Available online at www.sciencedirect.com
ScienceDirect
Information Processing and Management 43 (2007) 1281–1293
INFORMATION PROCESSING & MANAGEMENT
www.elsevier.com/locate/infoproman

Using the revised EM algorithm to remove noisy data for improving the one-against-the-rest method in binary text classification

Hyoungdong Han [a], Youngjoong Ko [b,*], Jungyun Seo [a]

[a] *Department of Computer Science and Program of Integrated Biotechnology, Sogang University, Sinsu-dong 1, Mapo-gu, Seoul 121-742, Republic of Korea*
[b] *Department of Computer Engineering, Dong-A University, 840 Hadan 2-dong, Saha-gu, Busan 604-714, Republic of Korea*

Received 7 March 2006; received in revised form 8 November 2006; accepted 9 November 2006
Available online 18 January 2007

Abstract

Automatic text classification is the problem of automatically assigning predefined categories to free text documents, thus allowing for less manual labors required by traditional classification methods. When we apply binary classification to multi-class classification for text classification, we usually use the one-against-the-rest method. In this method, if a document belongs to a particular category, the document is regarded as a positive example of that category; otherwise, the document is regarded as a negative example. Finally, each category has a positive data set and a negative data set. But, this one-against-the-rest method has a problem. That is, the documents of a negative data set are not labeled manually, while those of a positive set are labeled by human. Therefore, the negative data set probably includes a lot of noisy data. In this paper, we propose that the sliding window technique and the revised EM (Expectation Maximization) algorithm are applied to binary text classification for solving this problem. As a result, we can improve binary text classification through extracting potentially noisy documents from the negative data set using the sliding window technique and removing actually noisy documents using the revised EM algorithm. The results of our experiments showed that our method achieved better performance than the original one-against-the-rest method in all the data sets and all the classifiers used in the experiments.

Keywords: Binary text classification; The one-against-the-rest method; The EM algorithm; The sliding window technique

1. Introduction

Text classification, or the task of automatically assigning semantic categories to natural language texts, has become one of the key methods for organizing online information. It is a basic building block in a wide range

* Corresponding author. Tel.: +82 51 200 7782; fax: +82 51 200 7783.
E-mail addresses: hdhan@sogang.ac.kr (H. Han), yjko@dau.ac.kr (Y. Ko), seojy@sogang.ac.kr (J. Seo).

0306-4573/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ipm.2006.11.003

of applications. For example, directories like Yahoo! Categorize Web pages by topic, online newspapers customize themselves to a particular user's reading preferences, and routing agents at service hotlines forward incoming emails to appropriate experts by contents. To organize training examples for learning tasks, binary setting or multi-class setting can be used. As the binary setting consists of only two classes, it is the simplest, yet most important formulation of the learning problem. These two classes can be composed of "relevant (positive)" and "non-relevant (negative)" for information retrieval applications (Joachims, 2002).

Generally, some classification tasks involve more than two classes. When we apply the binary setting to the multi-class setting with more than two classes, there is a problem that the multi-class setting consists of only positive examples of each category; each category does not have negative examples. In order to solve this problem, the one-against-the-rest method has been used in many cases (Hsu & Lin, 2002; Zadrozny & Elkan, 2001; Zadrozny & Elkan, 2002); it can reduce a multi-class problem into many binary tasks. Fig. 1 shows how the multi-class setting with four categories (e.g. 'politics', 'economics', 'society', and 'sports') changed into the binary settings using the one-against-the-rest method. For example, the documents of the 'politic' category are considered as positive examples and those of the other categories are as negative examples. That is, while all the documents of a category are generated as positive examples by hand, documents that do not belong to the category regard as negative examples indirectly. This labeling task concentrates on only selecting positive examples for each category, and it does not label the negative examples which have the opposite meaning of counterpart positive category directly. Thus the negative data set in the one-against-the-rest method probably include noisy examples. In addition, because the negative data set consists of the different distributions of positive examples from various categories, it is hard to be considered as the exact negative examples of each category.

These noisy documents can be one of the major causes of decreasing the performance for binary text classification. Thus classifiers need to efficiently handle these noisy documents to achieve the high performance. There are two problems to efficiently remove them from training data as follows:

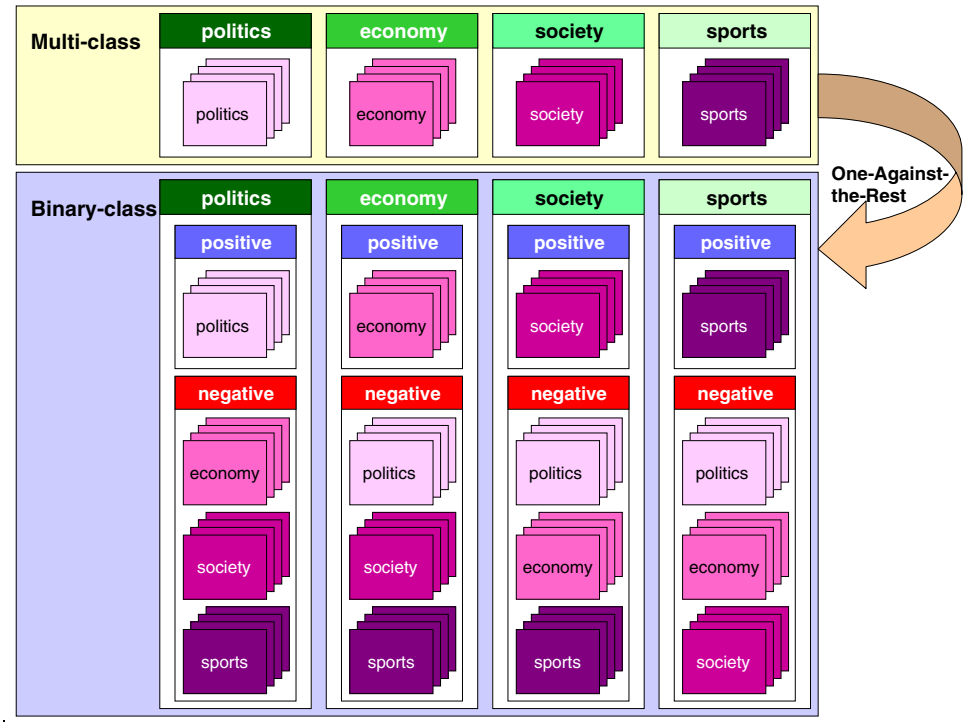

Fig. 1. Organizing the training data set by using the one-against-the-rest method.

(1) "*How can we find a boundary area containing many noisy documents?*"
The noisy documents are usually located in a boundary area between positive documents and negative documents. The sliding window technique and entropy are employed to effectively detect the boundary area. By estimating the entropy of mixed positive and negative documents in a window size, the boundary area are found and all the documents in the area are regards as unlabeled data (as candidate documents for noisy data).
(2) "*How can we deal with noisy documents found from the boundary?*"
This paper exploits the revised EM (Expectation Maximization) algorithm to efficiently handle unlabeled documents which are extracted from the previous step. The revised EM algorithm provides us the solution to extract and remove noisy documents from unlabeled data.

The rest of this paper is organized as follows. Section 2 presents previous related work. In Section 3, we explain the proposed method in detail. Section 4 is devoted to the analysis of the empirical results. Section 5 describes conclusions and future work.

2. Related work

In the literature, there are various studies that aim to obtain reliable training data to improve text classifiers in binary text classification. Some studies focused on learning from positive data and unlabled data (Li & Liu, 2003; Liu, Lee, Yu, & Li, 2002; Yu, Han, & Chang, 2002). An alternative method is to employ active learning for text classification (Roy & McCallum, 2001; Schohn & Cohn, 2000; Tong & Koller, 2001).

Liu et al. studied the problem of classification with only partial information, one class of labeled (positive) documents, and a set of mixed documents (Liu et al., 2002). The main idea of this method is to first use a spy technique to identify some reliable negative documents from the unlabeled set. They theoretically showed that there is enough information in positive and unlabeled data to build accurate classifiers and proposed a novel technique to solve the problem by utilizing the EM algorithm (S-EM) with the Naive Bayes classification method.

Yu et al. proposed an SVM based technique (called PEBL) to classify Web pages given positive and unlabeled pages (Yu et al., 2002). The core idea is the same as that in Liu et al. (2002), i.e. (1) identifying a set of reliable negative documents from the unlabeled set (called strong negative documents in PEBL), and (2) building a classifier using SVM. Strong negative documents are ones that do not contain any features of the positive data. After a set of strong negative documents is identified, SVM is iteratively applied to the construction of a classifier.

Li and Liu proposed a system which combines the Rocchio model with the SVM technique (Li & Liu, 2003). This system consists of two steps: (1) extracting some reliable negative documents from the unlabeled set, (2) applying SVM iteratively to building a classifier. In first step, Li gives two methods for finding reliable negative documents; Rocchio and Rocchio with clustering. The clustering technique that Li uses is k-means. The experiment using Rocchio with clustering showed better result than an experiment using Rocchio.

Tong and Koller introduced a new algorithm for performing active learning with SVM (Tong & Koller, 2001). By taking advantage of the duality between parameter space and feature space, they arrived at three algorithms that attempt to reduce version space as much as possible at each query. These three algorithms can provide considerable gains in both inductive and transductive settings.

Roy and McCallum presented an active learning method that directly optimizes expected future errors (Roy & McCallum, 2001). This becomes feasible by taking a sampling approach to estimating the expected reduction in error due to the labeling of a query.

Schohn and Cohn described a simple active learning heuristic which greatly enhances the generalization behavior of SVMs (Schohn & Cohn, 2000). They achieved better performance from a small subset of the data than all available data is used.

While many previous studies achieved the improved performance on specific classifiers such as SVM, we focus on improving the one-against-the-rest method, which can be applied to all kinds of classifiers. As a result, the proposed method also obtained remarkable improved performance on all the classifiers (k-NN, Naive Bayes, Rocchio, SVM) and all the test data sets (Reuters, WebKB, Newsgroups) in our experiments. This shows that the proposed method can contribute to improvement for text classification generally.

3. The proposed binary text classification method

This section will explain the proposed approach in detail. It consists of the following four steps: (1) applying the one-against-the-rest method, (2) calculating prediction scores, (3) calculating entropy using the sliding window technique, (4) the revised EM algorithm.

3.1. The one-against-the-rest method

In the one-against-the-rest method, the documents of one category are regarded as positive examples and the documents of the other categories as negative examples (Hsu & Lin, 2002; Zadrozny & Elkan, 2001; Zadrozny & Elkan, 2002). In order to set up training data into binary classification, multi-class setting is reformed into the binary setting using the one-against-the-rest method such as Fig. 1.

