

# Stability Analysis of ECOC Kernel Machines

Aijun Xue, Xiaodan Wang<sup>(✉)</sup>, and Xiaolong Fu

Department of Computer Science, Air Force Engineering University,  
Changle East Road, Xi'an 710051, China  
Wang\_afeu@126.com

**Abstract.** Error correcting output codes kernel machines (ECOC kernel machines) are ensemble of kernel machines based on ECOC decomposition methods. How to improve the generalization capability of this framework is an open problem. In this paper, we discussed the condition for generalization in terms of the stability property of ECOC kernel machines. Here we provide a proof for the result that an ECOC kernel machine has the pointwise hypothesis stability. This stability property can be calculated by training on the training dataset once and has clear and meaningful formulation. It can be applied to tune the kernel parameters in model selection and design good matrixes for ECOC kernel machines.

**Keywords:** Error correcting output codes (ECOC) · Kernel machines  
Pointwise hypothesis stability · Generalization error bound

## 1 Introduction

Many real-world pattern recognition applications are aim to map the instances into a set of classes. To deal with such problems, the efficient way is to decompose the multiclass problems into a set of binary classification problems [1]. In this case, many techniques addressing the binary classification task can be used to solve the complex multiclass problems. There are many different approaches for recasting the multiclass problem into a series of smaller binary classification tasks, such as one-versus-all (OVA), one-versus-one (OVO) [2, 3], and error correcting output codes (ECOC) [4, 5]. One-versus-all which is the simplest approach considers the comparison between each class and the others, and takes the maximum output as the final classification decision. One-versus-one approach addresses the comparison of all possible pairs of classes, and obtains the final classification result by means of a voting procedure. Error correcting output codes (ECOC) is a general framework to achieve this decomposition, which was presented by Dietterich and Bakiri [4]. Due to the error correcting capability [6–8], ECOC has been successfully applied to a wide range of applications [9–12].

When the ECOC framework is applied in practice, we expect that this framework will have an accuracy classification result. Generally speaking, training the learning algorithms on the empirical datasets to predict the unlabeled samples precisely is one of the key points for machine learning. The ECOC framework can be seen as a special learning algorithm, which is an ensemble of some binary classifiers. How to improve and evaluate the generalization capability of this framework is an open problem and

some works have been done. Masulli and Valentini [13] attempted to analyze the main factors affecting the effectiveness of ECOC methods experimentally, and their analysis showed that all these factors concurred to the effectiveness of ECOC methods in a not straightforward way and interacted between them. Garcia-Pedrajas and Fyfe [14] proposed an evolutionary approach to the design of output codes, which took five different aspects into account, but it was difficult to define an apparent relationship between these aspects and the generalization ability of ECOC methods. These results underscore the difficulty of this problem.

On the other hand, some works focused on just one or two aspects that were related to generalization performance of ECOC methods. Pujol and Radeva [15] focused on obtaining the partitions which had the maximum class discrimination. Escalera and Tax [16] used the subclass information to guarantee that the base classifier can split each subgroup of classes. Ali-Bagheri [17, 18] aimed to improve independency among base classifiers using different feature subsets for each base classifiers. Angel-Bautista and Escalera [19] presented a novel genetic strategy to obtain the better dichotomizers. These works provide us some new ways to the ECOC framework research. However, it is a pity that there is little formal justification to support these improvements. The previous works show that it is difficult to determine what is the intrinsic factors affecting the effectiveness of ECOC methods.

To improve the generalization capability of ECOC framework, what we need to do is to go back to the nature of ECOC framework. The nature is it is just a special learning algorithm. Note that in this case, we do not discuss the decoding strategy. A central question for machine learning is to determine conditions under which a learning algorithm will generalize from its finite training set to novel samples [20]. A milestone in learning theory was empirical risk minimization (ERM) based on minimizing the training error. Furthermore, Valentini [21] proposed the upper bounds on the training error of ECOC kernel machines. In this case, the problem is changed to how to minimize the difference between the training error and the generalization error. Fortunately, Poggio and Rifkin [20] discussed the conditions for generalization in terms of a precise stability property of the learning process. Bousquet and Elisseeff [22] showed how to use the stability to derive bounds on the generalization error based on the training error. The stability of a learning algorithm can be a bridge between the training error and the generalization error, and as a powerful tool in ECOC framework research.

In other word, if the learning algorithm has minimization training error and is more stable, the learning algorithm can have better generalization capability. It is our purpose in this paper to discuss the stability of ECOC framework with kernel machines. The most contribution of our work is that we obtain the result that ECOC kernel machines have pointwise hypothesis stability. This stability can be calculated by training on the training dataset once, and has clear and meaningful formulation. It can be applied to tune the kernel parameters in model selection and design good matrixes for ECOC kernel machines.

The rest of this paper is organized as follows: Sect. 2 provides a brief introduction to the ECOC decomposition methods and background on kernel machines. The pointwise hypothesis stability of ECOC kernel machines is explained in detail in

