# Adaptive Error-Correcting Output Codes

**Guoqiang Zhong, Mohamed Cheriet**
Synchromedia Laboratory for Multimedia Communication in Telepresence,
École de Technologie Supérieure, Montréal, Québec H3C 1K3, Canada.
E-mail: guoqiang.zhong@synchromedia.ca, mohamed.cheriet@etsmtl.ca.

## Abstract

Error-correcting output codes (ECOC) are a successful technique to combine a set of binary classifiers for multi-class learning problems. However, in traditional ECOC framework, all the base classifiers are trained independently according to the defined ECOC matrix. In this paper, we reformulate the ECOC models from the perspective of multi-task learning, where the binary classifiers are learned in a common subspace of data. This novel model can be considered as an adaptive generalization of the traditional ECOC framework. It simultaneously optimizes the representation of data as well as the binary classifiers. More importantly, it builds a bridge between the ECOC framework and multi-task learning for multi-class learning problems. To deal with complex data, we also present the kernel extension of the proposed model. Extensive empirical study on 14 data sets from UCI machine learning repository and the USPS handwritten digits recognition application demonstrates the effectiveness and efficiency of our model.

## 1 Introduction

In machine learning and pattern recognition research, many issues can be formulated as multi-class learning problems. To address these problems, numerous approaches have been proposed, such as error-correcting output codes (ECOC) [Dietterich and Bakiri, 1995], boosting [Schapire, 1999] and random forests [Breiman, 2001]. Among others, ECOC represent a successful framework to combine a set of binary classifiers. Due to their simplicity and efficiency, ECOC methods have been widely used to face verification [Kittler *et al.*, 2001], text recognition [Ghani, 2001], and digits classification [Zhou and Suen, 2005].

The ECOC framework includes two steps: coding and decoding. The coding strategies include one-versus-all [Nilsson, 1965], one-versus-one [Hastie and Tibshirani, 1998], discriminant ECOC (DECOC) [Pujol *et al.*, 2006], ECOC-optimizing node embedding (ECOCONE) [Escalera and Pujol, 2006], and so on. Among them, one-versus-all and one-versus-one are problem-independent ECOC coding design strategies, whilst DECOC and ECOCONE are problem-dependent. The commonly used decoding strategies are Hamming decoding [Nilsson, 1965] and Euclidean decoding [Hastie and Tibshirani, 1998]. Some researchers have introduced loss-based function [Allwein *et al.*, 2000] or probabilities [Passerini *et al.*, 2004; Dekel and Singer, 2002] in decoding. Recently, Escalera et al. [Escalera *et al.*, 2010] proposed two novel ternary ECOC decoding strategies, $\beta$-density decoding and loss-weighted decoding, and showed their advantages over state-of-the-art decoding strategies. To the best of our knowledge, however, there is no work that attempts to integrate representation learning into the ECOC framework. That is, all the existing approaches train the binary classifiers in the original data space and combine them based on the applied decoding strategy, but no one aims to explore the intrinsic geometric structure of the data. Alternatively, learning intrinsic representation of high dimensional data may significantly benefit the classification accuracy.

To learn intrinsic representations of data, many dimensionality reduction approaches have been developed, under the assumption that the observed data reside on a low dimensional subspace or sub-manifold embedded in the original space. Traditional linear dimensionality reduction methods, such as principal components analysis (PCA) [Jolliffe, 1986] and linear discriminant analysis (LDA) [Fisher, 1936], have been widely used for finding a linear subspace of the data. However, they may fail to discover the intrinsic low dimensional structure when data lie on a nonlinear manifold. The kernel extension of linear dimensionality reduction methods usually work well on complex data, but seriously suffer from the high computational complexity problem on large scale data sets. Since the publication of two seminal manifold learning algorithms, isometric feature mapping (Isomap) [Tenenbaum *et al.*, 2000] and locally linear embedding (LLE) [Roweis and Saul, 2000], a plenty of nonlinear manifold learning methods have been proposed. However, most of them are unsupervised and cannot deal with the out-of-sample problem easily [Bengio *et al.*, 2003].

To combine the idea of representation learning and ECOC, we present an adaptive ECOC model in this paper. We consider the training of each binary classifier in the ECOC framework as a task, and reformulate the ECOC problem as a multiple correlated tasks learning problem. Furthermore, we learn a common subspace for all the data belonging to different classes, and perform classification of the learned binary classifiers in this subspace. Hence, our model simultaneously learns the informative representation of the data and the effective binary classifiers. More importantly, it builds a bridge between the ECOC framework and multi-task learning for the multi-class learning problems. Although the formulated model is not jointly convex with respect to the projection matrix and classifiers, a gradient descent optimization procedure with curvilinear search can be used to solve it, which is efficient and can deliver satisfactory classification results.

The rest of this paper is organized as follows: We introduce the ECOC framework in Section 2, and present the formulation and optimization of our model in Section 3. Experimental results on datasets from UCI machine learning repository and handwritten digits recognition are shown in Section 4. In Section 5, we conclude this paper with remarks.

## 2 Error-correcting output codes (ECOC)

In this section, we introduce the notation used in this paper at first, and then give a brief introduction to the ECOC framework.

