SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

January 2016, Vol. 59 012107:1–012107:14
doi: 10.1007/s11432-015-5321-y

Constructing ECOC based on confusion matrix for multiclass learning problems

Jindeng ZHOU*, Yun YANG, Mingjie ZHANG & Haibo XING

Science and Technology on Complex Aviation Systems Simulation Laboratory, Beijing 100076*, China*

Received December 24, 2014; accepted February 11, 2015; published online December 4, 2015

Abstract In the pattern recognition field, error-correcting output codes (ECOC) are a powerful tool to fuse any number of binary classifiers to model multiclass problems, and the research of encoding based on data is attracting more and more attention. In this paper, we are going to propose a new encoding method for constructing subclass Error-Correcting Output Codes, which was first introduced by Escalera et al. To achieve this goal, we first obtain the correlation between each pair of classes with the help of confusion matrix. Then, we select the most easily separated subclasses for classification by following Fisher's principle. At last, we were able to obtain binary partitions based on subclasses. After finishing this work, a new data-driven coding matrix-Subclass ECOC will be achieved. Experimental results on University of CaliforniaIrvine data sets and three kinds of high resolution range profile data sets with logistic linear classifier and support vector machine as the binary classifiers show that our approach can provide a better performance and the robustness of classification with a little longer but acceptable code length.

Keywords machine learning, multiclass classification, error correcting output codes, subclass partition, confusion matrix

Citation Zhou J D, Yang Y, Zhang M J, et al. Constructing ECOC based on confusion matrix for multiclass learning problems. Sci China Inf Sci, 2016, 59(1): 012107, doi: 10.1007/s11432-015-5321-y

1 Introduction

The binary classification problem in pattern recognition has achieved many research results so far, and there are lots of state-of-the-art dichotomizers, such as support vector machine (SVM), Back Propagation (BP), and Adaboost, which have been in use for a long time. However, it is still an open issue about how to keep or get closer to the classification performance when we extend dichotomizer into multiclass classifier. For this problem, a popular approach is to reduce a multiclass problem into a set of binary problems and fuse these dichotomizers' results by voting rules. As a kind of multiclass classification framework, error-correcting output codes (ECOC) can decompose a multiclass problem into a set of binary problems effectively, then construct dichotomizer for each binary partition. In so doing, the state-of-the-art binary classifiers can be used for multiclass classification [1]. The ECOC has been successfully applied to a wide range of fields as reported in human face recognition [2,3], text recognition [4], and traffic sign recognition [5,6].

* Corresponding author (email: zhoujin198417@163.com)

In the domain of multiclass classification based on ECOC, the construction of the data-driven error-correcting output code is a special hotspot. The basic idea of these methods is to use the training data to guide the training process and, thus, to construct the coding matrix. Focusing on the binary problems that better fit the decision boundaries of a given data set, there are lots of researches that have been done by many famous scholars in this area. For example, Alpaydin et al. [7] proposed the back propagation algorithm by analyzing the data's feature in 1999. Utschick et al. [8] constructed the data-driven coding matrix by optimizing a maximal likelihood function with the rule of expectation maximization algorithm in 2001. Pujol et al. [6] put forward a discriminant ECOC (DECOC) which is based on the embedding of discriminant tree structures derived from the problem domain in 2006. The binary trees are built by looking for the partition that maximizes the mutual information between the data and their respective class labels. Escalera and Tax present a novel strategy to model multiclass classification problems using subclass information in the ECOC framework (subclass ECOC) in 2008. In this framework, complex problems are solved by splitting the original set of classes into subclasses and embedding the binary problems in a data-dependent ECOC design [9]. Recently, a general extension of the ECOC framework for the online learning scenario is shown by Escalera et al. [10], and this is the first time we are discussing about the learning ability of online ECOC. Simeone et al. [11] presented a way of introducing a reject rule to ECOC for constructing a classification system with rejection area. Nicolás et al. [12] made an empirical study of the general assumptions in the field that have not been fully assessed for ECOC, and it is also the first time we discuss some items that are more specific and some ones that are easily overlooked such as the influence of the base learner on the performance, the independence of binary classifiers, the relationship between binary classifier error and coding performance, and so on. Furthermore, there are some more new improvements in data-dependent ECOC demonstrating the suitability of the ECOC methodology to deal with multiclass classification problems [13]. As you can see in these references, all of these improvements promote the development of ECOC greatly.

In this paper, we present a novel strategy to construct a data-driven ECOC based on confusion matrix, which is entitled confusion matrix superclass ECOC (CMSECOC in short), and different from its original version shown by Escalera et al. in [9]. A primal difference is that the subclasses in the latter are obtained by splitting the original classes. However, in this paper, the subclasses are obtained by integrating the original classes. To distinguish from the former, we call these subclasses as superclasses and rename the subclass ECOC as superclass ECOC. In this way, the superclass which contains abundant information about similarity and difference among classes can be obtained. Thus, the new method proposed in this paper will be more effective and simpler. To achieve this goal, we use the confusion matrix as the basis of the measurement for classes' separability. Then, by abiding Fisher's principle, the classes with maximal margin will be separated into different superclass, and the ones with high similarity can be integrated into one superclass. In the end, the superclass ECOC can be obtained by combining these superclasses under some rules.

2 ECOC

2.1 Concepts

The motive for using ECOC to solve multiclass problem is that it can solve the problem by decomposing a complex multiclass classification into a set of binary classification. Binary ECOC is an original decomposing framework, and there are two symbols in this coding matrix which stand for a positive class and a negative class in each binary problem. By taking account of a special case, there are some classes that may be unhelpful to train some binary classifiers. In such a case, these classes should be ignored, and the corresponding codebits in matrix will be replaced by a zero symbol. Because of introducing zero symbols into binary ECOC, the binary ECOC is renamed to ternary ECOC. In Figure 1, the four state-of-the-art ECOCs are shown as (a) one-versus-all, (b) dense random [14], (c) one-versus-one, and (d) sparse random [14]. Each row of the binary ECOC matrix acts as a code word or label for class C_i $(i = 1, 2, 3, 4)$, and each column presents one kind of binary partition of the sample. The code word '1', '-1', and '0'

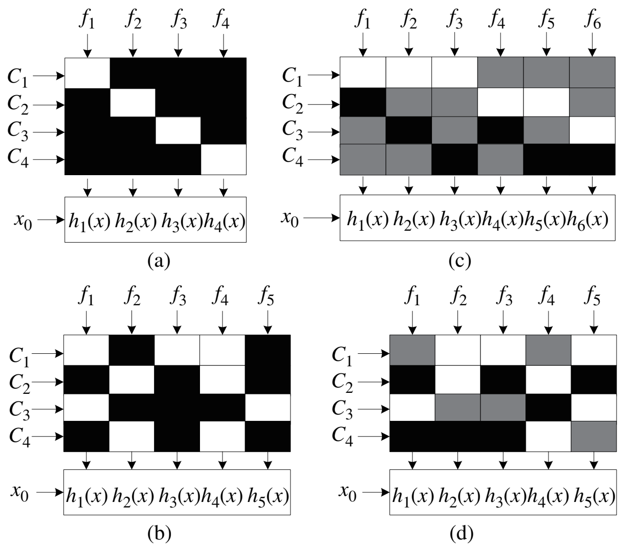


Figure 1 Four state-of-the-art ECOCs. Binary ECOC: (a) one-versus-all, and (b) dense random. Ternary ECOC: (c) one-versus-one, and (d) spare random.