3.2. Calculating prediction scores

The goal of this and the following sections is to find a boundary area which denotes a region including many noisy documents. First of all, using a positive data set and a negative data set for each category from the one-against-the-rest method, we can learn a Naive Bayes (NB) classifier and we can obtain a prediction score for each document by the following formula.

$$\text{Prediction\_Score}(c_i|d_j) = \frac{P(\text{Positive}|d_j)}{P(\text{Positive}|d_j) + P(\text{Negative}|d_j)} \tag{1}$$

where c_i means a category and d_j means a document of c_i. $P(\text{Positive}|d_j)$ means a probability of the document d_j to be positive in c_i, and $P(\text{Negative}|d_j)$ means a probability of the document d_j to be negative in c_i.

According to these prediction scores, the entire documents of each category are sorted out in the descending order. Probabilities, $P(\text{Positive}|d_j)$ and $P(\text{Negative}|d_j)$, of formula (1) is generally calculated by the Naive Bayes formula as follows (Craven et al., 2000; Ko & Seo, 2004; Lewis, 1998):

$$\begin{aligned} P(\text{Positive}|d_j) &= \frac{P(\text{Positive})P(d_j|\text{Positive})}{P(d_j)} = P(\text{Positive})\prod_{i=1}^{T} P(t_i|\text{Positive})^{N(t_i|d_j)} \\ &\propto \frac{\log(P(\text{Positive}))}{n} + \sum_{i=1}^{|T|} P(t_i|d_j)\log\left(\frac{P(t_i|\text{Positive})}{P(t_i|d_j)}\right) \end{aligned} \tag{2}$$

where t_i is the i-th word in the vocabulary, T is the size of the vocabulary, and $N(t_i|d_j)$ is the frequency of word t_i in document d_j.

Note that we use the Naive Bayes formula as the base in the proposed method because it has a strong foundation for EM and is more efficient.

3.3. Calculating entropy using the sliding window technique

In our method, a boundary can be detected in a block with the most mixed degree of positive and negative documents. The sliding window technique is first used to detect the block (Lee, Lin, & Chen, 2001). In this technique, windows of a certain size are sliding from the top document to the last document in a list ordered by the prediction scores. An entropy value is calculated for estimating the mixed degree of each window as follows (Mitchell, 1997):

$$\text{Entropy}(W) = -p_+\log_2 p_+ - p_-\log_2 p_- \tag{3}$$

where, given a window (W), p_+ is the proportion of positive documents in W and p_- is the proportion of negative documents in W. For example, if a window of five documents has three positive documents and two negative documents, the proportions of positive documents and negative documents are $\frac{3}{5}$ and $\frac{2}{5}$ respectively. Thus

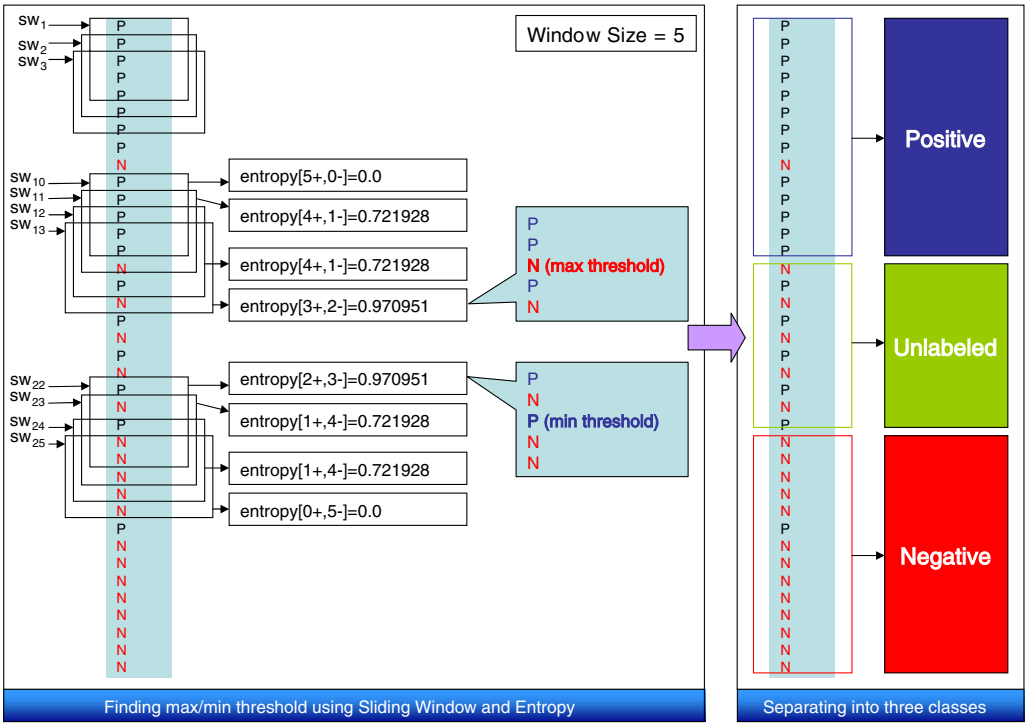

Fig. 2. Finding the boundary area using the sliding window technique and entropy.

the final estimated entropy value is calculated by this formula ($\text{Entropy}(W) = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} = 0.970951$). This value is the highest entropy value when using five documents as window size.

Two windows with the highest entropy value are picked up; one window is firstly detected from the top and the other is firstly detected from the bottom. If there are no window or only one window with the highest entropy value, windows with the next highest entropy value become targets of the selected windows. Then maximum (*max*) and minimum (*min*) threshold values can be searched from selected windows, respectively. The *max* threshold value is found as the highest prediction score of a negative document in the former window and the *min* threshold value is as the lowest prediction score of a positive document in the latter window. The left side of Fig. 2 explains how to set up *max* and *min* threshold values.