### 2.1 Notation

We use boldface uppercase letters, such as $\mathbf{A}$, to denote matrices, and boldface lowercase letters, such as $\mathbf{z}$, to denote vectors. The $i$th row and $j$th column of a matrix $\mathbf{A}$ are defined as $\mathbf{A}_{i*}$ and $\mathbf{A}_{*j}$, respectively. $\mathbf{A}_{ij}$ denotes the element of $\mathbf{A}$ at the $i$th row and $j$th column. $\mathbf{z}_i$ is the $i$th element of a vector $\mathbf{z}$. We use $\mathbf{A}^T$ to denote the transpose of $\mathbf{A}$, and $\mathrm{tr}$ to denote the trace of $\mathbf{A}$. $|\mathbf{A}_{cl}|$ is the absolute value of $\mathbf{A}_{cl}$, and $\|\mathbf{A}\|_F$ is the Frobenius norm of $\mathbf{A}$.

For ECOC based models, we use $\mathbf{M} \in \{-1, 0, 1\}^{C\times L}$ to denote the ECOC matrix, where $L$ is the length of codewords. The $i$th row of $\mathbf{M}$, $\mathbf{M}_{i*}$, presents the codeword of class $i$. And meantime, the columns of $\mathbf{M}$ present the partition of the classes, where a symbol '0' means no need to take into account the corresponding class during training the binary classifier. Consequently, each of the $L$ base classifiers will be constructed according to the corresponding column of $\mathbf{M}$. If $\mathbf{M}_{cl} = 1$ (or $-1$), the samples associated with class $c$ will be treated as the positive (or negative) class for the $l$th binary classifier. $\mathbf{M}_{cl} = 0$ indicates that the samples associated with class $c$ will not be used for constructing the $l$th classifier. We use $\mathbf{w}_l$ and $\mathbf{b}_l$ to denote the coefficients and the bias of the $l$th linear binary classifier. We define $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_L\}$, and $\mathbf{b} = \{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_L\}$.

For multi-class classification problems, we are often given a set of samples $\{\{\mathbf{X}_i, \mathbf{y}_i\} \in \Re^D \times \Re^1, i = 1, 2, \dots, N\}$, where $\mathbf{y}_i \in \{1, 2, \dots, C\}$, $C$ is the number of classes and $N$ is the number of samples. In this work, we consider the training of each binary classifier as a learning task, and denote $\mathbf{X}^l$ as the training data for task $l$, $l = 1, 2, \dots, L$. Furthermore, we use $\mathbf{y}^l$ to denote the class labels for the data in task $l$.

### 2.2 The ECOC framework

As mentioned above, the ECOC framework include two steps: coding and decoding. In the coding step, an ECOC matrix, $\mathbf{M}$, is defined or learned from data. And then, the binary classifiers are trained according to the partition of the classes in each column of $\mathbf{M}$. In the decoding step, all the test data are fed in the binary classifiers, and each classifier provides a unique vote as positive class or negative class. One decoding strategy, such as Hamming distance or lost-based decoding, can be used to decide to which class the test datum should belong.

Even though having been widely used in many applications, the ECOC framework suffers from two critical problems: the base classifiers are trained independently, that is, the binary classifiers cannot get help from each other; and the training is performed in the original space, and doesn't try to explore the intrinsic structure of the data. To address these two problems, in this paper, we present an adaptive ECOC model.

## 3 Adaptive ECOC (AECOC)

In this section, we introduce our model, AECOC, in detail, including its formulation, optimization and decoding. Further, we discuss the relationship between AECOC and existing multi-task learning approaches, and present the differences between them.

### 3.1 Formulation

Here, we assume all the data have a low-dimensional subspace representation. For the $l$-th task, the linear predictive function is derived from the subspace as follows

$$f_l(\mathbf{X}^l) = \mathbf{X}^l \mathbf{A} \mathbf{w}_l + \mathbf{b}_l \tag{1}$$

where $\mathbf{w}_l \in \Re^d$ is the linear weight vector, $\mathbf{b}_l \in \Re$ is the bias parameter, $\mathbf{A} \in \Re^{D\times d}$ is the linear transformation matrix that projects the input data onto the low-dimensional subspace, and $d$ is the dimensionality of the subspace. The transformation matrix, $\mathbf{A}$, has orthogonal columns such that $\mathbf{A}^\top \mathbf{A} = \mathbf{I}$, where $\mathbf{I}$ is an identity matrix. To jointly learn the binary classifiers and the low dimensional representation of data, we conduct training by minimizing the following regularized loss function over the model parameters $\{\mathbf{A}, \mathbf{w}_l, \mathbf{b}_l |\, l = 1, \dots, L\}$,

$$\sum_{l=1}^{L} \mathcal{L}(f_l(\mathbf{X}^l), \mathbf{y}^l) + \alpha_l \parallel \mathbf{w}_l \parallel^2 + \gamma\, \mathrm{Reg}(\mathbf{A}), \tag{2}$$

subject to the constraints $\mathbf{A}^\top \mathbf{A} = \mathbf{I}$. Here, $\mathcal{L}(\cdot, \cdot)$ is a general loss function, $\mathrm{Reg}(\mathbf{A})$ is a regularization term with respect to $\mathbf{A}$, $\{\alpha_l\}, 1 = 1, \dots, L$ and $\gamma$ are tradeoff parameters. By conducting multi-task training, we expect the subspace representations can both capture the task specific discriminative information and benefit the learning of each binary classifiers.

In this work, we consider a least square loss function and a trace regularization, i.e.,

$$\mathcal{L}(f_l(\mathbf{X}^l), \mathbf{y}^l) = \parallel \mathbf{X}^l \mathbf{A} \mathbf{w}_l + \mathbf{b}_l - \mathbf{y}^l \parallel^2, \tag{3}$$

$$\mathrm{Reg}(\mathbf{A}) = \mathrm{tr}(\mathbf{A}). \tag{4}$$

Here, the reason for that we use the trace regularization is to learn the intrinsic representation automatically from data.