are shown in white, black, and gray. In the training phase, each column of the matrix partitions the training data into two superclasses according to the value of the corresponding binary element, and a binary classifier f_i $(i = 1, 2, \ldots, 4, (5, 6))$ is obtained as a result [14]. For example, in Figure 1(d), when the base classifier f_3 is trained, classes C_2 and C_4 are used as positive classes, C_1 is used as negative class, and the rest of the classes C_3 in gray will be refused to take part in the training progress. Then, we can get five resulting binary classifiers $\{f_1, f_2, f_3, f_4, f_5\}$ in turns. In testing phase, a given testing sample will be classified by the four base classifiers mentioned before. The output is a code-word vector $(x_1, x_2, x_3, x_4, x_5)$, where $x_j \in \{-1, 1\}$. Finally, we use some decoding rules (fusion strategy) to make a fusion of all binary classification results, and then get the final classification result.

2.2 CMSECOC analysis

As pointed in [9], the key point of SECOC is how to generate the subgroups of problems that are split into simpler ones until the base classifier is able to learn the original problem. In this way, multiclass problems that cannot be modeled using the original set of classes are modeled without the need of using more complex classifiers. There are two guidelines that we need to take into consideration about the construction of CMSECOC: first, each one in the set of superclass can be separated easily from the rest of the ones (i.e., find the best superclass as shown in dashed box of Figure 2), then the result based on classifiers trained according to the relationship of superclasses can be obtained from a better performance. Second, the capability of error-correcting system among each of the classes in superclass as shown in each solid box of Figure 2 should be better, which means the redundancy information for classes in every superclass is the most reasonable.

Following the above-mentioned consideration, to construct CMSECOC, the class separability based on the training samples should be calculated quantificationally for the first order. To finish it, the measure of class separability is the key point. In the way of [9], mutual information is regarded as the concrete measure, and the sequential forward floating search based on maximizing the mutual information was used to generate the subclasses. Furthermore, the state-of-the-art measure methods of class separability include: (1) the measure based on geometric distance in feature space, for example, deviation matrix S_W, S_B, S_T, and their modified form $\mathrm{Tr}\lceil S_W^{-1} S_B \rceil, |S_B|/|S_W|$ [15]; (2) the measure based on probability distribution, for example, Bhattcharyya criterion J_B, Chernoff criterion J_C, and divergence criterion J_D [16]; and (3) the measure based on posterior probability, for example, the Shannon entropy J_H and the simpler one-general entropy [17]. It is clear that the method based on geometric distance is easily

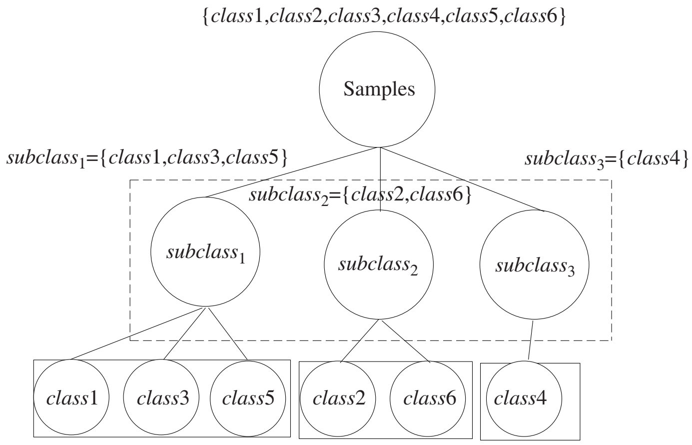


Figure 2 A superclass partition framework of six-class samples, and the elements in dashed boxes are the most easily separated. On the contrary, the elements in solid box are separated with most difficulty because they belong to the same superclass.

understood and less time-consuming, and it does not require abundant samples or the information of prior probability distribution to measure the class separability. However, it cannot clarify the overlaps of classes in each superclass. Although the method based on probability distribution or posterior probability can easily clarify the overlaps, it still highly depends on samples and probability information, so we have to select a measure method and the corresponding calculation strategy which is suitable for a given problem.

Furthermore, it is obvious that the classification error of the base classifier as shown in solid box of Figure 2 will be large, so it is necessary to enhance the error-correcting capability of these base classifiers. It is worth noting that the error-correcting capability depends on the degree of redundancy; the better the error-correcting ability, the larger the degree of redundancy, which requires longer coding length. Then, how to set the degree of redundancy should be carefully considered. An effective way is to take the separability among classes as a key to address the above-mentioned problem. Next, what we should do is to find the way of measure for class separability, which is the one among superclasses and the one among classes in each superclass. In the following sections, we will particularly discuss the issue, and then apply the approach to constructing CMSECOC.

3 Construction of CMSECOC based on confusion matrix

From Subsection 2.2 we know that, it is a crucial step for the metric of class separability. As to this point, confusion matrix was introduced to act as the measure. The correlation between each pair of classes is gained with the help of confusion matrix. Then, abiding by Fisher's principle, the superclasses containing better separability can be obtained. At last, we were able to obtain binary partitions based on the set of superclass. After finishing this works, a data-driven coding matrix will be ready.

3.1 Class separability measure based on confusion matrix

It is well known that a special recognition system has its own characteristics of errors, which are based on its underlying understanding of how some classes could often be mistaken for others. This kind of recognition characteristics are represented by a confusion matrix. From the viewpoint of knowledge, a confusion matrix could be regarded as the prior knowledge of a recognizer (classifier). In this section, we consider the class separability as the prior knowledge as mentioned before. Let us assume that there are N classes with T_i samples for each one ($i = 1, \ldots, N$) in pattern space D. Then, the confusion matrix of classifier C can be written as follows:

$$\begin{bmatrix} \mathrm{cm}_{11} & \mathrm{cm}_{12} & \cdots & \mathrm{cm}_{1i} & \cdots & \mathrm{cm}_{1N} \\ \mathrm{cm}_{21} & \mathrm{cm}_{22} & \cdots & \mathrm{cm}_{2i} & \cdots & \mathrm{cm}_{2N} \\ \vdots & \vdots & \ddots & \cdots & \ddots & \vdots \\ \mathrm{cm}_{i1} & \mathrm{cm}_{i2} & \cdots & \mathrm{cm}_{ii} & \cdots & \mathrm{cm}_{iN} \\ \vdots & \vdots & \ddots & \cdots & \ddots & \vdots \\ \mathrm{cm}_{N1} & \mathrm{cm}_{N2} & \cdots & \mathrm{cm}_{Ni} & \cdots & \mathrm{cm}_{NN} \end{bmatrix}, \tag{1}$$

where cm_{ij} denotes the percent of samples of class i which are assigned to class j. The row index i stands for the real class label of the sample, and the column index j stands for the predicted class label. The diagonal elements signify the classification accuracy of each class for classifier C, while the non-diagonal elements are the error rates of classification.