We regard the documents between *max* and *min* threshold values as unlabeled documents. These documents are considered as potentially noisy documents.

Now three classes for training documents of each category are constructed just like the right side of Fig. 2: definitely positive documents, unlabeled documents, definitely negative documents. By applying the revised EM algorithm to these three data sets, we can extract actual noisy documents and remove them.

3.4. The revised EM algorithm

In this paper, the EM algorithm is used to pick out noisy documents from unlabeled data and to remove them. The general EM algorithm consists of two steps, the *Expectation* step and the *Maximization* step (Dempster, Laird, & Rubin, 1997). This algorithm first trains a classifier using the available labeled documents and labels the unlabeled documents by hard classification (*Expectation* (E or E') step). It then trains a new classifier using the labels of all the documents (*Maximization* (M) step), and iterates to convergence. The Naive Bayes classifier is used in the two steps of the EM algorithm. Fig. 3 shows how the EM algorithm is revised in our method.

E'-step is reformed to effectively remove the noise documents located in the boundary area. Unlike original E-step, it does not assign an unlabeled document, d_u, to the positive data set, P, because it regards d_u as

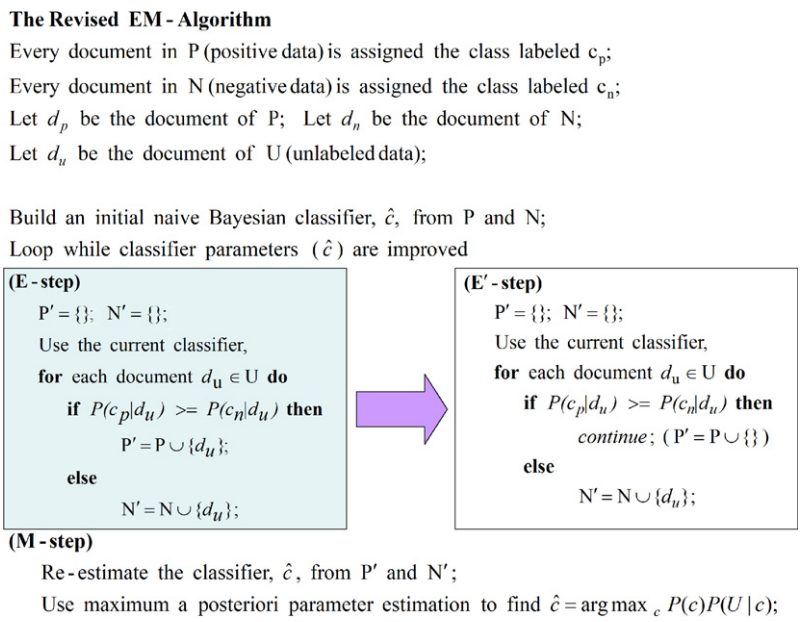

Fig. 3. The revised EM algorithm.

another noisy document; since positive documents are labeled by hand and have enough information for a category, additional positive documents can decrease performance. Finally, we can learn the text classifiers with binary training data generated by the revised EM algorithm.

4. Empirical evaluation

In this section, we provide empirical evidences that the proposed method is effective in improving binary text classification. We present experimental results with three different test data sets: UseNet newsgroups (*Newsgroups*), web pages (*WebKB*), and newswire articles (*Reuters*). Results show that the proposed method outperforms the original one-against-the-rest method.

4.1. Data sets and experimental settings

The *Newsgroups* data set, collected by Ken Lang, contains about 20,000 articles evenly divided among 20 UseNet discussion groups (McCallum & Nigam, 1998). Many of the categories fall into confusable clusters; for example, five of them are comp. *Discussion groups, and three of them discuss about religion. After removing words that occur only once or on a stop word list, the average training data vocabulary over all five folds has 51,325 words (with no stemming).

The second data set comes from the *WebKB* project at CMU (Craven et al., 2000). This data set contains web pages gathered from university computer science departments. The pages are divided into seven categories: course, faculty, project, student, department, staff, and other. In this paper, we used the four most populous entity-representing categories: course, faculty, project, and student. The resulting data set consists of 4198 pages with an average vocabulary of 18,742 words over all five folds. It is an uneven data set; the largest category has 1641 pages and the smallest one has 503 pages.

The *Reuters 21578* Distribution 1.0 data set consists of 12,902 articles and 90 topic categories from the Reuters newswire. In our experiments, we used only the ten most populous categories out of the 90 topic categories to identify the news topic. Since the documents in this data set can have multiple category labels, each category is traditionally evaluated with a binary classifier. In our experimental setting, when one is used as the positive category, the other nine categories are regarded as the negative ones. To split train/test data, we follow a stan-

dard 'ModApte' split. The standard 'ModApte' train/test split divides the articles by time. We used all the words inside the title and body, a stoplist, and no stemming. The vocabulary from training data has 21,023 words.

For fair evaluation in Newsgroups and WebKB, we used the five-fold cross-validation method. That is, each data set was split into five subsets, and each subset was used once as test data in a particular run, while the remaining subsets were used as training data for that run. The split into training and test sets for each run was the same for all the classifiers. Therefore, all the results of the experiments are averages of five runs.