Hence we get the following optimization problem

$$\begin{aligned} \min_{\mathbf{A},\mathbf{w}_l,\mathbf{b}_l} \quad & \sum_{l=1}^{L} \parallel \mathbf{X}^l\mathbf{A}\mathbf{w}_l + \mathbf{b}_l - \mathbf{y}^l \parallel^2 \\ + \quad & \alpha_l \parallel \mathbf{w}_l \parallel^2 + \gamma \mathrm{tr}(\mathbf{A}) \\ s.t. \quad & \mathbf{A}^T\mathbf{A} = \mathbf{I}. \end{aligned} \tag{5}$$

Below we show that the optimal $\mathbf{w}_l, \mathbf{b}_l$ can be solved in terms of $\mathbf{A}$ from the optimization problem.

**Lemma 1** *The optimal $\{\mathbf{w}_l^*, \mathbf{b}_l^*\}_{l=1}^L$ that solve the optimization problem in Eq. (5) is given by*

$$\mathbf{w}_l^* = (\mathbf{A}^\top\mathbf{X}^{l\top}\mathbf{H}\mathbf{X}^l\mathbf{A} + \alpha_l\mathbf{I})^{-1}\mathbf{A}^\top\mathbf{X}^{l\top}\mathbf{H}\mathbf{y}^l, \tag{6}$$

*and*

$$\mathbf{b}_l^* = \frac{1}{N_l}\mathbf{1}^\top(\mathbf{y}^l - \mathbf{X}^l\mathbf{A}\mathbf{w}_l^*), \tag{7}$$

*for $l = 1, \ldots, L$, where $\mathbf{H} = \mathbf{I} - \frac{1}{N_l}\mathbf{1}\mathbf{1}^\top$, $\mathbf{1}$ denotes a column vector of length $N_l$ with all ones and $N_l$ is the number of training data in task $l$.*

**Proof:** Taking the derivatives of the objective function in Eq. (5) with respect to $\mathbf{b}_l$, and setting it to zero, we obtain

$$\mathbf{b}_l = \frac{1}{N_l}\mathbf{1}^\top(\mathbf{y}^l - \mathbf{X}^l\mathbf{A}\mathbf{w}_l) \tag{8}$$

for $l = 1, \ldots, L$. Substituting it back into Eq. (5), we have a new objective function as below

$$\sum_{l=1}^{L} \parallel \mathbf{H}(\mathbf{X}^l\mathbf{A}\mathbf{w}_l - \mathbf{y}^l) \parallel^2 + \alpha_l \parallel \mathbf{w}_l \parallel^2 + \gamma\mathrm{tr}(\mathbf{A}). \tag{9}$$

Then taking derivatives of this new objective function with respect to $\mathbf{w}_l$, and setting it to zeros, we obtain

$$\mathbf{w}_l = (\mathbf{A}^\top\mathbf{X}^{l\top}\mathbf{H}\mathbf{X}^l\mathbf{A} + \alpha_l\mathbf{I})^{-1}\mathbf{A}^\top\mathbf{X}^{l\top}\mathbf{H}\mathbf{y}^l, \tag{10}$$

for $l = 1, \ldots, L$. ■

Following Lemma 1, the objective function in Eq. (5) can be rewritten as below by replacing $\{\mathbf{w}_l, \mathbf{b}_l\}$

$$\begin{aligned} \mathcal{L}_1(\mathbf{A}) \quad = \quad & 2\mathbf{y}^{l\top}\mathbf{H}\mathbf{y}^l + \gamma\mathrm{tr}(\mathbf{A}) \qquad (11) \\ - \quad & \sum_{l=1}^{L} \mathbf{z}_l^\top\mathbf{A}(\mathbf{A}^\top\mathbf{M}_l\mathbf{A} + \alpha_l\mathbf{I})^{-1}\mathbf{A}^\top\mathbf{z}_l, \end{aligned}$$

where $\mathbf{M}_l$ and $\mathbf{z}_l$ are defined as

$$\mathbf{M}_l = \mathbf{X}^{l\top}\mathbf{H}\mathbf{X}^l, \tag{12}$$

and

$$\mathbf{z}_l = \mathbf{X}^{l\top}\mathbf{H}\mathbf{y}^l. \tag{13}$$

Hence the optimization problem in Eq. (5) can be equivalently re-expressed as

$$\begin{aligned} \min_{\mathbf{A}} \quad & \mathcal{L}_1(\mathbf{A}) \\ s.t. \quad & \mathbf{A}^\top\mathbf{A} = \mathbf{I}. \end{aligned} \tag{14}$$

The problem above is a non-convex optimization problem. Nevertheless, the gradient of the objective function with respect to $\mathbf{A}$ can be easily computed, and it is

$$\begin{aligned} \nabla_{\mathbf{A}}\mathcal{L}_1(\mathbf{A}) \quad = \quad & \gamma\mathbf{I} - 2\mathbf{z}_l\mathbf{z}_l^\top\mathbf{A}(\mathbf{A}^\top\mathbf{M}_l\mathbf{A} + \alpha_l\mathbf{I})^{-1} \qquad (15) \\ & + 2\mathbf{M}_l\mathbf{A}(\mathbf{A}^\top\mathbf{M}_l\mathbf{A} + \alpha_l\mathbf{I})^{-1} \\ & \mathbf{A}^\top\mathbf{z}_l\mathbf{z}_l^\top\mathbf{A}(\mathbf{A}^\top\mathbf{M}_l\mathbf{A} + \alpha_l\mathbf{I})^{-1}. \end{aligned}$$

## 3.2 Optimization

The non-convex optimization problem (14) is generally difficult to optimize due to the orthogonal constraints. In this work, we use a gradient descent optimization procedure with curvilinear search [Wen and Yin, 2010] to solve it for a local optimal solution.