The classification accuracy of each class can be written as follows:

$$R_i = \mathrm{cm}_{ii},\ i = 1, \dots, N. \tag{2}$$

The classification error rate of each class can be written as follows:

$$W_i = \sum_{i=1}^{n} \mathrm{cm}_{ij} = 1 - \mathrm{cm}_{ii} = 1 - R_i. \tag{3}$$

If one class is similar to another, it will be easily misjudged from the opposite side. According to the research of [18], each row of confusion matrix C_i $(i = 1, \dots, N)$ reflects the probability that the samples of class i are classified into each class. So, we can define a similarity measurement matrix TM_2, the elements of which are the upper triangular ones of the L_2 metric matrix of CM, and the others are zeros [19]. Then, the definition can be written as follows:

$$\mathrm{tm}_{ij} = \begin{cases} 0, & i > j, \\ (C_i - C_j)^2 = \sum_{k=1}^{N} (\mathrm{cm}_{ik} - \mathrm{cm}_{jk})^2, & i \leqslant j. \end{cases} \tag{4}$$

The matrix of TM_2 reflects the similarity between each pair of classes. The smaller the value of tm_{ij} is, the more similar each pair of classes is, and the more easily they will be misclassified.

3.2 Superclass partition based on Fisher criterion

From the analysis in Subsection 2.2, we know that there are two advantages of CMSECOC. One is the better separability among superclasses, and the other is the higher capability of error correction in each superclass.

To achieve the above-mentioned goal, we should find out the partitions. The classes with high correlation should be assigned into the same superclass, and those with low correlation should be assigned into different superclasses. According to Fisher criterion, the classification could achieve the best performance when the samples have both the minimal inner-class distance and the maximal inter-class distance, so the superclass partition generated by Fisher criterion meets the demand of the first metric of CMSECOC naturally. What about the capability of error correction? For this problem, we consider the partition for inner class of each superclass that will get the higher ability of correcting in Subsection 3.3. In this section, we take the measure of class separability as key point and discuss the strategy of superclass partition based on Fisher criterion. Then, the resulting set of superclass which constitutes the final coding matrix can be obtained.

The steps of superclass partition are as follows:

Input: training sample R, pre-classifier C, the combining threshold α, the parting threshold β

Step 1. Classify the R by C and obtain the confusion matrix $\mathrm{CM}(C, R)$.

Step 2. Calculate the dissimilarity matrix based on (4), and normalize it with the following equation:

$$\acute{\mathrm{tm}}_{ij} = \mathrm{tm}_{ij}/t_{\max}, \text{ and } t_{\max} = \max_{i,j}(\mathrm{tm}_{ij}). \quad (5)$$

Step 3. Then, deal with the element in TM_2 as follows:

Step 3.1. If $\acute{\mathrm{tm}}_{ij} \leqslant \alpha$, it means that class i is very similar to pattern j. Then, combine the two classes to a set. After doing that, a superclass, i.e., $subclass_k = \{C_i, C_j\}$, comes into being.

Step 3.2. If $\acute{\mathrm{tm}}_{ij} \geqslant \beta$, it means that class i and class j have low correlations. Then, the two classes should be separated to get two superclasses, that is, $subclass_k = \{C_i\}$ and $subclass_{k+1} = \{C_j\}$.

Step 3.3. If $\alpha \leqslant \acute{\mathrm{tm}}_{ij} \leqslant \beta$, it means that the two classes' correlation is not clear. Then, we should reconsider the partition based on existent superclasses as follows:

Step 3.3.1. If there exist two superclasses, for example, $subclass_p$ and $subclass_q$, and $C_i \in subclass_p$, $C_j \in subclass_q$, then do not generate any new superclass.

Step 3.3.2. If only for class C_i or C_j, there exists a superclass which meets the conditions that $C_i \in subclass_p$ (or $C_j \in subclass_q$), then generate a new superclass $subclass_k = \{C_j\}$ (or $subclass_k = \{C_i\}$).

Step 3.3.3. If there does not exist any superclass to which class C_i and C_j belong to, then generate new superclasses $subclass_k = \{C_i\}$ and $subclass_{k+1} = \{C_j\}$.

Step 3.4. For two arbitrary superclasses $subclass_p$ and $subclass_q$, if $subclass_p \cap subclass_q \neq \emptyset$, then integrate the two superclasses into one.

Output: The set of superclasses $J = \{subclass_1, subclass_2, \ldots, subclass_k, \ldots, subclass_t\}$.

The superclass partition algorithm is shown as Algorithm 1.

Algorithm 1 Superclass partition algorithm

INPUT: combining threshold α, parting threshold β;
OUTPUT: superclass;
Require: $superclass = \emptyset, count = 1, length = 0, Issuperclass = true$;

```
for i = 1 to N do
  for j = 1 to N do
    if tm_ij < α then
      superclass[count++] = {C_i, C_j};
    else if tm_ij > β then
      superclass[count++] = {C_i};
      superclass[count++] = {C_j};
    else
      length = count − 1;
      For C_i and C_j do as following respectively, take C_i for example;
      for k = 1 to length do
        if C_i = superclass_k then
          Issuperclass=false;
          break;
        end if
      end for
      if Issuperclass = true then
        superclass[count++] = {C_i};
      end if
    end if
  end for
end for
Note: Both i and j denote the class identifier.
```

3.3 Construction of CMSECOC

In Subsection 3.2, we have discussed the strategy of superclass partition, based on which we can get the wanted superclasses with the help of Fisher criterion. In this section, we will analyze the construction of CMSECOC based on the superclasses learnt before.

To construct binary partitions which compose the columns of the ECOC matrix, there are two situations we have to consider: the recognition among superclasses and the recognition among inner classes of each superclass. We note that the degree of separability for inter-superclass (i.e., the degree of dissimilarity among superclasses) is large enough to distinguish one superclass from others, so the "one-versus-all" strategy will be adopted for construction. In so doing, we choose one superclass as the positive class, and the rest as negative class. Then, the binary partitions for superclasses' recognition can be obtained to form the columns of CMSECOC.

The "one-versus-one" strategy is used when classifying the inner classes of each superclass. In so doing, two classes selected from a superclass are used to act as the positive class and negative one, respectively, with the rest being ignored. There are two reasons for doing so. The first is that the classes which belong to the same superclass have higher similarity, and the resulting classification problems will be difficult to model, so we pay attention to the only two classes in superclasses each time with the encoding strategy of "one-versus-one". For another, to reduce the error of base classifiers, as we analyzed in Subsection 2.2, the reasonable increasing for class redundancy can enhance the capability of error correction for the recognition of inner-superclass. Next, we will discuss its usage in practice.