In the preprocessing step to extract features from each document, the content words were extracted from all the documents by the Brill POS tagger (Brill, 1995). Words with noun or verb POS tags were considered as content words. In addition, we implemented conventional classifiers for experiments: k-NN, Naive Bayes (NB), Rocchio, and SVM. The text classifiers except SVM can handle multi-class problem directly. But most text classification tasks fall into the multi-label setting. Unlike in the multi-class case, there is no one-to-one correspondence between category and document. That is, each document can be in multiple, exactly one, or no category at all such as the Reuters data set. This motivates that a multi-label task can also be split up into a set of binary classification tasks by using the one-against-the-rest strategy (Joachims, 2002). The k in k-NN was set to 30, and $\alpha = 16$ and $\beta = 4$ were used in the Rocchio classifier (Ko, Park, & Seo, 2004). For SVM, we used the linear model offered by $\text{SVM}^{\text{light}}$.

As performance measures, the standard definition of recall and precision is used. The *micro-averaging method* and *the macro-averaging method* are applied for evaluating performance average across categories (Yang, Slattery, & Ghani, 2002). Results are reported as the precision-recall BEP (BreakEven Points), which is a standard information retrieval measure for binary classification; given a ranking of documents, the precision-recall breakeven point is the value at which precision and recall are equal (Joachims, 1998; Ko & Seo, 2004; Yang, 1999).

4.2. Experimental results

This section provides empirical evidences for the effectiveness of the proposed method. The experimental results show that the proposed method achieved better performance than the original one-against-the-rest method in all the three training data sets and all the four classifiers.

4.2.1. The experiments for setting parameters

The purpose of experiments in this section is to make a decision about setting several parameters before main experiments are conducted. The data for these experiments is composed of 5408 training documents and 1802 validation documents, which are created from 6490 Reuter training documents. The Naive Bayes classifier is selected for these experiments, and the basis system denotes the Naive Bayes classifier using the original one-against-the-rest method. The parameter adjustment depends on the data set of application domain. However, since finding optimal parameters for each domain is a very tedious task, we attempt to recommend the range of optimal parameters. Thus we first found the optimal parameters on the Reuters data set and used them for the other data sets (Newsgroups and WebKB). As a result, remarkable improvements were also achieved on the Newgroups and WebKB data sets even though the optimal parameters were not obtained from them directly.

4.2.1.1. Verifying the effectiveness of positive documents in the one-against-the-rest method. As we explained in Section 3.4, we suppose that if a sufficient amount of positive data is provided by manually labeling task, it has the sufficient information for a category. Thus we think that any modification of the positive data by machine may not be helpful to the improvement of text classification. The experiments in this subsection give proofs for our assumption. In first experiment, we verify the E'-step of the revised EM algorithm proposed in the Section 3.4; the E'-step of the revised EM algorithm is based on our assumption. The performances of the original EM algorithm (E-step) is compared to that of the revised EM algorithm (E'-step) in Table 1.

As you can see in Table 1, we achieved higher performance in the revised EM algorithm than in the original EM algorithm.

Table 1
The comparison of the original EM algorithm and the revised EM algorithm

	Basis system	The original EM algorithm (E-step)	The revised EM algorithm (E′-step)
Micro-avg. BEP	89.75	90.52	*92.31*

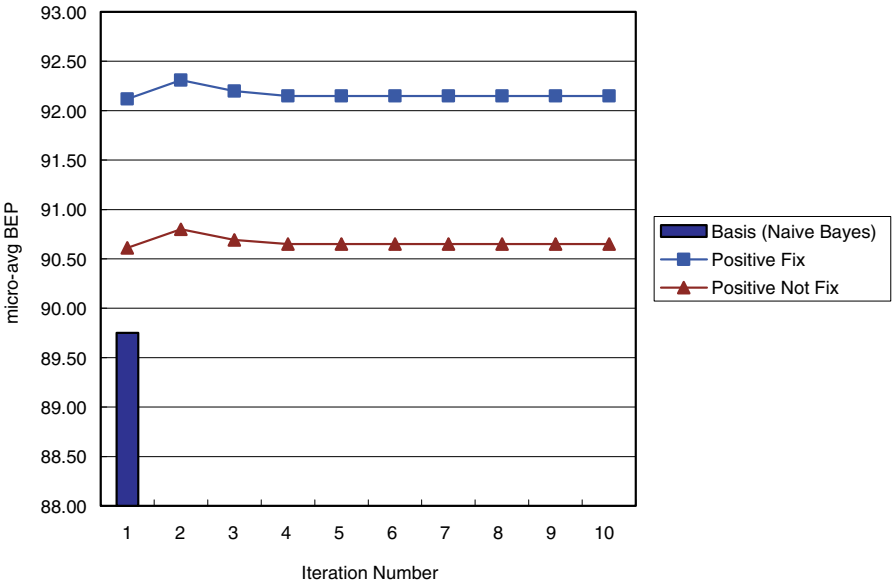

Fig. 4. The performance changes of the proposed method according to iteration numbers in the '*Positive Fix*' and '*Positive Not Fix*' cases.

For further evaluation, the next experiment tests whether it is more effective that the whole positive documents are used as positive training data without any modification; all the positive documents are regarded as only definitely positive ones and they are never classified as the unlabeled data or the definitely negative data among three classes shown in Fig. 2. This experimental setting is denoted as '*Positive Fix*' and otherwise as '*Positive Not Fix*'. Fig. 4 shows the difference of performances between the '*Positive Fix*' case and '*Positive Not Fix*' case. In this experiment, the revised EM algorithm (E'-step) was applied and the performance changes according to the iteration number were observed.

