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 relationship among classifiers. However, as these classifiers are established on the same training data, there may be some inherent relationship 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 data sets from the UCI machine learning repository as well as two data sets from real-world image recognition tasks. The experimental results demonstrate the efficacy of the proposed method in comparison 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, error-correcting output coding (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 multi-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 set 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.

In traditional ECOC-based methods, multiple binary classifiers are usually directly taken from many existing classifiers (e.g., SVM, AdaBoost) and then separately [3, 7-11], implying that the inherent relationship 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 relationship among classifiers. For instance, Fig. 1 (a) illustrates the pair-wise linear correlation coefficients among weight vectors of SVM base classifiers using one-versus-all (OVA) encoding on vehicle data set 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 relationship among classifiers can improve the generalization performances of binary classifiers.

![Fig. 1. Pair-wise linear correlation coefficients among weight vectors of SVM binary classifiers using OVA encoding strategy on vehicle data set.](image)

In this work, we explore to mine and utilize the inherent relationship among binary classifiers to boost the performances of individual binary classifiers in ECOC. 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 relationship among these classifiers are formulated into a unified objective function. Then, we develop an efficient alternating optimization algorithm to solve the proposed objective function. Experimental results validate our intuition that exploiting the relationship among binary classifiers can boost the performances of ECOC-based multi-class learning algorithms.

The reminder 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 and the corresponding optimization algorithm in detail. Experimental results and discussions are given in Section 4 and Section 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-1)/2 binary sub-problems are generated [15]. The prediction of a new instance is performed by voting of all corresponding binary classifiers. Dietterich et al. [3] proposed a general (binary) error-correcting output codes (ECOC) framework, where each class is given a 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 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, 17-22]. For the encoding procedure, there are several standard 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 instance, Pujol et al. [8] developed the 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 the ECOC method. Escalera et al. [9] modeled complex problems by splitting the original set of classes into sub-classes, and embedding the binary problems in a subclass 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 et al. [18] presented the ternary AdaBoost and the ternary Bradley-Terry model-based decoding methods.

On the other hand, comparing to the large amount of literatures 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, 28, 29], 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. [27] 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, these methods do not consider the relationship...
among base classifiers. Intuitively, exploiting such a relationship can improve the generalization performance of individual classifiers, and thus boost the performances of ECOC-based learning methods. In this work, we explore how to mine and utilize such relationship through a joint binary classifier learning method. It’s 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 notations 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 Section 3.3 and Section 3.4, respectively.

3.1 Notations

Denote $C$ as the number of classes and $D$ as the feature dimension of an instance. Let $P \in \{-1, 0, +1\}^{C \times L}$ 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 $P$, each of $L$ binary classifiers will be constructed respectively. To be specific, if $P_{cl} = 1$ (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 $P_{cl} = 0$, data points associated with class $c$ will not be used to construct the $l$-th classifier.

Denote $x_i \in \mathbb{R}^D$ as the $i$-th instance of $l$-th classification problem. For each of the $L$ binary classifiers, we want to learn a linear function $f(x_i) = w_i^T x_i + b_i$, where $w_i$ is the weight vector and $b_i$ is the bias term, respectively.

3.2 Objective Function

Follow the notations in the previous sub-section, and denote $W = [w_1, w_2, ..., w_L]$ where $w_i \in \mathbb{R}^D$ ($i = 1, ..., L$). Let $b = [b_1, ..., b_L]^T$ and $N_l$ as the number of instances in the $l$-th classification problem. Given the training data set $X_l = [x_{1l}, ..., x_{N_l}]$ and their corresponding class label $y_l = [y_{1l}, ..., y_{N_l}]$ for the $l$-th binary classifier, we propose the following optimization problem to learn these classifiers jointly:

$$\min_W \sum_{l=1}^{L} \sum_{i=1}^{N_l} \text{loss}(y_{ii}, f_l(x_{il}^l)) + \lambda_1 \sum_{l=1}^{L} V(w_l)$$  \hspace{1cm} (1)

where the term $\text{loss}(y_{ii}, f_l(x_{il}^l))$ is a loss function that measures the mismatch between $y_{ii}$ and the predicted value $f_l(x_{il}^l)$, $V(w_l)$ is a regularizer that controls the complexity of weight vector, and $\lambda_1$ 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 classifiers 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 $l_2$-norm regularizer in this paper, and thus the optimization problem defined in Eq. (1) can be rewritten as follows:

$$\min_{W:||\rho||_2 \leq 1} \sum_{l=1}^{L} \sum_{i=1}^{N_l} |y_{ii} - f_l(x_{il}^l)|^2 + \frac{\lambda_2}{2} \text{tr}(WW^T)$$

s.t. $y_{ii}^l = (w_i^T x_{il}^l + b_i) = 1 - \rho_i^l$, $\forall i, l = 1, ..., L$.  \hspace{1cm} (2)

where $\rho_i^l$ is a slack variable.

