

Text Classification from Positive and Unlabeled Data using Misclassified Data Correction

Fumiyo Fukumoto and Yoshimi Suzuki and Suguru Matsuyoshi

Interdisciplinary Graduate School of Medicine and Engineering

University of Yamanashi, Kofu, 400-8511, JAPAN

{fukumoto, ysuzuki, sugurum}@yamanashi.ac.jp

Abstract

This paper addresses the problem of dealing with a collection of labeled training documents, especially annotating negative training documents and presents a method of text classification from positive and unlabeled data. We applied an error detection and correction technique to the results of positive and negative documents classified by the Support Vector Machines (SVM). The results using Reuters documents showed that the method was comparable to the current state-of-the-art biased-SVM method as the F-score obtained by our method was 0.627 and biased-SVM was 0.614.

1 Introduction

Text classification using machine learning (ML) techniques with a small number of labeled data has become more important with the rapid increase in volume of online documents. Quite a lot of learning techniques *e.g.*, semi-supervised learning, self-training, and active learning have been proposed. Blum *et al.* proposed a semi-supervised learning approach called the Graph Mincut algorithm which uses a small number of positive and negative examples and assigns values to unlabeled examples in a way that optimizes consistency in a nearest-neighbor sense (Blum *et al.*, 2001). Cabrera *et al.* described a method for self-training text categorization using the Web as the corpus (Cabrera *et al.*, 2009). The method extracts unlabeled documents automatically from the Web and applies an enriched self-training for constructing the classifier.

Several authors have attempted to improve classification accuracy using only positive and unlabeled data (Yu *et al.*, 2002; Ho *et al.*, 2011). Liu *et al.* proposed a method called biased-SVM that

uses soft-margin SVM as the underlying classifiers (Liu *et al.*, 2003). Elkan and Noto proposed a theoretically justified method (Elkan and Noto, 2008). They showed that under the assumption that the labeled documents are selected randomly from the positive documents, a classifier trained on positive and unlabeled documents predicts probabilities that differ by only a constant factor from the true conditional probabilities of being positive. They reported that the results were comparable to the current state-of-the-art biased SVM method. The methods of Liu *et al.* and Elkan *et al.* model a region containing most of the available positive data. However, these methods are sensitive to the parameter values, especially the small size of labeled data presents special difficulties in tuning the parameters to produce optimal results.

In this paper, we propose a method for eliminating the need for manually collecting training documents, especially annotating negative training documents based on supervised ML techniques. Our goal is to eliminate the need for manually collecting training documents, and hopefully achieve classification accuracy from positive and unlabeled data as high as that from labeled positive and labeled negative data. Like much previous work on semi-supervised ML, we apply SVM to the positive and unlabeled data, and add the classification results to the training data. The difference is that before adding the classification results, we applied the MisClassified data Detection and Correction (MCDC) technique to the results of SVM learning in order to improve classification accuracy obtained by the final classifiers.

2 Framework of the System

The MCDC method involves category error correction, *i.e.*, correction of misclassified candidates, while there are several strategies for automatically detecting lexical/syntactic errors in corpora (Abney *et al.*, 1999; Eskin, 2000; Dickinson and

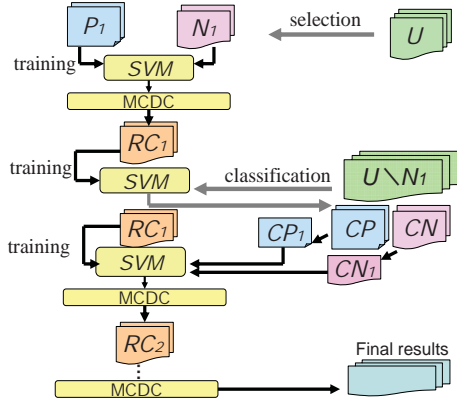


Figure 1: Overview of the system

Meurers., 2005; Boyd et al., 2008) or categorical data errors (Akoglu et al., 2013). The method first detects error candidates. As error candidates, we focus on support vectors (SVs) extracted from the training documents by SVM. Training by SVM is performed to find the optimal hyperplane consisting of SVs, and only the SVs affect the performance. Thus, if some training document reduces the overall performance of text classification because of an outlier, we can assume that the document is a SV.

Figure 1 illustrates our system. First, we randomly select documents from unlabeled data (U) where the number of documents is equal to that of the initial positive training documents (P_1). We set these selected documents to negative training documents (N_1), and apply SVM to learn classifiers. Next, we apply the MCDC technique to the results of SVM learning. For the result of correction (RC_1)¹, we train SVM classifiers, and classify the remaining unlabeled data ($U \setminus N_1$). For the result of classification, we randomly select positive (CP_1) and negative (CN_1) documents classified by SVM and add to the SVM training data (RC_1). We re-train SVM classifiers with the training documents, and apply the MCDC. The procedure is repeated until there are no unlabeled documents judged to be either positive or negative. Finally, the test data are classified using the final classifiers. In the following subsections, we present the MCDC procedure shown in Figure 2. It consists of three steps: extraction of misclassified candidates, estimation of error reduction, and correction of misclassified candidates.

