Neurocomputing 89 (2012) 158–167

Contents lists available at SciVerse ScienceDirect

Neurocomputing

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

# Can under-exploited structure of original-classes help ECOC-based multi-class classification?

Yunyun Wang [a], Songcan Chen [a,*], Hui Xue [b]

[a] School of Computer Science and Engineering, Nanjing University of Aeronautics & Astronautics, Nanjing 210016 , PR China
[b] School of Computer Science and Engineering, Southeast University, Nanjing 210096, PR China

ARTICLE INFO

*Article history:*
Received 13 July 2011
Received in revised form
29 November 2011
Accepted 26 February 2012
Communicated by Weifeng Liu
Available online 24 March 2012

*Keywords:*
Multi-class classification
Error correcting output codes
Support vector machine
Cluster assumption
Manifold assumption

ABSTRACT

Error correcting output codes (ECOC) is a popular framework for addressing multi-class classification problems by combing multiple binary sub-problems. In each binary sub-problem, at least one class is actually a "meta-class" consisting of multiple original classes, and treated as a single class in the learning process. This strategy brings a simple and common implementation of multi-class classification, but simultaneously, results in the under-exploitation of already-provided structure knowledge in individual original classes. In this paper, we present a new methodology to show that the utilization of such prior structure knowledge can further strengthen the performance of ECOCs, and the structure knowledge is formulated under the cluster and manifold assumptions, respectively. Finally, we validate our methodology on both toy and real benchmark datasets (UCI, face recognition and objective category), consequently validate the structure knowledge of individual original classes for ECOC-based multi-class classification.



## 1. Introduction

In real applications, we frequently encounter problems involving multi-class classification, in which observed data belong to more than two classes [1,2]. Examples for such applications include optical character recognition, text classification and medical analysis, etc..

There are mainly two independent lines of researches for designing multi-class classification methods. One line is "direct design", i.e., directly designing a multi-class classifier by adopting multi-class output encodings, typically including decision tree, neural network, logistic regression [3], least-squares classifier, and multi-class SVMs [4–6]. The other line is "(indirect) decomposition or ECOC design", i.e., decomposing the original multi-class problems into multiple binary sub-problems, which can be efficiently solved by any binary classification method [7–9], and then combining the results from all binary sub-classifiers for final classification. This strategy is simple and common, thus has brought an independent and broad area of researches. In this paper, we focus the second line.

The simplest decomposition strategy is One-Vs-All (OVA), in which each class is compared with all other ones, generating $C$ binary sub-problems (or corresponding binary classifiers), where $C$ is the number of classes. A new instance is then assigned to the class with the maximum classification score among all corresponding binary classifiers. Friedman [10] suggested the One-Vs-One (OVO) strategy, in which all pairs of classes are compared, resulting in $C(C-1)/2$ binary sub-problems, and the prediction for a new instance is implemented by voting of all corresponding binary classifiers. Dietterich et al. [11] developed the general (binary) error correcting output codes (ECOC) framework, in which each class is given a $N$-length error correcting output codeword with each component valued from $\{-1, +1\}$, and those codewords for individual classes have the optimal separation between each other. Arranging those codewords as rows, a $C \times N$-size codeword matrix is constructed, whose individual columns indicate the class-set partitions for the $N$ generated binary sub-problems, respectively. For a new instance, a $N$-length code can be obtained from the corresponding binary classifiers, and the instance is classified to the "closest" class measured by Hamming distance between the instance code and individual class codewords. Allwein et al. [7] extended the ECOC framework and developed ternary ECOC, in which each component in the codeword matrix is allowed to take values from $\{-1, +1, 0\}$, and the zero-value indicates that the corresponding class is not considered in the current binary sub-problem. The prediction of any new instance adopts a loss-based function instead of the original Hamming distance. It is ternary ECOC that covers OVA, OVO and ECOC in a unified framework.

Later, new improvements have been developed for ECOC and focus on both the designs of its encoding (w.r.t. the construction

* Corresponding author. Tel.: +86 25 848 96481; fax: +86 25 844 98 069.
*E-mail addresses:* wangyunyun@nuaa.edu.cn (Y. Wang), s.chen@nuaa.edu.cn (S. Chen), hxue@seu.edu.cn (H. Xue).

0925-2312/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2012.02.035

of codeword matrix) [2,8,9] and decoding (w.r.t. the prediction of new instances) [12–14] strategies. For the encoding strategy, many researchers attempted to adapt the encoding process to the learning problem at hand, or more specifically, utilized the prior knowledge of the current learning problem to develop problem-dependent codeword matrices [8,15]. For example, Pujol et al. [8] sacrificed the optimal codeword separation in favor of class discrimination in the class-set partitions. Escalera et al. [9] modeled complex classification problems by splitting the complex classes into several subclasses. Pujol et al. [2] extended any initial codeword matrix by adding new binary classifiers that focus on the difficult-to-split classes. Orthogonal to the work over the encoding process, other researchers concentrated more on the decoding process. Hastie et al. [12] adopted the Bradley–Terry (BT) model to develop a new decoding method, which integrates the results of the binary classifiers in OVO into a single estimate of class membership probabilities. Later, Zadrozny et al. [13] extended such a decoding method to any ternary coding scheme. Luo et al. [16] weighted the output space of each base classifier so that the distance function of decoding is adapted. While Takenouchi et al. [14] considered the inconsistency between the encoding and decoding processes in ternary ECOC approaches, and developed ternary AdaBoost and ternary BT model-based decoding method to make them consistent with each other.

Although those ECOC approaches work well in multi-class classification, there is still prior knowledge under-exploited in their implementations. Specifically, for binary ECOC approaches, in each binary sub-problem, at least one class is actually a "meta-class" consisting of several original classes, and for ternary ECOC approaches except OVO, such a "meta-class" exists in at least one sub-problem. In the learning process, each "meta-class" will be treated as a single class, which brings a simple and common implementation of multi-class classification, but simultaneously results in the under-exploitation of already-provided structure knowledge in individual original classes. It is well-known that for any learning method, its generalization depends on both the representation of data and exploitation of prior knowledge for the current learning problem [17], as a result, for better generalization performance, we should explore as much prior knowledge as possible, let alone the knowledge provided already. In this paper, we present a methodology to show that utilizing such prior structure knowledge in implementing ECOCs can further strengthen the multi-class classification performance. The structure knowledge here is formulated under assumptions that data distribution follows the cluster and manifold structures, respectively, corresponding to incorporating manners similar to those in structural regularized SVM (SRSVM) [18] and Laplacian SVM (LapSVM) [19], respectively. Finally, we validate our methodology by comparison with the baselines on both toy and real benchmark datasets (UCI, face recognition and objective category), consequently validate the structure knowledge of individual original classes for ECOC-based multi-class classification. The contributions of this paper are summarized as follows,

- Point out the common under-exploitation of already-provided structure knowledge in the implementation of the off-the-shelf ECOCs (except OVO).
- Provide a general methodology for incorporating such structure knowledge into the implementation of ECOCs, which can be applied to any ECOC, as well as their improvements.
- Develop a more effective multi-class classifier than its original design, which is exactly consistent with the theory that the generalization of any learning method depends on both the representation of data and exploitation of prior knowledge [17], consequently validate the prior structure of individual original classes for ECOC-based multi-class classification.

In this paper, our aim is to utilize the already-provided structure knowledge in individual original classes, respectively under the cluster and manifold assumptions, to show that it is helpful for ECOC-based multi-class classification. Of course, other formulations for the structure knowledge, or other prior knowledge in individual original classes can also be exploited for boosting the multi-class classification performance.

The rest of the paper is organized as follows, Section 2 introduces the ECOC framework for multi-class classification. Section 3 presents the proposed methodology. Section 4 shows the comparison experiments. Section 5 draws some conclusions.

## 2. Error correcting output codes

Error correcting output codes [11] is a general framework for solving multi-class classification problems by combining multiple binary classifiers. The main idea is to assign the given classes a set of corresponding Error correcting output codewords with the optimal separation between each other. The components of those codewords are valued from $\{-1, +1\}$. Then a codeword matrix $\mathbf{M}\in\{-1, +1\}^{C\times N}$ can be constructed with individual rows corresponding to the codewords for the $C$ classes, respectively, and individual columns indicating the class-set partitions for the $N$ generated binary sub-problems (or corresponding binary classifiers), respectively. After applying the $N$ binary classifiers, a $N$-length code is generated for each new instance, and the instance is assigned to the "closest" class according to the Hamming distance between the instance code and codewords of individual classes. The framework was then extended by allowing the components of $\mathbf{M}$ to take values from $\{-1, +1, 0\}$ [7], in which the zero-value indicates that the corresponding class is not considered in the current binary sub-problem.