Take the example in Subsection 3.2 for explanation. As we know, the three superclasses are $subclass_1 = \{C_1, C_3\}$, $subclass_2 = \{C_2, C_4\}$, and $subclass_3 = \{C_5\}$. Using the "one-versus-all" strategy to make the partitions for inter-superclass, we can get the coding matrix as follows:

$$M_1 = \begin{bmatrix} 1 & -1 & -1 \\ -1 & 1 & -1 \\ 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \end{bmatrix}. \tag{6}$$

Next, we can get the coding matrix using "one-versus-one" strategy for inner-superclass follows:

$$M_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \\ 0 & 0 \end{bmatrix}. \tag{7}$$

At last, the CMSECOC can be achieved by combining the two matrixes as follows:

$$M = \begin{bmatrix} 1 & -1 & -1 & 1 & 0 \\ -1 & 1 & -1 & 0 & 1 \\ 1 & -1 & -1 & -1 & 0 \\ -1 & 1 & -1 & 0 & -1 \\ -1 & -1 & 1 & 0 & 0 \end{bmatrix}. \tag{8}$$

So far, an approach to constructing CMSECOC based on confusion matrix has come into being. Next, we will validate the classification performance through experiments.

4 Experiments

4.1 Experimental data set

In this section, we will validate our proposed approaches using two kinds of data sets: UCI data set [20] and high resolution range profile (HRRP) data set [21]. The characteristics of the UCI data sets are shown in Table 1. Meanwhile, the principal component analysis (PCA) is used to reduce the dimensionality to promote classification speed. On the other hand, the data set of HRRP used consists of three airplanes: B-52, Farmer, and Fishbed. It was acquired with zoom models in a microwave anechoic chamber, and

Table 1 Similarity measurement matrix of five-class samples

Problem	Train	Attributes	Classes	
(a)	Yeast	1484	8	10
(b)	Segmentation	2310	19	7
(c)	Satimage	6435	36	6
(d)	Glass	214	9	7
(e)	Vehicle	846	18	4
(f)	Zoo	101	18	7
(g)	Wine	178	13	3
(h)	Vowel	990	10	11
(i)	Ecoli	336	8	8
(j)	Iris	150	4	3

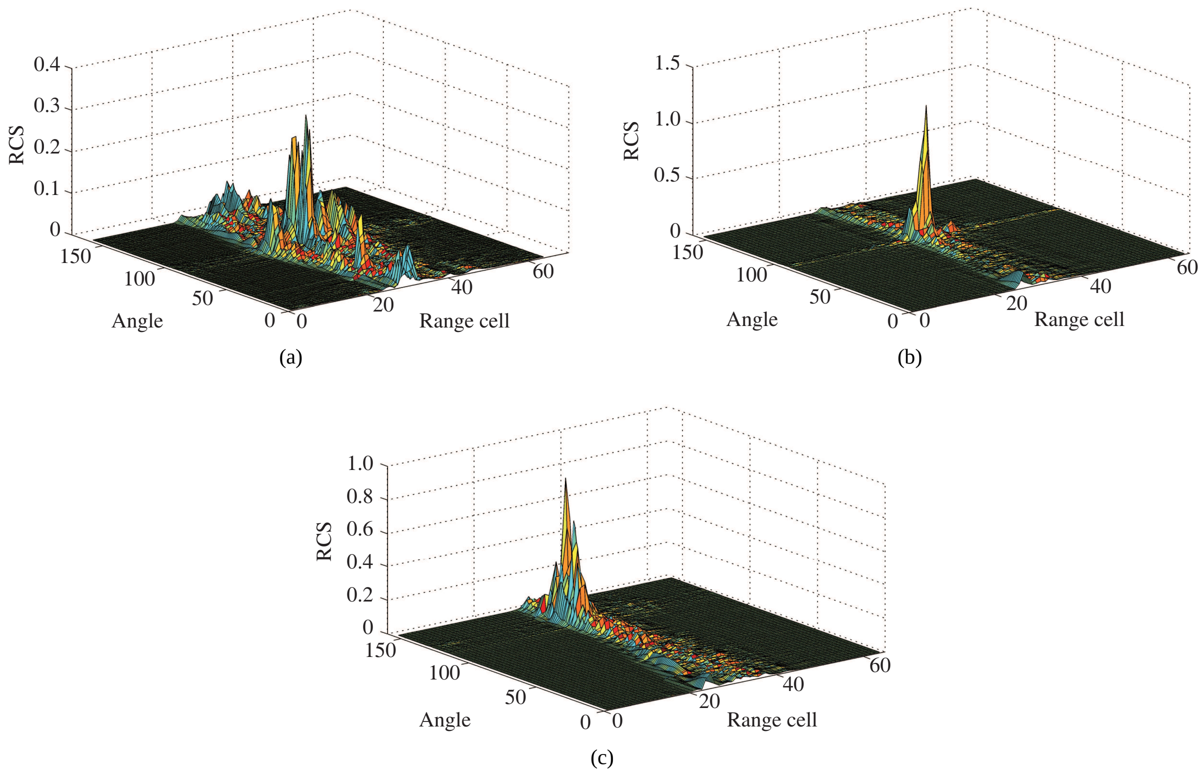


Figure 3 (Color online) 1D radar HRRP data sets of three different kinds of planes under different angles. (a) B-52; (b) Farmer; (c) Fishbed.

it was composed of data in the range of 0°–155°. There are 322 location data for B-52, 311 for Farmer and 451 for Fishbed. Each data sample is described by 64 attributes, namely range cells. The 1D range profiles with the imaging angles are given in Figure 3.

4.2 Experimental design

To evaluate the classification performance of CMSECOC based on confusion matrix, we compared the results with the state-of-the-art ECOC in UCI data sets, and those coding matrices are: one-versus-all, one-versus-one, dense random, sparse random, DECOC, and SECOC as mentioned before. The random matrices were selected from a set of 20000 randomly generated matrices with $p(-1) = p(+1) = 0.5$ for the dense random matrix and $p(-1) = p(+1) = p(0) = 1/3$ for the sparse random matrix. Meanwhile, four state-of-the-art decoding strategies will be applied to each coding strategy for a convincing validation. These state-of-the-art decoding strategies are Hamming decoding [1], Euclidean decoding [22], linear loss-weight decoding, and exponential loss-weight decoding [5]. The parameters of the decoding strategies are the predefined or default values given by the authors. When we are comparing the effects of different coding strategies, we consider two kinds of base classifiers, one is logistic linear classifier (LOGLC) and the other is SVM with polynomial kernel $k(x, x_i) = [x, x_i + 1]^q$. The regularization parameter C and the kernel parameter q are selected by K-fold cross-validation ($K = 5$) [23]. The range of values allowed for q parameter is 1–6, but there is no limit to parameter C.

Furthermore, the CMSECOC was applied for target recognition based on three different planes' HRRP. In this experiment, we pick up three different angle ranges to evaluate the performance in practice ($0°$–$100°, 80°$–$155°$, and $0°$–$155°$). To simplify the experimental operation, the SVM was used for the base classifier with the parameters the same as aforementioned ones, and the Hamming decoding was used for decoding strategy. At last, as discussed in Subsection 3.2, the composition of the superclasses also depends on the values of two thresholds (α and β). To see the sensitivity of the results with small changes in the threshold values, we will make an extensional experiment with different values of two aforementioned thresholds in classification progress. After finishing it, we can summarize the attentions paid in choosing the thresholds.

