparison of JCL with another two methods for base classifier learning is also shown in the online supplementary material.

4.4 Comparison with Joint Learning Methods

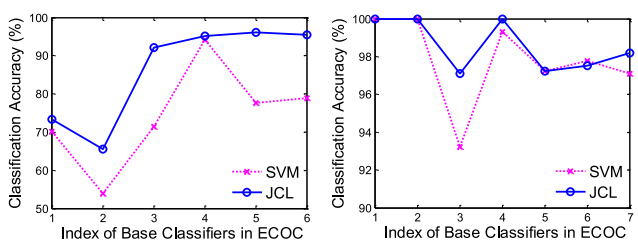
In this sub-section, we compare the proposed JCL method with two existing joint learning based ECOC methods (i.e., ECOCone [29] and JECOC [19]), which learn the coding matrix and the binary classifiers simultaneously. The corresponding experimental results using LLW decoding strategy on UCI datasets are shown in Table 3. Here, $JCL_{ECOCone}$ and $SVM_{ECOCone}$ denote JCL and SVM using the coding matrix learned from ECOCone, respectively. Similarly, JCL_{JECOC} and SVM_{JECOC} denote JCL and SVM using the coding matrix learned from JECOC, respectively.

From Table 3, one can see that, on most datasets, the proposed $JCL_{ECOCone}$ and JCL_{JECOC} methods outperform ECOCone and JECOC, respectively. Especially, the improvement of $JCL_{ECOCone}$ over ECOCone in terms of average accuracy is most significant. In addition, Table 3 shows that in most cases $JCL_{ECOCone}$ and JCL_{JECOC} outperform $SVM_{ECOCone}$ and SVM_{JECOC}, respectively. These results further validate the efficiency of JCL as a general base classifier learning method for ECOC-based multi-class classification.

5 Discussions

It is important to analyze the possible reason for the advantage of our proposed JCL method over traditional methods. Accordingly, in this section, we perform two extra groups of experiments. Specifically, in the first group of experiments, we investigate the classification accuracies achieved by all binary classifiers. In the second one, we compute the number of bit-errors (simultaneous errors) committed by binary classifiers on each test data point, following the work in [37]. The OVA encoding and HD decoding strategies are used in these experiments. The corresponding results of two groups of experiments are shown in Figs. 2 and 3, respectively.

From Fig. 2, one can see that the classification accuracies achieved by multiple binary classifiers in JCL are usually higher than those of SVM on both the *Glass* and the *Zoo* datasets.



(a) *Glass* (b) *Zoo*

Fig. 2. Classification accuracies of multiple binary classifiers on *Glass* and *Zoo datasets*.

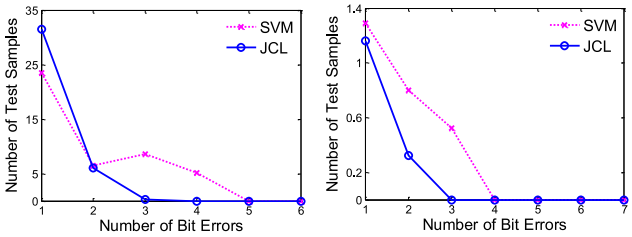


Fig. 3. Error distribution of multiple binary classifiers on *Glass* and *Zoo datasets*.

Specifically, the average accuracies on the *Glass* dataset achieved by JCL and SVM are 86.28 and 74.36 percent, respectively, and those of JCL and SVM on the *Zoo* dataset are 98.58 and 97.81 percent, respectively. The underlying reason for the advantage of JCL over SVM could be that, as we do via the regularization terms in the JCL model in Eq. (4), exploiting the relationships among multiple binary classifiers can help improve the learning performance of individual classifiers.

On the other hand, it can be observed from Fig. 3 that the simultaneous multi-bits errors made by the proposed JCL method are relatively much less than those of SVM on both *Glass* and *Zoo* datasets, although the one-bit errors made by JCL are higher than those of SVM on *Glass* dataset. According to [7], ECOC only succeeds if the errors made in the individual bit positions are relatively uncorrelated, so that the number of simultaneous errors in many bit positions is small. Otherwise, ECOC will not be able to correct these errors. The above experimental results demonstrate that the proposed JCL method can produce less simultaneous multi-bits errors than SVM, which can partly explain the reason for the improvement of the reported results achieved by JCL over traditional ones.