In each iteration of the gradient descent procedure, given the current feasible point $\mathbf{A}$, the gradients can be computed using (15), such that

$$\mathbf{G} = \nabla_{\mathbf{A}}\mathcal{L}_1(\mathbf{A}). \tag{16}$$

We then compute two skew-symmetric matrices

$$\mathbf{F} = \mathbf{G}\mathbf{A}^\top - \mathbf{A}\mathbf{G}^\top. \tag{17}$$

It is easy to see $\mathbf{F}^\top = -\mathbf{F}$. The next new point can be searched as a curvilinear function of a step size variable $\tau$, such as

$$\mathbf{Q}(\tau) = (\mathbf{I} + \frac{\tau}{2}\mathbf{F})^{-1}(\mathbf{I} - \frac{\tau}{2}\mathbf{F})\mathbf{A} \tag{18}$$

It is easy to verify that $\mathbf{Q}(\tau)^\top\mathbf{Q}(\tau) = \mathbf{I}$ for all $\tau \in \Re$. Thus we can stay in the feasible region along the curve defined by $\tau$. Moreover, $\frac{d}{d\tau}\mathbf{Q}(0)$ is equal to the projections of $(-\mathbf{G})$ onto the tangent space $\mathcal{Q} = \{\mathbf{A} : \mathbf{A}^\top\mathbf{A} = \mathbf{I}\}$ at the current point $\mathbf{A}$. Hence $\mathbf{Q}(\tau)_{\tau \geq 0}$ is a descent path in the close neighborhood of the current point. We thus apply a similar strategy as the standard backtracking line search to find a proper step size $\tau$ using curvilinear search, while guaranteeing the iterations to converge to a stationary point. We determine a proper step size $\tau$ based on the well-known Barzilai-Borwein (BB) step size [Barzilai and Borwein, 1988].

## 3.3 Decoding

After the optimization of $\mathbf{A}$, we can obtain the binary classifiers based on Lemma 1. Specifically, given an unseen test datum, $\hat{\mathbf{x}} \in \Re^D$, we compute the prediction values using the binary classifiers as below

$$\mathbf{f}_l(\hat{\mathbf{x}}) = \hat{\mathbf{x}}^\top\mathbf{A}\mathbf{w}_l + \mathbf{b}_l, \tag{19}$$

and output the predicted label $\mathrm{sign}(\mathbf{f}_l(\hat{\mathbf{x}}))$, where $\mathrm{sign}(\cdot)$ is the signum function. Following this, any ECOC decoding strategy can be applied to obtain the predicted label of $\hat{\mathbf{x}}$.

## 3.4 Kernel AECOC

So far we have only considered the linear case for AECOC. In this section, we will apply the kernel trick to provide a nonlinear extension of the algorithm presented above. The optimization problem for the kernel extension is essentially the same as problem (5), with the only difference being that the data point $\mathbf{x}$ is mapped to $\phi(\mathbf{x})$ in a reproducing kernel Hilbert space, where $\phi(\cdot)$ denotes the feature map. Then the corresponding kernel function $\mathbf{K}(\cdot,\cdot)$ satisfies $\mathbf{K}(\mathbf{x}_1, \mathbf{x}_2) = \phi(\mathbf{x}_1)^T\phi(\mathbf{x}_2)$.

Suppose the $p$-th column of the projection matrix $\mathbf{A}$ in the reproducing kernel Hilbert space can be written as $\mathbf{A}_{*p} = \sum_{i=1}^N \omega_i^p\phi(\mathbf{x}_i)$. According to the representer theorem, we have $\phi(\mathbf{X})\mathbf{A} = \mathbf{K}\boldsymbol{\Omega}$, where $\boldsymbol{\Omega} = \{\omega^1, \ldots, \omega^d\}$ and $d$ is

Table 1: Details of the UCI data sets (T: training samples; A: attributes; C: classes).

| Problem | ♯ of T | ♯ of A | ♯ of C | Problem | ♯ of T | ♯ of A | ♯ of C |
|---|---|---|---|---|---|---|---|
| Dermatology | 366 | 34 | 6 | Segmentation | 2310 | 19 | 7 |
| Ecoli | 336 | 8 | 8 | Thyroid | 215 | 5 | 3 |
| Glass | 214 | 9 | 7 | Vehicle | 846 | 18 | 4 |
| Iris | 150 | 4 | 3 | Waveform21 | 5000 | 21 | 3 |
| Optdigits | 5620 | 64 | 10 | Waveform21 | 5000 | 40 | 3 |
| Pendigits | 10992 | 16 | 10 | Wine | 178 | 13 | 3 |
| Satimage | 6435 | 36 | 7 | Yeast | 1484 | 8 | 10 |

Table 2: Classification accuracy and standard deviation obtained by AECOC and the compared ECOC approaches on 14 UCI data sets and the USPS date set. The best results are highlighted in boldface.

