

# Enhancing the Performance of Centroid Classifier by ECOC and Model Refinement

Songbo Tan, Gaowei Wu, and Xueqi Cheng

Key Laboratory of Network,  
Institute of Computing Technology,  
Beijing, China

tansongbo@{software.ict.ac.cn, gmail.com}, {wgu, cxq}@ict.ac.cn

**Abstract.** With the aim of improving the performance of centroid text classifier, we attempt to make use of the advantages of Error-Correcting Output Codes (ECOC) strategy. The framework is to decompose one multi-class problem into multiple binary problems and then learn the individual binary classification problems by centroid classifier. However, this kind of decomposition incurs considerable bias for centroid classifier, which results in noticeable degradation of performance for centroid classifier. In order to address this issue, we use Model-Refinement strategy to adjust this so-called bias. The basic idea is to take advantage of misclassified examples in the training data to iteratively refine and adjust the centroids of text data. The experimental results reveal that Model-Refinement strategy can dramatically decrease the bias introduced by ECOC, and the combined classifier is comparable to or even better than SVM classifier in performance.

## 1 Introduction

With the advent of the Web and the enormous growth of digital content in Internet, databases, and archives, text categorization has received more and more attention in information retrieval and natural language processing community. To this date, numerous machine-learning approaches have been introduced to deal with text classification [1-6, 11-12, 19-24].

In recent years, ECOC has been applied to boost naïve bayes, decision tree and SVM classifier for text data [7-10]. Following this research direction, in this work, we explore the use of ECOC to enhance the performance of centroid classifier. The framework we adopted is to decompose one multi-class problem into multiple binary problems and then use centroid classifier to learn the individual binary classification problems.

However, this kind of decomposition incurs considerable bias [11-13] for centroid classifier. In substance, centroid classifier [2, 21] relies on a simple decision rule that a given document should be assigned a particular class if the similarity (or distance) of this document to the centroid of the class is the largest (or smallest). This decision rule is based on a straightforward assumption that the documents in one category should share some similarities with each other. However, this hypothesis is often broken by ECOC on the grounds that it ignores the similarities of original classes when disassembling one multi-class problem into multiple binary problems.

In order to attack this problem, we use Model-Refinement strategy [11-12] to reduce this so-called bias. The basic idea is to take advantage of misclassified examples in the training data to iteratively refine and adjust the centroids. This technique is very flexible, which only needs one classification method and there is no change to the method in any way. The empirical evaluation shows that Model-Refinement strategy can dramatically reduce the bias and boost the performance of centroid classifier. From the perspective of mathematics, we justified that with respect to a linearly separable problem, the Model-Refinement strategy converges to the optimal solution after finite online updates.

To examine the performance of proposed method, we conduct an extensive experiment on two commonly used datasets, i.e., Newsgroup and Industry Sector. The results indicate that Model-Refinement strategy can dramatically decrease the bias introduced by ECOC, and the resulted classifier is comparable to or even better than SVM classifier in performance.

The rest of this paper is constructed as follows: Next section presents related work on applying ECOC to text classification. ECOC algorithm is described in section 3. In section 4, we present the proposed method. Experimental results are given in section 5. Finally section 6 concludes this paper.

## 2 Related Work

In this section, we present the related work on applying ECOC to text classification. ECOC has been applied to boost Naïve Bayes, Decision Tree, SVM and Co-Training [7-10].

Berger [7] made the first attempt to explore the application of ECOC to Naïve Bayes and Decision Tree for text categorization. He conducted his experiments on four datasets: Newsgroup, WebKB, Yahoo Science and Yahoo Health. His experiment showed that with sufficiently high bit  $n$ , combining a Decision Tree (Naïve Bayes) to an ECOC classifier can improve the performance over the one-vs.-rest Decision Tree (Naïve Bayes) approach. He also gave some theoretical evidences for the use of random codes rather than error-correcting codes.

Following the spirit of Berger, Ghani [8] explored the use of different kinds of codes, namely Error-Correcting Codes, Random Codes, Domain and Data-specific codes. Experiments conducted on Industry Sector show a reduction in classification error by up to 66% over Naive Bayes Classifier. Further more, he also gave empirical evidence for using error-correcting codes rather than random codes.

In 2002, Ghani [9] continued his research in this direction. He developed a framework to incorporate unlabeled data in ECOC setup by first decomposing multi-class problems into multiple binary problems and then using Co-Training to learn the individual binary classification problems. Experiments show that this strategy is especially useful for text classification tasks with a large number of categories and outperforms other semi-supervised learning techniques such as EM and Co-Training. In addition to being highly accurate, this method utilizes the Hamming distance from ECOC to provide high-precision results. He also present results with algorithms other than Co-Training in this framework and show that Co-Training is uniquely suited to work well within ECOC.