Following the work in [30], we resort to the column covariance matrix of $W$ to model the relationship among $w_i$'s. Accordingly, we get the following objective function:

$$\min_{W, b, M:||\rho||_2 \leq 1} \sum_{l=1}^{L} \sum_{i=1}^{N_l} |y_{ii}^l|^2 + \frac{\lambda_1}{2} \text{tr}(WW^T) + \frac{\lambda_2}{2} \text{tr}(WM^{-1}W^T)$$

s.t. $y_{ii}^l = (w_i^T x_{il}^l + b_i) = 1 - \rho_i^l$, $\forall i, l = 1, ..., L$

$M \succeq 0$, $\text{tr}(M) = 1$.  \hspace{1cm} (3)

where $\lambda_1$ and $\lambda_2$ are two regularization parameters, and $M$ is the column covariance matrix of the weight matrix $W$. It’s worth noting that the last term $\text{tr}(WM^{-1}W^T)$ is used to model the correlation relation among individual binary classifiers, where the inverse covariance matrix $M^{-1}$ plays a role of coupling pairs of weight vectors. The term $\text{tr}(M) = 1$ in the constraint is used to further penalize the complexity of $W$, and the constraint $M \succeq 0$ is used to restrict $M$ as positive semi-definite because it denotes the covariance matrix.

Furthermore, to avoid the data imbalance problem where one binary classifier with so many instances dominates the empirical loss, we reformulate the problem defined in Eq. (3) as follows:

$$\min_{W, b, M:||\rho||_2 \leq 1} \sum_{l=1}^{L} \frac{1}{N_l} \sum_{i=1}^{N_l} |y_{ii}^l|^2 + \frac{\lambda_1}{2} \text{tr}(WW^T) + \frac{\lambda_2}{2} \text{tr}(WM^{-1}W^T)$$

s.t. $y_{ii}^l = (w_i^T x_{il}^l + b_i) = 1 - \rho_i^l$, $\forall i, l = 1, ..., L$

$M \succeq 0$, $\text{tr}(M) = 1$.  \hspace{1cm} (4)

which is called joint binary classifier learning (JCL) model in this paper.

It’s worth noting that the proposed model is different from support vector machine (SVM) [31], although they use the same loss function. The main difference is that traditional SVM binary classifiers for ECOC are trained separately ignoring the relationship among classifiers, while the proposed method aims to exploit such relationship 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].

It is easy to find that the proposed model in Eq. (4) is jointly convex with respect to $W, b$ and $M$. For reasons of efficiency, an alternating optimization algorithm is
developed to solve the objective function, with more
details described in the following sub-section.

3.3 Alternating Optimization Algorithm

In Eq. (4), there are three variables to be optimized, which
are jointly convex. In what follows, we will present an
efficient alternating optimization algorithm for solving it.

The first step is to optimize $W$ and $b$ given a fixed $M$,
and the second step is to optimize $M$ when $W$ and $b$
are fixed.

(1) Optimize $W$ and $b$ when $M$ is fixed

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

$$
\min_{W,b,\rho_i} \sum_{l=1}^N \frac{1}{2} \|y_l - \langle Wx_l, b \rangle \|^2 + \frac{1}{2} \rho_i \|W\|_F^2 + \frac{1}{2} \|W - M\|_F^2
$$

s.t. $y_l(x_l, b) = 1 - \rho_i$, $\forall l, l = 1, \ldots, L$  

(5)

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

$$
G = \sum_{l=1}^N \frac{1}{2} \alpha_i \|y_l(x_l, b) - 1 + \rho_i\|
$$

After calculating the gradient of $G$ with respect to $W,
b_i$, and $\rho_i$, and setting them to 0, we get the following
equations:

$$
\frac{\partial G}{\partial W} = 0 \\
\frac{\partial G}{\partial b_i} = 0 \\
\frac{\partial G}{\partial \rho_i} = 0
$$