| Data sets | OnevsOne | OnevsAll | DECOC | ECOCONE | AECOC |
|---|---|---|---|---|---|
| Dermatology | **0.9699 ± 0.0203** | 0.9426 ± 0.0226 | 0.9645 ± 0.0122 | 0.9645 ± 0.0123 | **0.9699 ± 0.0203** |
| Ecoli | **0.8432 ± 0.0434** | 0.7171 ± 0.0557 | 0.8236 ± 0.0268 | 0.8207 ± 0.0387 | 0.8347 ± 0.0320 |
| Glass | 0.6235 ± 0.0403 | 0.5112 ± 0.0820 | 0.5469 ± 0.0555 | 0.5470 ± 0.0439 | **0.6414 ± 0.0605** |
| Iris | 0.9600 ± 0.0365 | 0.9333 ± 0.0527 | 0.9333 ± 0.0527 | 0.9600 ± 0.0365 | **0.9667 ± 0.0236** |
| Optdigits | **0.9801 ± 0.0039** | 0.8601 ± 0.0109 | 0.8493 ± 0.0086 | 0.8541 ± 0.0088 | 0.9795 ± 0.0038 |
| Pendigits | **0.9684 ± 0.0026** | 0.7159 ± 0.0042 | 0.7715 ± 0.0200 | 0.7821 ± 0.0232 | **0.9684 ± 0.0025** |
| Satimage | 0.8570 ± 0.0079 | 0.7535 ± 0.0099 | 0.8057 ± 0.0167 | 0.7826 ± 0.0367 | **0.8572 ± 0.0090** |
| Segmentation | 0.9268 ± 0.0103 | 0.8645 ± 0.0248 | 0.8026 ± 0.0334 | 0.8398 ± 0.0182 | **0.9329 ± 0.0142** |
| Thyroid | 0.9023 ± 0.0345 | 0.8651 ± 0.0504 | 0.8512 ± 0.0510 | 0.8791 ± 0.0382 | **0.9163 ± 0.0265** |
| Vehicle | 0.7660 ± 0.0275 | 0.7258 ± 0.0158 | 0.7659 ± 0.0355 | 0.7176 ± 0.0735 | **0.7908 ± 0.0308** |
| Waveform21 | **0.8620 ± 0.0105** | 0.8144 ± 0.0162 | 0.8246 ± 0.0263 | 0.8246 ± 0.0263 | 0.8618 ± 0.0108 |
| Waveform40 | **0.8626 ± 0.0099** | 0.8176 ± 0.0219 | 0.8218 ± 0.0058 | 0.8142 ± 0.0270 | 0.8616 ± 0.0102 |
| Wine | 0.9886 ± 0.0156 | 0.9775 ± 0.0126 | 0.9717 ± 0.0202 | 0.9830 ± 0.0255 | **0.9887 ± 0.0154** |
| Yeast | 0.5803 ± 0.0074 | 0.4312 ± 0.0254 | 0.5158 ± 0.0317 | 0.5258 ± 0.0265 | **0.5809 ± 0.0100** |
| USPS | **0.9492 ± 0.0038** | 0.8093 ± 0.0093 | 0.8311 ± 0.0177 | 0.8316 ± 0.0166 | 0.9464 ± 0.0049 |
| Mean rank | 1.6000 | 4.4333 | 3.9333 | 3.6000 | **1.4333** |

the dimensionality of a low dimensional subspace of the reproducing kernel Hilbert space. Note that, since the dimensionality of the reproducing kernel Hilbert space may be infinity, the trace regularization needs to be changed to other regularization term, such as $\gamma\mathbf{\Omega}^T\mathbf{KLK}\mathbf{\Omega}$, where $\mathbf{L}$ is the so called graph Laplacian. However, for other optimization procedures, they are quite similar to that in the linear case. Hence, we omit the details here.

### 3.5 Relationship with existing multi-task learning approaches

In literature, several methods have been proposed for the multi-task learning problems. Some methods work on the assumption that task parameters lie in a low dimensional subspace, such that common features for all tasks are learned under a regularization framework [Argyriou *et al.*, 2008; Liu *et al.*, 2009]. Some other methods work on the assumption that tasks are clustered and parameters of tasks within a cluster lie close to each other, such that relationship between tasks are studied [Thrun and O'Sullivan, 1996; Xue *et al.*, 2007; Kumar and III, 2012]. There also exist probabilistic models which attempt to learn full task covariance matrix and use it to learn the predictor functions [Zhang and Yeung, 2010].

In this paper, we propose an adaptive generalization of the ECOC framework, to deal with multi-class learning problems. Even though the formulation of our AECOC model is similar to previous multi-task learning approaches, the motivation and optimization of AECOC are both distinct from existing multi-task learning approaches. On one hand, we reformulate the ECOC model from the perspective of multi-task learning, which builds a bridge between the ECOC framework and multi-task learning models. On the other hand, we introduce a gradient descent optimization procedure with curvilinear search [Wen and Yin, 2010] to optimize the AECOC model, which is efficient and can deliver satisfactory results.

## 4 Experiments

To demonstrate the effectiveness and efficiency of our proposed model, AECOC, we compared it with some related works on 14 data sets from UCI machine learning repository[1], and the USPS handwritten digits recognition application[2]. The details of the UCI data sets are shown in Table 1. The USPS data set includes 9298 samples from 10

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

[2]http://www.cs.nyu.edu/∼roweis/data.html

Table 3: Classification accuracy and standard deviation obtained by AECOC and the compared dimensionality reduction approaches on 14 UCI data sets and the USPS date set. The best results are highlighted in boldface.