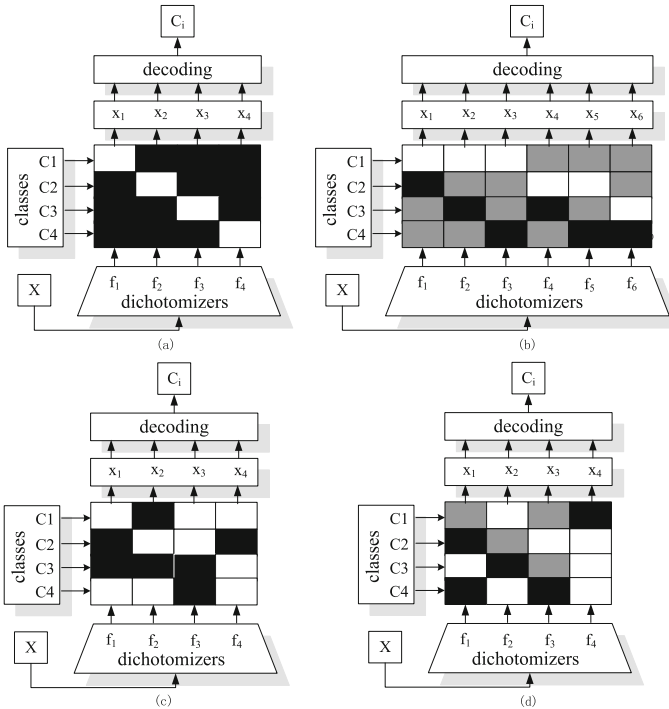
Sect. 3. In Sect. 4, some applications of pointwise hypothesis stability are presented. Finally, Sect. 5 draws the main conclusions of the paper.

## 2 ECOC Kernel Machines

### 2.1 ECOC

The ECOC framework consists of two stages: encoding and decoding. At the coding stage, the main task is to construct a coding matrix. Each column of the coding matrix is the bipartition (groups of classes), which indicates one binary problem. Each row of the coding matrix is the codeword for each class, where the bit implies the label of the class for a binary problem [23]. The base classifier is obtained by training the binary problem. Binary ECOC is the original framework. It has two symbols  $\{+1, -1\}$  in the coding matrix, which represent the negative class and the positive class in one binary problem. When the coding step attracted special attention, Allwen et al. [5] introduced zero as the third symbol, which indicated that the corresponding class would be ignored for a binary problem. The classical ECOCs are shown in Fig. 1.

In Fig. 1, the white, black and gray regions of the coding matrix stand for the symbol “1”, “-1” and “0” respectively. For example, in Fig. 1(c) when classifier  $f_1$  is



**Fig. 1.** Four classical ECOCs, binary ECOC: (a) one-versus-all; (c) dense random; ternary ECOC: (b) one-versus-one; (d) sparse random

trained, classes  $C_1$  and  $C_4$  are seen as the positive class and classes  $C_2$  and  $C_3$  as the negative class. In the same way, finally, we obtain a classifier vector  $\{f_1, f_2, f_3, f_4\}$ .

Given a test sample, the base classifiers output a codeword. Note that the obtained codeword cannot take the value zero since the output of the base classifier is “+1” or “-1”. At the decoding stage, this codeword is compared with the codewords defined in the coding matrix, and the test sample is assigned to the class corresponding to the closest codeword. Usually, this comparison is implemented by the Hamming and the Euclidean decoding distance. Specially, Allwen et al. [5] showed the advantage of using a loss-based function of the margin of the output of the base classifier. For example, in Fig. 1(c) given the test sample  $X$ , the classifier vector output a codeword  $\{x_1, x_2, x_3, x_4\}$ . Then, the final classification result is obtained by a given decoding strategy.

### 2.2 Kernel Machines

We assume that the training set is  $Z^m = \{(x_i, y_i)_{i=1}^m \in \{X \times \{-1, 1\}\}^m\}$  and  $\ell : R \rightarrow R$  is a loss function. Kernel machines are the minimizers of functionals of the form

$$F[f; Z^m] = \frac{1}{m} \sum_{i=1}^m \ell(y_i f(x_i)) + \lambda \|f\|_K^2 \tag{1}$$

where  $\lambda$  is a positive constant named regularization parameter. The minimization of functional in (1) is done in a Reproducing Kernel Hilbert Space (RKHS)  $\mathcal{H}$  defined by a symmetric and positive definite kernel  $K : X \times X \rightarrow R$ , and  $\|f\|_K^2$  is the norm of a function  $f : X \rightarrow R$  belonging to  $\mathcal{H}$ .

If  $\ell$  is convex, the minimizer of functional in (1) is unique and has the form

$$f(x) = \sum_{i=1}^m \alpha_i y_i K(x_i, x) \tag{2}$$

The coefficients  $\alpha_i$  are computed by solving an optimization problem whose form is determined by the loss function  $\ell$ . For example, in SVMs, the soft-margin loss is  $\ell(yf(x)) = |1 - yf(x)|_+$ , where  $|x|_+ = x$  if  $x > 0$  and zero otherwise. In this case, the  $\alpha_i$  is the solution of a quadratic programming problem with constraints  $\alpha_i \in [0, 1/2m\lambda]$ . A peculiar property of an SVM is that, usually, only few data points have nonzero coefficients  $\alpha_i$ . These points are named support vectors.

## 3 Pointwise Hypothesis Stability of ECOC Kernel Machines

### 3.1 Pointwise Hypothesis Stability

The stability of one learning algorithm can be used to get bounds on the generalization error [22]. Here we focus on the stability with respect to changes in the training set. Firstly, we introduce some notations.

A training dataset is given as follows

$$S = \{z_1 = (x_1, y_1), \dots, z_m = (x_m, y_m)\} \quad (3)$$

By removing the  $i$ th element, the changed training dataset is given as:

$$S^{\setminus i} = \{z_1, \dots, z_{i-1}, z_{i+1}, \dots, z_m\} \quad (4)$$

Then, we denote by  $f$  the function trained on the set  $S$ , and  $f^{\setminus i}$  the function trained on the set  $S^{\setminus i}$ . The definition of pointwise hypothesis stability is given in Definition 1.