(6)

where $\alpha_i$ is the $i$-th column vector of $L \times L$ identity
matrix $I$. Combining Eqs. (7) - (9) into the constraint in
Eq. (5), we obtain the following linear system:

$$
\begin{pmatrix}
I & \frac{1}{2} \Lambda \\
\frac{1}{2} \Lambda & 0 \end{pmatrix}
\begin{pmatrix}
\alpha_L \cr \alpha_1 \end{pmatrix} = \begin{pmatrix}
\frac{1}{2} \Lambda \cr 0 \end{pmatrix}
$$

(10)

where the matrix $\Lambda$ is diagonal with elements value $N_l$
if the corresponding instance belongs to the $l$-th binary
sub-problem, and $\alpha = [\alpha_1, \ldots, \alpha_n, \ldots, \alpha_n]$. Note
that $K$ is a kernel matrix on all instances for binary
classifiers, with element defined as follows:

$$
k(x_l, x_l') = \langle \Phi(x_l), \Phi(x_l') \rangle
$$

Given the label vector $y_l$ of training data in the $l$-th
classification problem, $Q_{12}$ and $Q_{21}$ are expressed in the
following:

$$
Q_{12} = Q_{21} = \begin{pmatrix}
y_l & 0 & 0 \\
0 & \cdots & 0 \\
0 & 0 & y_l
\end{pmatrix}
$$

(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

$$
\min_{\alpha} \frac{1}{2} \alpha^T \left( K + \frac{1}{2} \Lambda \right) \alpha - \sum_{l=1}^N \alpha_i \|y_l\|
$$

s.t. $\sum_{i=1}^N \alpha_i = 0, \forall i, l = 1, \ldots, L$  

(13)

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

(2) Optimize $M$ when $W$ and $b$ are fixed

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

$$
\min_M tr(W^T W)
$$

s.t. $M \geq 0, tr(M) = 1$  

(14)

Denote $A = W^T W$, and then we have

$$
tr(M^T A) = tr((M - A)^T A) tr(M - A) \geq \frac{1}{2} tr((M - A)^2)
$$

(15)

The first inequality holds because of the last constraint
$tr(M) = 1$ in Eq. (14). The last inequality holds because of the
Cauchy-Schwarz inequality for the Frobenius norm.
In addition, $tr(M^T A)$ attains its minimum value if and only if
$M^T A = aM$ for some constant $a$ and $tr(M) = 1$. Then we can get an analytical solution for Eq. (14), which is:

$$
M = \frac{W^T W}{tr(W^T W)}
$$

(16)

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 $W$, $b$, and $M$,
the prediction for test instances can be performed. Given a
test instance $z$ of the $l$-th sub-problem, its label can be
predicted through the following:

$$
f_l(z) = \text{sign} \left( \sum_{i=1}^N \alpha_i y_l \Phi(x_l) z_i + b_i \right)
$$

(17)

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 of the proposed algorithm.

The optimization problem for the kernel extension is the
same as problem (4), with the only difference being that
the data point $x_l$ is mapped to $\Phi(x_l)$ in some
reproducing kernel Hilbert space where $\Phi(\cdot)$ denotes
the feature mapping. And the corresponding kernel function
$k(\cdot, \cdot)$ satisfies $k(x_l, x_2) = \Phi(x_l)^T \Phi(x_2)$.  

Similar to the linear case, 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 $k_{\phi}(x_l, x_l') = \phi(x_l)^T \phi(x_l')$. The rest is the same as the
linear case. In the second step, the change is required in
the calculation of $W^T W$ as in the form of Eq. (7). In
this way, we have $W^T W = \sum_{l=1}^L \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_l y_j k_{\phi}(x_i, x_j)$.
\((\lambda_1 \mathbf{M} + \lambda_2 \mathbf{I}_2)^{-1} \mathbf{M} (\mathbf{e}_i)^T \mathbf{M} (\lambda_1 \mathbf{M} + \lambda_2 \mathbf{I}_2)^{-1} \). We omit the details here since it is similar to the calculation of above linear kernel version.

4 Experiments

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

4.1 Data sets

First, we perform classification experiments on eleven data sets from UCI machine learning repository [12]. Characteristics of these data sets are shown in Table 1.