| Data sets | PCA | LDA | LPP | MFA | AECOC |
|---|---|---|---|---|---|
| Dermatology | 0.9070 ± 0.0247 | 0.9536 ± 0.0121 | 0.9071 ± 0.0150 | **0.9726 ± 0.0217** | 0.9699 ± 0.0203 |
| Ecoli | 0.7983 ± 0.0376 | 0.8123 ± 0.0187 | 0.7898 ± 0.0595 | 0.8095 ± 0.0350 | **0.8347 ± 0.0320** |
| Glass | 0.6500 ± 0.1074 | 0.6093 ± 0.0853 | 0.6357 ± 0.1619 | **0.6723 ± 0.0514** | 0.6414 ± 0.0605 |
| Iris | 0.9400 ± 0.0435 | **0.9667 ± 0.0236** | 0.9467 ± 0.0506 | **0.9667 ± 0.0236** | **0.9667 ± 0.0236** |
| Optdigits | 0.9726 ± 0.0050 | 0.9219 ± 0.0117 | 0.9324 ± 0.0333 | 0.9644 ± 0.0013 | **0.9795 ± 0.0038** |
| Pendigits | 0.9881 ± 0.0009 | 0.9843 ± 0.0020 | 0.9901 ± 0.0041 | **0.9928 ± 0.0019** | 0.9684 ± 0.0025 |
| Satimage | **0.8862 ± 0.0085** | 0.8587 ± 0.0051 | 0.8401 ± 0.0079 | 0.8622 ± 0.0073 | 0.8572 ± 0.0090 |
| Segmentation | **0.9619 ± 0.0132** | 0.5160 ± 0.3145 | 0.9199 ± 0.0164 | 0.6221 ± 0.0466 | 0.9329 ± 0.0142 |
| Thyroid | 0.9581 ± 0.0104 | 0.9581 ± 0.0303 | **0.9860 ± 0.0208** | 0.9488 ± 0.0303 | 0.9163 ± 0.0265 |
| Vehicle | 0.5083 ± 0.0507 | 0.7282 ± 0.0299 | 0.4929 ± 0.0244 | 0.7389 ± 0.0635 | **0.7908 ± 0.0308** |
| Waveform21 | 0.8062 ± 0.0107 | 0.8230 ± 0.0161 | 0.5344 ± 0.0193 | 0.8144 ± 0.0140 | **0.8618 ± 0.0108** |
| Waveform40 | 0.8170 ± 0.0062 | 0.8118 ± 0.0036 | 0.5168 ± 0.0207 | 0.8224 ± 0.0153 | **0.8616 ± 0.0102** |
| Wine | 0.9549 ± 0.0320 | **0.9887 ± 0.0154** | 0.8703 ± 0.0600 | 0.9771 ± 0.0373 | **0.9887 ± 0.0154** |
| Yeast | 0.5265 ± 0.0348 | 0.5319 ± 0.0282 | 0.5050 ± 0.0179 | 0.5239 ± 0.0167 | **0.5809 ± 0.0100** |
| USPS | 0.8976 ± 0.0039 | 0.9234 ± 0.0037 | 0.8599 ± 0.0042 | 0.9155 ± 0.0067 | **0.9464 ± 0.0049** |
| Mean rank | 3.1667 | 3.0667 | 4.1333 | 2.5333 | **2.1000** |

classes of handwritten digits. The dimensionality of the data is 256. The compared methods include four state-of-the-all ECOC approaches, four classic dimensionality reduction approaches, one multi-task learning approach and one deep neural network model. For all the data sets, the features were normalized within $[0, 1]$. The average classification accuracy and standard deviation based on 5-fold cross validation are reported. In the following, we show the experimental results in detail.

### 4.1 Comparison to state-of-the-art ECOC approaches

In this experiment, we compared AECOC with four ECOC coding design methods. They are one-versus-one (OneVsOne) [Hastie and Tibshirani, 1998], one-versus-all (OneVsAll) [Nilsson, 1965], discriminant ECOC (DECOC) [Pujol *et al.*, 2006], and ECOC-optimizing node embedding (ECOCONE) [Escalera and Pujol, 2006]. Here, OneVsOne and OneVsAll are two problem-independent ECOC design strategies, while DECOC and ECOCONE are both problem-dependent design methods. Although any ECOC coding strategy can be used to formulate the AECOC model, for simplicity, we employed the OneVsOne coding design in this experiment. As mentioned in previous section, the training of each binary classifier was considered as a task, and all the tasks are correlated in the AECOC framework. For all the methods including AECOC, we used linear least square support vector machine (LSSVM) as base classifiers. The linear loss-weighted (LLW) decoding strategy [Escalera *et al.*, 2010] was used for all the coding designs. Following the parameter settings of [Escalera *et al.*, 2010], the regularization parameters, $\alpha$, for all the compared ECOC methods were empirically tested and set to $\frac{1}{2}$. For AECOC, $\gamma$ was set to 1, and $\alpha_i$'s, $i = 1, 2, \ldots, L$, were set to 0.1. This parameter setting might not be optimal for all the data sets. However, it's fair to use fixed parameters for all the compared methods, as weakness in the classification results will also be shared. Moreover, it's much easier for the replication of the experiments.

Table 2 shows the classification accuracies and standard deviations obtained on the UCI data sets, and the USPS data set. To evaluate the significance of the performance differences, we conducted the Friedman [Friedman, 1937; 1940] and the Nemenyi test [Nemenyi, 1963] as suggested by [Demsar, 2006], with a confidence value 0.05 on the classification results presented in Table 2. The statistical comparison results show that AECOC is significantly better than all the compared ECOC methods except for the OneVsOne coding design. However, in terms of the classification accuracy and the mean rank, we can see that AECOC improves the OneVsOne coding on most of the tested data sets.