**Definition 1 (Pointwise Hypothesis Stability).** An algorithm  $A$  has pointwise hypothesis stability  $\delta$  with respect to the loss function  $\ell$  if the following holds

$$\forall S \in Z^m, \forall i \in \{1, \dots, m\}, \mathbb{E}_S \left[ \left| \ell(f, z_i) - \ell(f^{\setminus i}, z_i) \right| \right] \leq \delta \quad (5)$$

### 3.2 Multiclass Loss Function

Based on Definition 1, the loss function of ECOC kernel machines is needed. Let the coding matrix be  $M \in \{-1, 0, 1\}^{k \times l}$ , where  $k$  is the number of class and  $l$  is the number of binary classifier.  $m_{ps}$  is the code bit which indicates the label of class  $p$  in the  $s$ th binary classifier. The vector function formed by the binary classifiers is  $f(x) = \{f_1(x), \dots, f_l(x)\}$ .

The multiclass margin of a sample  $(x_i, y_i) \in X \times \{1, \dots, k\}$  can be written as [24]

$$g(x_i, y_i) = d_L(m_p, f(x_i)) - d_L(m_{y_i}, f(x_i)) \quad (6)$$

where  $d_L(m_p, f(x_i)) = \sum_{s=1}^l L(m_{ps} f_s(x_i))$  is the linear loss-based decoding function with  $L(m_{ps} f_s(x_i)) = -m_{ps} f_s(x_i)$ , and  $p = \arg \min_{q \neq y_i} d_L(m_q, f(x_i))$ .

Note that the multiclass margin is positive when sample  $x_i$  is classified correctly. Considering a loss function is typically a nondecreasing function of the margin, the linear-margin loss function can be defined as  $\ell(f, z) = -g(x, y)$ .

### 3.3 Pointwise Hypothesis Stability of ECOC Kernel Machines

In this section, we present the proof of pointwise hypothesis stability of ECOC kernel machines. To this end, we first need the following lemma [24].

**Lemma 1.** Let  $f$  be the kernel machine as defined in (2) obtained by solving (1). Let  $f^{\setminus i}$  be the solution of (1) found when the data point  $(x_i, y_i)$  is removed from the training set. We have

$$y_i f(x_i) - \alpha_i G_{ii} \leq y_i f^{\setminus i}(x_i) \leq y_i f(x_i) \tag{7}$$

where  $G_{ij} = K(x_i, x_j)$ .

By applying Lemma 1 simultaneously to each kernel machine used in the ECOC procedure, Inequality (7) can be rewritten as

$$f_s^{\setminus i}(x_i) = f_s(x_i) - \lambda_s m_{y_i s}, \quad s \in \{1, \dots, l\} \tag{8}$$

where  $\lambda_s$  is a parameter in  $[0, \alpha_i^s G_{ii}^s]$ .

**Theorem 1.** Let  $M \in \{-1, 1\}^{k \times l}$  be the code matrix and  $f_s(x) = \sum_{i=1}^m \alpha_i^s m_{y_i s} K^s(x_i, x)$  be the  $s$  th binary classifier, where  $\alpha_i \in [0, C]$  with  $C = 1/(2m\lambda)$ . The multiclass loss function is  $\ell(f, z) = -g(x, y)$ . Let  $\kappa = \max_s \max_i G_{ii}^s = \max_s \max_i K^s(x_i, x_i)$ .  $f$  is the vector function derived by the ECOC kernel machines based on the coding matrix  $M$ . Decoding strategy is set as linear loss-based decoding function.  $\theta(\cdot)$  is the Heavyside function:  $\theta(x) = 1$  if  $x > 0$  and zero otherwise. Thus, ECOC kernel machines have pointwise hypothesis stability with

$$E_S \left[ \left| \ell(f, z_i) - \ell(f^{\setminus i}, z_i) \right| \right] \leq \frac{2\kappa C}{m} \sum_{i=1}^m \sum_{s=1}^l \theta(\alpha_i^s) \tag{9}$$

**Proof.** In order to prove that ECOC kernel machines has pointwise hypothesis stability, we have to find the bound for  $E_S \left[ \left| \ell(f, z_i) - \ell(f^{\setminus i}, z_i) \right| \right]$ . Firstly, for  $\forall S \in Z^m, \forall z_i = (x_i, y_i) \in S$ , we have

$$E_S \left[ \left| \ell(f, z_i) - \ell(f^{\setminus i}, z_i) \right| \right] = \frac{1}{m} \sum_{i=1}^m \left| \ell(f, z_i) - \ell(f^{\setminus i}, z_i) \right| = \frac{1}{m} \sum_{i=1}^m |g(x_i, y_i) - g^{\setminus i}(x_i, y_i)|$$

So, if we want to bound  $E_S \left[ \left| \ell(f, z_i) - \ell(f^{\setminus i}, z_i) \right| \right]$ , firstly we can bound  $|g(x_i, y_i) - g^{\setminus i}(x_i, y_i)|$ .

The above problem can be divided into two parts. On one hand, from the definition of  $g(x, y)$  and linear loss-based decoding function, we have

$$g^{\setminus i}(x_i, y_i) = m_{y_i} f^{\setminus i}(x_i) - m_{p^{\setminus i}} f^{\setminus i}(x_i) \tag{10}$$

where  $p^{\setminus i} = \arg \max_{q \neq y_i} m_q f^{\setminus i}(x_i)$ .

$$g(x_i, y_i) = m_{y_i} f(x_i) - m_p f(x_i) \tag{11}$$

where  $p = \arg \max_{q \neq y_i} m_q f(x_i)$ .