In 2002, Rennie [10] compared Naive Bayes and Support Vector Machines using ECOC on the task of multi-class text classification. The experimental results show that the Support Vector Machine can perform multi-class text classification very effectively when used as part of an ECOC scheme. Rennie argues that its improved ability to perform binary classification gives it much lower error scores than Naive Bayes.

### 3 Error-Correcting Output Codes

Error-Correcting Output Codes (ECOC) is a form of combination of multiple classifiers [8]. The ECOC method is borrowed from data transmitting task in communication. Its main idea is to add redundancy to the data being learned (transmitted) so that even if some errors occur due to the biases (noises) in the learning process (channel), the data can be correctly classified (received) in prediction stage (at the other end). It works by converting a multi-class supervised learning problem into a large number ( $L$ ) of two-class supervised learning problems [8]. Any learning algorithm that can handle two-class learning problems, such as Naïve Bayes [3], can then be applied to learn each of these  $L$  problems.  $L$  can then be thought of as the length of the codewords with one bit in each codeword for each classifier. The ECOC algorithm is outlined in Figure 1.

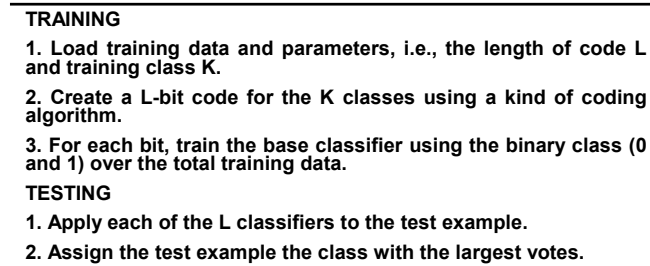


Fig. 1. Outline of ECOC

Different from the use of ECOC in communication tasks, the use in classification tasks requires not only the rows of a code to be well-separated, but also the columns to be well-separated as well. The reason behind the rows being well-separated is obvious, since we want codewords or classes to be maximally far apart from each other, but the column separation is necessary because the functions being learned by the learner for each bit should be uncorrelated so that the errors in each bit are independent of each other [8].

There are three commonly used coding strategies: code theory based method [16], random method [7], and data-specific method. Ghani's work [8] indicates that code theory based method, such as BCH, performs the best. Further more, BCH can insure the code to be well-separated in row. As a result, we only use BCH in this work.

### 4 Methodology

This section presents the rationale of why we combine ECOC with Model-Refinement strategy. First, we illustrate why ECOC brings bias for centroid classifier. Then explain why Model-Refinement strategy can modify this kind of bias. Finally we outline the combined ECOC algorithm.

#### 4.1 The Bias Incurred by ECOC for Centroid Classifier

Centroid classifier is a linear, simple and yet efficient method for text categorization. The basic idea of centroid classifier is to construct a centroid  $C_i$  for each class  $c_i$  using formula (1) where  $d$  denotes one document vector and  $|z|$  indicates the cardinality of set  $z$ . In substance, centroid classifier makes a simple decision rule (formula (2)) that a given document should be assigned a particular class if the similarity (or distance) of this document to the centroid of the class is the largest (or smallest). This rule is based on a straightforward assumption: the documents in one category should share some similarities with each other.

$$C_i = \frac{1}{|c_i|} \sum_{d \in c_i} d \tag{1}$$

$$c = \arg \max_{c_i} \left( \frac{d \cdot C_i}{\|d\|_2 \|C_i\|_2} \right) \tag{2}$$

For example, the single-topic documents involved with “sport” or “education” can meet with the presumption; while the hybrid documents involved with “sport” as well as “education” break this supposition.

As such, ECOC based centroid classifier also breaks this hypothesis. This is because ECOC ignores the similarities of original classes when producing binary problems. In this scenario, many different classes are often merged into one category. For example, the class “sport” and “education” may be assembled into one class. As a result, the assumption will inevitably be broken.

Let’s take a simple multi-class classification task with 12 classes. After coding the original classes, we obtain the dataset as in Figure 2. Class 0 consists of 6 original categories, and class 1 contains another 6 categories. Then we calculate the centroids

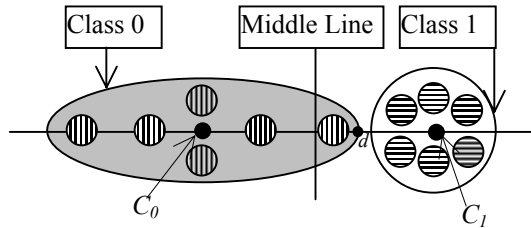


Fig. 2. Original Centroids of Merged Class 0 and Class 1

of merged class 0 and merged class 1 using formula (1), and draw a Middle Line that is the perpendicular bisector of the line between the two centroids.