According to the analysis for the choice of pre-classifier in Subsection 3.2, the k-nearest neighbor classifier (the parameter of k is 3) will be used to get the confusion matrix. To evaluate the performance of the different experiments, we apply stratified 10-fold cross-validation and test for the confidence interval at 95% with a two-tailed t-test, and the calculating formula is given as follows:

$$\frac{|\overline{x} - \mu|}{\sigma/\sqrt{n}} \geqslant t_{0.025(n-1)}, \tag{9}$$

where μ and σ indicate mean and variance, respectively, and, $t_{0.025}(9) = 2.2622$. Besides, we use the statistical Nemenyi test to look for significant differences between the method performances.

4.3 Experimental results and analysis

In the following sections, two results with different data sets are shown. Meanwhile, we deeply analyze the cause of them to get valuable results.

4.3.1 *UCI data set*

Four decoding strategies and two kinds of base classifiers were used to make comparisons between CMSECOC and six state-of-the-art coding methods mentioned before in this part of experiment. After finishing this experiment, we performed a total of 800 10-fold tests. Tables 2–5 list the classification error rate for different decoding strategies based on SVM classifier. The best performance per data set is highlighted in boldface. Meanwhile, each coding matrix's length is below the error rate. From the results in tables, we can see that the classification error rates got by CMSECOC are smaller than other state-of-the-art data-dependent codes and data-independent codes most of the time. However, it is worth noting that the SECOC can also achieve good classification results sometimes. The most likely reason is that the original classes have distinctive features, and they can be directly split into different superclasses. For this reason, SECOC can easily get effective dichotomizers and reach an ideal classification effect at last. Besides, we note that the coding length of CMSECOC is longer than the other types, so the reason is the usage of "one-versus-one" coding strategy to design redundancy code for the classes with high similarity. On the other hand, compared with the other coding strategies, the coding length of CMSECOC does not change more, but it can reduce the error rate obviously, so the redundancy code design is feasible.

To get the statistical experimental conclusions, we use rank-sum test to analyze the results. The mean rank is calculated as follows:

$$R_j = \frac{1}{J} \sum\nolimits_i r_i^j, \tag{10}$$

where r_i^j is the coding rank for i problem based on j decoding strategy, and J is the experiment time of each approach, which is four decoding strategies multiplied by 10 kinds of UCI data. Table 6 shows the

Table 2 Classification error rates for the confidence interval at 95% based on Hamming distance and SVM

	One vs. all	One vs. one	Dense	Sparse	DECOC	SECOC	CMSECOC
(a)	78.33 ± 14.73 10×10	56.55 ± 5.91 10×45	58.53 ± 5.73 10×10	52.94 ± 7.02 10×10	55.05 ± 7.01 10×9	45.96 ± 11.46 18×26	$\mathbf{43.90 \pm 13.80}$ **10×11**
(b)	8.61 ± 4.21 7×7	9.26 ± 3.54 7×21	7.23 ± 2.85 7×7	8.59 ± 2.87 7×7	9.57 ± 3.58 7×6	6.64 ± 3.48 12×16	$\mathbf{5.44 \pm 4.94}$ **7×12**
(c)	22.82 ± 5.72 6×6	18.18 ± 4.74 6×15	17.70 ± 3.90 6×6	19.64 ± 3.72 6×6	19.29 ± 4.88 6×5	20.04 ± 3.54 10×13	$\mathbf{17.31 \pm 4.03}$ **6×12**
(d)	13.78 ± 10.30 7×7	12.19 ± 9.69 7×21	12.58 ± 9.79 7×7	12.36 ± 8.75 7×7	12.30 ± 8.55 7×6	$\mathbf{11.39 \pm 7.06}$ **11×13**	13.46 ± 9.10 7×11
(e)	22.87 ± 4.61 4×4	23.53 ± 6.07 4×6	22.71 ± 6.35 4×4	24.72 ± 7.18 4×4	23.05 ± 7.34 4×3	$\mathbf{21.91 \pm 6.06}$ **6×8**	22.69 ± 5.37 4×6
(f)	31.92 ± 11.52 7×7	31.05 ± 9.40 7×21	28.51 ± 10.10 7×7	30.05 ± 9.20 7×7	$\mathbf{27.35 \pm 8.77}$ **7×6**	28.61 ± 9.84 10×10	28.30 ± 11.16 7×10
(g)	15.62 ± 10.35 3×3	19.17 ± 13.75 3×3	15.25 ± 9.10 3×3	18.48 ± 12.07 3×3	16.30 ± 13.68 3×2	16.07 ± 13.06 3×2	$\mathbf{14.42 \pm 12.84}$ **3×4**
(h)	2.13 ± 1.40 11×11	1.80 ± 2.41 11×55	2.16 ± 2.03 11×11	2.25 ± 2.01 11×11	2.97 ± 1.13 11×10	$\mathbf{1.77 \pm 1.93}$ **16×22**	2.22 ± 1.78 11×12
(i)	40.05 ± 6.94 8×8	34.31 ± 10.53 8×28	$\mathbf{33.04 \pm 8.20}$ **8×8**	50.31 ± 13.64 8×8	50.51 ± 18.19 8×7	33.80 ± 7.34 12×11	35.80 ± 7.48 8×7
(j)	4.91 ± 6.26 3×3	5.18 ± 4.92 3×3	2.79 ± 6.05 3×3	3.60 ± 4.04 3×3	3.93 ± 3.83 3×2	3.92 ± 6.87 3×2	$\mathbf{2.27 \pm 5.81}$ **3×4**

Table 3 Classification error rates for the confidence interval at 95% based on Euclidean distance and SVM

	One vs. all	One vs. one	Dense	Sparse	DECOC	SECOC	CMSECOC
(a)	65.90 ± 5.19 10×10	56.17 ± 5.68 10×45	57.40 ± 7.26 10×10	57.59 ± 4.53 10×10	56.15 ± 6.33 10×9	56.58 ± 7.87 18×26	$\mathbf{48.99 \pm 5.67}$ **10×11**
(b)	31.36 ± 3.71 7×7	8.96 ± 2.28 7×21	24.92 ± 3.44 7×7	14.77 ± 4.68 7×7	23.66 ± 1.14 7×6	20.03 ± 4.31 12×16	$\mathbf{11.42 \pm 3.72}$ **7×12**
(c)	53.23 ± 5.21 6×6	38.58 ± 2.74 6×15	36.50 ± 2.95 6×6	22.25 ± 2.84 6×6	37.95 ± 1.62 6×5	$\mathbf{20.39 \pm 3.28}$ **10×13**	20.83 ± 4.72 6×12
(d)	33.73 ± 15.02 7×7	25.41 ± 13.42 7×21	20.00 ± 9.57 7×7	18.64 ± 9.07 7×7	17.81 ± 10.54 7×6	$\mathbf{16.94 \pm 11.31}$ **11×13**	19.51 ± 10.82 7×11
(e)	33.87 ± 8.96 4×4	30.48 ± 5.91 4×6	$\mathbf{28.78 \pm 7.04}$ **4×4**	42.07 ± 6.71 4×4	31.10 ± 5.83 4×3	36.97 ± 7.64 6×8	31.27 ± 8.67 4×6
(f)	48.80 ± 11.23 7×7	35.17 ± 9.51 7×21	32.88 ± 8.58 7×7	37.12 ± 10.25 7×7	32.77 ± 9.86 7×6	41.84 ± 13.13 10×15	$\mathbf{30.38 \pm 9.22}$ **7×10**
(g)	20.54 ± 5.45 3×3	15.22 ± 7.66 3×3	12.65 ± 5.62 3×3	18.31 ± 7.17 3×3	14.84 ± 4.69 3×2	$\mathbf{10.75 \pm 6.99}$ **3×2**	15.67 ± 4.54 3×4
(h)	21.85 ± 3.55 11×11	33.58 ± 3.05 11×55	19.20 ± 3.30 11×11	32.84 ± 3.88 11×11	18.82 ± 4.32 11×10	$\mathbf{18.75 \pm 3.01}$ **16×22**	25.75 ± 4.19 11×12
(i)	41.15 ± 10.90 8×8	68.09 ± 10.72 8×28	37.57 ± 6.16 8×8	37.57 ± 6.28 8×8	$\mathbf{35.52 \pm 6.34}$ **8×7**	51.67 ± 8.38 12×15	36.71 ± 7.39 8×7
(j)	9.14 ± 6.27 3×3	7.01 ± 10.26 3×3	8.37 ± 7.92 3×3	9.77 ± 8.61 3×3	7.50 ± 8.12 3×2	5.99 ± 5.95 3×2	$\mathbf{4.50 \pm 6.39}$ **3×4**