Given a dataset $\{x_i,y_i\}_{i=1}^n$ where each $y_i\in\{1,\ldots C\}$, $C>2$, and codeword matrix $\mathbf{M}\in\{-1, +1, 0\}^{C\times N}$, suppose the binary classification method implements the Tikhonov regularization [20,21], then with a linear decision function $f(x)=w^Tx+b$, the $j$th binary classifier can be formulated as

$$\min_{w_j,b_j}\sum_{i=1}^{n}\ell(w_j^Tx_i+b_j,\mathbf{M}_{y_ij})+\frac{\lambda}{2}||w_j||^2 \tag{1}$$

where $\mathbf{M}_{y_ij}$ denotes the true label of $x_i$ in the $j$th binary classifier, $\ell(\cdot,\cdot)$ is a loss function between the classification score and true label of any instance, and $\lambda$ is a regularization parameter balancing between the classification of training instances and complexity of the learning model.

After applying the $N$ binary classifiers, a $N$-length code can be obtained for each new instance consisting of the corresponding classification scores, and the instance is assigned to the "closest" class according to some distance metric between the instance code and individual class codewords. Based on the commonly-used Hamming distance, the prediction for an arbitrary new instance $x$ can be formulated as

$$\hat{y}=\underset{r=1\ldots C}{\operatorname{argmin}}\sum_{j=1}^{N}\left(\frac{1-sign(\mathbf{M}_{rj}f_j(x))}{2}\right) \tag{2}$$

where $f_j(x)$ denotes the classification score for instance $x$ by the $j$th binary classifier.

## 3. Methodology

In this section, we present the methodology for utilizing the structure knowledge of individual original classes in the implementation of ECOCs, assuming that data distribution obeys the cluster and manifold structures, respectively. In what follows, we

first formulate the structure knowledge, and then present our proposed methodology in separated sub-sections.

### 3.1. Prior knowledge

In each binary sub-problem of ECOCs, each class is actually a "meta-class" consisting of multiple original classes, and treated as a single class. This strategy brings a simple strategy for applying any binary classification method to multi-class problems, but simultaneously, leaves the prior structure knowledge of individual original classes under-exploited, which would also be helpful for multi-class classification. In this paper, we aim to verify that such prior knowledge is helpful for ECOC-based multi-class classification.

For formulating the structure knowledge, some assumption for data distribution should be adopted. There are mainly two such assumptions in literature, i.e., the cluster and manifold assumptions [18,22]. The cluster (structure) assumption assumes that data are distributed within several clusters, and instances in the same cluster should share similar classification outputs. Here we adopt the class granularity,[1] i.e., view each original class as a single cluster, and formulate the structure knowledge for the $k$th (original) class under the cluster assumption as

$$\begin{aligned}&\sum_{x_i\in\Omega_k}\left(f(x_i)-\frac{1}{|\Omega_k|}\sum_{x_j\in\Omega_k}f(x_j)\right)^2\\&=\sum_{x_i\in\Omega_k}\left(w^Tx_i-\frac{1}{|\Omega_k|}\sum_{x_j\in\Omega_k}w^T_{x_j}\right)^2\\&=w^T\sum_{x_i\in\Omega_k}(x_i-u_k)(x_i-u_k)^Tw\\&\triangleq w^T\mathbf{V}_kw\end{aligned}\tag{3}$$

where $\Omega_k$ denotes the set of instances belonging to the $k$th class, $|\Omega_k|$ denotes the number of instances in $\Omega_k$. $u_k$ and $\mathbf{V}_k$ denote the instance mean and covariance matrix of the $k$th class, respectively [18]. From the cluster assumption, one should minimize Eq. (3) to guarantee that instances in the same class with a cluster structure should be as similar as possible in the output space, which is equivalent to minimizing the within-class compactness[2] under the cluster assumption.

The manifold (structure) assumption assumes that data are distributed on some low dimensional manifold, which can be captured by a Laplacian graph with its nodes representing instances and edge weights representing similarities between instances. Similar instances should share similar classification outputs according to the graph Laplacian. For the $k$th (original) class, the structure knowledge under the manifold assumption can be formulated as

$$\begin{aligned}&\sum_{x_s,x_t\in\Omega_k}\mathbf{S}^k_{st}(f(x_s)-f(x_t))^2\\&=w^T\sum_{x_s,x_t\in\Omega_k}\mathbf{S}^k_{st}(x_s-x_t)(x_s-x_t)^Tw\\&\triangleq w^T\mathbf{X}_k\mathbf{L}_k{\mathbf{X}_k}^Tw\end{aligned}\tag{4}$$

where $\mathbf{X}_k$ denotes the data matrix for the $k$th class. $\mathbf{L}_k=\mathbf{D}^k-\mathbf{S}^k$ is the Laplacian matrix for the $k$th class, $\mathbf{D}^k$ is a diagonal matrix with its component written as $\mathbf{D}^k_{ss}=\sum_{x_t\in\Omega_k}\mathbf{S}^k_{st}$, and $\mathbf{S}^k_{st}$ describes the similarity between $x_s$ and $x_t$ over the Laplacian graph, which is usually defined as

$$S^k_{st}=\begin{cases}1\text{ or }e^{((-\|x_s-x_t\|^2)/2\sigma^2)} & \text{if } x_s,x_t\in\Omega_k\\0 & \text{else}\end{cases}\tag{5}$$

where $\sigma$ is a weight parameter [19]. From the manifold assumption, one should minimize Eq. (5) to guarantee that instances in the same class with a manifold structure should be as similar as possible in the output space, which is equivalent to minimizing the within-class compactness of individual original classes under the manifold assumption.

In what follows, we will present the methodology for incorporating such structure knowledge into ECOCs, assuming that data distribution follows the cluster and manifold structures in separated sub-sections, respectively.

### 3.2. Structure-exploited ECOC with cluster assumption

Given dataset $\{x_i,y_i\}^n_{i=1}$, where each $y_i\in\{1,\dots C\}$, $C>2$, and codeword matrix $\mathbf{M}\in\{-1,\ +1,\ 0\}^{C\times N}$, suppose there are $C_j$ original classes involved in the $j$th binary sub-problem, or more specifically, $C_j$ original classes whose corresponding labels are non-zero in the $j$th binary sub-problem. With a linear decision function $f(x)=w^Tx+b$, through utilizing the structure knowledge under the cluster assumption, the new optimization problem for the $j$th binary classifier can be formulated as

$$\min_{w_j,b_j}\sum_{i=1}^n\ell(w_j^Tx_i+b_j,\mathbf{M}_{y_ij})+\lambda||w_j||^2+\frac{\lambda_s}{2}w_j^T\mathbf{V}^jw_j\tag{6}$$

where $\mathbf{V}^j=\sum_{k=1}^{C_j}\mathbf{V}^j_k$, $\mathbf{V}^j_k$ is the covariance matrix for the $k$th original class involved in the $j$th binary classifier, and $\lambda_s$ is the regularization parameter regulating the relative importance of the structure incorporated.

When adopting the hinge loss as in SVM, Eq. (6) can be reformulated as

$$\begin{aligned}\min_{w_j,b_j,\xi_{ij}}\quad&\frac{1}{2}||w_j||^2+\lambda\sum_{i=1}^n\xi_{ij}+\frac{\lambda_s}{2}{w_j}^T\mathbf{V}^jw_j\\\text{s.t.}\quad&\mathbf{M}_{y_ij}(w_j^Tx_i+b_j)\geq1-\xi_{ij}\\&\xi_{ij}\geq0,\forall i=1\dots n.\end{aligned}\tag{7}$$

Further, by adopting the standard method of Lagrange multipliers, the dual problem of Eq. (7) can be formulated as

$$\begin{aligned}\max_{\alpha^j_1,\dots\alpha^j_s}\quad&\sum_{s=1}^n\alpha^j_s-\frac{1}{2}\sum_{s=1}^n\sum_{t=1}^n\alpha^j_s\alpha^j_t\mathbf{M}_{y_sj}\mathbf{M}_{y_tj}x_s^T(\mathbf{I}+\lambda_s\mathbf{V}^j)^{-1}x_t\\\text{s.t.}\quad&\sum_{s=1}^n\alpha^j_s\mathbf{M}_{y_sj}=0\\&0\leq\alpha^j_s\leq\lambda,\forall s=1\dots n\end{aligned}\tag{8}$$

where $\{\alpha^j_s\}^n_{s=1}$ denote the Lagrange multipliers for the $j$th binary classifier. One can easily observe that Eq. (8) replaces each $x_s^Tx_t$ in the original hinge-loss ECOC methods by $x_s^T(\mathbf{I}+\lambda_s\mathbf{V}^j)^{-1}x_t$. Since the inner product can be viewed as a similarity criteria between instances, the new method actually defines a new similarity criteria considering different weights for individual features by $(\mathbf{I}+\lambda_s\mathbf{V}^j)^{-1}$, or more specifically, by the sum of covariance matrices with parameter$\lambda_s$. The optimization problem in Eq. (8) is a QP problem which can be solved by any standard QP solvers. Then the classification score for an arbitrary instance $x$ by the $j$th binary classifier can be formulated as

$$f_j(x)=\sum_{s=1}^n\alpha^j_s\mathbf{M}_{y_sj}x_s^T(\mathbf{I}+\lambda_s\mathbf{V}^j)^{-1}x+b_j\tag{9}$$

Finally, those obtained classification scores by individual binary sub-classifiers are combined for the prediction of $x$.