### 4.2 Comparison to dimensionality reduction approaches

In this experiment, we compared AECOC with some classical dimensionality reduction methods, including principal components analysis (PCA) [Jolliffe, 1986], linear discriminant analysis (LDA), locality preserving projections (LPP) [He and Niyogi, 2003] and marginal Fisher analysis (MFA). Among them, PCA and LPP are unsupervised learning methods, while LDA and MFA are supervised methods. From the other viewpoint, PCA and LDA are global learning methods, whilst LPP and MFA are locality-based learning methods. Since the dimensionality of the LDA subspace is at most $C - 1$, the dimensionality of the targeted subspace for all the dimensionality reduction approaches was set to $d = \min\{D - 1, C - 1\}$, where $D$ is the dimensionality of the original data and $C$ is the number of classes. Based on our empirical study, since LPP and MFA are locality-based dimensionality reduction approaches, 1-nearest neighbor classifier generally performs better than or comparable with ECOC methods in the learned subspace of LPP and

MFA. Hence, we use 1-nearest neighbor classifier to evaluate the classification rates for all the dimensionality reduction approaches.

Table 3 shows the classification accuracies and standard deviations obtained on the UCI data sets and the USPS data set. To evaluate the significance of the performance differences, we conducted the Friedman [Friedman, 1937; 1940] and the Nemenyi test [Nemenyi, 1963] with a confidence value 0.05 on the classification results shown in Table 3. The statistical comparison results show that AECOC is significantly better than LPP, and at least comparable with other approaches. Moreover, we can see that, as the base classifiers in the AECOC model have effectively explored the distribution of the classes and each one benefits from the learned information of others, AECOC performs very well and delivers promising classification results on most of the data sets.

### 4.3 Comparison to multi-task learning and deep learning models

In this experiment, we compared AECOC with one state-of-the-art multi-task learning model, called multi-task relationship learning (MTRL) [Zhang and Yeung, 2010], as well as a deep neural network model, called stacked denoising autoencoders (SDAE) [Vincent *et al.*, 2010]. The code of MTRL can be found at the author's webpage[3], while the deep learning toolbox is available at the GitHub website[4]. For MTRL and AECOC, we formulated both the models according to the one-versus-one ECOC coding design, where the training of each binary classifier was taken into account as a task. For SDAE, we used a four layer neural network. The numbers of units in each layer were $D$, $D$, $d$ and $C$, respectively, where $D$, $d$ and $C$ are defined as same as the above subsection. The number of epoches for unsupervised pre-training and supervised fine-tuning were empirically set to 400 and 5000. Due to the high computational complexity of MTRL, we only implemented experiments on three small size UCI data sets, which include Dermatology, Iris and Wine.

Fig. 1 shows the accuracy obtained by each learning method. It is easy to see, AECOC outperforms the other two compared methods. Since the data size is small, the SDAE model cannot be trained sufficiently. Hence, it doesn't perform as well as AECOC and MTRL. Table 4 shows the training time of SDAE, MTRL and AECOC on the UCI data sets. We can see that the training of AECOC is very efficient. Basically, it has a computation cost of $O(Nd^2+d^3)$, where $N$ is the number of training data and $d$ is the dimensionality of the low dimensional subspace. This computational complexity is similar to that of SDAE, which is $O(N)$. However, compared to those of AECOC and SDAE, both the time and space complexity of MTRL are much higher.

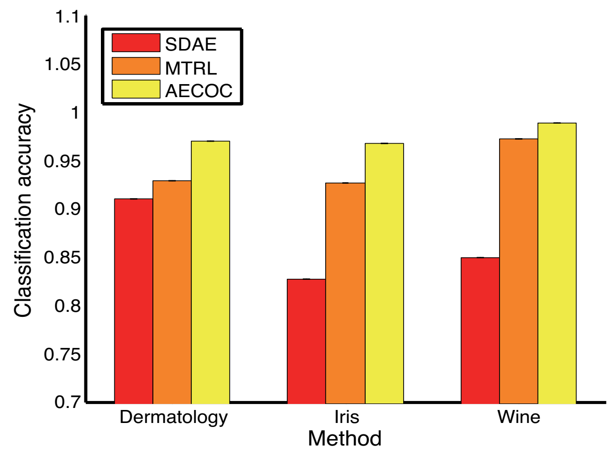


Figure 1: Classification accuracy obtained by SDAE, MTRL and AECOC on three UCI data sets.

Table 4: Training time of SDAE, MTRL and AECOC on three UCI data sets. The lowest ones are highlighted in boldface.

| Data sets | SDAE | MTRL | AECOC |
|---|---|---|---|
| Dermatology | **0.0136** | 7.0280 | 0.2770 |
| Iris | 0.0105 | 0.1032 | **0.0081** |
| Wine | **0.0111** | 0.1424 | 0.0146 |

## 5 Conclusion

In this paper, we have proposed an adaptive generalization of the ECOC framework, called AECOC, for multi-class learning problems. Unlike traditional ECOC methods, in the model of AECOC, we consider the training of the binary classifiers as correlated tasks, and all the tasks are combined to solve a multi-class learning problem. Particularly, AECOC learns the base classifiers and the intrinsic representation of the data simultaneously, and builds a bridge between the ECOC framework and multi-task learning models. Extensive experiments on 14 data sets from UCI machine learning repository and the USPS handwritten digits recognition application demonstrate the superiority of AECOC over state-of-the-art ECOC methods, dimensionality reduction methods, multi-task learning methods and deep learning methods. In future work, we will investigate the integration of the learning of ECOC coding design into the AECOC model, to further improve the performance of AECOC on multi-class learning problems.

[3]http://www.comp.hkbu.edu.hk/ yuzhang/

[4]https://github.com/rasmusbergpalm/DeepLearnToolbox

## Acknowledgments

This work was partially supported by the Social Sciences and Humanities Research Council of Canada (SSHRC) and the Natural Sciences and Engineering Research Council of Canada (NSERC). The authors thank the reviewers for their insightful comments and constructive suggestions.