Moreover, from the definition of  $p$ , we can get

$$m_p f(x_i) \geq m_{p^{\vee i}} f(x_i) \quad (12)$$

(11) minus (10), there is

$$g(x_i, y_i) - g^{\vee i}(x_i, y_i) = \sum_{s=1}^l \left[ m_{y_i s} f_s(x_i) - m_{p s} f_s(x_i) - m_{y_i s} f_s^{\vee i}(x_i) + m_{p^{\vee i} s} f_s^{\vee i}(x_i) \right] \quad (13)$$

Now, applying (8), we can get

$$\begin{aligned} g(x_i, y_i) - g^{\vee i}(x_i, y_i) &= \sum_{s=1}^l \left[ m_{y_i s} f_s(x_i) - m_{p s} f_s(x_i) - m_{y_i s} (f_s(x_i) - \lambda_s m_{y_i s}) + m_{p^{\vee i} s} (f_s(x_i) - \lambda_s m_{y_i s}) \right] \\ &= \sum_{s=1}^l \left[ (m_{p^{\vee i} s} - m_{p s}) f_s(x_i) + (m_{y_i s} - m_{p^{\vee i} s}) \lambda_s m_{y_i s} \right] \end{aligned} \quad (14)$$

Considering (12), we can write the following inequality

$$g(x_i, y_i) - g^{\vee i}(x_i, y_i) \leq \sum_{s=1}^l (m_{y_i s} - m_{p^{\vee i} s}) \lambda_s m_{y_i s} \leq \sum_{s=1}^l 2\lambda_s \leq \sum_{s=1}^l 2\alpha_i^s G_{ii}^s \leq 2\kappa \sum_{s=1}^l \alpha_i^s \quad (15)$$

Note that the second inequality is just because of  $m_{y_i s} \in \{-1, 1\}$ , and then

$$0 \leq (m_{y_i s} - m_{p^{\vee i} s}) m_{y_i s} \leq 2 \quad (16)$$

Considering  $\alpha_i^s \in [0, C]$ , and  $\alpha_i^s > 0$  indicates  $z_i$  is a support vector, for  $\forall (x_i, y_i) \in S$ ,

$$g(x_i, y_i) - g^{\vee i}(x_i, y_i) \leq 2\kappa \sum_{s=1}^l \alpha_i^s \leq 2\kappa C \sum_{s=1}^l \theta(\alpha_i^s) \quad (17)$$

On the other hand, due to (16) and  $\lambda_s \geq 0$ , we have

$$\sum_{s=1}^l (m_{y_i s} - m_{p^{\vee i} s}) \lambda_s m_{y_i s} \geq 0 \quad (18)$$

From (14), we can get

$$\begin{aligned}
 g(x_i, y_i) - g^{\setminus i}(x_i, y_i) &\geq \sum_{s=1}^l (m_{p^{\setminus i}s} - m_{ps})f_s(x_i) \\
 &= \sum_{s=1}^l [m_{p^{\setminus i}s}f_s(x_i) - m_{p^{\setminus i}s}f_s^{\setminus i}(x_i)] + \sum_{s=1}^l [m_{p^{\setminus i}s}f_s^{\setminus i}(x_i) - m_{ps}f_s(x_i)] \\
 &\geq \sum_{s=1}^l [m_{p^{\setminus i}s}f_s(x_i) - m_{p^{\setminus i}s}f_s^{\setminus i}(x_i)] + \sum_{s=1}^l [m_{ps}(f_s^{\setminus i}(x_i) - f_s(x_i))] \quad (19) \\
 &= \sum_{s=1}^l [m_{p^{\setminus i}s}(f_s(x_i) - f_s^{\setminus i}(x_i))] - \sum_{s=1}^l [m_{ps}(f_s(x_i) - f_s^{\setminus i}(x_i))] \\
 &= \sum_{s=1}^l (m_{p^{\setminus i}s} - m_{ps})\lambda_s m_{y_i s}
 \end{aligned}$$

Moreover,  $\lambda_s \geq 0$  and  $(m_{p^{\setminus i}s} - m_{ps})m_{y_i s} \geq -2$ , the following inequalities are given

$$g(x_i, y_i) - g^{\setminus i}(x_i, y_i) \geq \sum_{s=1}^l (m_{p^{\setminus i}s} - m_{ps})\lambda_s m_{y_i s} \geq -2 \sum_{s=1}^l \lambda_s \geq -2\kappa \sum_{s=1}^l \alpha_i^s \quad (20)$$

So, for  $\forall (x_i, y_i) \in \mathcal{S}$ ,

$$g(x_i, y_i) - g^{\setminus i}(x_i, y_i) \geq -2\kappa \sum_{s=1}^l \alpha_i^s \geq -2\kappa C \sum_{s=1}^l \theta(\alpha_i^s) \quad (21)$$

And then, we get the bound

$$\forall i \in \{1, \dots, m\}, \quad |g(x_i, y_i) - g^{\setminus i}(x_i, y_i)| \leq 2\kappa C \sum_{s=1}^l \theta(\alpha_i^s) \quad (22)$$

Finally, we prove that ECOC kernel machines has pointwise hypothesis stability with

$$\mathbb{E}_S \left[ \left| \ell(f, z_i) - \ell(f^{\setminus i}, z_i) \right| \right] \leq \frac{2\kappa C}{m} \sum_{i=1}^m \sum_{s=1}^l \theta(\alpha_i^s) \quad \blacksquare$$

**Remark.** The parameters  $\alpha_i^s$  indicate if point  $x_i$  is a support vector for the  $s$ th kernel machine, which depend on the solution of the machines trained on the full dataset (so training the machines once will suffice). Our result indicates that pointwise hypothesis stability  $\delta$  is related with all samples in the training set. For binary ECOC the number of training samples of every dichotomy is the same, thus, there exists parameter  $\alpha_i$  for every  $z_i$ . But, for ternary ECOC the number of training samples of every dichotomy is different, thus, many training samples are not considered in the training phase of a kernel machine. Considering the parameters  $\alpha_i$  indicate if point  $x_i$  is a support vector, so, the parameters  $\alpha_i$  for these ignored points can be seen as zero just as that these



points are not support vectors for kernel machines. This hypothesis is an open problem and will be discussed in our future work.