As a result, '*Positive Fix*' setting obtained better performance than '*Positive Not Fix*' in all the iteration numbers, and the best performance was achieved in second iteration. Thus our basic assumption, that any modification of the positive data by machine may not be helpful, is verified through these experiments. The semi-supervised models such as our revised EM algorithm find a better local maximum than unsupervised models since their initialization is closer to the desired one. As shown in Fig. 4, the performances drop with EM iterations before converging. This decrease generally happens except when only a limited amount of hand-labeled example is available (Merialdo, 1994). In addition, Liu et al. also used two iterations for their S-EM algorithm (Liu et al., 2002). Therefore, the revised EM algorithm with two iterations and '*Positive Fix*' setting is applied to the following experiments.

4.2.1.2. Comparing the performances in different window sizes. We here observe the performance changes according to several window sizes used in the sliding window technique. Table 2 shows the performances when windows sizes are 3, 5, and 7. As a result, the window size is fixed as 5.

Table 2
The comparison of performances in each window size

Window size	3	5	7
Micro-avg. BEP	91.54	*92.31*	92.10

4.2.2. The experimental results to verify the proposed method in each text classifier and each data set

To evaluate the effectiveness of the proposed method, we implemented four different text classifiers (Naive Bayes, *k*-NN, Rocchio, and SVM). And the performance of the original one-against-the-rest method is compared to that of the proposed method on three test data sets. Tables 3–5 show the experimental results from each text classifier in the Reuters data set, the WebKB data set, and the Newsgroups data set, respectively. As a result, the proposed method achieved better performances than the original method over all the classifiers and all the data sets. Note that the proposed method obtained the improved performances in even all the categories of each data set. This is an obvious proof that the proposed method is more effective than the original one-against-the-rest method.

As shown in Tables 3–5, SVM achieved less improvement than the other classifiers. It is caused by the fact that the performance of SVM using the original one-against-the-rest method is too high in all the data sets. Note that it is more difficult to improve a classifier with higher performance.

Table 3
Results in the Reuters data set

Category	Classifier							
	k-NN		NB		Rocchio		SVM	
	Original method	Proposed method	Original method	Proposed method	Original method	Proposed method	Original method	Proposed method
Acq	96.64	96.64	96.80	97.89	96.52	97.80	97.64	97.64
Corn	64.44	71.65	60.71	71.58	57.14	65.24	87.50	89.29
Crude	89.27	92.56	88.89	92.45	87.30	90.15	88.89	91.01
Earn	97.50	98.75	96.96	98.14	96.78	96.87	98.80	98.80
Grain	90.27	91.26	89.93	90.94	89.93	91.28	95.30	96.64
Interest	75.68	81.67	76.34	76.67	68.70	74.81	84.73	84.73
Money	78.66	80.59	79.33	80.90	74.30	77.65	82.68	82.68
Ship	86.64	87.59	83.15	92.26	78.65	83.15	88.76	89.89
Trade	82.53	86.33	77.12	92.37	60.17	80.73	89.83	90.68
Wheat	65.00	67.28	63.38	70.67	66.20	77.83	84.51	85.92
Micro-avg.BEP	91.47	94.27 (*+3.06*)	90.80	93.86 (*+3.37*)	89.24	91.80 (*+2.86*)	94.66	95.52 (*+0.91*)
Macro-avg.BEP	82.66	85.43 (*+3.34*)	81.26	86.38 (*+6.31*)	77.56	83.55 (*+7.71*)	89.86	90.72 (*+0.96*)

Table 4
Results in the WebKB data set

Category	Classifier							
	k-NN		NB		Rocchio		SVM	
	Original method	Proposed method	Original method	Proposed method	Original method	Proposed method	Original method	Proposed method
Course	81.59	84.07	83.46	85.67	83.02	86.01	90.15	91.35
Faculty	82.82	86.25	84.25	87.83	84.29	87.90	92.06	92.46
Project	80.21	83.67	81.22	83.27	82.69	85.24	89.76	90.43
Student	83.92	88.23	85.39	89.37	84.87	89.00	94.12	94.45
Micro-avg.BEP	84.97	86.74 (*+2.08*)	85.67	87.21 (*+1.8*)	86.52	88.26 (*+2.01*)	92.12	92.64 (*+0.56*)
Macro-avg.BEP	82.13	85.55 (*+4.16*)	83.58	86.53 (*+3.53*)	83.71	87.03 (*+3.96*)	91.52	92.17 (*+0.71*)

Table 5
Results in the Newsgroups data set

Category	Classifier							
	k-NN		NB		Rocchio		SVMs	
	Original method	Proposed method	Original method	Proposed method	Original method	Proposed method	Original method	Proposed method
Atheism	73.73	75.49	75.59	76.80	71.57	73.72	77.42	79.77
Graphics	79.85	80.41	74.18	76.69	74.38	78.03	81.91	83.66
Windows.misc	77.78	79.84	65.69	73.70	76.51	78.16	83.78	85.73
Pc.hardware	72.58	77.74	68.98	71.79	69.44	71.39	78.69	80.44
Mac.hardware	80.27	85.53	81.35	84.26	83.51	84.86	87.53	88.78
Windows.x	87.20	88.26	85.82	87.43	85.66	86.81	89.22	90.07
Forsale	73.20	80.56	79.89	82.20	79.87	80.82	82.76	83.11
Autos	88.25	90.01	89.43	91.34	88.72	91.67	92.84	93.70
Motorcycles	93.12	94.58	94.58	95.19	93.34	94.09	96.49	97.14
Baseball	97.38	97.94	96.78	97.49	95.86	97.71	96.26	97.11
Hockey	97.57	98.43	98.20	98.83	96.23	96.88	98.58	99.06
Crypt	95.29	96.65	93.81	94.82	90.19	91.27	96.29	97.19
Electronics	76.36	78.02	76.44	78.25	73.52	77.27	84.27	86.42
Med	92.30	94.36	92.32	92.53	89.95	91.50	93.51	94.46
Space	94.43	95.19	92.24	93.45	91.16	92.11	94.64	95.39
Christian	89.24	94.30	92.46	92.97	82.42	84.17	98.29	98.76
Guns	94.76	95.02	84.27	84.78	79.53	82.58	87.41	89.16
Midest	93.16	95.52	92.12	93.83	93.38	94.93	94.32	95.07
Politics.misc	73.81	76.97	71.43	72.54	72.26	74.21	76.64	78.09
Religion.misc	61.39	65.85	51.86	52.27	42.54	49.29	63.87	68.52
Micro-avg.BEP	86.07	87.96 (*+2.19*)	83.17	84.86 (*+2.03*)	82.84	84.48 (*+1.98*)	88.34	89.08 (*+0.84*)
Macro-avg.BEP	84.58	87.03 (*+2.89*)	82.87	84.55 (*+2.03*)	81.5	83.57 (*+2.54*)	87.73	89.08 (*+1.53*)