According to the decision rule (formula (2)) of centroid classifier, the examples of class 0 on the right of the Middle Line will be misclassified into class 1. This is the mechanism why ECOC can bring bias for centroid classifier. In other words, the ECOC method conflicts with the assumption of centroid classifier to some degree.

#### 4.2 Why Model-Refinement Strategy Can Reduce This Bias?

In order to decrease this kind of bias, we employ the Model-Refinement strategy to adjust the class representatives, i.e., the centroids. The basic idea of Model-Refinement strategy is to make use of training errors to adjust class centroids so that the biases can be reduced gradually, and then the training-set error rate can also be reduced gradually.

For example, if document  $d$  of class 0 is misclassified into class 1, both centroid  $C_0$  and  $C_1$  should be moved right by the following formulas (3-4) respectively,

$$C_0^* = C_0 + \eta \cdot d . \quad (3)$$

$$C_1^* = C_1 - \eta \cdot d . \quad (4)$$

where  $\eta$  ( $0 < \eta < 1$ ) is the *Learning Rate* which controls the step-size of updating operation. The former formula (3) is called as “drag” formula and the latter (4) is called as “push” formula.

---

```

1. Load training data and parameters;
2. Calculate centroid for each class;
3. For iter=1 to MaxIteration Do
  3.1 For each document  $d$  in training set Do
    3.1.1 Classify  $d$  labeled “ $A_1$ ” into class “ $A_2$ ”;
    3.1.2 If ( $A_1 \neq A_2$ ) Do
      Drag centroid of class  $A_1$  to  $d$  using formula (3);
      Push centroid of class  $A_2$  against  $d$  using formula (4);

```

---

Fig. 3. Outline of Model-Refinement Strategy

The Model-Refinement strategy for centroid classifier is outlined in Figure 3 where *MaxIteration* denotes the pre-defined steps for iteration. The time requirement of Model-Refinement strategy is  $O(MTKW)$  where  $M$  denotes the iteration steps,  $T$  denotes the size of training set and  $W$  denotes the size of vocabulary.

With this so-called move operation,  $C_0$  and  $C_1$  are both moving right gradually. At the end of this kind of move operation (see Figure 4), no example of class 0 locates at the right of Middle Line so no example will be misclassified.

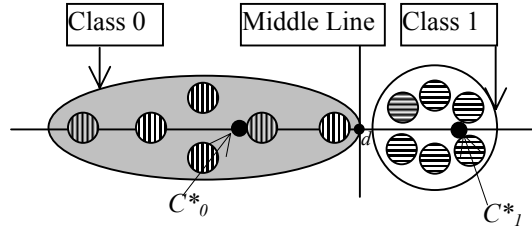


Fig. 4. Refined Centroids of Merged Class 0 and Class 1

### 4.3 The Combination of ECOC and Model-Refinement Strategy for Centroid Classifier

In this subsection, we present the outline (Figure 5) of combining ECOC with Model-Refinement strategy for centroid classifier. In substance, the improved ECOC combines the strengths of ECOC and Model-Refinement strategy. ECOC research in ensemble learning techniques has shown that it is well suited for classification tasks with a large number of categories. On the other hand, Model-Refinement strategy has proved to be an effective approach to reduce the bias of base classifier, that is to say, it can dramatically boost the performance of the base classifier.

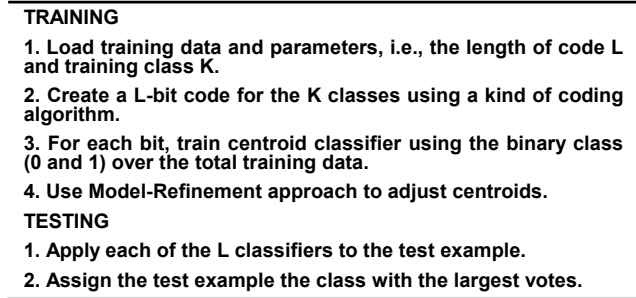


Fig. 5. Outline of combining ECOC with Model-Refinement strategy

### 4.4 The Convergence Analysis of Model-Refinement Strategy

Given a training set  $S = \bigcup_{i=1}^K S_i$ , where  $K$  denotes the number of training classes, and  $S_i$  denotes the training examples of class  $i$ . In the following analysis, we suppose the data is 2-norm bounded, that is,  $\forall d \in S, \|d\|_2 \leq R (R > 0)$ . Since the size of training set is finite, this assumption always holds.

**Definition 1.** We said a training set  $S$  is a linearly separable problem if there exists  $\{C_1^{opt}, C_2^{opt}, \dots, C_K^{opt}\}$  so that  $\forall i \in [1, K]$  satisfies,

$$C_i^{opt} d - C_j^{opt} d \geq \gamma (\gamma > 0) \text{ where } d \in S_i, j \neq i.$$

**Theorem 1.** With respect to linearly separable problem, if we select an appropriate learning parameter  $\eta$ , the Model-Refinement strategy converges to the optimal solution  $\{C_i^{opt}\}$  after finite online updates.