## References

[Allwein *et al.*, 2000] E. L. Allwein, R. E. Schapire, and Y. Singer. Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers. *Journal of Machine Learning Research*, 1:113–141, 2000.

[Argyriou *et al.*, 2008] A. Argyriou, T. Evgeniou, and M. Pontil. Convex Multi-Task Feature Learning. *Machine Learning*, 73(3):243–272, 2008.

[Barzilai and Borwein, 1988] J. Barzilai and J. M. Borwein. Two-point sStep Size Gradient Methods. *IMA J. Numer. Anal.*, 8:141–148, 1988.

[Bengio *et al.*, 2003] Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N. Le Roux, and M. Ouimet. Out-of-Sample Extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering. In *NIPS*, 2003.

[Breiman, 2001] L. Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.

[Dekel and Singer, 2002] O. Dekel and Y. Singer. Multiclass Learning by Probabilistic Embeddings. In *NIPS*, pages 945–952, 2002.

[Demsar, 2006] Janez Demsar. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[Dietterich and Bakiri, 1995] T. G. Dietterich and G. Bakiri. Solving Multiclass Learning Problems via Error-Correcting Output Codes. *J. Artif. Intell. Res. (JAIR)*, 2:263–286, 1995.

[Escalera and Pujol, 2006] S. Escalera and O. Pujol. ECOC-ONE: A Novel Coding and Decoding Strategy. In *ICPR*, pages 578–581, 2006.

[Escalera *et al.*, 2010] S. Escalera, O. Pujol, and P. Radeva. On the Decoding Process in Ternary Error-Correcting Output Codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):120–134, 2010.

[Fisher, 1936] R. A. Fisher. The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*, 7:179–188, 1936.

[Friedman, 1937] M. Friedman. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *Journal of the American Statistical Association*, 32:675–701, 1937.

[Friedman, 1940] M. Friedman. A Comparison of Alternative Tests of Significance for the Problem of m Rankings. *Annals of Mathematical Statistics*, 11:86–92, 1940.

[Ghani, 2001] R. Ghani. Combining Labeled and Unlabeled Data for Text Classification with a Large Number of Categories. In *ICDM*, pages 597–598, 2001.

[Hastie and Tibshirani, 1998] T. Hastie and R. Tibshirani. Classification by Pairwise Coupling. *Annals of Statistics*, 26(2):451–471, 1998.

[He and Niyogi, 2003] X. He and P. Niyogi. Locality Preserving Projections. In *NIPS*, 2003.

[Jolliffe, 1986] I.T. Jolliffe. *Principal Component Analysis*. New York: Springer-Verlag, 1986.

[Kittler *et al.*, 2001] J. Kittler, R. Ghaderi, T. Windeatt, and J. Matas. Face Verification Using Error Correcting Output Codes. In *CVPR (1)*, pages 755–760, 2001.

[Kumar and III, 2012] A. Kumar and H. Daumé III. Learning Task Grouping and Overlap in Multi-task Learning. In *ICML*, 2012.

[Liu *et al.*, 2009] J. Liu, S. Ji, and J. Ye. Multi-Task Feature Learning Via Efficient $l_{2,1}$-Norm Minimization. In *UAI*, pages 339–348, 2009.

[Nemenyi, 1963] P. B. Nemenyi. *Distribution-Free Multiple Comparisons*. PhD thesis, Princeton University, 1963.

[Nilsson, 1965] N. J. Nilsson. *Learning Machines*. McGraw-Hill, 1965.

[Passerini *et al.*, 2004] A. Passerini, M. Pontil, and P Frasconi. New Results on Error Correcting Output Codes of Kernel Machines. *IEEE Transactions on Neural Networks*, pages 45–54, 2004.

[Pujol *et al.*, 2006] O. Pujol, P. Radeva, and J. Vitrià. Discriminant ECOC: A Heuristic Method for Application Dependent Design of Error Correcting Output Codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(6):1007–1012, 2006.

[Roweis and Saul, 2000] S. T. Roweis and L. K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290(5500):2323–2326, 2000.

[Schapire, 1999] R. E. Schapire. A Brief Introduction to Boosting. In *IJCAI*, pages 1401–1406, 1999.

[Tenenbaum *et al.*, 2000] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, 2000.

[Thrun and O'Sullivan, 1996] S. Thrun and J. O'Sullivan. Discovering Structure in Multiple Learning Tasks: The TC Algorithm. In *ICML*, pages 489–497, 1996.

[Vincent *et al.*, 2010] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *Journal of Machine Learning Research*, 11:3371–3408, 2010.

[Wen and Yin, 2010] Z. Wen and W. Yin. A Feasible Method for Optimization with Orthogonality Constraints. Technical report, Rice University, 2010.

[Xue *et al.*, 2007] Y Xue, X Liao, L Carin, and B Krishnapuram. Multi-Task Learning for Classification with Dirichlet Process Priors. *Journal of Machine Learning Research*, 8:35–63, 2007.

[Zhang and Yeung, 2010] Y. Zhang and D.-Y. Yeung. A Convex Formulation for Learning Task Relationships in Multi-Task Learning. In *UAI*, pages 733–442, 2010.

[Zhou and Suen, 2005] J. Zhou and C. Y. Suen. Unconstrained Numeral Pair Recognition Using Enhanced Error Correcting Output Coding: A Holistic Approach. In *ICDAR*, pages 484–488, 2005.