## 4 Application of Pointwise Hypothesis Stability

In this section, we introduce two aspects of application of pointwise hypothesis stability: model selection and good coding matrixes design for ECOC kernel machines.

### 4.1 Model Selection

When a coding matrix  $M$  is given, in order to have a better generalization performance we should tune the kernel parameters to find the better binary classifiers. Importantly, we need to evaluate the generalization error to check if the tuned kernel parameter is the best one. However, it is difficult to calculate the generalization error, due to the unknown distribution of the data. We have to estimate the generalization error from the available dataset. This available dataset is often defined as the training dataset. On the training dataset we can only obtain the training error or the empirical error. Previously, the better classifier is selected by empirical error minimization, which is known as empirical risk minimization (ERM). But this always leads to an overfitting problem, which means that although the classifier has the minimum empirical error, it has a bad generalization performance. There must be a gap between the empirical error and the generalization error. Fortunately, Bousquet and Elisseeff [22] have given the relation between stability and generalization. They give the generalization error bound in Theorem 2. That is to say the difference between the empirical error and the generalization error can be measured by the stability of a learning algorithm.

**Theorem 2.** For any learning algorithm  $A$  with pointwise hypothesis stability  $\delta$  with respect to a loss function  $\ell$  such that  $0 \leq \ell(f, z) \leq B$ , we have

$$R(A, S) \leq R_{emp}(A, S) + \sqrt{\frac{B^2 + 12Bm\delta}{2m}} \quad (23)$$

where  $R(A, S)$  is the generalization error and  $R_{emp}(A, S)$  is the empirical error.

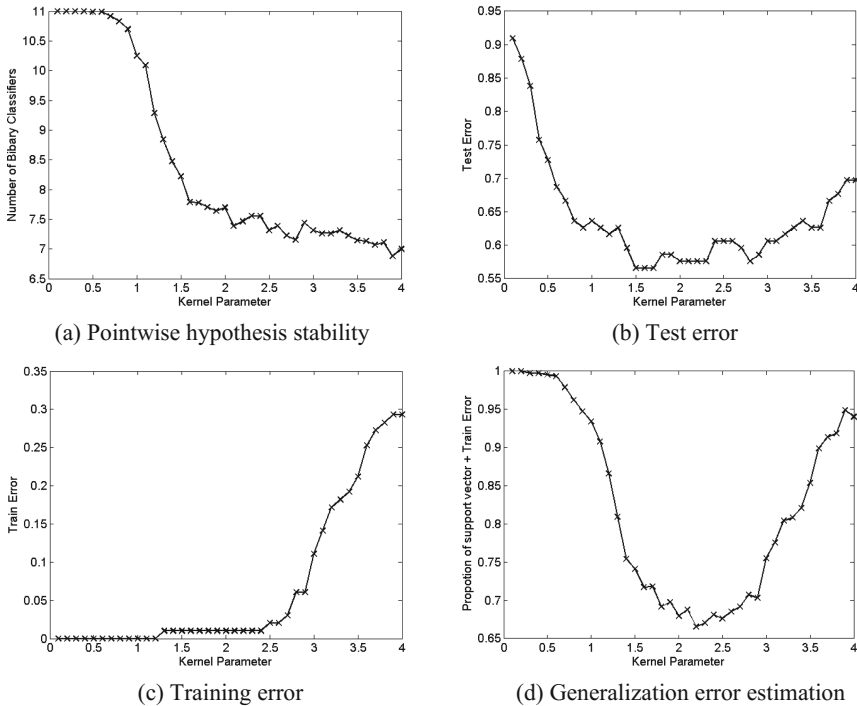
Note that for a loss function we can always find a suitable upper bound, for example, a large enough bound. Considering we tune the kernel parameters with the same loss function, the upper bound can be seen as a constant argument. In this case, the generalization error bound is just affected by the empirical error on the training dataset and the pointwise hypothesis stability of the learning algorithm.

In order to present the application of pointwise hypothesis stability in model selection, we carry out the experiments on the UCI datasets [25]. Table 1 shows a summary of the datasets used in the experiments. We take the experiment on the vowel dataset as an instance. To reduce the computational complexity, we use 10-fold cross validation to split the whole dataset into 10 parts, and select one part as the training dataset and another part as the test dataset. Moreover, we do not discuss the parameter  $C$  which is related to the regularization parameter  $\lambda$ , and treat it as a constant argument.

**Table 1.** Summary of the used datasets.

Dataset	#Instances	#Features	#Classes
glass	214	9	6
vehicle	846	18	4
zoo	101	18	7
ecoli	336	8	8
vowel	990	13	11
letter	1214	16	26
iris	150	4	3

SVMs are trained on a Gaussian kernel  $K(x, x_i) = \exp\{-|x - x_i|^2 / \sigma^2\}$ . So, we have  $\kappa = \max_s \max_i G_{ii}^s = \max_s \max_i K^s(x_i, x_i) = 1$ . In this case, computing the pointwise hypothesis stability means computing the average number of support vectors for each sample on all binary classifiers. Note that we focus on searching for the best value of the kernel parameter  $\sigma$  of the Gaussian kernel. Finally, the parameters are set as  $C = 10$  and  $\sigma \in [0.1, 4]$  sampled with step 0.1. The used coding strategy is one-versus-all. Figure 2 shows the experimental result.



