Using Error-Correcting Output Codes with Model-Refinement to Boost Centroid Text Classifier

Songbo Tan
Information Security Center, ICT, P.O. Box 2704, Beijing, 100080, China
tansongbo@software.ict.ac.cn, tansongbo@gmail.com

Abstract

In this work, we investigate the use of error-correcting output codes (ECOC) for boosting 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. 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 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 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

In recent years, ECOC has been applied to boost the naïve bayes, decision tree and SVM classifier for text data (Berger 1999, Ghani 2000, Ghani 2002, Rennie et al. 2001). Following this research direction, in this work, we explore the use of ECOC to enhance the performance of centroid classifier (Han et al. 2000). To the best of our knowledge, no previous work has been conducted on exactly this problem. 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 (Liu et al. 2002) for centroid classifier. In substance, centroid classifier (Han et al. 2000) 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 violated 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 (Tan et al. 2005) 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.

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 can dramatically decrease the bias introduce by ECOC, and the resulted classifier is comparable to or even better than SVM classifier in performance.

2. Error-Correcting Output Coding

Error-Correcting Output Coding (ECOC) is a form of combination of multiple classifiers (Ghani 2000). It works by converting a multi-class supervised learning problem into a large number (L) of two-class supervised learning problems (Ghani 2000). Any learning algorithm that can handle two-class learning problems, such as Naïve Bayes (Sebastiani 2002), can then be applied to learn each of these L problems. L can then be thought of as the length of the codewords

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 81–84,
Prague, June 2007.

with one bit in each codeword for each classifier. The ECOC algorithm is outlined in Figure 1.

TRAINING

1 Load training data and parameters, i.e., the length of code L and training class K.

2 Create a L-bit code for the K classes using a kind of coding algorithm.

3 For each bit, train the base classifier using the binary class (0 and 1) over the total training data.

TESTING

1 Apply each of the L classifiers to the test example.

2 Assign the test example the class with the largest votes.

Figure 1: Outline of ECOC

3. Methodology

3.1 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 Figure 2. Class 0 consists of 6 original categories, and class 1 contains another 6 categories. Then we calculate the centroids 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.

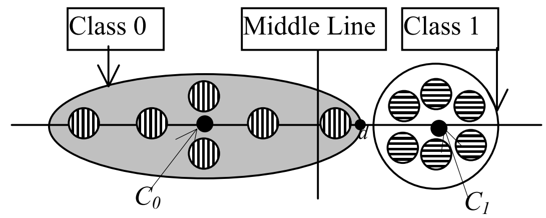

Figure 2: Original Centroids of Merged Class 0 and Class 1

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.

3.2 Why Model-Refinement can reduce this bias?

In order to decrease this kind of bias, we employ the Model-Refinement to adjust the class representative, i.e., the centroids. The basic idea of Model-Refinement 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.

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!=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);

Figure 3: Outline of Model-Refinement Strategy

For example, if document d of class 1 is misclassified into class 2, both centroids C_1 and C_2 should be moved right by the following formulas (3-4) respectively,

$$C_1^* = C_1 + \eta \cdot d \tag{3}$$

$$C_2^* = C_2 - \eta \cdot d \tag{4}$$

where η ($0<\eta<1$) is the *Learning Rate* which controls the step-size of updating operation.

The Model-Refinement for centroid classifier is outlined in Figure 3 where *MaxIteration* denotes the pre-defined steps for iteration. More details can be found in (Tan et al. 2005). The time requirement of Model-Refinement is $O(MTKW)$ where M denotes the iteration steps.

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.

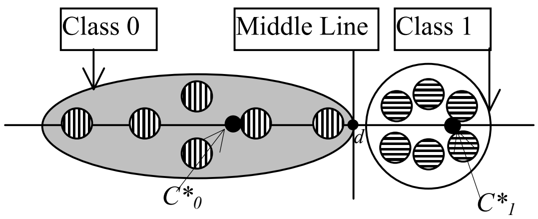

Figure 4: Refined Centroids of Merged Class 0 and Class 1

3.3 The combination of ECOC and Model-Refinement for centroid classifier

In this subsection, we present the outline (Figure 5) of combining ECOC and Model-Refinement for centroid classifier. In substance, the improved ECOC combines the strengths of ECOC and Model-Refinement. 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 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.

TRAINING

1 Load training data and parameters, i.e., the length of code L and training class K.

2 Create a L-bit code for the K classes using a kind of coding algorithm.

3 For each bit, train centroid classifier using the binary class (0 and 1) over the total training data.

4 Use Model-Refinement approach to adjust centroids.

TESTING

1 Apply each of the L classifiers to the test example.

2 Assign the test example the class with the largest votes.

Figure 5: Outline of combining ECOC and Model-Refinement

4. Experiment Results

4.1 Datasets

In our experiment, we use two corpora: NewsGroup[1], and Industry Sector[2].

NewsGroup The NewsGroup 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 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.

4.2 Experimental Design

To evaluate a text classification system, we use MicroF1 and MacroF1 measures (Chai et al. 2002). We employ Information Gain as feature selection method because it consistently performs well in most cases (Yang et al. 1997). We employ TFIDF (Sebastiani 2002) to compute feature weight. For SVM classifier we employ SVMTorch. (www.idiap.ch/~bengio/projects/SVMTorch.html).

4.3 Comparison and Analysis

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

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.

On the other hand, the combination of ECOC and Model-Refinement makes a significant performance improvement over centroid classifier.

1 www-2.cs.cmu.edu/afs/cs/project/theo-11/www/wwkb.

2 www-2.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/.

On Newsgroup, it beats centroid classifier by 4 percents; on Sector-48, it beats centroid classifier by 11 percents. More encouraging, it yields better performance than SVM classifier on Sector-48. This improvement also indicates that Model-Refinement can effectively reduce the bias incurred by ECOC.

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	**0.9122**	0.8948
NewsGroup	0.8371	0.8697	0.8085	**0.8788**	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	**0.9138**	0.8970
NewsGroup	0.8331	0.8661	0.7936	0.8757	0.8759

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

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

Bit / Dataset	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

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

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.

5. Conclusion Remarks

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

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 makes a considerable performance improvement over traditional centroid classifier, and even performs comparably with SVM classifier.

References

Berger, A. *Error-correcting output coding for text classification*. In Proceedings of IJCAI, 1999.

Chai, K., Chieu, H. and Ng, H. *Bayesian online classifiers for text classification and filtering*. SIGIR. 2002, 97-104

Ghani, R. *Using error-correcting codes for text classification*. ICML. 2000

Ghani, R. *Combining labeled and unlabeled data for multiclass text categorization*. ICML. 2002

Han, E. and Karypis, G. *Centroid-Based Document Classification Analysis & Experimental Result*. PKDD. 2000.

Liu, Y., Yang, Y. and Carbonell, J. *Boosting to Correct Inductive Bias in Text Classification*. CIKM. 2002, 348-355

Rennie, J. and Rifkin, R. *Improving multiclass text classification with the support vector machine*. In MIT. AI Memo AIM-2001-026, 2001.

Sebastiani, F. *Machine learning in automated text categorization*. ACM Computing Surveys, 2002,34(1): 1-47.

Tan, S., Cheng, X., Ghanem, M., Wang, B. and Xu, H. *A novel refinement approach for text categorization*. CIKM. 2005, 469-476

Yang, Y. and Pedersen, J. *A Comparative Study on Feature Selection in Text Categorization*. ICML. 1997, 412-420.