**Proof:** In the iteration  $t$ , assume example  $d(d \in S_A)$  is a misclassified example, that is,  $C_A^t d - C_B^t d < 0$ , where  $C_B^t$  denote the most similar centroid to  $d$  with the different label. Then,

$$\begin{aligned}
& \sum_{i=1}^K \|C_i^{t+1} - C_i^{opt}\|^2 = \sum_{i \neq A, B} \|C_i^t - C_i^{opt}\|^2 \\
& + \|C_A^t + \eta d - C_A^{opt}\|^2 + \|C_B^t - \eta d - C_B^{opt}\|^2 \\
& = \sum_{i=1}^K \|C_i^t - C_i^{opt}\|^2 + 2\eta^2 \|d\|^2 + 2\eta d(C_A^t - C_A^{opt}) - 2\eta d(C_B^t - C_B^{opt}) \\
& = \sum_{i=1}^K \|C_i^t - C_i^{opt}\|^2 + 2\eta^2 R^2 + 2\eta d(C_A^t - C_B^t) - 2\eta d(C_A^{opt} - C_B^{opt}) \\
& \leq \sum_{i=1}^K \|C_i^t - C_i^{opt}\|^2 + 2\eta^2 R^2 - 2\eta\gamma
\end{aligned}$$

In this time, as long as select  $\eta < \gamma/R^2$ , we can guarantee that  $\sum_{i=1}^K \|C_i^{t+1} - C_i^{opt}\|^2 < \sum_{i=1}^K \|C_i^t - C_i^{opt}\|^2$ . In other words, after each update, class centroid  $C_i^t$  approaches optimal centroid  $C_i^{opt}$ .

Furthermore, if select an appropriate  $\rho$  so as to  $0 < \rho < (\eta\gamma - \eta^2 R^2)$ , then,

$$\sum_{i=1}^K \|C_i^t - C_i^{opt}\|^2 < \sum_{i=1}^K \|C_i^{t-1} - C_i^{opt}\|^2 - 2\rho = \sum_{i=1}^K \|C_i^0 - C_i^{opt}\|^2 - 2t\rho.$$

Obviously,  $\sum_{i=1}^K \|C_i^t - C_i^{opt}\|^2 \geq 0$ , let  $\zeta = \sum_{i=1}^K \|C_i^0 - C_i^{opt}\|^2$ , then,

$$\zeta - 2t\rho > 0, \text{ that is, } t < \frac{\zeta}{2\rho}. \quad \square$$

**Lemma 1.**  $\left(\sum_{i=1}^K a_i\right)^2 \leq K \sum_{i=1}^K a_i^2$  when  $a_i \geq 0$ .

**Proof:** 
$$\begin{aligned}
K \sum_{i=1}^K a_i^2 - \left(\sum_{i=1}^K a_i\right)^2 &= (K-1) \sum_{i=1}^K a_i^2 - 2 \sum_{i \neq j} a_i a_j \\
&= \sum_{i \neq j} (a_i - a_j)^2 \geq 0 \quad \square
\end{aligned}$$

**Theorem 2.** With respect to a linearly separable problem, the proposed method converges after finite online updates using any learning parameter  $\eta(\eta > 0)$ .

**Proof:** In the iteration  $t$ , assume example  $d(d \in S_A)$  is a misclassified example or small-margin example, that is,  $C_A^t d - C_B^t d < \delta (0 < \delta < \gamma)$ , where  $C_B^t$  denote the most similar centroid to  $d$  with the different label. Then,

$$\begin{aligned} \sum_{i=1}^K C_i^{t+1} C_i^{opt} &= \sum_{i \neq A, B} C_i^t C_i^{opt} + (C_A^t + \eta d) A_A^{opt} + (C_B^t - \eta d) C_B^{opt} \\ &= \sum_{i=1}^K C_i^t C_i^{opt} + \eta d (C_A^{opt} - C_B^{opt}) \\ &\geq \sum_{i=1}^K C_i^t C_i^{opt} + \eta \gamma, \end{aligned}$$

which indicates,

$$\sum_{i=1}^K C_i^t C_i^{opt} \geq \sum_{i=1}^K C_i^0 C_i^{opt} + t \eta \gamma \quad (5)$$