<table>
<thead>
<tr>
<th>Data set</th>
<th>#class</th>
<th>#Instance</th>
<th>#feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Balance</td>
<td>3</td>
<td>823</td>
<td>4</td>
</tr>
<tr>
<td>Cmc</td>
<td>3</td>
<td>1473</td>
<td>9</td>
</tr>
<tr>
<td>Tae</td>
<td>3</td>
<td>151</td>
<td>5</td>
</tr>
<tr>
<td>Wine</td>
<td>3</td>
<td>178</td>
<td>13</td>
</tr>
<tr>
<td>Thyroid</td>
<td>3</td>
<td>215</td>
<td>5</td>
</tr>
<tr>
<td>Vehicle</td>
<td>4</td>
<td>846</td>
<td>18</td>
</tr>
<tr>
<td>Dermatology</td>
<td>6</td>
<td>366</td>
<td>33</td>
</tr>
<tr>
<td>Glass</td>
<td>6</td>
<td>214</td>
<td>10</td>
</tr>
<tr>
<td>Zoo</td>
<td>7</td>
<td>101</td>
<td>17</td>
</tr>
<tr>
<td>Ecoli</td>
<td>8</td>
<td>336</td>
<td>8</td>
</tr>
<tr>
<td>Vowel</td>
<td>11</td>
<td>990</td>
<td>10</td>
</tr>
</tbody>
</table>

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

Texture. The texture data set used in this paper is a collection of texture images from the well-known Outex data sets [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). Three kinds of subjects are included, which are (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 preprocessing, 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 [27] and JECCOC [19]), with results given in Section 4.4. It’s worth noting that both ECOCone and JECCOC 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 5-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 5-fold cross-validation on the training data. The parameters \(\lambda_1\) and \(\lambda_2\) for the proposed JCL model are chosen from \(\{10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}\}\) through 5-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 5-fold cross-validation from \(\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\} \times \sigma\), where \(\sigma\) 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 5-fold cross validation. We also carry out the paired \(t\)-test with significance level 95\% to compare different methods. We further show the
From Table 2, one can observe that the proposed JCL method achieves the overall best average accuracies compared to SVM, no matter what encoding strategies are used. Specifically, JCL consistently and significantly outperforms SVM on both image recognition data sets (i.e., Texture and AD), and achieves better performances than SVM in most cases on UCI data sets. For example, JCL significantly outperforms SVM on 11 out of 13 data sets using OVA encoding method. This indicates our proposed JCL method, which exploits the inherent relationship among base classifiers, can help improve the classification performances of ECOC-based multi-class learning methods.

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 other decoding strategy (i.e., HD) and also obtained significant performance improvements, with detailed results reported in the online supplementary material. Moreover, the comparison of JCL with two other base classifier learning methods 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 [27] and JECOC [19]) that learn the coding matrix and base classifiers simultaneously. The corresponding experimental results using LLW decoding strategy on UCI data sets 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.