However, when instances are linearly non-separable in the input space, the linear classifiers would provide poor performance [23]. In those cases, kernelization provides an alternative solution

[1] Class granularity refers to that data structure within each class is depicted by a single cluster, or in other words, each class is viewed as a single cluster.

[2] Within-class compactness describes the similarity of classification outputs for instances within the same class.

by projecting those instances from the input space to a higher (or even infinite) dimension kernel space, in which instances are more likely to be linearly separable [24]. In what follows, we will provide the kernelized version for the new method. Suppose a nonlinear (implicit) kernel mapping $\phi:\mathcal{R}^d\rightarrow\mathscr{H}$, where $\mathscr{H}$ denotes the high dimension kernel space, the new optimization problem in the kernel space can be written as

$$\begin{aligned}\min_{w_j,b_j,\xi_{ij}}\quad & \frac{1}{2}||w_j||^2+\lambda\sum_{i=1}^{n}\xi_{ij}+\frac{\lambda_s}{2}w_j^T\mathbf{V}^{j\Phi}w_j\\ \text{s.t.}\quad & \mathbf{M}_{y_ij}(w_j^T\phi(x_i)+b_j)\geq 1-\xi_{ij}\\ & \xi_{ij}\geq 0,\forall s=1...n\end{aligned}\tag{10}$$

where $\mathbf{V}^{j\Phi}=\sum_{k=1}^{C_j}\mathbf{V}_k^{j\Phi}$ denotes the sum of covariance matrices for individual original classes in the kernel space. The dual problem of Eq. (10) can be formulated as

$$\begin{aligned}\max_{\alpha_1^j,\ldots\alpha_s^j}\quad & \sum_{s=1}^{n}\alpha_s^j-\frac{1}{2}\sum_{s=1}^{n}\sum_{t=1}^{n}\alpha_s^j\alpha_t^j\mathbf{M}_{y_sj}\mathbf{M}_{y_tj}\phi(x_s)^T(\mathbf{I}+\lambda_s\mathbf{V}^{j\Phi})^{-1}\phi(x_t)\\ \text{s.t.}\quad & \sum_{s=1}^{n}\alpha_s^j\mathbf{M}_{y_sj}=0\\ & 0\leq\alpha_s^j\leq\lambda,\forall s=1...n\end{aligned}\tag{11}$$

which can be further written as

$$\begin{aligned}\max_{\boldsymbol{\alpha}^j}\quad & \boldsymbol{\alpha}^{j^T}\vec{\mathbf{1}}_n-\tfrac{1}{2}\boldsymbol{\alpha}^{j^T}\left[(\mathbf{K}_j-\lambda_s\tilde{\mathbf{K}}_j\boldsymbol{\Lambda}(\boldsymbol{\Lambda}+\lambda_s\boldsymbol{\Lambda}\hat{\mathbf{K}}_j\boldsymbol{\Lambda})^{-1}\boldsymbol{\Lambda}\tilde{\mathbf{K}}_j^T)\circ\mathbf{M}_{\mathbf{y}j}\mathbf{M}_{\mathbf{y}j}^T\right]\boldsymbol{\alpha}^j\\ \text{s.t.}\quad & \boldsymbol{\alpha}^{j^T}\mathbf{M}_{\mathbf{y}j}=0\\ & \vec{\mathbf{0}}_n\leq\boldsymbol{\alpha}^j\leq\lambda\vec{\mathbf{1}}_n\end{aligned}\tag{12}$$

where $\vec{\mathbf{0}}_n$ and $\vec{\mathbf{1}}_n$ denote the $n$-dimensional all-zero and all-one vectors, respectively, $\boldsymbol{\alpha}^j$denotes the vector of Lagrange multipliers, and $\mathbf{M}_{\mathbf{y}j}$ denotes the label vector for given instances in the $j$th binary sub-problem. $\mathbf{K}_j=\langle\mathbf{X}^j,\mathbf{X}^j\rangle_{\mathscr{H}}$ denotes the kernel matrix over given instances, $\tilde{\mathbf{K}}_j=\langle\mathbf{X}^j,\mathbf{P}^j\rangle_{\mathscr{H}}$ denotes the kernel matrix over the given instances and instances arranged by the sequence of original classes, and $\hat{\mathbf{K}}_j=\langle\mathbf{P}^j,\mathbf{P}^j\rangle_{\mathscr{H}}$ denotes the kernel matrix over instances arranged by the sequence of original classes in the $j$th binary classifier. The detailed derivation from Eq. (11) to Eq. (12) can be found in Appendix A.

### 3.3. Structure-exploited ECOC with manifold assumption

With a linear decision function $f(x)=w^Tx+b$, through utilizing the structure knowledge under the manifold assumption, the new optimization problem for the $j$th binary classifier can be formulated as

$$\begin{aligned}\min_{w_j,b_j,\xi_{ij}}\quad & \frac{1}{2}||w_j||^2+\lambda\sum_{i=1}^{n}\xi_{ij}+\frac{\lambda_s}{2}w_j^T\mathbf{X}^j\mathbf{L}^j\mathbf{X}^{jT}w_j\\ \text{s.t.}\quad & \mathbf{M}_{y_ij}(w_j^Tx_i+b_j)\geq 1-\xi_{ij}\\ & \xi_{ij}\geq 0,\forall i=1...n\end{aligned}\tag{13}$$

where $\mathbf{X}^j$ and $\mathbf{L}^j$ denote the data matrix and Laplacian matrix for the $j$th classifier, respectively. The dual problem of Eq. (13) can be formulated as

$$\begin{aligned}\max_{\alpha_1^j,\ldots\alpha_s^j}\quad & \sum_{s=1}^{n}\alpha_s^j-\frac{1}{2}\sum_{s=1}^{n}\sum_{t=1}^{n}\alpha_s^j\alpha_t^j\mathbf{M}_{y_sj}\mathbf{M}_{y_tj}x_s^T(\mathbf{I}+\lambda_s\mathbf{X}^j\mathbf{L}^j\mathbf{X}^{jT})^{-1}x_t\\ \text{s.t.}\quad & \sum_{s=1}^{n}\alpha_s^j\mathbf{M}_{y_sj}=0\\ & 0\leq\alpha_s^j\leq\lambda,\quad\forall s=1...n\end{aligned}\tag{14}$$

It can be easily observed that Eq. (14) replaces each original $x_s^Tx_t$ by $x_s^T(\mathbf{I}+\lambda_s\mathbf{X}^j\mathbf{L}^j\mathbf{X}^{jT})^{-1}x_t$, amounting to defining a new similarity criteria between instances, which actually considers different weights for individual features by the Laplacian matrix with parameter $\lambda_s$. The optimization problem in Eq. (14) is a QP problem, which can be solved by any standard QP solvers. Then the classification score for an arbitrary instance $x$ by the $j$th binary classifier can be formulated as

$$f_j(x)=\sum_{s=1}^{n}\alpha_s^j\mathbf{M}_{y_sj}x_s^T(\mathbf{I}+\lambda_s\mathbf{X}^j\mathbf{L}^j\mathbf{X}^{jT})^{-1}x+b_j\tag{15}$$

Similarly, suppose a nonlinear (implicit) kernel mapping $\phi:\mathcal{R}^d\rightarrow\mathscr{H}$, the new optimization problem in the kernel space can be written as

$$\begin{aligned}\max_{\alpha_1^j,\ldots\alpha_s^j}\quad & \sum_{s=1}^{n}\alpha_s^j-\frac{1}{2}\sum_{s=1}^{n}\\ & \sum_{t=1}^{n}\alpha_s^j\alpha_t^j\mathbf{M}_{y_sj}\mathbf{M}_{y_tj}\phi(x_s)^T(\mathbf{I}+\lambda_s\mathbf{X}^{j\Phi}\mathbf{L}^j\mathbf{X}^{j\Phi^T})^{-1}\phi(x_t)\\ \text{s.t.}\quad & \sum_{s=1}^{n}\alpha_s^j\mathbf{M}_{y_sj}=0\\ & 0\leq\alpha_s^j\leq\lambda,\forall s=1,...,n\end{aligned}\tag{16}$$

where $\mathbf{X}^{j\Phi}$denotes data matrix for the $j$th binary sub-problem in the kernel space. Eq. (16) can be further formulated as

$$\begin{aligned}\max_{\boldsymbol{\alpha}^j}\quad & \boldsymbol{\alpha}^{j^T}\vec{\mathbf{1}}_n-\frac{1}{2}\boldsymbol{\alpha}^{j^T}\left[(\mathbf{K}_j-\lambda_s\mathbf{K}_j\mathbf{L}^j(\mathbf{L}^j+\lambda_s\mathbf{L}^j\mathbf{K}_j\mathbf{L}^j)^{-1}\mathbf{L}^j\mathbf{K}_j^T)\circ\mathbf{M}_{\mathbf{y}j}\mathbf{M}_{\mathbf{y}j}^T\right]\boldsymbol{\alpha}^j\\ \text{s.t.}\quad & \boldsymbol{\alpha}^{j^T}\mathbf{M}_{\mathbf{y}j}=0\\ & \vec{\mathbf{0}}_n\leq\boldsymbol{\alpha}^j\leq\lambda\vec{\mathbf{1}}_n\end{aligned}\tag{17}$$

and the detailed derivation can be seen in Appendix B.