In the same way,

$$\begin{aligned} \sum_{i=1}^K \|C_i^{t+1}\|^2 &= \sum_{i \neq A, B} \|C_i^t\|^2 + \|C_A^t + \eta d\|^2 + \|C_B^t - \eta d\|^2 \\ &= \sum_{i=1}^K \|C_i^t\|^2 + 2\eta^2 \|d\|^2 + 2\eta d (C_A^t - C_B^t) \\ &\leq \sum_{i=1}^K \|C_i^t\|^2 + 2\eta^2 R^2 + 2\eta \delta, \end{aligned}$$

therefore,

$$\sum_{i=1}^K \|C_i^t\|^2 \leq \sum_{i=1}^K \|C_i^0\|^2 + 2t(\eta^2 R^2 + \eta \delta)$$

let  $\tau = \max_i \|C_i^{opt}\|$ , then,

$$\sum_{i=1}^K C_i^t C_i^{opt} \leq \sum_{i=1}^K \|C_i^t\| \cdot \|C_i^{opt}\| \leq \tau \sum_{i=1}^K \|C_i^t\|.$$

According to **lemma 1**,

$$\begin{aligned} \sum_{i=1}^K C_i^t C_i^{opt} &\leq \tau \sum_{i=1}^K \|C_i^t\| \\ &\leq \tau \left( K \sum_{i=1}^K \|C_i^t\|^2 \right)^{1/2} \\ &\leq \tau \sqrt{K} \left( \sum_{i=1}^K \|C_i^0\|^2 + 2t(\eta^2 R^2 + \eta \delta) \right)^{1/2} \\ &\leq \tau \sqrt{K} \left( \sum_{i=1}^K \|C_i^0\|^2 \right)^{1/2} + \tau \sqrt{2K(\eta^2 R^2 + \eta \delta)t} \quad (6) \end{aligned}$$

According to (5) and (6), we obtain,

$$\tau\sqrt{K}\left(\sum_{i=1}^K\|C_i^0\|^2\right)^{1/2} + \tau\sqrt{2K(\eta^2R^2 + \eta\delta)}t \geq \sum_{i=1}^K C_i^0 C_i^{opt} + t\eta\gamma.$$

Obviously, if above inequality holds,  $t$  must be finite. That is to say, the Model-Refinement strategy converges after finite online updates.  $\square$

## 5 Experiment Results

In this section, we introduce the experimental data set, evaluation metrics, experiment settings and present the experimental results.

### 5.1 Datasets

In our experiment, we use two corpora: NewsGroup<sup>1</sup>, and Industry Sector<sup>2</sup>.

**20NewsGroup.** The 20Newsgroup (20NG) dataset contains approximately 20,000 articles evenly divided among 20 Usenet newsgroups. We use a subset consisting of total categories and 19,446 documents.

**Industry Sector.** The Industry Section dataset is based on the data made available by Market Guide, Inc. ([www.marketguide.com](http://www.marketguide.com)). The set consists of company homepages that are categorized in a hierarchy of industry sectors, but we disregard the hierarchy. There were 9,637 documents in the dataset, which were divided into 105 classes. We use a subset called as Sector-48 consisting of 48 categories and in all 4,581 documents.

### 5.2 The Performance Measure

To evaluate a text classification system, we use the F1 measure introduced by van Rijsbergen [15]. This measure combines recall and precision in the following way:

$$Recall = \frac{\text{number of correct positive predictions}}{\text{number of positive examples}}$$

$$Precision = \frac{\text{number of correct positive predictions}}{\text{number of positive predictions}}$$

$$F1 = \frac{2 \times Recall \times Precision}{(Recall + Precision)}$$

For ease of comparison, we summarize the F1 scores over the different categories using the Micro- and Macro-averages of F1 scores [17]:

$$Micro-F1 = F1 \text{ over categories and documents}$$

$$Macro-F1 = \text{average of within-category F1 values}$$

<sup>1</sup> [www-2.cs.cmu.edu/afs/cs/project/theo-11/www/wwkb](http://www-2.cs.cmu.edu/afs/cs/project/theo-11/www/wwkb)

<sup>2</sup> [www-2.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/](http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/)

The MicroF1 and MacroF1 emphasize the performance of the system on common and rare categories respectively. Using these averages, we can observe the effect of different kinds of data on a classification system [18].

### 5.3 Experimental Design

We evenly split the each dataset into two parts. Then we use one part for training and the remaining second for test. We perform the train-test procedure two times and use the average of the two performances as final result. This is so called two-fold cross validation.

In order to remove redundant features and save running time, we employ Information Gain as feature selection method because it consistently performs well in most cases [14].

We employ TFIDF as input features. The formula for calculating the TFIDF can be written as follows:

$$W(t, d) = \frac{tf(t, d) \times \log(N / n_t)}{\sqrt{\sum_{t \in d} [tf(t, d) \times \log(N / n_t)]^2}} \quad (7)$$

where  $N$  is the total number of training documents, and  $n_t$  is the number of documents containing the word  $t$ .  $tf(t, d)$  indicates the occurrences of word  $t$  in document  $d$ .