results of rank positions for each decoding strategy, where the bold values represent the minimum ones, which present the coding approach with the least rank value.

From Table 6, we can see that the rank position of CMSECOC is the smallest and the rank position of SECOC is little larger, while the rank position of one-versus-all is the largest. Overall, the coding type of SECOC is more effective than others. To check for the statistically significant methods, we use the Nemenyi test (two techniques are significantly different if the corresponding average rankings differ by at

Table 4 Classification error rates for the confidence interval at 95% based on linear loss-weight function and SVM

	One vs. all	One vs. one	Dense	Sparse	DECOC	SECOC	CMSECOC
(a)	71.80 ± 2.63 10×10	57.46 ± 4.63 10×45	56.13 ± 4.64 10×10	55.73 ± 4.03 10×10	55.52 ± 3.34 10×9	**55.39 ± 2.76** **18×26**	56.67 ± 3.12 10×11
(b)	54.85 ± 0.88 7×7	14.17 ± 2.45 7×21	13.30 ± 2.34 7×7	22.60 ± 3.94 7×7	36.49 ± 2.96 7×6	13.14 ± 3.89 12×16	**12.69 ± 2.17** **7×12**
(c)	24.54 ± 3.46 6×6	22.24 ± 5.38 6×15	25.17 ± 2.67 6×6	22.15 ± 4.06 6×6	24.86 ± 4.47 6×5	**21.35 ± 3.18** **10×13**	23.30 ± 3.63 6×12
(d)	28.57 ± 7.65 7×7	27.70 ± 10.52 7×21	26.67 ± 9.21 7×7	27.65 ± 7.37 7×7	27.39 ± 6.99 7×6	**26.38 ± 9.42** **11×13**	26.79 ± 9.33 7×11
(e)	20.69 ± 5.07 4×4	22.59 ± 4.20 4×6	21.27 ± 4.90 4×4	21.46 ± 6.74 4×4	21.49 ± 4.46 4×3	**18.73 ± 5.82** **6×8**	19.85 ± 7.07 4×6
(f)	32.13 ± 11.25 7×7	32.11 ± 13.36 7×21	27.53 ± 14.51 7×7	25.73 ± 11.18 7×7	28.36 ± 13.55 7×6	28.10 ± 15.18 10×15	**27.19 ± 13.01** **7×10**
(g)	12.78 ± 10.82 3×3	12.33 ± 8.91 3×3	13.99 ± 10.14 3×3	12.92 ± 7.82 3×3	11.87 ± 8.89 3×2	11.52 ± 8.92 3×2	**11.01 ± 7.19** **3×4**
(h)	20.85 ± 6.54 11×11	19.04 ± 4.39 11×55	19.44 ± 4.03 11×11	20.98 ± 4.16 11×11	18.80 ± 3.57 11×10	**16.14 ± 3.64** **16×22**	18.01 ± 6.27 11×12
(i)	41.72 ± 10.73 8×8	**34.95 ± 10.88** **8×28**	35.21 ± 12.43 8×8	36.10 ± 8.41 8×8	36.65 ± 10.61 8×7	36.14 ± 10.76 12×15	37.76 ± 9.06 8×7
(j)	5.60 ± 10.94 3×3	7.18 ± 5.50 3×3	7.18 ± 7.99 3×3	7.98 ± 6.64 3×3	9.21 ± 6.10 3×2	7.99 ± 7.64 3×2	**5.23 ± 8.71** **3×4**

Table 5 Classification error rates for the confidence interval at 95% based on linear loss-weight function and SVM

	One vs. all	One vs. one	Dense	Sparse	DECOC	SECOC	CMSECOC
(a)	86.10 ± 2.66 10×10	74.84 ± 3.49 10×45	81.38 ± 6.53 10×10	84.87 ± 2.96 10×10	48.95 ± 8.60 10×9	**43.24 ± 6.50** **18×26**	44.75 ± 7.45 10×11
(b)	8.12 ± 3.66 7×7	9.03 ± 2.82 7×21	8.45 ± 4.07 7×7	11.67 ± 3.79 7×7	14.05 ± 3.71 7×6	**7.51 ± 3.55** **12×16**	7.78 ± 3.89 7×12
(c)	24.00 ± 3.46 6×6	25.27 ± 3.82 6×15	24.24 ± 3.04 6×6	23.82 ± 3.71 6×6	23.96 ± 5.15 6×5	**21.76 ± 4.25** **10×13**	22.61 ± 5.59 6×12
(d)	10.91 ± 8.81 7×7	13.47 ± 8.73 7×21	8.80 ± 8.82 7×7	12.27 ± 11.82 7×7	8.90 ± 9.52 7×6	12.65 ± 11.48 11×13	**8.58 ± 11.92** **7×11**
(e)	25.25 ± 7.62 4×4	25.07 ± 6.88 4×6	26.17 ± 6.53 4×4	25.72 ± 4.33 4×4	23.77 ± 3.91 4×3	**20.34 ± 5.15** **6×8**	20.77 ± 6.12 4×6
(f)	34.52 ± 12.03 7×7	35.28 ± 12.67 7×21	33.90 ± 13.03 7×7	32.13 ± 11.16 7×7	34.13 ± 12.60 7×6	**30.87 ± 13.91** **10×15**	33.44 ± 9.89 7×10
(g)	12.35 ± 9.36 3×3	12.21 ± 7.34 3×3	12.92 ± 6.86 3×3	11.73 ± 7.49 3×3	12.50 ± 6.20 3×2	11.07 ± 9.29 3×2	**10.23 ± 7.56** **3×4**
(h)	7.79 ± 2.58 11×11	5.54 ± 2.14 11×55	7.45 ± 1.54 11×11	6.26 ± 2.66 11×11	5.69 ± 4.16 11×10	4.28 ± 2.84 16×22	**5.00 ± 2.58** **11×12**
(i)	35.31 ± 8.05 8×8	36.17 ± 9.41 8×28	35.24 ± 9.19 8×8	32.65 ± 10.43 8×8	30.32 ± 10.20 8×7	30.34 ± 9.24 12×15	**29.04 ± 11.25** **8×7**
(j)	5.51 ± 8.33 3×3	**2.65 ± 5.48** **3×3**	4.19 ± 6.43 3×3	3.90 ± 6.19 3×3	4.19 ± 5.79 3×2	2.46 ± 6.92 3×2	3.07 ± 6.74 3×4