In addition, we also show the learned relationships among binary classifiers in the online supplementary material, from which one can see that the relationships learned from the proposed JCL model includes positive, negative and uncorrelated relation. It demonstrates that the proposed method can model rich structure in binary classifiers for ECOC-based methods.

6 Conclusion

In traditional ECOC-based multi-class classification methods, binary classifiers for multiple sub-problems are usually trained separately, which ignores the inherent relationships among binary classifiers. In this work, we explore to mine and utilize such relationships to improve the learning performances of individual classifiers in ECOC-based methods. To be specific, we first propose a JCL method, by formulating the training of binary classifiers and the learning of their relationships into a unified objective function. Then, we develop an alternating optimization algorithm, as well as a kernel extension of the proposed JCL model. Finally, we evaluate the proposed method on eleven UCI datasets and two datasets from real-world multi-class image recognition tasks. The experimental results demonstrate that exploiting the relationships among binary classifiers can promote the generalization performances of individual classifiers, and, thus, boost the learning performances of ECOC-based methods in multi-class classification problems.

In the current work, we lean binary classifier and their relationships jointly, using a given coding matrix. It is interesting to learn the coding matrix, binary classifiers and their relationship simultaneously from data, which will be our future work.

Acknowledgments

The authors would like to thank the editor and the anonymous reviewers for their useful comments and contributions for the improvement of the paper. This work was supported by the National Natural Science Foundation of China (Nos. 61422204, 61473149, 61375057), the Jiangsu Natural Science Foundation for Distinguished Young Scholar (No. BK20130034), the Specialized Research Fund for the Doctoral Program of Higher Education (Nos. 20123218110009, 20133218110032), the NUAA Fundamental Research Funds (No. NE2013105), the Jiangsu Natural Science Foundation (No. BK20131298), and the Jiangsu Qinglan Project of China. D. Zhang is the corresponding author of this paper.