### Table 3: Comparison with Joint Learning Methods (%)

<table>
<thead>
<tr>
<th>Data set</th>
<th>ECOCone</th>
<th>SVM_ECOCone</th>
<th>JCL_ECOCone</th>
<th>JECOC</th>
<th>SVM_JECOC</th>
<th>JCL_JECOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Balance</td>
<td>89.62(0.03)</td>
<td>87.06(0.02)</td>
<td>88.74(0.04)</td>
<td>91.68(0.02)</td>
<td>87.69(0.03)</td>
<td>88.41(0.05)</td>
</tr>
<tr>
<td>Cmc</td>
<td>48.89(0.05)</td>
<td>45.07(0.04)</td>
<td>49.67(0.09)</td>
<td>52.27(0.05)</td>
<td>52.89(0.03)</td>
<td>55.22(0.05)</td>
</tr>
<tr>
<td>Taue</td>
<td>40.75(0.08)</td>
<td>46.62(0.03)</td>
<td>47.48(0.23)</td>
<td>50.0(0.05)</td>
<td>52.64(0.41)</td>
<td>53.44(0.43)</td>
</tr>
<tr>
<td>Wine</td>
<td>93.75(0.10)</td>
<td>96.88(0.14)</td>
<td>98.52(0.03)</td>
<td>96.58(0.06)</td>
<td>98.22(0.03)</td>
<td>99.68(0.01)</td>
</tr>
<tr>
<td>Thyroid</td>
<td>92.55(0.06)</td>
<td>86.04(0.10)</td>
<td>88.40(0.10)</td>
<td>95.34(0.09)</td>
<td>91.16(0.06)</td>
<td>93.14(0.15)</td>
</tr>
<tr>
<td>Vehicle</td>
<td>65.45(0.11)</td>
<td>75.38(0.09)</td>
<td>95.95(0.01)</td>
<td>78.72(0.03)</td>
<td>78.34(0.08)</td>
<td>80.45(0.02)</td>
</tr>
<tr>
<td>Dermatology</td>
<td>64.93(2.46)</td>
<td>95.81(0.05)</td>
<td>96.19(0.87)</td>
<td>96.06(0.07)</td>
<td>95.78(0.03)</td>
<td>96.83(0.03)</td>
</tr>
<tr>
<td>Glass</td>
<td>63.15(1.77)</td>
<td>62.74(0.47)</td>
<td>65.00(0.20)</td>
<td>64.22(0.12)</td>
<td>65.22(0.07)</td>
<td>68.75(0.25)</td>
</tr>
<tr>
<td>Zoo</td>
<td>78.54(2.92)</td>
<td>88.21(1.19)</td>
<td>87.50(0.32)</td>
<td>94.44(0.21)</td>
<td>94.22(0.10)</td>
<td>95.83(0.07)</td>
</tr>
<tr>
<td>Ecoli</td>
<td>65.85(2.37)</td>
<td>80.89(0.56)</td>
<td>81.93(1.08)</td>
<td>87.39(0.12)</td>
<td>87.66(1.06)</td>
<td>87.41(0.02)</td>
</tr>
<tr>
<td>Vowel</td>
<td>32.52(0.52)</td>
<td>49.29(0.44)</td>
<td>49.91(0.40)</td>
<td>67.98(0.21)</td>
<td>64.84(0.15)</td>
<td>78.65(0.10)</td>
</tr>
<tr>
<td>Average</td>
<td>76.23(0.16)</td>
<td>77.04(0.35)</td>
<td>72.84(0.22)</td>
<td>76.10(0.17)</td>
<td>74.24(0.23)</td>
<td>75.28(0.18)</td>
</tr>
<tr>
<td>W/T/L</td>
<td>8/2/3</td>
<td>-</td>
<td>11/1/1</td>
<td>-</td>
<td>8/5/0</td>
<td>8/3/2</td>
</tr>
</tbody>
</table>