Table 6 Each method's corresponding rank-sum mean

Coding	One vs. all	One vs. one	Dense	Sparse	DECOC	SECOC	CMSECOC
HD	5.40	4.70	3.10	4.90	4.80	2.70	**2.40**
AED	6.20	4.30	3.70	4.90	2.90	3.20	**2.70**
LLB	5.60	4.40	4.00	4.20	4.70	**2.30**	2.70
ELB	5.50	5.00	5.10	4.40	4.20	**1.80**	1.90
Global rank	3.24	2.63	2.27	2.63	2.37	1.43	**1.39**

Table 7 Classification error rates of HRRP between 0° and 80°

	One vs. all	One vs. one	Dense	Sparse	DECOC	SECOC	CMSECOC
B-52	59.08 ± 1.97	51.60 ± 3.55	$\mathbf{27.88 \pm 5.65}$	92.15 ± 2.58	35.18 ± 3.25	77.90 ± 3.14	30.49 ± 1.75
Farmer	28.72 ± 1.79	38.77 ± 2.12	22.04 ± 2.55	58.81 ± 1.81	16.65 ± 1.93	$\mathbf{15.51 \pm 1.55}$	17.86 ± 2.23
Fishbed	28.19 ± 1.35	30.25 ± 2.86	32.61 ± 1.87	26.67 ± 1.94	22.19 ± 1.94	19.99 ± 1.55	$\mathbf{18.25 \pm 2.34}$

Table 8 Classification error rates of HRRP between 80° and 155°

	One vs. all	One vs. one	Dense	Sparse	DECOC	SECOC	CMSECOC
B-52	28.55 ± 2.42	21.90 ± 3.61	33.09 ± 6.00	75.32 ± 3.15	25.94 ± 4.74	23.96 ± 3.32	$\mathbf{21.65 \pm 2.53}$
Farmer	43.55 ± 1.58	31.82 ± 3.45	56.79 ± 2.84	40.25 ± 3.05	$\mathbf{27.21 \pm 1.94}$	28.59 ± 3.31	27.29 ± 1.99
Fishbed	21.83 ± 2.01	20.67 ± 2.85	36.11 ± 2.90	20.50 ± 2.42	16.99 ± 3.20	18.90 ± 2.67	$\mathbf{15.87 \pm 3.44}$

Table 9 Classification error rates of HRRP between 0° and 155°

	One vs. all	One vs. one	Dense	Sparse	DECOC	SECOC	CMSECOC
B-52	52.45 ± 2.24	49.19 ± 3.99	70.32 ± 6.03	47.33 ± 3.12	27.73 ± 3.66	29.13 ± 3.87	$\mathbf{25.35 \pm 3.18}$
Farmer	54.57 ± 2.12	42.31 ± 3.33	54.47 ± 3.78	63.43 ± 4.43	45.67 ± 2.85	41.56 ± 2.65	$\mathbf{37.25 \pm 3.38}$
Fishbed	58.75 ± 3.06	39.83 ± 3.77	42.45 ± 3.09	36.97 ± 2.78	42.47 ± 3.21	38.40 ± 3.81	$\mathbf{35.56 \pm 4.65}$

least the critical difference (CD) value [24]) as follows:

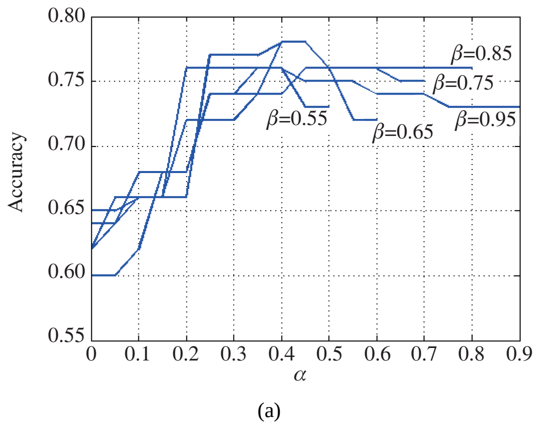
$$\mathrm{CD} = q_\alpha \sqrt{\frac{k\,(k+1)}{6J}}, \tag{11}$$

where q_α is based on the Studentized range statistic divided by $\sqrt{2}$, k is the number of methods to be validated, and J is the total number of experiments performed. In our case, we compare seven methods with a confidence value $\alpha = 0.10$ and $q_{0.10} = 1.372$. Substituting this in (11), we obtain a CD value of 0.663.

Now reviewing Table 6, we can observe that our proposed methods have a difference superior to the critical values of other methods in most of the cases. Because of that, we can argue that the CMSECOC based on confusion matrix of this paper is significantly better than other ones without SECOC at 90% of the confidence interval in the present experiments. Note that the performance of SECOC coding strategy is also better than other ones in most of the cases. The detailed description of reasons for this can be found in [9].

4.3.2 *HRRP data set*

In this section, we have finished three experiments on the HRRP data set according to the imaging angles like 0°–80°, 80°–155°, and 0°–155°, respectively, using the proposed coding strategies. The classification error rates of different coding strategies for HRRP data set with three kinds of angle ranges are shown in Tables 7–9. From the tables, we can see that the classification error rates of the targets in small angle range (0°–80°and 80°–155°) are smaller than those in large angle range (0°–155°) using all the methods. On the data set with angle range being 0°–80°, the level of performance achieved by CMSECOC is similar to the other data-driven ECOC such as DECOC and SECOC, while its classification performance still outperforms the ones independent of data set. On the data set with angle range being 80°–155°, the method of CMSECOC performs much better than the other methods, no matter if that is data-driven one or not. It is worth noting that CMSECOC has the best performance for each target when the data set is in the angle range of 0°–155°, although the classification error rate is larger than in any other angle range. For that, we can give a conclusion that the CMSECOC has higher generalization capability than the other coding strategies when being used for classification in data sets containing noise.



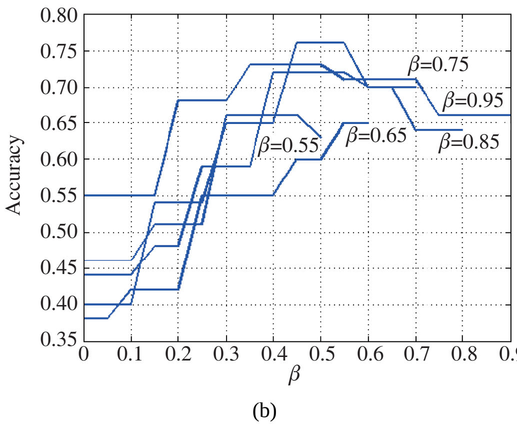


Figure 4 (Color online) Comparison of the CMSECOC performance using LOGLC on the UCI data set for different parameters: combining threshold α and the parting threshold β. (a) Glass data set; (b) Iris data set.