Clearly, the presented methodology can be applied to any ECOC approaches including OVO (in that case, one actually considers the individual structure of the two classes in each binary sub-problem). Moreover, other loss functions can also be adopted in this methodology.

## 4. Experiments

In this section, we validate the proposed methodology over both toy and real benchmark datasets. Through adopting the hinge loss function, we actually adopt SVM as the base binary classifier, which is among the most powerful and commonly-used binary classification methods, and we resort to the LIBSVM [25] toolbox for its learning. For the base ECOC approaches, we adopt OVA using the maximum-classification-score strategy for testing new instances [1], and ternary DECOC using Hamming distance for testing new instances [8]. We resort to the ECOCs library [15] for their implementations. Then the compared baselines are denoted as OVA_SVM and DECOC_SVM, and the corresponding new methods are CS_OSVM, MS_OSVM, CS_DSVM and MS_DSVM, which denote OVA_SVM incorporating structure knowledge under the cluster and manifold assumptions, respectively, and DECOC_SVM incorporating structure knowledge under the cluster and manifold assumptions, respectively. The RBF kernel is adopted over the toy dataset, and both linear and RBF kernels are adopted over the real datasets. The regularization parameters $\lambda$ and $\lambda_s$ are both selected by 5-fold cross validation from {0.001, 0.01, 0.1, 1, 10, 100, 1000}, and the width parameter of the RBF kernel is selected by 5-fold cross validation from {0.001, 0.01, 0.1, 1, 10, 100, 1000}$\times\sigma_0$, where $\sigma_0$ is the average distance between training instances. In what follows, we will present experiments on toy and real datasets in separate sub-sections, respectively.

### 4.1. Toy problem

In this sub-section, we evaluate the new methods by illustrations over two toy datasets, respectively following the cluster and manifold distribution structures.

#### 4.1.1. Cluster-structured toy dataset

For illustrating CS_OSVM and CS_DSVM (under the cluster assumption), we adopt a two-dimensional five-class toy dataset, in which each class follows a Guassian distribution. Each class has

40 instances, half of which are selected for training and the rest for testing, the attributes of this dataset is described in Table 1.

The training and testing performances of those compared classifiers are shown in Table 2, in which the bold value indicates the best performance among the three compared classifiers (OVA_SVM, CS_OSVM and MS_OSVM; DECOC_SVM, CS_DSVM and MS_DSVM). From Table 2, we can observe that both CS_OSVM and MS_OSVM perform better than OVA_SVM, and similarly, both CS_DSVM and MS_DSVM perform better than DECOC-SVM on both training and testing datasets. At the same time, CS_OSVM and CS_DSVM perform better than MS_OSVM and MS_DSVM, respectively. As a result, the utilization of already-provided structure knowledge in individual original classes can help improve the classification performance. Moreover, when real data distribution follows the cluster structure, methods utilizing structure knowledge under the cluster assumption will lead to better performance than those under the manifold assumption.

The superiority of CS_OSVM and CS_DSVM here can also be observed in Fig. 1, in which the decision boundaries of CS_OSVM and CS_DSVM are displayed, respectively, with those of OVA_SVM and DECOC_SVM as the baselines. As shown in Fig. 1, the instances in class 1 are distributed in a sphere, while the instances in class 3 are distributed in an ellipsoid with a horizontal long-axis, and the instances in class 4 are distributed in an ellipsoid with a vertical long-axis, consequently, the decision boundary (between class 1 and 3, 1 and 4) should be closer to class 1. At the same time, the instances in class 2 are distributed in an ellipsoid with a vertical long-axis, and the instances in class 5 are distributed in an ellipsoid with a horizontal long-axis, consequently, the decision boundary (between class 1 and 2, 1 and 5) should be farther to class 1. As a result, from Fig. 1, the decision boundaries of both CS_OSVM and CS_DSVM better capture the (cluster) structure of individual original classes, consequently are able to classify more instances correctly.

#### 4.1.2. Manifold-structured toy dataset

For illustrating MS_OSVM and MS_DSVM (under the manifold assumption), we adopt a two-dimensional toy dataset consisting of three concentric-circles, one circle each class. Each class has 200 instances, a half for training and the rest for testing.

The training and testing performances of individual classifiers are shown in Table 3, in which the bold value indicates the best performance among the three compared classifiers (OVA_SVM, CS_OSVM and MS_OSVM; DECOC_SVM, CS_DSVM and MS_DSVM). From Table 3, we can observe that both CS_OSVM and MS_OSVM perform better than OVA_SVM, and similarly, both CS_DSVM and MS_DSVM perform better than DECOC_SVM on both training and testing datasets. Moreover, MS_OSVM and MS_DSVM perform better than CS_OSVM and CS_DSVM, respectively. As a result, the already-provided structure knowledge in individual original classes is helpful for ECOC-based multi-class classification. At the same time, when real data distribution obeys the manifold assumption, methods utilizing structure knowledge under the manifold assumption will lead to better performance than those under the cluster assumption.

The superiority of MS_OSVM and MS_DSVM here can also be observed in Fig. 2, in which the decision boundaries of MS_OSVM and MS_DSVM are shown, respectively, with those of OVA_SVM and DECOC_SVM as the baselines. From Fig. 2, both MS_OSVM and MS_DSVM better capture the real data distribution than OVA_SVM and DECOC_SVM, respectively, thus are both able to classify more instances correctly.

### 4.2. Real problems

Those methods are further compared over both UCI and image recognition datasets, and the corresponding results are provided in separated sub-sections, respectively.

#### 4.2.1. UCI datasets

The attributes of the 12 multi-class UCI datasets are shown in Table 4, and the comparison results using the linear and RBF kernels are reported in Tables 5 and 6, respectively. In both Tables 5 and 6, the bold value in each row indicates that CS_OSVM/MS_OSVM performs better than OVA_SVM, or CS_DSVM/MS_DSVM performs better than DECOC_SVM, the value further marked by "∗" indicates that the better classifier achieves significant improvement by $t$-test (with the confidence interval at 95%), and the underlined value indicates the best performance among the three compared methods (OVA_SVM, CS_OSVM and MS_OSVM; DECOC_SVM, CS_DSVM and MS_DSVM).

From both Tables 5 and 6, we can make several observations as follows,

- When linear kernel is adopted, both CS_OSVM and MS_OSVM perform better than OVA_SVM, and similarly, both CS_DSVM and MS_DSVM perform better than DECOC-SVM. More specifically, CS_OSVM performs better than OVA_SVM on 10 out of the 12 datasets, achieves significant improvements on 5 ones, and performs comparably on the other 2 datasets. MS_OSVM achieves better performances than OVA_SVM on 11 datasets, significant improvements on 4 ones, and comparable performances on the other one dataset. At the same time, CS_DSVM performs better than DECOC_SVM on 10 datasets, achieves

**Table 1**
Attributes of the toy dataset.

| Class index | 1 ("×") | 2 ("∗") | 3 ("◇") | 4 ("·") | 5 ("○") |
|---|---|---|---|---|---|
| Mean | [0,0] | [1,1] | [−1, 1] | [0,2] | [0,1] |
| Covariance | $\begin{bmatrix}0.2 & 0\\ 0 & 0.03\end{bmatrix}$ | $\begin{bmatrix}0.03 & 0\\ 0 & 0.2\end{bmatrix}$ | $\begin{bmatrix}0.2 & 0\\ 0 & 0.03\end{bmatrix}$ | $\begin{bmatrix}0.03 & 0\\ 0 & 0.2\end{bmatrix}$ | $\begin{bmatrix}0.1 & 0\\ 0 & 0.1\end{bmatrix}$ |
| Instance | 40 instances in each class, 50% for training, and 50% for testing | | | | |