¹The manually annotated positive examples are not corrected.

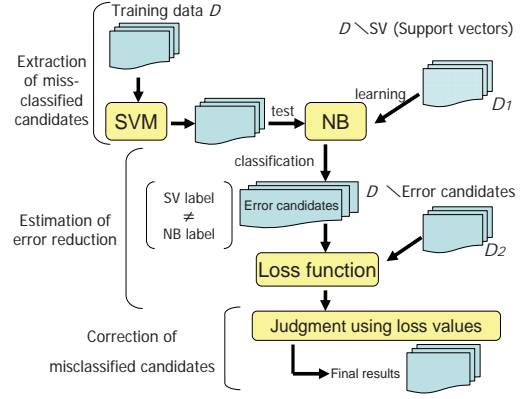


Figure 2: The MCDC procedure

2.1 Extraction of misclassified candidates

Let D be a set of training documents and $\mathbf{x}_k \in \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\}$ be a SV of negative or positive documents obtained by SVM. We remove $\cup_{k=1}^m \mathbf{x}_k$ from the training documents D . The resulting $D \setminus \cup_{k=1}^m \mathbf{x}_k$ is used for training Naive Bayes (NB) (McCallum, 2001), leading to a classification model. This classification model is tested on each \mathbf{x}_k , and assigns a positive or negative label. If the label is different from that assigned to \mathbf{x}_k , we declare \mathbf{x}_k an error candidate.

2.2 Estimation of error reduction

We detect misclassified data from the extracted candidates by estimating error reduction. The estimation of error reduction is often used in active learning. The earliest work is the method of Roy and McCallum (Roy and McCallum, 2001). They proposed a method that directly optimizes expected future error by log-loss or 0-1 loss, using the entropy of the posterior class distribution on a sample of unlabeled documents. We used their method to detect misclassified data. Specifically, we estimated future error rate by log-loss function. It uses the entropy of the posterior class distribution on a sample of the unlabeled documents. A loss function is defined by Eq (1).

$$E_{\hat{P}_{D_2 \cup (\mathbf{x}_k, y_k)}} = -\frac{1}{|X|} \sum_{x \in X} \sum_{y \in Y} P(y|x) \times \log(\hat{P}_{D_2 \cup (\mathbf{x}_k, y_k)}(y|x)). \quad (1)$$

Eq (1) denotes the expected error of the learner. $P(y | x)$ denotes the true distribution of output classes $y \in Y$ given inputs x . X denotes a

set of test documents. $\hat{P}_{D_2 \cup (\mathbf{x}_k, y_k)}(y | x)$ shows the learner’s prediction, and D_2 denotes the training documents D except for the error candidates $\cup_{k=1}^l \mathbf{x}_k$. If the value of Eq (1) is sufficiently small, the learner’s prediction is close to the true output distribution.

We used bagging to reduce variance of $P(y | x)$ as it is unknown for each test document x . More precisely, from the training documents D , a different training set consisting of positive and negative documents is created². The learner then creates a new classifier from the training documents. The procedure is repeated m times³, and the final class posterior for an instance is taken to be the unweighted average of the class posteriori for each of the classifiers.

2.3 Correction of misclassified candidates

For each error candidate \mathbf{x}_k , we calculated the expected error of the learner, $E_{\hat{P}_{D_2 \cup (\mathbf{x}_k, y_{k_old})}}$ and $E_{\hat{P}_{D_2 \cup (\mathbf{x}_k, y_{k_new})}}$ by using Eq (1). Here, y_{k_old} refers to the original label assigned to \mathbf{x}_k , and y_{k_new} is the resulting category label estimated by NB classifiers. If the value of the latter is smaller than that of the former, we declare the document x_k to be misclassified, *i.e.*, the label y_{k_old} is an error, and its true label is y_{k_new} . Otherwise, the label of \mathbf{x}_k is y_{k_old} .

3 Experiments

3.1 Experimental setup

We chose the 1996 Reuters data (Reuters, 2000) for evaluation. After eliminating unlabeled documents, we divided these into three. The data (20,000 documents) extracted from 20 Aug to 19 Sept is used as training data indicating positive and unlabeled documents. We set the range of δ from 0.1 to 0.9 to create a wide range of scenarios, where δ refers to the ratio of documents from the positive class first selected from a fold as the positive set. The rest of the positive and negative documents are used as unlabeled data. We used categories assigned to more than 100 documents in the training data as it is necessary to examine a wide range of δ values. These categories are 88 in all. The data from 20 Sept to 19 Nov is used

²We set the number of negative documents extracted randomly from the unlabeled documents to the same number of positive training documents.