4.3.3 *An extensional experiment*

To discuss the sensitivity of the construction of CMSECOC with small changes in the threshold values, we performed an experiment using the UCI Glass and Ecoli data sets with different threshold values of combining threshold α and the parting threshold β. In this experiment, the values for α are varied between 0 and 0.9, increasing 0.05 per step. For each value of α, the values for β are $\{0.55, 0.65, 0.75, 0.85, 0.95\}$. The results of these experiments using LOGLC as the base classifier and linear loss-weight decoding as the decoding strategy are shown graphically in Figure 4.

From Figure 4, we observe that there are always some pairs of values of α and β, with which the CMSECOC can obtain better classification performance. When the value of α is increased, more classes can be fused into superclasses by decreasing the degree of complexity for classification, thereby enhancing the classification accuracy. However, as the value of α is increasing step by step, the difficulty in classifying classes of each superclass will be enlarged. This is precisely why the classification accuracy first ascended and then descended. Also, we can find that the effect of classification influenced by the parameter β presents the same trend as the one of α. For example, in Glass data set, the largest classification accuracy can be obtained by setting the parameters as follows: $\{\alpha = 0.4, \beta = 0.75; \alpha = 0.5, \beta = 0.75; \alpha = 0.6, \beta = 0.75; \alpha = 0.4, \beta = 0.8; \alpha = 0.5, \beta = 0.8\}$; while in Iris data set, the optimum settings are $\{\alpha = 0.45, \beta = 0.75; \alpha = 0.5, \beta = 0.75; \alpha = 0.55, \beta = 0.75\}$. In a word, when we use CMSECOC as the coding strategy, there always exists a better choice of combining threshold α and the parting threshold β for a special data set, and the final decision for the choice should be made by taking account of both dataset and experience.

5 Conclusion

The ECOC based on data is a research hotspot of using ECOC to model the multiclass problem. In this paper, we first use confusion matrix as similarity measurement to find the superclasses based on the Fisher criterion. Then, the classes with high similarity are assigned together, while those classes with low similarity are assigned into different superclasses. Next, the "one-versus-all" decoding strategy was used to classify the different superclasses to avoid the unnecessary redundancy. Note that classifying classes in the superclasses is much difficult to do, so the "one-versus-one" strategy is used to design the partition, thus making each classification focus on just two classes in a superclass to reduce the complexity of decision boundary and increase redundancy. In the end, all binary partitions are combined to get the CMSECOC. The UCI and HRRP data sets are applied to validate the efficiency of CMSECOC in decreasing the error rate of classification. The results show that the classification performance of CMSECOC is better than other state-of-the-art coding designs in most cases, which means the CMSECOC is more suitable for multiclass problem.

Conflict of interest The authors declare that they have no conflict of interest.

References

1 Dietterich T G, Bakiri G. Solving multiclass learning problems via error-correcting output codes. J Artif Intell Res, 1995, 2: 263–286
2 Windeatt T, Ardeshir G. Boosted ECOC ensembles for face recognition. In: Proceedings of International Conference on Visual Information Engineering. California: AAAI Press, 2003. 165–168
3 Windeatt T, Smith R S. Weighted decoding ECOC for facial action unit classification. In: Proceedings of International Conference on Artificial Intelligence. California: AAAI Press, 2008. 26–30
4 Ghani R. Combining labeled and unlabeled data for text classification with a large number of categories. In: Proceedings of International Conference on Data Mining. California: AAAI Press, 2001. 597–598
5 Escalera S, Pujol O, Radeva P. On the decoding process in ternary errorcorrecting output codes. IEEE Trans Patt Anal Mach Intell, 2010, 32: 120–134
6 Pujol O, Radeva P, Vitria J. Discriminant ECOC: a heuristic method for application dependent design of error correcting output codes. IEEE Trans Patt Anal Mach Intell, 2006, 28: 1001–1007
7 Alpaydin E, Mayoraz E. Learning error-correcting output codes from data. In: Proceedings of International Conference on Artificial Neural Networks. California: AAAI Press, 1999. 743–748
8 Utschick W, Weichselberger W. Stochastic organization of output codes in multiclass learning problems. Neural Comput, 2001, 13: 1065–1102
9 Escalera S, Tax D M J, Pujol O, et al. Subclass problem-dependent design of error-correcting output codes. IEEE Trans Patt Anal Mach Intell, 2008, 30: 1–14
10 Escalera S, Masip D, Puertas E, et al. Online error correcting output codes. Patt Recognit Lett, 2011, 32: 458–467
11 Simeone P, Marrocco C, Tortorella F. Design of reject rules for ECOC classification systems. Patt Recognit, 2012, 2: 863–875
12 García-Pedrajas N, Ortiz-Boyer D. An empirical study of binary classifier fusion methods for multiclass classification. Inf Fusion, 2011, 2: 111–130
13 Zhou J, Suen C. Unconstrained numeral pair recognition using enhanced error correcting output coding: a holistic approach. In: Proceedings of International Conference on Document Analysis and Recognition. California: AAAI Press, 2005. 484–488
14 Allwein E, Schapire R, Singer Y. Reducing multiclass to binary: a unifying approach for margin classifiers. Mach Learn Res, 2002, 1: 113–141
15 Ruda H, Snorrason M, Shue D. Framework for automatic target recognition optimization. Cambridge Charles River Analytics Technical Report No. R96451. 1997
16 Jain A K, Duin R P W, Mao J C. Statistical pattern recognition: a review. IEEE Trans Patt Anal Mach Intell, 2000, 22: 4–37
17 Sun J X. Modern Pattern Recognition (in Chinese). Changsha: National Defense Industry Press, 2002. 112–118
18 Song R, Zhang J, Xia S P, et al. An adaptive classification method of BP-NN group based classification system and its application (in Chinese). Acta Electron Sin, 2001, 29: 1950–1953
19 Logan J D. Applied Mathematics. 2nd ed. Hoboken: Wiley-Interscience Ltd., 1996. 54–60
20 UCI Machine Learning Repository. School of Information and Computer Sciences, University of California, Irvine, 2007
21 Zhang Y X, Wang X D, Yao X, et al. HRRP recognition for polarization radar based on Bagging-SVM dynamic ensemble (in Chinese). Syst Eng Electron, 2012, 34: 1366–1372
22 Hastie T, Tibshirani R. Classification by pairwise grouping. In: Jordan M I, Kearns M J, Solla S A, eds. Advances in Neural Information Processing Systems 10. Cambridge: MIT Press, 1998. 451–471
23 Chapelle O, Vapnik V, Bousquet O, et al. Choosing multiple parameters for support machines. Mach Learn, 2002, 46: 131–159
24 Demsar J. Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res, 2006, 2: 1–30