4.3. Discussions

4.3.1. The analysis of the performance with regards to the number of positive data and the property of categories

In this section, we first observe the relationship between the performance improvement and the number of positive documents using the Reuters data set. As shown in Table 6, the proposed method obtained remarkable improvement in the small size of categories: especially '*corn*', '*wheat*', and '*trade*' categories. The results from Tables 3–5 can become another proof; the performance differences using the macro-averaging measure are bigger than ones using the micro-averaging measure. Moreover, we could look at anther phenomenon from the results of Table 6. Categories with comprehensive contents such as '*trade*' showed much improvement and categories with mutually similar contents such as '*corn*' and '*wheat*' also did. It is caused by the fact

Table 6
The improvement scores according to the number of positive documents

Category	Positive document number	Negative document number	*k*-NN	NB	Rocchio	SVM
Corn	181	6309	+11.18	+17.9	+14.17	+2.04
Ship	197	6293	+1.09	+10.95	+5.72	+1.27
Wheat	212	6278	+3.5	+11.50	+17.56	+1.68
Interest	347	6143	+7.92	+0.43	+8.89	0
Trade	369	6121	+4.6	+19.77	+34.16	+0.94
Crude	393	6101	+3.68	+4.0	+3.26	+2.38
Grain	433	6057	+1.09	+1.12	+1.5	+1.4
Money	538	5952	+2.88	+1.97	+4.5	0
Acq	1650	4840	0	+1.12	+1.32	0
Earn	2877	3613	+1.29	+1.21	+0.09	0

that these categories can have a lot of noisy documents in the negative data set. As a result, the proposed method is more effective when there are insufficient positive data and ambiguous categories. Actually, we can frequently meet these cases in many application areas. Therefore, we believe that the proposed method is usefully applied to binary text classification applications to improve their performance.

4.3.2. The analysis of the changes of cohesion scores when applying the proposed method

A simple additional experiment was conducted to prove that the proposed method removes the noisy documents properly. Salton argued that a collection of small tightly clustered documents with wide separation between individual clusters should produce the best performance (Salton, Yang, & Wang, 1975). In this light, we employed the method used by Salton et al. (1975) to verify the proposed method.

We define the cohesion within a negative data set and the cohesion between positive and negative data sets. In binary text classification, each category consists of a positive data set and a negative data set. The cohesion within a negative data set is a measure for similarity values between documents in the negative data set of a category; note that, since a positive data set is not changed in the proposed method, only the cohesion score of the negative data set is calculated. The cohesion between positive and negative data sets is a measure for similarities between positive and negative data sets in a category. The former is calculated by formula (5), and the latter is calculated by formula (6):

$$\vec{C}_k = \frac{1}{|I_k|}\sum_{\vec{d}\in I_k}\vec{d}, \quad \vec{C}_{\text{glob}} = \frac{1}{|D|}\sum_{k=1}^{2}|I_k|\cdot\vec{C}_k \tag{4}$$

$$Co_{\text{within}} = \frac{1}{|I_2|}\sum_{\vec{d}\in I_k}\vec{d}\cdot\vec{C}_{2(\text{negative})} \tag{5}$$

$$Co_{\text{between}} = \frac{1}{|D|}\sum_{k=1}^{2}|I_k|(\vec{C}_{\text{glob}}\cdot\vec{C}_k) \tag{6}$$

where D denotes the total training data set of a category: the positive ($k=1$) and negative ($k=2$) data sets, I_k denotes k-th training data set, $\vec{C}_k$ denotes a centroid vector of k-th training data set, and $\vec{C}_{\text{glob}}$ denotes a centroid vector of the total training data.

In formulae (4), $\vec{C}_k$ is described by the mean vector of each document set, and $\vec{C}_{\text{glob}}$ is represented by the mean vector of all training vectors. The cohesion within a negative data set in formula (5) is calculated by averaging cosine similarity values between $\vec{C}_{2(\text{negative})}$ and each document of the negative data set, and the cohesion between positive and negative data sets in formula (6) is calculated by averaging cosine similarity values between $\vec{C}_{\text{glob}}$ and $\vec{C}_k$.