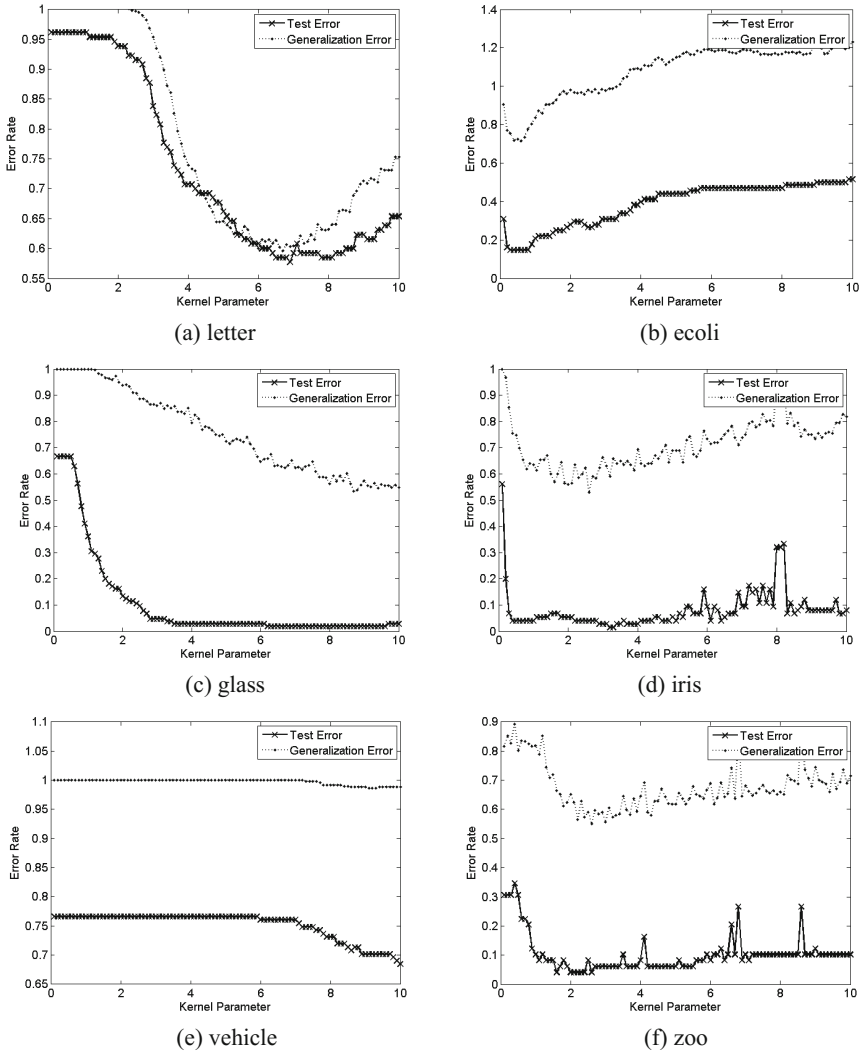
**Fig. 2.** Experimental result on the vowel dataset

Figure 2(c) plots the training error against different kernel parameter  $\sigma$ . We can observe that when the kernel parameter  $\sigma$  is smaller than 1.3, the training error is the minimum value, equal to zero. However, in Fig. 2(b) the test error is not at the least level, although there is a downward trend. This is just the overfitting problem. In other word, the minimum training error not always leads to the minimum test error. Intuitively, if we combine two figures together [Fig. 2(a) and (c)], the combination will have the same trend with the test error, which is just a validation to Theorem 2. To finish this combination, we calculate the proportion that the number of support vectors for each sample takes on the number of binary classifiers, which can be seen as the possibility of being a support vector for each sample, and regard it as the pointwise hypothesis stability. Refer to Theorem 2, Fig. 2(d) shows the estimation of generalization error against different kernel parameter  $\sigma$ . We can observe that the minimum of the generalization error estimate is very close to the minimum of the test error, although there is a slight deviation. Moreover, Fig. 3 shows the comparison between the generalization error and the test error on several UCI datasets. The comparison result shows that the generalization error estimated by pointwise hypothesis stability can be used to select the better value of kernel parameter in the model selection.

## 4.2 Good Coding Matrixes Design

On the other hand, the coding matrix design problem is that given a set of binary classifiers, finding a matrix which has better generalization performance. Cramer and Singer [26] have proven that this problem is NP-complete. As an alternative way, many works have focused on the problem dependent design for coding matrixes, such as, Discriminant ECOC [15] and Subclass ECOC [16], which can be a promising approach in the future. For the moment, these problem dependent designs are implemented to achieve a specific criterion. For example, Discriminant ECOC is designed to maximize a discriminative criterion, and Subclass ECOC is designed to guarantee that the base classifier is capable of splitting each subgroup of classes. In this case, the key point for problem dependent design is to find a criterion, which will lead to a better generalization performance. However, there is less formal justification to find it. The difficulty for the problem dependent design is that there is no an apparent relationship between the generalization performance and the property of coding matrix. That is to say if we know which property of coding matrix will lead to a better generalization performance, we can take this property as the criterion in the problem dependent design, such as, the discriminative criterion.