For experiments involving SVM we employed SVMtorch, which uses one-versus-the-rest decomposition and can directly deal with multi-class classification problems. ([www.idiap.ch/~bengio/projects/SVMtorch.html](http://www.idiap.ch/~bengio/projects/SVMtorch.html)). Particularly, it has been specifically tailored for large-scale problems.

### 5.4 Comparison and Analysis

Table 1 and table 2 show the performance comparison of different methods on two datasets when using 10,000 features. For ECOC, we use 63-bit BCH coding; for Model-Refinement strategy, we fix its *MaxIteration* and *LearningRate* as 8 and 0.01 respectively. For brevity, we use MR to denote Model-Refinement strategy.

From the two tables, we can observe that ECOC indeed brings significant bias for centroid classifier, which results in considerable decrease in accuracy. Especially on sector-48, the bias reduces the MicroF1 of centroid classifier from 0.7985 to 0.6422.

**Table 1.** The MicroF1 of different methods

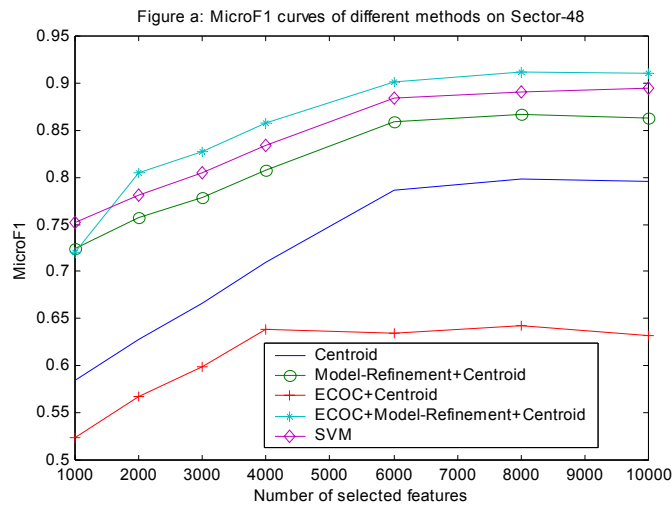
Method \ Dataset	Centroid	MR +Centroid	ECOC +Centroid	ECOC + MR +Centroid	SVM
Sector-48	0.7985	0.8671	0.6422	<b>0.9122</b>	0.8948
NewsGroup	0.8371	0.8697	0.8085	<b>0.8788</b>	0.8777

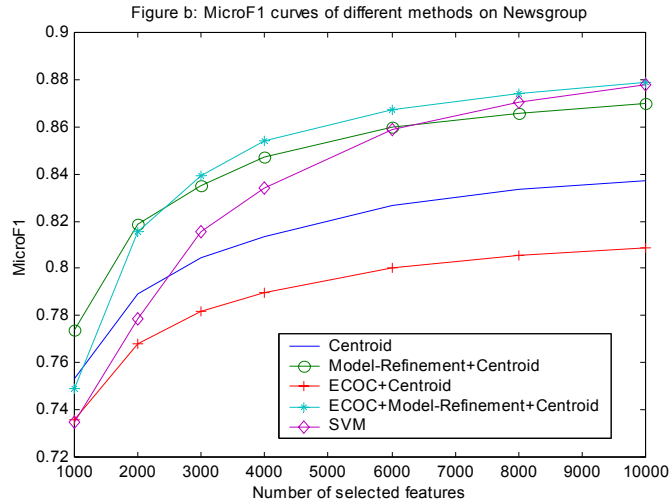
**Table 2.** The MacroF1 of different methods

Method \ Dataset	Centroid	MR +Centroid	ECOC +Centroid	ECOC + MR +Centroid	SVM
Sector-48	0.8097	0.8701	0.6559	<b>0.9138</b>	0.8970
NewsGroup	0.8331	0.8661	0.7936	0.8757	0.8759

On the other hand, the combination of ECOC and Model-Refinement strategy makes a considerable performance improvement over centroid classifier. On NewsGroup, it beats centroid classifier by 4 percents; on Sector-48, it beats centroid classifier by 11 percents. More encouragingly, it yields better performance than SVM classifier on Sector-48. This improvement also indicates that Model-Refinement strategy can effectively reduce the bias incurred by ECOC.

Figure 6 displays the MicroF1 curves of different methods vs. the number of features. For ECOC, we use 63-bit BCH coding; for Model-Refinement strategy, we fix its *MaxIteration* and *LearningRate* as 8 and 0.01 respectively. From this figure, we can observe that the combination of ECOC and Model-Refinement strategy delivers consistent top-notch performance on two datasets, especially when the number of features is larger than 3000.