**Table 2**
The performances of compared classifiers on both training and testing datasets.

| Methods | OVA_SVM | CS_OSVM | MS_OSVM | DECOC_SVM | CS_DSVM | MS_DSVM |
|---|---|---|---|---|---|---|
| Tra. acc. | 0.92 | **0.94** | 0.93 | 0.93 | **0.94** | 0.93 |
| Tes. acc. | 0.87 | **0.89** | 0.88 | 0.86 | **0.89** | 0.87 |

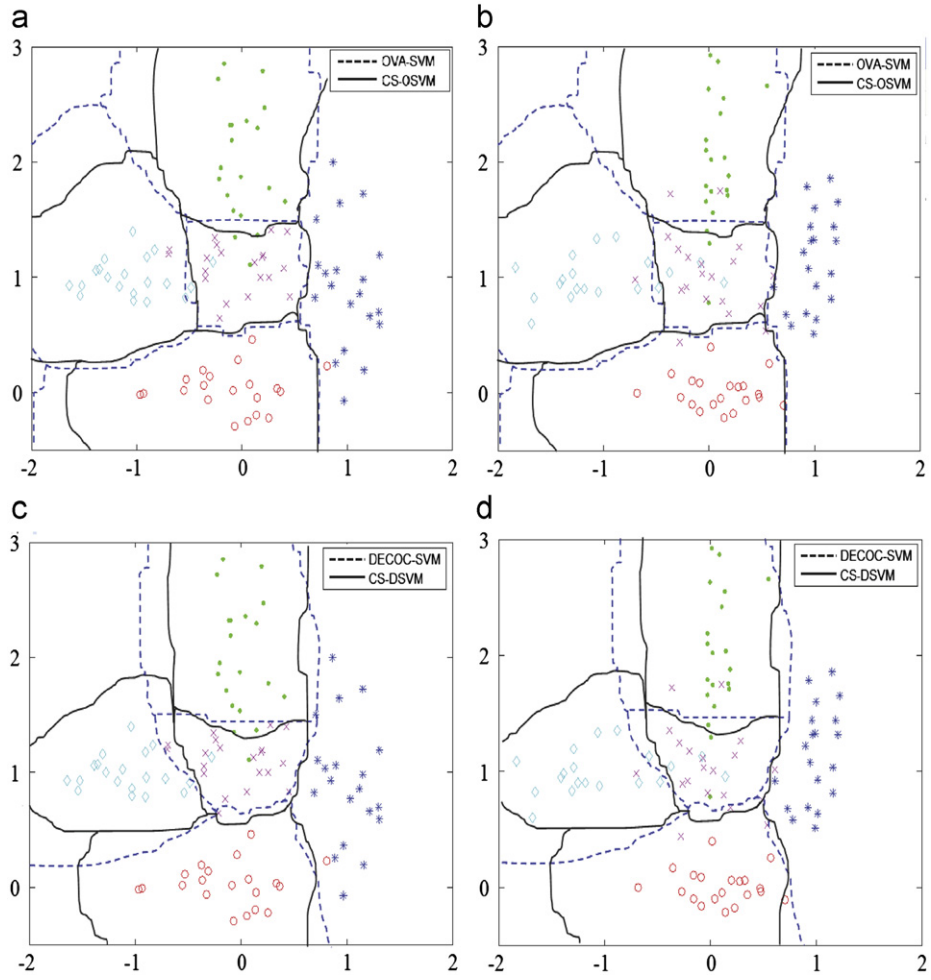


**Fig. 1.** The respective decision boundaries of OVA_SVM and CS_OSVM on (a) training and (b) testing datasets, as well as the respective decision boundaries of DECOC_SVM and CS_DSVM on (c) training and (d) testing datasets.

**Table 3**
The performances of compared classifiers on both training and testing datasets.

| Methods | OVA_SVM | CS_OSVM | MS_OSVM | DECOC_SVM | CS_DSVM | MS_DSVM |
|---|---|---|---|---|---|---|
| Tra. acc. | 0.96 | 0.9667 | **0.9733** | 0.9667 | 0.97 | **0.9767** |
| Tes. acc. | 0.9033 | 0.9133 | **0.9333** | 0.94 | 0.9464 | **0.9533** |

significant improvements on 5 ones, and performs comparably on the other 2 datasets. MS_DSVM also achieves better performances on 10 datasets, significant improvements on 6 ones, and comparable performances on the other 2 dataset. As a result, incorporating the structure knowledge of individual original classes can improve the performances of ECOCs over most datasets when linear kernel is adopted.

- When RBF kernel is adopted, both CS_OSVM and MS_OSVM perform better than OVA_SVM, and similarly, both CS_DSVM and MS_DSVM perform better than DECOC-SVM. More specifically, CS_OSVM performs better than OVA_SVM on 11 datasets, achieves significant improvements on 4 ones, and performs comparably on the other one datasets. MS_OSVM achieves better performances than OVA_SVM on 10 datasets, significant improvements on 4 ones, and comparable performances on the other 2 dataset. At the same time, CS_DSVM performs better than DECOC_SVM on 10 datasets, achieves significant improvements on 3 ones, and performs comparably on the other 2 datasets. MS_DSVM also achieves better performances on 11 datasets, significant improvements on 4 ones, and comparable performances on the other one dataset. As a result, incorporating the structure knowledge of individual original classes can improve the performances of ECOCs over most datasets when RBF kernel is adopted.
- CS_OSVM and CS_DSVM perform comparably with MS_OSVM and MS_DSVM with both linear and RBF kernels, respectively. More specifically, when linear kernel is adopted, CS_OSVM performs the best among the three compared methods over 6 datasets, MS_OSVM performs the best over 5 datasets. At the same time, CS_DSVM achieves the best performances among the three compared methods over 6 datasets, MS_DSVM performs the best over 6 datasets. When RBF kernel is adopted, CS_OSVM performs the best over 6 datasets, MS_OSVM performs the best over 6 datasets. At the same time, CS_DSVM performs the best over 6 datasets, MS_DSVM performs the best over 6 datasets. As a result, both cluster and manifold assumption are reasonable in real datasets. When data distribution is closer to cluster structures, methods incorporating structure knowledge with cluster

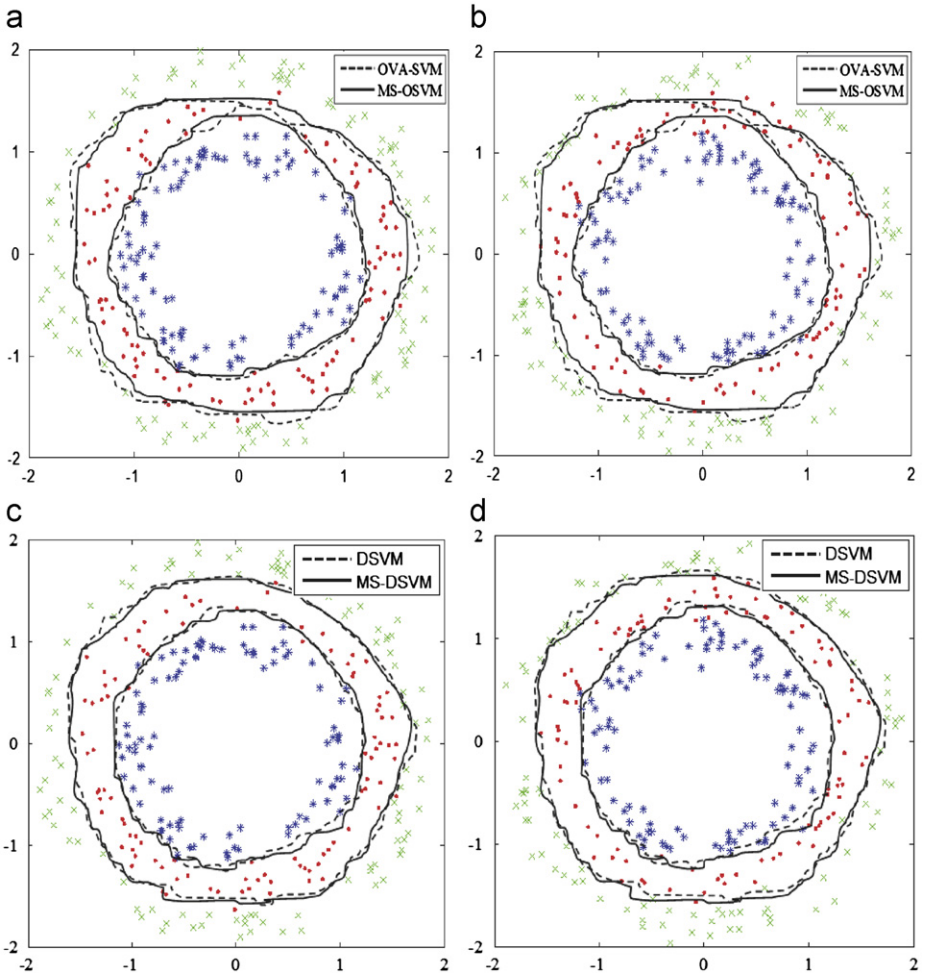


**Fig. 2.** The respective decision boundaries of OVA_SVM and MS_OSVM on (a) training and (b) testing datasets, as well as the respective decision boundaries of DECOC_SVM and MS_DSVM on (c) training and (d) testing datasets.