Fortunately, we think that pointwise hypothesis stability will make a certain process for the problem dependent design. Theorem 2 shows the difference between the generalization error and the empirical error. It is sure that if we reduce this difference, we will obtain the better generalization performance. Note that in this paper we do not discuss how to get minimum empirical error, because this problem needs a more detailed work. Just as in Sect. 4.1, we also take the upper bound for the loss function as the constant argument. So, the difference is just affected by pointwise hypothesis stability of the learning algorithm. In this case, the minimization for this difference is equal to the minimization for pointwise hypothesis stability.



**Fig. 3.** Comparison between the generalization error and the test error

Refer to Theorem 1, pointwise hypothesis stability for ECOC kernel machines can be written as follows:

$$\delta = \frac{2\kappa C}{m} \sum_{i=1}^m \sum_{s=1}^l \theta(\alpha_i^s) \tag{24}$$

Considering that we just take care the coding matrix design, parameters, such as,  $\kappa, C, m$ , can be seen as the constant arguments. So, pointwise hypothesis stability is determined by  $\sum_{s=1}^l \theta(\alpha_i^s)$ .

Note that  $l$  is the codeword length or the number of binary classifiers. Now we discuss that when the codeword length is given, what we should do to minimize the pointwise hypothesis stability. In this case,  $\sum_{s=1}^l \theta(\alpha_i^s)$  can be seen as the possibility for one sample  $(x_i, y_i)$  to be support vectors among all binary classifiers. Reducing the pointwise hypothesis stability means to reduce the possibility for all samples in the training dataset. On the other hand, the support vectors are the samples on the separating surface. In other word, the separating surface is represented by the support vectors. Complex separating surface needs more support vectors, which often means that the two class groups are difficult to be split. If two class groups have maximum class discrimination, the separating surface will be simple and the possibility for one sample to be a support vector will be reduced. That is to say if we want to reduce the pointwise hypothesis stability and have a better generalization performance for ECOC kernel machines, we should design the coding matrix which has high discrimination power. This also validates that the Discriminant ECOC has the advantage to have better generalization performance.

In order to validate the relationship between the discriminative criterion in problem dependent design of coding matrix and the pointwise hypothesis stability, we carry out a simple experiment on the synthetic dataset. The synthetic dataset generated randomly has four classes. There are 100 samples for each class. The feature vector of each class has two dimensions: Feature1 and Feature2. The probability density function for each class is defined as follows:

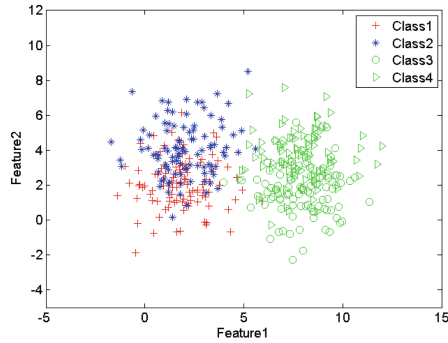
$$p(x|class_i) = \frac{1}{2\pi|\Sigma_i|^{1/2}} \exp\left[-\frac{1}{2}(x - \mu_i)^T \Sigma_i^{-1}(x - \mu_i)\right] \quad i = 1, 2, \dots, 4 \quad (25)$$

where the parameters are shown in Table 2. Figure 4 shows the distribution of four classes.

**Table 2.** Parameters for synthetic dataset

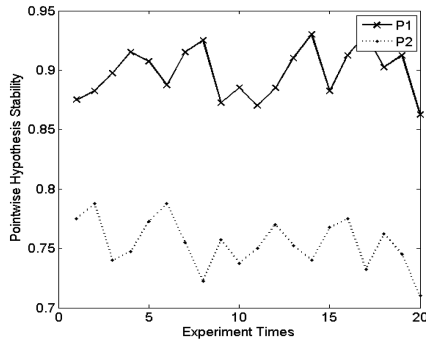
Class	Mean vectors	Covariance matrices
$C_1$	$\mu_1 = (2, 2)^T$	$\Sigma_1 = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$
$C_2$	$\mu_2 = (2, 4)^T$	$\Sigma_2 = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$
$C_3$	$\mu_3 = (8, 2)^T$	$\Sigma_3 = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$
$C_4$	$\mu_4 = (8, 4)^T$	$\Sigma_4 = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$

Simply, we take the design of one column of the coding matrix into consideration. In this experiment, we compare two columns with different discrimination power. Intuitively, in Fig. 4 we can observe that the class groups  $\{\{Class1, Class2\}, \{Class3, Class4\}\}$  can be split more easily than the class groups  $\{\{Class1, Class3\},$



**Fig. 4.** The synthetic dataset with four classes

$\{Class2, Class4\}$ . So, one column is set as  $P1 = [1, -1, 1, -1]^T$  and the other column is set as  $P2 = [1, 1, -1, -1]^T$ , with more class discrimination. The pointwise hypothesis stability for each column is calculated by the proportion that the support vectors take up all the samples in the synthetic dataset, which is proportional to the possibility of one sample to be a support vector. Figure 5 shows the fluctuation curves of pointwise hypothesis stability of two columns against different experiment times. We can see that the pointwise hypothesis stability of  $P2$  is smaller than that of  $P1$ . This experiment proves that the maximization of discriminative criterion can lead to have smaller pointwise hypothesis stability, which may lead to a better generalization performance finally.



**Fig. 5.** Pointwise hypothesis stability of two columns

However, the design of good coding matrix is a complex problem, which is determined by many different factors, such as, the codeword length and the minimum hamming distance. Furthermore, these factors may work in an intersectant way. For example, if we only achieve the minimum pointwise hypothesis stability, maybe we will get a bad training error rate. So, the design of good coding matrix needs a tradeoff among several factors or criterions.