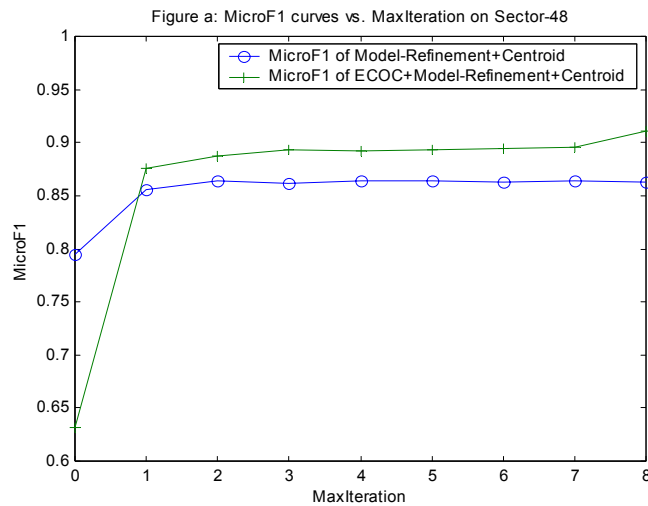
**Fig. 6(a).** MicroF1 vs. the number of features on Sector-48



**Fig. 6(b).** MicroF1 vs. the number of features on Newsgroup

Figure 7 presents the MicroF1 curves of different methods vs. iteration. For ECOC, we use 63-bit BCH coding; for Model-Refinement strategy, we fix its *LearningRate* as 0.01; the number of feature is fixed as 10,000.

As we increase the iteration for Model-Refinement strategy, the combination of ECOC and Model-Refinement strategy shows an improved performance that is to be expected. This result verifies the fact that Model-Refinement strategy can dramatically reduce the bias that ECOC brings to centroid classifier.



**Fig. 7(a).** MicroF1 of different methods vs. iteration on Sector-48

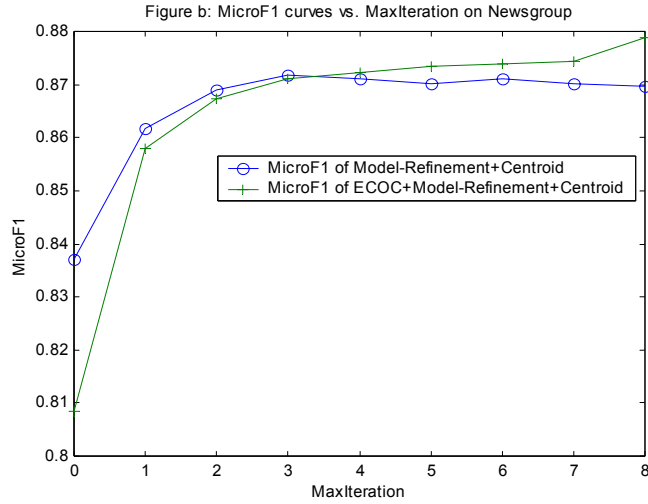


Fig. 7(b). MicroF1 of different methods vs. iteration on Newsgroup

It is worth noticing that “0” means no centroid adjustment is used. That is to say, the method “Model-Refinement+Centroid” equals to centroid classifier; the method “ECOC+Model-Refinement+Centroid” equals to “ECOC+Centroid” classifier. A noticeable observation is that the first round of centroid adjustment makes the largest performance improvement.

Table 3 and 4 report the classification accuracy of combining ECOC with Model-Refinement strategy on two datasets vs. the length BCH coding. For Model-Refinement strategy, we fix its *MaxIteration* and *LearningRate* as 8 and 0.01 respectively; the number of features is fixed as 10,000.

We can clearly observe that increasing the length of the codes increases the classification accuracy. However, the increase in accuracy is not directly proportional to the increase in the length of the code. As the codes get larger, the accuracies start leveling off as we can observe from the two tables.

This phenomenon is also observed in Ghani’s work [8] and he gave an explanation for this case: The longer a code is, the more separated the individual codewords can be, thus having a larger minimum Hamming distance and improving the error-correcting ability.

Table 3. The MicroF1 vs. the length of BCH coding

Dataset \ Bit	15bit	31bit	63bit
Sector-48	0.8461	0.8948	0.9105
NewsGroup	0.8463	0.8745	0.8788

**Table 4.** The MacroF1 vs. the length of BCH coding

Dataset \ Bit	15bit	31bit	63bit
Sector-48	0.8459	0.8961	0.9122
NewsGroup	0.8430	0.8714	0.8757

## 6 Conclusion Remarks

In this work, we examine the use of ECOC for improving centroid text classifier. The implementation framework is to decompose one multi-class problem into multiple binary problems and then learn the individual binary classification problems by centroid classifier. Meanwhile, Model-Refinement strategy is employed to reduce the bias incurred by ECOC. Furthermore, we present the theoretical justification and analysis for Model-Refinement strategy.