**Table 4**
The attributes of the 12 multi-class UCI datasets.

| Dataset | Class | Instance | Feature | Dataset | Class | Instance | Feature |
|---|---|---|---|---|---|---|---|
| Balance | 3 | 625 | 4 | Segmentation | 7 | 2310 | 19 |
| Cmc | 3 | 1473 | 9 | Tae | 3 | 151 | 5 |
| Dermatology | 6 | 366 | 33 | Vehicle | 4 | 846 | 18 |
| Ecoli | 8 | 336 | 8 | Vowel | 11 | 990 | 10 |
| Glass | 7 | 214 | 10 | Wine | 3 | 178 | 13 |
| Iris | 3 | 150 | 4 | Yeast | 10 | 1484 | 8 |

**Table 5**
The comparison results on multi-class UCI datasets using the linear kernel.

| Dataset (LINEAR) | OVA_SVM | CS_OSVM | MS_OSVM | DECOC_SVM | CS_DSVM | MS_DSVM |
|---|---|---|---|---|---|---|
| Balance | 88.1766 ± 0.0310 | **88.6396 ± 0.0106** | **88.8736 ± 0.0207** | 87.4644 ± 0.0060 | **89.4231 ± 0.0038*** | **88.7978 ± 0.0038*** |
| Cmc | 47.5737 ± .0060 | **49.8413 ± .0152*** | **47.9858 ± .0094** | 47.2102 ± .0121 | 47.1209 ± .0128 | **47.3462 ± .0116** |
| Dermatology | 97.1154 ± 0.0127 | **97.3901 ± 0.0041** | **97.4952 ± 0.0092** | 96.1770 ± 0.0134 | **97.2759 ± 0.0104*** | **97.8327 ± 0.0096*** |
| Ecoli | 87.7500 ± 0.0145 | 87.2982 ± 0.0098 | 87.5613 ± 0.0104 | 86.1446 ± 0.0017 | **86.5783 ± 0.0129** | 85.8731 ± 0.0213 |
| Glass | 92.8027 ± 0.0980 | **94.1633 ± 0.0184*** | **93.7321 ± 0.0336** | 94.4218 ± 0.0406 | **95.9184 ± 0.0082*** | **95.8362 ± 0.0104*** |
| Iris | 95.0000 ± 0.0495 | 94.8333 ± 0.0378 | **95.1823 ± 0.0209** | 95.6667 ± 0.0190 | **96.8333 ± 0.0175** | **96.9637 ± 0.0112*** |
| Segmentation | 91.6017 ± 0.0242 | **92.6883 ± 0.0187** | **92.5412 ± 0.0203** | 92.8973 ± 0.0327 | **93.0465 ± 0.0201** | **93.2142 ± 0.0198** |
| Tae | 52.7619 ± 0.0821 | **53.7143 ± 0.0339** | **53.8952 ± 0.0224*** | 49.7143 ± 0.1583 | **50.0952 ± 0.0643** | **50.5347 ± 0.0306** |
| Vehicle | 77.7962 ± 0.0221 | **78.2227 ± 0.0210** | **78.4142 ± 0.0189** | 77.5924 ± 0.0216 | 77.1327 ± 0.0187 | **77.6421 ± 0.0146** |
| Vowel | 51.2121 ± 0.0224 | **52.5758 ± 0.0281*** | **52.3354 ± 0.0196*** | 57.4411 ± 0.0026 | **60.6061 ± 0.0306*** | **59.4241 ± 0.0096*** |
| Wine | 96.5909 ± .0148 | **97.7273 ± 0.0074*** | **97.2826 ± 0.0096*** | 96.4545 ± 0.0129 | **97.7273 ± 0.0086*** | **96.8487 ± 0.0074*** |
| Yeast | 53.0447 ± .0718 | **55.4804 ± .0366*** | **55.2628 ± .0428*** | 57.9161 ± .0015 | **58.1863 ± .0012** | 57.7895 ± .0026 |

**Table 6**
The comparison results on multi-class UCI datasets using the RBF kernel.

| Dataset (RBF) | OVA_SVM | CS_OSVM | MS_OSVM | DECOC_SVM | CS_DSVM | MS_DSVM |
|---|---|---|---|---|---|---|
| Balance | 94.7736 ± 0.0074 | 94.7456 ± 0.0082 | **94.8126 ± 0.0081** | 94.4632 ± 0.0571 | **96.5759 ± 0.0178*** | **95.7862 ± 0.0442*** |
| Cmc | 54.3878 ± 0.0232 | **56.2245 ± 0.0188*** | **56.1108 ± 0.0201*** | 53.5123 ± 0.0306 | **53.7325 ± 0.0301** | **53.5758 ± 0.0372** |
| Dermatology | 96.9388 ± 0.0280 | **97.4097 ± 0.0128** | **97.8863 ± 0.0142** | 96.7712 ± 0.0161 | **97.7083 ± 0.0101** | **97.9328 ± 0.0093*** |
| Ecoli | 87.7912 ± 0.0299 | **88.2579 ± 0.0167** | 87.5368 ± 0.0232 | 88.0506 ± 0.0415 | **88.2341 ± 0.0397** | **88.4321 ± 0.0286** |
| Glass | 98.0952 ± 0.0104 | **99.1430 ± 0.0097*** | **99.2481 ± 0.0093*** | 98.7632 ± 0.0122 | **99.0809 ± 0.0016** | **98.9643 ± 0.0093** |
| Iris | 97.6333 ± 0.0546 | **98.8200 ± 0.0276** | **98.1245 ± 0.0208** | 98.1312 ± 0.0284 | **98.9342 ± 0.0111** | **98.7459 ± 0.0192** |
| Segmentation | 95.1515 ± 0.0602 | **95.7215 ± 0.0319*** | **95.9623 ± 0.0408*** | 94.6517 ± 0.0438 | **95.9872 ± 0.0329** | **96.1978 ± 0.0225** |
| Tae | 55.3324 ± 0.0602 | **56.2767 ± 0.0720** | **57.3328 ± 0.0348*** | 54.1312 ± 0.3010 | **55.6748 ± 0.2717** | **55.9912 ± 0.2862*** |
| Vehicle | 81.0394 ± 0.0573 | **82.4948 ± 0.0426*** | **81.6956 ± 0.0565** | 82.0120 ± 0.0133 | **83.5758 ± 0.0179*** | **82.1108 ± 0.0142** |
| Vowel | 96.8687 ± 0.0072 | **97.3737 ± 0.0086** | **97.1816 ± 0.0076** | 97.4212 ± 0.0037 | 97.3203 ± 0.0018 | 97.0102 ± 0.0036 |
| Wine | 96.1410 ± 0.0293 | **96.5987 ± 0.0201** | 95.7859 ± 0.0332 | 94.7739 ± 0.0407 | **96.1432 ± 0.0121*** | **95.8832 ± 0.0221*** |
| Yeast | 61.2991 ± .0619 | **61.5020 ± .0484** | **61.8742 ± .0339** | 59.6663 ± .0023 | 59.2346 ± 0.0033 | **59.7975 ± .0016** |


**Fig. 3.** Sample images of 20 objectives in the COIL-20 database.

assumption would bring better performance, otherwise, when data distribution is closer to manifold structures, methods incorporating structure knowledge with manifold assumption would perform better. However, selecting a suitable assumption actually depends on more prior knowledge about real data distribution.

- In both tables, the performances of OVA are comparable with those of DECOC on most datasets adopted, which is consistent with the conclusion that when binary classifiers are well-tuned regularized classifiers such as SVM, a simple OVA scheme is as accurate as any other approaches [1,26].

*4.2.2. Image recognition*

Image recognition is a popular classification task in pattern recognition. In this sub-section, we use two image recognition datasets, i.e., COIL-20 [27] and Yale [28], corresponding to objective and face recognition, respectively, to validate our proposed methods.

COIL-20 is a database containing gray-scale images of 20 objects, as shown in Fig. 3 [29]. The objects were placed on a motorized turntable against a black background. The turntable was rotated through 360 degrees to vary the object pose with respect to a fixed camera. Images of the objects were taken at pose intervals of 5 degrees, which corresponds to 72 images per object. In our experiments, we resize each image to $32 \times 32$ pixels. For each object, we randomly select 10 images for training, and the rest for testing. The process along with the corresponding classifier learning is repeated 20 times, and the average performances are reported in Fig. 5.