As shown in Table 7, we can observe the high cohesion within a negative data set and the low cohesion between positive and negative data sets in each category when using the proposed method. We can find

Table 7
The experimental results of the cohesion within a negative data set and cohesion between positive and negative data sets

Category	Cohesion within a negative data set		Cohesion between positive and negative data sets	
	Original method	Proposed method	Original method	Proposed method
Corn	0.04336	0.05079	0.01101	0.00867
Ship	0.05458	0.06113	0.01419	0.01188
Wheat	0.04469	0.05034	0.01087	0.00905
Interest	0.05117	0.05334	0.01180	0.01091
Trade	0.05156	0.06119	0.01421	0.01102
Crude	0.05745	0.06123	0.01422	0.01270
Grain	0.05797	0.06047	0.01399	0.01285
Money	0.05267	0.05545	0.01245	0.01133
Acq	0.06140	0.06395	0.01506	0.01383
Earn	0.06148	0.06407	0.01510	0.01385
Average	0.05363	*0.05820*	0.01329	*0.01161*

out that our proposed method reforms the vector space for a better performance: the high cohesion within a negative data set and the low cohesion between positive and negative data sets. Using the proposed method, the document vectors in a negative data set are located more closely and positive and negative data sets are separated more widely.

5. Conclusions

In this paper, we proposed a new method for binary data setting in binary text classification, which revised the original one-against-the-rest method using the sliding window technique and the revised EM algorithm.

The experimental results showed that the proposed method produced the significant improved performance in all four kinds of text classifiers and all three kinds of data sets. Especially, the decreasing performances in any category of all the data sets were not detected from our experiments. This result proves the effectiveness of the proposed method in binary text classification. In Section 4.3, we verify the proposed method in that it can be more useful method in real application areas and it can reform the document vector space for better performance in binary text categorization. As a result, the proposed method can provide much improvement when it is used in real binary text classification applications instead of the one-against-the-rest method.

Acknowledgement

This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD) (KRF-2006-331-D00536).

References

Brill, E. (1995). Transformation-based error-driven learning and natural language processing: a case study in part of speech tagging. *Computational Linguistics, 21*(4), 543–566.

Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., et al. (2000). Learning to construct knowledge bases from the world wide web. *Artificial Intelligence, 118*(1–2), 69–113.

Dempster, A., Laird, N. M., & Rubin, D. (1997). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B, 39*(1), 1–38.

Hsu, C. W., & Lin, C. J. (2002). A comparison of methods of multi-class support vector machines. *IEEE Transactions on Neural Networks, 13*, 415–425.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of European conference on machine learning (ECML)* (pp. 137–142). Springer.

Joachims, T. (2002). *Learning to classify text using support vector machines*. Kluwer Academic Publishers.

Ko, Y., Park, J., & Seo, J. (2004). Improving text categorization using the importance of sentences. *Information Processing and Management, 40*(1), 65–79.

Ko, Y., & Seo, J. (2004). Learning with unlabeled data for text categorization using a bootstrapping and a feature projection technique. In *Proceedings of the 42nd annual meeting of the association for computational linguistics (ACL 2004)*, pp. 255–262.

Ko, Y., & Seo, J. (2004). Using the feature projection technique based on a normalized voting method for text categorization. *Information Processing and Management, 40*(2), 191–208.

Lee, C.H., Lin, C.R., & Chen, M.S. (2001). Sliding-window filtering: an efficient algorithm for incremental mining. In *Proceedings of the tenth international conference on information and knowledge management*, pp. 263–270.

Li, X., & Liu, B. (2003). Learning to classify text using positive and unlabeled data. In *Proceedings of eighteenth international joint conference on artificial intelligence (IJCAI-03)*, pp. 587–594.

Liu, B., Lee, W.S., Yu, P.S., & Li, X. (2002). Partially Supervised Classification of Text Documents. In *Proceedings of the nineteenth international conference on machine learning (ICML, 2000)*, Sydney, Australia, pp. 8–12.

Lewis, D.D. (1998). Naive (bayes) at forty: The independence assumption in information retrieval. In *Proceedings of European conference on machine learning*.

McCallum, A., & Nigam, K. (1998). A comparison of event models for Naive Bayes text classification. *AAAI'98 workshop on learning for text categorization*, pp. 41–48.

Merialdo, B. (1994). Tagging english text with a probabilistic model. *Computational Linguisitcs, 20*(2), 155–171.

Mitchell, T. (1997). *Machine learning*. New York: McGraw-Hill.

Roy, N., & McCallum, A. (2001). Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of 18th international conference on machine learning*, pp. 441–448.

Salton, G., Yang, C., & Wang, A. (1975). A vector space model for automatic indexing. *Communications of the ACM, 18*(11), 613–620.

Schohn, G., & Cohn, D. (2000). Less is more: active learning with support vector machines. In *Proceedings of 17th international conference on machine learning*, pp. 839–846.

Tong, S., & Koller, D. (2001). Support vector machine active learning with application to text classification. *Journal of Machine Learning Research, 2*, 45–66.

Yang, Y. (1999). An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval, 1*(1/2), 67–88.

Yang, Y., Slattery, S., & Ghani, R. (2002). A study of approaches to hypertext categorization. *Journal of Intelligent Information Systems, 18*(2).

Yu, H., Han, J., & Chang, K. (2002). PEBL: positive example based learning for web page classification using SVM. In *Proceedings of international conference on knowledge discovery and data mining (KDD-02)*.

Zadrozny, B., & Elkan, C. (2001). Obtaining calibrated probability estimates from decision trees and naïve bayesian classifiers. In *Proceedings of the Eighteenth International Conference on Machine Learning*.

Zadrozny, B., & Elkan, C. (2002). Reducing multi-class to binary by coupling probability estimates. In *Proceedings of International Conference on Knowledge Discovery and Data Mining (KDD-02)*.