In order to investigate the effectiveness and robustness of proposed method, we conduct an extensive experiment on two commonly used corpora, i.e., Industry Sector and Newsgroup. The experimental results indicate that the combination of ECOC with Model-Refinement strategy makes a considerable performance improvement over traditional centroid classifier, and even performs comparably with SVM classifier.

The results reported here are not necessarily the best that can be achieved. Our future effort is to seek new techniques to enhance the performance of ECOC for centroid text classifier. Additionally, we will investigate the effectiveness of proposed method on multi-label text classification problems.

## Acknowledgments

This work was mainly supported by two funds, i.e., 0704021000 and 60803085, and one another project, i.e., 2004CB318109.

## References

1. Yang, Y., Lin, X.: A re-examination of text categorization methods. In: SIGIR, pp. 42–49 (1999)
2. Han, E., Karypis, G.: Centroid-Based Document Classification Analysis & Experimental Result. In: Zighed, D.A., Komorowski, J., Żytkow, J.M. (eds.) PKDD 2000. LNCS (LNAI), vol. 1910, pp. 424–431. Springer, Heidelberg (2000)
3. McCallum, A., Nigam, K.: A Comparison of Event Models for Naive Bayes Text Classification. In: AAAI/ICML 1998 Workshop on Learning for Text Categorization, pp. 41–48. AAAI Press, Menlo Park (1998)
4. van Mun, P.P.T.M.: Text Classification in Information Retrieval using Winnow, <http://citeseer.ist.psu.edu/cs>

5. Tan, S.: Neighbor-weighted K-nearest neighbor for unbalanced text corpus. *Expert Systems With Applications* 28(4), 667–671 (2005)
6. Joachims, T.: Text categorization with support vector machines: learning with many relevant features. In: Nédellec, C., Rouveirol, C. (eds.) *ECML 1998*. LNCS, vol. 1398, pp. 137–142. Springer, Heidelberg (1998)
7. Berger, A.: Error-Correcting Output Coding for text classification. In: *IJCAI (1999)*
8. Ghani, R.: Using error-correcting codes for text classification. In: *ICML (2000)*
9. Ghani, R.: Combining labeled and unlabeled data for multiclass text categorization. In: *ICML (2002)*
10. Rennie, J., Rifkin, R.: Improving multiclass text classification with the support vector machine. In: *AI Memo AIM-2001-026*. MIT, Cambridge (2001)
11. Tan, S., Cheng, X., Ghanem, M., Wang, B., Xu, H.: A novel refinement approach for text categorization. In: *CIKM 2005*, pp. 469–476 (2005)
12. Tan, S.: An Effective Refinement Strategy for KNN Text Classifier. *Expert Systems With Applications* 30(2), 290–298 (2006)
13. Wu, H., Phang, T.H., Liu, B., Li, X.: A Refinement Approach to Handling Model Misfit in Text Categorization. In: *SIGKDD*, pp. 207–216 (2002)
14. Yang, Y., Pedersen, J.O.: A Comparative Study on Feature Selection in Text Categorization. In: *ICML*, pp. 412–420 (1997)
15. van Rijsbergen, C.: *Information Retrieval*. Butterworths, London (1979)
16. Peterson, W., Weldon, E.: *Error-Correcting Codes*. MIT Press, Cambridge (1972)
17. Lewis, D., Schapire, R., Callan, J., Papka, R.: Training algorithms for linear text classifiers. In: *SIGIR*, pp. 298–306 (1996)
18. Chai, K., Ng, H., Chieu, H.: Bayesian online classifiers for text classification and filtering. In: *SIGIR*, pp. 97–104 (2002)
19. Shankar, S., Karypis, G.: Weight adjustment schemes for a centroid-based classifier. In: *TextMining Workshop, KDD (2000)*
20. Godbole, S., Sarawagi, S., Chakrabarti, S.: Scaling multi-class support vector machine using inter-class confusion. In: *SIGKDD*, pp. 513–518 (2002)
21. Vapnik, V.: *Statistical Learning Theory*. John Wiley and Sons, New York (1998)
22. Apte, C., Damerau, F., Weiss, S.: Text mining with decision rules and decision trees. In: *Proceedings of the Workshop with Conference on Automated Learning and Discovery: Learning from text and the Web (1998)*
23. Schapire, R., Singer, Y.: Boostexter: A boosting-based system for text categorization. *Machine Learning* 39, 135–168 (2000)
24. Schapire, R., Freund, Y., Bartlett, P., Lee, W.: Boosting the Margin: A New Explanation for the Effectiveness of Voting Method. *The Annals of Statistics* 26(5), 1651–1686 (1998)