References

[1] T. Gao and D. Koller, "Discriminative learning of relaxed hierarchy for large-scale visual recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 2072–2079, 2011.
[2] K. Crammer and C. Gentile, "Multiclass classification with bandit feedback using adaptive regularization," *Mach. Learning*, vol. 90, pp. 347–383, 2013.
[3] T. G. Dietterich and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes," *J. Artif. Intell. Res.*, vol. 2, pp. 263–286, 1995.
[4] M. M. David, "Statistical decision-tree models for parsing," in *Proc. 33rd Ann. Meeting Assoc. Comput. Linguistics*, pp. 276–283, 1995.
[5] M. T. Hagan, H. B. Demuth, and M. H. Beale, *Neural Network Design, PWS Pub. Co.*, 1996.
[6] T.-F. Wu, C.-J. Lin, and R. C. Weng, "Probability estimates for multi-class classification by pairwise coupling," *J. Mach. Learning Res.*, vol. 5, pp. 975–1005, 2004.
[7] W. W. Peterson and E. J. Weldon, *Error-Correcting Codes*, Cambridge MIT Press, 1972.
[8] O. Pujol, P. Radeva, and J. Vitria, "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.
[9] 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.
[10] M. Cissé, T. Artières, and P. Gallinari, "Learning compact class codes for fast inference in large multi class classification," in *Proc. IEEE Int. Conf. Mach. Learning*, pp. 506–520, 2012.
[11] T. Takenouchi and S. Ishii, "A unified framework of binary classifiers ensemble for multi-class classification," in *Proc. IEEE Int. Conf. Neural Inform. Process.*, vol. 7664, pp. 375–382, 2012.
[12] A. Frank and A. Asuncion, "UCI Machine Learning Repository," ed. Irvine School Inf. Comput. Sci., Univ. California, Berkeley, CA, USA, 2007.
[13] T. Gao and D. Koller, "Discriminative learning of relaxed hierarchy for large-scale visual recognition," in *Proc. IEEE Int. Conf. Comput. Vision*, pp. 2072–2079, 2011.
[14] N. J. Nilsson, *Learning Machines*. New York: McGraw-Hill, 1965.
[15] T. Hastie and R. Tibshirani, "Classification by pairwise coupling," *Annals Stat.*, vol. 26, pp. 451–471, 1998.
[16] E. L. Allwein, R. E. Schapire, Y. Singer, and P. Kaelbling, "Reducing multiclass to binary: A unifying approach for margin classifiers," *J. Mach. Learning Res.*, vol. 1, pp. 113–141, 2001.
[17] J. D. Zhou, X. D. Wang, H. J. Zhou, J. M. Zhang, and N. Jia, "Decoding design based on posterior probabilities in ternary error-correcting output codes," *Pattern Recognit.*, vol. 45, pp. 1802–1818, 2012.
[18] T. Takenouchi and S. Ishii, "Ternary Bradley-Terry model-based decoding for multi-class classification and its extensions," *Mach. Learning*, vol. 85, pp. 249–272, 2011.
[19] G. Q. Zhong, K. Z. Huang, and C. L. Liu, "Joint learning of error-correcting output codes and dichotomizers from data," *Neural Comput. Appl.*, vol. 21, pp. 715–724, 2012.
[20] C. Koby and S. Yoram, "On the learnability and design of output codes for multiclass problems," *Mach. Learning*, vol. 47, pp. 201–233, 2002.
[21] M. Á. Bautista, S. Escalera, X. Baró, and O. Pujol, "On the design of an ecoc-compliant genetic algorithm," *Pattern Recognit.*, vol. 47, pp. 865–884, 2014.
[22] M. Ali Bagheri, Q. Gao, and S. Escalera, "A genetic-based subspace analysis method for improving error-correcting output coding," *Pattern Recognit.*, vol. 46, pp. 2830–2839, 2013.
[23] 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, pp. 1759–1768, 2007.
[24] N. Hatami, "Thinned-ECOC ensemble based on sequential code shrinking," *Expert Syst. Appl.*, vol. 39, pp. 936–947, 2012.
[25] B. Zadrozny, "Reducing multiclass to binary by coupling probability estimates," *Proc. Adv. Neural Inf. Process. Syst.*, pp. 1041–1048, 2001.
[26] 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.
[27] R. Rifkin and A. Klautau, "In defense of one-vs-all classification," *J. Mach. Learning Res.*, vol. 5, pp. 101–141, 2004.
[28] Y. Wang, S. Chen, and H. Xue, "Can under-exploited structure of original-classes help ecoc-based multi-class classification?," *Neurocomputing*, vol. 89, pp. 158–167, 2012.

[29] O. Pujol, S. Escalera, and P. Radeva, "An incremental node embedding technique for error correcting output codes," *Pattern Recognit.*, vol. 41, pp. 713–725, 2008.
[30] Y. Zhang and D.-Y. Yeung, "A convex formulation for learning task relationships in multi-task learning," in *Proc. IEEE Int. Conf. Uncertainty Artif. Intell.*, pp. 733–742, 2010.
[31] T. Evgeniou and M. Pontil, "Regularized multi–task learning," in *Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery Data Mining*, Seattle, WA, USA, 2004, pp. 109–117.
[32] M. A. Hearst, S. Dumais, E. Osman, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intell. Syst. Their Appl.*, vol. 13, no. 4, pp. 18–28, Jul./Aug. 1998.
[33] T. Ojala, T. Mäenpää, M. Pietikäinen, J. Viertola, J. Kyllönen, and S. Huovinen, "Outex-new framework for empirical evaluation of texture analysis algorithms," in *Proc. IEEE Int. Conf. Pattern Recognit.*, 2002, pp. 701–706.
[34] A. L. da Cunha, J. Zhou, and M. N. Do, "The nonsubsampled contourlet transform: Theory, design, and applications," *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 3089–3101, Oct. 2006.
[35] D. Zhang, Y. Wang, L. Zhou, H. Yuan, and D. Shen, "Multimodal classification of Alzheimer's disease and mild cognitive impairment," *NeuroImage*, vol. 55, pp. 856–867, 2011.
[36] S. Escalera, O. Pujol, and P. Radeva, "Error-correcting ouput codes library," *J. Mach. Learning Res.*, vol. 11, pp. 661–664, 2010.
[37] E. B. Kong and T. G. Dietterich, "Error-correcting output coding corrects bias and variance," *Int. Conf. Mach. Learning*, 1995.