## 5 Conclusion

We provide a proof for the result that an ECOC kernel machines has the pointwise hypothesis stability. In our proof, the stability is determined by the coefficients which can be calculated by training the machines once on the training dataset, and it is easy to be applied in practice. Note that the stability can be seen as the difference between the training error and the generalization error. Minimizing this gap can help to reduce the generalization error. Finally, the applications of this stability in model selection and good coding matrixes design for ECOC kernel machines are presented. How to take both the training error and pointwise hypothesis stability into consideration in good coding matrixes design will be a meaningful direction to get better generalization capability, which will be discussed in our future research works.

**Acknowledgments.** This work is supported by National Science Foundation of China under grant 61273275.

## References

1. Garcia-Pedrajas, N., Ortiz-Boyer, D.: An empirical study of binary classifier fusion methods for multiclass classification. *Inf. Fusion* **12**(2), 111–130 (2011)
2. Anand, R., Mehrotra, K., Mohan, C.K., Ranka, S.: Efficient classification for multiclass problems using modular neural networks. *IEEE Trans. Neural Netw.* **6**(1), 117–124 (1995)
3. Clark, P., Boswell, R.: Rule induction with CN2: some recent improvements. In: Kodratoff, Y. (ed.) *EWSL 1991*. LNCS, vol. 482, pp. 151–163. Springer, Heidelberg (1991). <https://doi.org/10.1007/BFb0017011>
4. Dietterich, T., Bakiri, G.: Solving multiclass learning problems via error-correcting output codes. *J. Artif. Intell. Res.* **2**, 263–268 (1995)
5. Allwein, E.L., Schapire, R.E., Singer, Y.: Reducing multiclass to binary: a unifying approach for margin classifiers. *J. Mach. Learn. Res.* **1**, 1113–1141 (2001)
6. Kong, E., Dietterich, T.: Error-correcting output coding corrects bias and variance. In: Prieditis, A., Lemmer, J. (eds.) *Machine Learning: Proceedings of the Twelfth International Conference on Machine Learning*, pp. 313–321 (1995)
7. Kong, E., Dietterich, T.: Why error-correcting output coding works with decision trees. Technical report, Department of Computer Science, Oregon State University, Corvallis, OR (1995)
8. Windeatt, T., Ghaderi, R.: Coding and decoding strategies for multi-class learning problems. *Inf. Fusion* **4**(1), 11–21 (2003)
9. David, A., Lerner, B.: Support vector machine-based image classification for genetic syndrome diagnosis. *Pattern Recogn. Lett.* **26**(8), 1029–1038 (2005)
10. Ubeyli, E.D.: Multiclass support vector machines for diagnosis of erythematous diseases. *Expert Syst. Appl.* **35**(4), 1733–1740 (2008)
11. Kittler, J., Ghaderi, R., Windeatt, T., Matas, J.: Face verification via error correcting output codes. *Image Vis. Comput.* **21**(13–14), 1163–1169 (2003)
12. Bagheri, M.A., Montazer, G.A., Escalera, S.: Error correcting output codes for multiclass classification: application to two image vision problems. In: 2012 16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP), Shiraz, Iran, pp. 508–513 (2012)

13. Masulli, F., Valentini, G.: Effectiveness of error correcting output coding methods in ensemble and monolithic learning machines. *Pattern Anal. Appl.* **6**, 285–300 (2003)
14. Garcia-Pedrajas, N., Fyfe, C.: Evolving output codes for multiclass problem. *IEEE Trans. Evol. Comput.* **12**(1), 93–106 (2008)
15. Pujol, O., Radeva, P., Vitria, J.: Discriminant ECOC: a heuristic method for application dependent design of error correcting output codes. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(6), 1007–1012 (2006)
16. Escalera, S., Tax, D.M.J., Pujol, O., Radeva, P., Duin, R.P.W.: Subclass problem-dependent design for error-correcting output codes. *IEEE Trans. Pattern Anal. Mach. Intell.* **30**(6), 1041–1053 (2008)
17. Ali-Bagheri, M., Ali-Montazer, G., Kabir, E.: A subspace approach to error correcting output codes. *Pattern Recogn. Lett.* **34**, 176–184 (2013)
18. Ali-Bagheri, M., Gao, Q., Escalera, S.: A genetic-based subspace analysis method for improving Error-Correcting Output Coding. *Pattern Recogn.* **46**, 2830–2839 (2013)
19. Angel-Bautista, M., Escalera, S., Baro, X., Pujol, O.: On the design of an ECOC-Compliant Genetic Algorithm. *Pattern Recogn.* **47**, 865–884 (2014)
20. Poggio, T., Rifkin, R., Mukherjee, S., Niyogi, P.: General conditions for predictivity in learning theory. *Nature* **428**(25), 419–422 (2004)
21. Valentini, G.: Upper bounds on the training error of ECOC SVM ensembles. Technical report DISI-TR-00-17, Dipartimento di Informatica e Science dell' Informazione, Università di Genova (2000)
22. Bousquet, O., Elisseeff, A.: Stability and generalization. *J. Mach. Learn. Res.* **2**, 499–526 (2002)
23. Escalera, S., Pujol, O., Radeva, P.: Separability of ternary codes for sparse designs of error-correcting output codes. *Pattern Recogn. Lett.* **30**(3), 285–297 (2009)
24. Passerini, A., Pontil, M., Frasconi, P.: New results on error correcting output codes of kernel machines. *IEEE Trans. Neural Netw.* **15**(1), 45–54 (2004)
25. Asuncion, A., Newman, D.: UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences (2007)
26. Crammer, K., Singer, Y.: On the learnability and design of output codes for multiclass problems. *Mach. Learn.* **47**, 201–233 (2002)