The Yale Face Database contains 165 grayscale images of 15 persons, 11 images per person. Images for each person are taken at different facial expressions or configurations. Fig. 4 [30] shows the 11 images for one of the 15 persons. In our experiments, we resize each image to $32 \times 32$ pixels. For each person, we randomly select 5 images for training, and the rest for testing. This process along with the corresponding classifier learning is repeated 20 times, and the average performances are reported in Fig. 6.

From both Figs. 5 and 6, we can observe that both CS_OSVM and MS_OSVM perform better than OVA_SVM, and both CS_DSVM and MS_DSVM perform better than DECOC-SVM, which implies

**Fig. 4.** Sample images for one person in the Yale dataset.

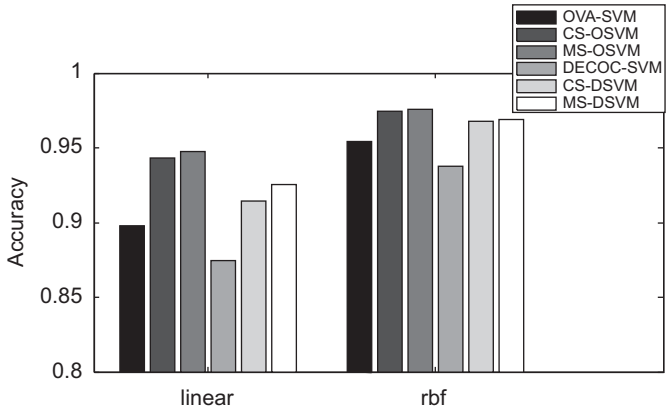


**Fig. 5.** Testing accuracies of compared methods with linear and RBF kernels on COIL-20 dataset, respectively.

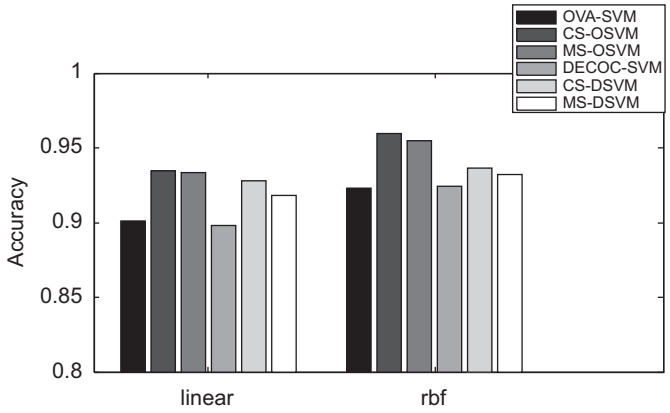


**Fig. 6.** Testing accuracies of compared methods with linear and RBF kernels on Yale dataset, respectively.

that the utilization of structure knowledge can boost the performance of ECOCs. Moreover, as shown in Fig. 5, the classification accuracies of MS_OSVM/MS-DSVM are better than those of CS_OSVM/CS_DSVM on COIL-20, since COIL-20 naturally implies a low-dimensional intrinsic manifold structure, on which the neighboring samples are small transformations of one another [31]. At the same time, as shown in Fig. 6, CS_OSVM/CS_DSVM yields better performances than MS_OSVM/MS-DSVM, implying that the Yale dataset is more likely prone to the cluster structure than the manifold structure.

## 5. Conclusion

In the off-the-shelf ECOC approaches, each class in the binary sub-problems could be a "meta-class" containing several original-classes, and treated as a single class in the implementation, naturally resulting in the under-exploitation of prior structure knowledge in individual original-classes. In this paper, we develop a methodology for utilizing such structure knowledge so as to show that it is helpful for ECOC-based multi-class classification. We formulate the structure knowledge under the cluster and manifold assumptions, respectively, corresponding to incorporating manners similar to those in SRSVM and LapSVM, respectively. Finally, we validate our methodology, and consequently the structure knowledge by encouraging results on both toy and real benchmark datasets. Of course, other formulations of structure knowledge or incorporating manners can also be adopted, as well as other forms of prior knowledge in individual original classes. At the same time, we adopt the class granularity in utilizing structure knowledge with the cluster assumption, while more delicate granularity, such as cluster-granularity,[3] can be used as well to further exploit the underlying sub-structures of clusters within each original class.

## Acknowledgments

This work is partially supported by Natural Science Foundations of China Grant Nos. 60973097, 61170151, 61035003 and 60905002, and NUAA Research Funding No. ns2010233.

## Appendix A

The optimization problem in Eq. (11) can be equivalently written as

$$\begin{aligned}\max_{\boldsymbol{\alpha}^j}\quad & \boldsymbol{\alpha}^{jT}\vec{\mathbf{1}}_n-\frac{1}{2}\boldsymbol{\alpha}^{jT}\left[\mathbf{X}^{j\Phi^T}(\mathbf{I}_n+\lambda_s\mathbf{V}^{j\Phi})^{-1}\mathbf{X}^{j\Phi}\circ\mathbf{M}_{\mathbf{y}j}\mathbf{M}_{\mathbf{y}j}^T\right]\boldsymbol{\alpha}^j\\ \text{s.t.}\quad & \boldsymbol{\alpha}^{jT}\mathbf{M}_{\mathbf{y}j}=0\\ & \vec{\mathbf{0}}_n\le\boldsymbol{\alpha}^j\le\lambda\vec{\mathbf{1}}_n\end{aligned}\tag{18}$$

and the $k$th covariance matrix in the $j$th binary sub-problem in the kernel space can be formulated as

$$\begin{aligned}\mathbf{V}_k^{j\Phi}&=\sum_{x_s\in\Omega_k^j}(\phi(x_s)-u_k^{j\Phi})(\phi(x_s)-u_k^{j\Phi})^T/|\Omega_k^j|\\&=\mathbf{X}_k^{j\Phi}\mathbf{X}_k^{j\Phi^T}/|\Omega_k^j|-\mathbf{X}_k^{j\Phi}\vec{\mathbf{1}}_{|\Omega_k^j|}\vec{\mathbf{1}}_{|\Omega_k^j|}^T\mathbf{X}_k^{j\Phi^T}\end{aligned}\tag{19}$$

where $\Omega_k^j$ denotes the set of instances belonging to the $k$th original-class in the $j$th binary sub-problem, $|\Omega_k^j|$ denotes the number of instances in $\Omega_k^j$, $\mathbf{X}_k^{j\Phi}$ and $u_k^{j\Phi}$ denote the data matrix and class mean of the $k$th class in the $j$th sub-problem in the kernel space, respectively, and $\vec{\mathbf{1}}_{|\Omega_k^j|}$ denotes a $|\Omega_k^j|$-dimensional vector with all components equaling to $1/|\Omega_k^j|$, then we have

$$\begin{aligned}\mathbf{V}^{j\Phi}&=\textstyle\sum_{k=1}^{C_j}\mathbf{V}_k^{j\Phi}\\&=\textstyle\sum_{k=1}^{C_j}\mathbf{X}_k^{j\Phi}\mathbf{X}_k^{j\Phi^T}/|\Omega_k^j|-\mathbf{X}_k^{j\Phi}\vec{\mathbf{1}}_{|\Omega_k^j|}\vec{\mathbf{1}}_{|\Omega_k^j|}^T\mathbf{X}_k^{j\Phi^T}\\&=\left[\mathbf{X}_1^{j\Phi}\cdots\mathbf{X}_{C_j}^{j\Phi}\right]\begin{bmatrix}\mathbf{I}_{|\Omega_1^j|}/|\Omega_1^j|-\vec{\mathbf{1}}_{|\Omega_1^j|}\vec{\mathbf{1}}_{|\Omega_1^j|}^T & & \\ & \ddots & \\ & & \mathbf{I}_{|\Omega_{C_j}^j|}/|\Omega_{C_j}^j|-\vec{\mathbf{1}}_{|\Omega_{C_j}^j|}\vec{\mathbf{1}}_{|\Omega_{C_j}^j|}^T\end{bmatrix}\begin{matrix}\mathbf{X}_1^{j\Phi^T}\\ \vdots\\ \mathbf{X}_{C_j}^{j\Phi^T}\end{matrix}\\&\triangleq\mathbf{P}^{j\Phi}\boldsymbol{\Lambda}\mathbf{P}^{j\Phi^T}\end{aligned}\tag{20}$$

where $\mathbf{I}_{|\Omega_1^j|}$ is a $|\Omega_k^j|\times|\Omega_k^j|$ diagonal matrix.

[3] Cluster granularity refers to that data structure within each class is depicted by a certain amount of clusters.