---

*Table 2: Results based on LLW decoding strategy using linear kernel (%)*

<table>
<thead>
<tr>
<th>Data set</th>
<th>SVM</th>
<th>JCL</th>
<th>SVM</th>
<th>JCL</th>
<th>SVM</th>
<th>JCL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Balance</td>
<td>87.69(0.03)</td>
<td>86.41(0.04)</td>
<td>91.67(0.02)</td>
<td>88.02(0.04)</td>
<td>88.04(0.03)</td>
<td>88.41(0.03)</td>
</tr>
<tr>
<td>Cmc</td>
<td>52.89(0.03)</td>
<td>53.43(0.04)</td>
<td>48.67(0.05)</td>
<td>52.47(0.18)</td>
<td>50.24(0.08)</td>
<td>50.38(0.07)</td>
</tr>
<tr>
<td>Taue</td>
<td>54.64(0.41)</td>
<td>51.64(0.21)</td>
<td>48.00(0.46)</td>
<td>54.64(0.47)</td>
<td>50.04(0.57)</td>
<td>53.62(0.72)</td>
</tr>
<tr>
<td>Wine</td>
<td>98.82(0.02)</td>
<td>99.52(0.01)</td>
<td>97.75(0.15)</td>
<td>97.76(0.05)</td>
<td>98.46(0.02)</td>
<td>98.93(0.05)</td>
</tr>
<tr>
<td>Thyroid</td>
<td>91.16(0.06)</td>
<td>88.33(0.49)</td>
<td>88.37(0.14)</td>
<td>89.51(0.07)</td>
<td>89.30(0.02)</td>
<td>90.70(0.22)</td>
</tr>
<tr>
<td>Vehicle</td>
<td>78.25(0.08)</td>
<td>80.22(0.02)</td>
<td>75.37(0.04)</td>
<td>78.58(0.01)</td>
<td>77.39(0.03)</td>
<td>76.98(0.02)</td>
</tr>
<tr>
<td>Dermatology</td>
<td>96.78(0.04)</td>
<td>97.74(0.03)</td>
<td>96.09(0.04)</td>
<td>98.30(0.01)</td>
<td>96.69(0.03)</td>
<td>97.50(0.01)</td>
</tr>
<tr>
<td>Glass</td>
<td>65.22(0.07)</td>
<td>67.35(0.40)</td>
<td>61.61(0.25)</td>
<td>65.74(0.48)</td>
<td>63.48(0.48)</td>
<td>64.85(0.17)</td>
</tr>
<tr>
<td>Zoo</td>
<td>92.22(0.72)</td>
<td>93.63(0.39)</td>
<td>93.22(0.40)</td>
<td>94.32(0.49)</td>
<td>87.06(1.09)</td>
<td>88.51(0.57)</td>
</tr>
<tr>
<td>Ecoli</td>
<td>87.66(0.16)</td>
<td>85.89(0.12)</td>
<td>77.64(0.41)</td>
<td>85.77(0.14)</td>
<td>84.57(0.07)</td>
<td>85.98(0.29)</td>
</tr>
<tr>
<td>Vowel</td>
<td>47.84(0.16)</td>
<td>49.49(0.10)</td>
<td>42.12(0.64)</td>
<td>47.07(0.12)</td>
<td>55.25(0.13)</td>
<td>56.36(0.06)</td>
</tr>
<tr>
<td>Texture</td>
<td>82.29(0.06)</td>
<td>84.36(0.04)</td>
<td>73.00(0.04)</td>
<td>77.50(0.03)</td>
<td>71.25(0.14)</td>
<td>72.56(0.13)</td>
</tr>
<tr>
<td>AD</td>
<td>55.61(0.30)</td>
<td>60.97(0.12)</td>
<td>53.42(0.20)</td>
<td>59.76(0.18)</td>
<td>53.30(0.26)</td>
<td>54.57(0.02)</td>
</tr>
</tbody>
</table>
From Table 3, one can see that, on most data sets, the proposed JCL\textsubscript{ECOCone} and JCL\textsubscript{ECOC} methods outperform ECOCone and JECOC, respectively. Especially, the improvement of JCL\textsubscript{ECOCone} over ECOCone in terms of average accuracy is most significant. In addition, Table 3 shows that in most cases JCL\textsubscript{ECOCone} and JCL\textsubscript{ECOC} outperform SVM\textsubscript{ECOCone} and SVM\textsubscript{ECOC}, 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 the proposed 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 of all binary classifiers, and 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 [38]. The OVA encoding and HD decoding strategies are used in the experiments. The corresponding results of two groups of experiments are shown in Fig. 2 and Fig. 3, respectively.

As can be seen from Fig. 2, one can see that the classification accuracies binary classifiers in JCL are usually higher than those of SVM on both the Glass and the Zoo data sets. Specifically, the average accuracies on the Glass data set achieved by JCL and SVM are 86.28% and 74.36%, respectively, and those of JCL and SVM on the zoo data set are 98.58% and 97.81%, respectively. It indicates that exploiting the relationship among classifiers can improve the classification accuracies of individual binary 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 data sets, although the one-bit errors made by JCL are higher than those of SVM on Glass data set. According to [7], ECOC only succeed 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.

Fig. 3. Error distribution of binary classifier on Glass and Zoo.

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

6 Conclusions

In traditional ECOC-based methods, binary classifiers for multiple sub-problems are usually trained separately, where the inherent relationship among classifiers is ignored. In this work, we explore to mine and utilize such relationship to improve the generalization performances of individual classifiers. To be specific, we first propose a joint binary classifier learning (JCL) method, by formulating the training of binary classifiers and the learning of their relationship into a unified objective function. Then, an efficient alternating optimization algorithm is developed to solve the proposed objective function. Finally, we evaluate the proposed method on eleven UCI data sets and two data sets from real-world multi-class image recognition tasks. The experimental results demonstrate that exploiting the relationship among binary classifiers can promote the generalization performances of individual classifiers and thus boost the performances of ECOC-based learning algorithms.

In the current work, we lean binary classifier and their relationship 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.

Acknowledgment

The authors wish to express their gratitude to the editor and all the referees for their useful comments and contribution for the improvement of the paper. This work was supported by the Jiangsu Natural Science Foundation for Distinguished Young Scholar (No. BK20130034), the
Specialized Research Fund for the Doctoral Program of Higher Education (Nos. 201232318110009 and 201332318110032), the NUAA Fundamental Research Funds (No. NE20130105), the Fundamental Research Funds for the Central Universities of China (No. NZ20133306), the Jiangsu Natural Science Foundation (No. BK20131298), and the National Natural Science Foundation of China under Grant 61375057 and 61072148.

REFERENCES