Further, from the Woodbury's formula [32],

$$(A+UBV)^{-1}=A^{-1}-A^{-1}UB(B+BVA^{-1}UB)^{-1}BVA^{-1} \tag{21}$$

we have

$$\begin{aligned}(\mathbf{I}+\lambda_s\mathbf{V}^{j\Phi})^{-1}&=(\mathbf{I}+\lambda_s\mathbf{P}^{\Phi}\mathbf{\Lambda}\mathbf{P}^{\Phi^T})^{-1}\\&=\mathbf{I}-\lambda_s\mathbf{P}^{\Phi}\mathbf{\Lambda}(\mathbf{\Lambda}+\lambda_s\mathbf{\Lambda}\mathbf{P}^{\Phi^T}\mathbf{P}^{\Phi}\mathbf{\Lambda})^{-1}\mathbf{\Lambda}\mathbf{P}^{\Phi^T}\end{aligned} \tag{22}$$

By substituting Eq. (22) into the objective function of Eq. (11), we have the formulation of Eq. (12).

## Appendix B

The optimization problem in Eq. (16) can be equivalently written as

$$\begin{aligned}\max_{\boldsymbol{\alpha}^j}\quad&\boldsymbol{\alpha}^{jT}\vec{\mathbf{1}}_n-\frac{1}{2}\boldsymbol{\alpha}^{jT}\left[\mathbf{X}^{j\Phi T}(\mathbf{I}+\lambda_s\mathbf{X}^{j\Phi}\mathbf{L}^j\mathbf{X}^{j\Phi^T})^{-1}\mathbf{X}^{j\Phi}\circ\mathbf{M}_{\mathbf{y}j}\mathbf{M}_{\mathbf{y}j}^T\right]\boldsymbol{\alpha}^j\\\text{s.t.}\quad&\boldsymbol{\alpha}^{jT}\mathbf{M}_{\mathbf{y}j}=0\\&\vec{\mathbf{0}}_n\le\boldsymbol{\alpha}^j\le\lambda\vec{\mathbf{1}}_n\end{aligned} \tag{23}$$

Further, from the Woodbury's formula [32], we have

$$(\mathbf{I}+\lambda_s\mathbf{X}^{j\Phi}\mathbf{L}^j\mathbf{X}^{j\Phi^T})^{-1}=\mathbf{I}-\lambda_s\mathbf{X}^{j\Phi}\mathbf{L}^j(\mathbf{L}^j+\lambda_s\mathbf{L}^j\mathbf{X}^{j\Phi^T}\mathbf{X}^{j\Phi}\mathbf{L}^j)^{-1}\mathbf{L}^j\mathbf{X}^{j\Phi^T} \tag{24}$$

By substituting Eq. (24) into the objective function of Eq. (16), we have the formulation of Eq. (17).

## References


[1] R. Rifkin, A. Klautau, In defense of One-Vs-All classification, J. Mach. Learn. Res. 5 (2004) 101–141.
[2] O. Pujol, S. Escalera, P. Radeva, An incremental node embedding technique for error correcting output codes, Pattern Recognit 41 (2008) 713–725.
[3] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, New York, 2006.
[4] J. Weston, C. Watkins, Multi-Class Support Vector Machines, in: Technical Report CSD-TR-98-04, University of London, Department of Computer Science, Royal Holloway, 1998.
[5] E. Bredensteiner, K.P. Bennett, Multicategory classification by support vector machines, Comput. Optim. Appl. 12 (1999) 53–79.
[6] K. Crammer, Y. Singer, On the algorithmic implementation of multiclass kernel-basd vector machines, J. Mech. Learn. Res. 2 (2001) 265–292.
[7] E. Allwein, R. Schapire, Y. Singer, Reducing multiclass to binary: A unifying approach for margin classifiers, J. Mach. Learn. Res. 1 (2002) 113–141.
[8] O. Pujol, P. Radeva, J. Vitria, Discriminant ECOC: a Heuristic method for application dependent design of error correcting output codes, IEEE Trans. Pattern Anal. Mach. Intell. 28 (2006) 1007–1012.
[9] S. Escalera, D.M.J. Tax, O. Pujol, P. Radeva, R.P.W. Duin, Subclass problem-dependent design for error correcting output codes, IEEE Trans. Pattern Anal. Mach. Intell. 30 (2008) 1041–1054.
[10] J.H. Friedman, Another Approach to Polychotomous Classification, Technical Report, Stanford University, 1996.
[11] T. Dietterich, G. Bakiri, Solving multiclass learning problems via error correcting output codes, J. Artif. Intell. Res. 2 (1995) 263–282.
[12] T. Hastie, R. Tibshirani, Classification by pairwise coupling, Ann. Stat. 26 (1998) 451–471.
[13] B. Zadrozny, Reducing multi class to binary by coupling probability estimates, in: Advances in Neural Information Processing Systems, MIT Press, Vancouver, British Columbia, Canada, 2001, pp. 1041–1048.
[14] T. Takenouchi, S. Ishii, Ternary Bradley–Terry model-based decoding for multi-class classification and its extensions Mach. learn. (2011) 1–24.
[15] S. Escalera, O. Pujol, P. Radeva, Error correcting ouput codes library, J. Mach. Learn. Res. 11 (2010) 661–664.
[16] D. Luo, R. Xiong, An improved error correcting output coding framework with kernel-based decoding, Neurocomputing 71 (2008) 3131–3139.
[17] O. Bousquet, S. Boucheron, G. Lugosi, Introduction to statistical learning theory, Adv. Lect. Mach. Learn. (2004) 169–207.
[18] H. Xue, S. Chen, Q. Yang, Structural regularized support vector machine: a framework for structural large margin classifier, IEEE Trans. Neural Networks 22 (2011) 573–584.
[19] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: a geometric framework for learning from labeled and unlabeled examples, J. Mach. Learn. Res. 7 (2006) 2399–2434.
[20] A.N. Tikhonov, On solving incorrectly posed problems and method of regularization, Dokl. Akademii Nauk USSR 151 (1963) 501–504.
[21] A.N. Tikhonov, V.Y. Aresnin, Solutions of Ill-Posed Problems, W.H. Winston, Washington, DC, 1977.
[22] X. Zhu, Semi-Supervised Learning literature survey, Computer Sciences, University of Wisconsin-Madison, MA, 2008.
[23] M.L. Minsky, S.A. Papert, Perceptrons (1969).
[24] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods, Cambridge University Press, UK, 2000.
[25] C.C. Chang, C.J. Lin, LIBSVM: A Library for Support Vector Machines, Software Available at http://www.csie.ntu.edu.tw/'cjlin/libsvm, 2001.
[26] Y. Shiraishi, Game theoretical analysis of the simple one-vs.-all classifier, Neurocomputing 71 (2008) 2747–2753.
[27] S.A. Nene, S.K. Nayar, H. Murase, Columbia Object Image Library (COIL-20), in: Technical Report CUCS-005-96, February, 1996.
[28] The Yale Face Database. Avaliable at http://cvc.yale.edu/projects/yalefaces/yalefaces.html.
[29] T. Sun, S. Chen, Locality preserving CCA with applications to data visualization and pose estimation, Image Vision Comput. 25 (2007) 531–543.
[30] B. Yang, S. Chen, Sample-dependent graph construction with application to dimensionality reduction, Neurocomputing 74 (2010) 301–314.
[31] A. Ghodsi, J. Huang, F. Southey, D. Schuurmans, Tangent-corrected embedding, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2005.
[32] M.A. Woodbury, Inverting modified matrices, in: memorandum report 42, Statistical Research Group, NJ, Princeton, 1950.


**Yunyun Wang** received the B.Sc. degree in computer applications from Anhui Normal University in 2006. Currently she is a Ph.D. student at the Department of Computer Science and Engineering in Nanjing University of Aeronautics and Astronautics (NUAA).

Her research interests focus on pattern recognition and machine learning.

**Songcan Chen** received the B.Sc. degree in mathematics from Hangzhou University (now merged into Zhejiang University) in 1983. In December 1985, he completed the M.Sc. degree in computer applications at Shanghai Jiaotong University and then worked at Nanjing University of Aeronautics and Astronautics (NUAA) in January 1986 as an assistant lecturer. There he received a Ph.D. degree in communication and information systems in 1997. Since 1998, as a full professor, he has been with the Department of Computer Science and Engineering at NUAA.

His research interests include pattern recognition, machine learning and neural computing. In these fields, he has authored or coauthored over 130 scientific journal papers.

**Hui Xue** received her B.S. degree in mathematics from Nanjing Normal University in 2002. In 2005, she received her M.S. degree in mathematics from Nanjing University of Aeronautics & Astronautics (NUAA). And she also received her Ph.D. degree in computer application technology at NUAA in 2008. Since 2009, as a university instructor, she has been with the School of Computer Science & Engineering at Southeast University.

Her research interests include pattern recognition, image processing and neural computing.