³We set the number of m to 100 in the experiments.

as a test set X , to estimate true output distribution. The remaining data consisting 607,259 from 20 Nov 1996 to 19 Aug 1997 is used as a test data for text classification. We obtained a vocabulary of 320,935 unique words after eliminating words which occur only once, stemming by a part-of-speech tagger (Schmid, 1995), and stop word removal. The number of categories per documents is 3.21 on average. We used the SVM-Light package (Joachims, 1998)⁴. We used a linear kernel and set all parameters to their default values.

We compared our method, MCDC with three baselines: (1) SVM, (2) Positive Example-Based Learning (PEBL) proposed by (Yu et al., 2002), and (3) biased-SVM (Liu et al., 2003). We chose PEBL because the convergence procedure is very similar to our framework. Biased-SVM is the state-of-the-art SVM method, and often used for comparison (Elkan and Noto, 2008). To make comparisons fair, all methods were based on a linear kernel. We randomly selected 1,000 positive and 1,000 negative documents classified by SVM and added to the SVM training data in each iteration⁵. For biased-SVM, we used training data and classified test documents directly. We empirically selected values of two parameters, “ c ” (trade-off between training error and margin) and “ j ”, *i.e.*, cost (cost-factor, by which training errors on positive examples) that optimized the F-score obtained by classification of test documents.

The positive training data in SVM are assigned to the target category. The negative training data are the remaining data except for the documents that were assigned to the target category, *i.e.*, this is the ideal method as we used all the training data with positive/negative labeled documents. The number of positive training data in other three methods depends on the value of δ , and the rest of the positive and negative documents were used as unlabeled data.

3.2 Text classification

Classification results for 88 categories are shown in Figure 3. Figure 3 shows micro-averaged F-score against the δ value. As expected, the results obtained by SVM were the best among all δ values. However, this is the ideal method that requires 20,000 documents labeled positive/negative, while other methods including our

⁴<http://svmlight.joachims.org>

⁵We set the number of documents up to 1,000.

Level (# of Cat)		SVM		PEBL		Biased-SVM		MCDC	
		Cat	F	Cat	F (Iter)	Cat	F (Iter)	Cat	F (Iter)
Top (22)	Best	GSPO	.955	GSPO	.802 (26)	CCAT	.939	GSPO	.946 (9)
	Worst	GODD	.099	GODD	.079 (6)	GODD	.038	GODD	.104 (4)
	Avg		.800		.475 (19)		.593		.619 (8)
Second (32)	Best	M14	.870	E71	.848 (7)	M14	.869	M14	.875 (9)
	Worst	C16	.297	E14	.161 (14)	C16	.148	C16	.150 (3)
	Avg		.667		.383 (22)		.588		.593 (7)
Third (33)	Best	M141	.878	C174	.792 (27)	M141	.887	M141	.885 (8)
	Worst	G152	.102	C331	.179 (16)	G155	.130	C331	.142 (6)
	Avg		.717		.313 (18)		.518		.557 (8)
Fourth (1)	–	C1511	.738	C1511	.481 (16)	C1511	.737	C1511	.719 (4)
Micro Avg F-score			.718		.428 (19)		.614		.627 (8)

Table 1: Classification performance ($\delta = 0.7$)

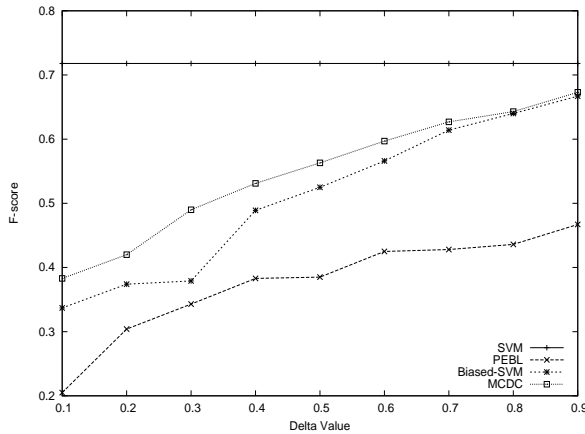


Figure 3: F-score against the value of δ

method used only positive and unlabeled documents. Overall performance obtained by MCDC was better for those obtained by PEBL and biased-SVM methods in all δ values, especially when the positive set was small, *e.g.*, $\delta = 0.3$, the improvement of MCDC over biased-SVM and PEBL was significant.

Table 1 shows the results obtained by each method with a δ value of 0.7. “Level” indicates each level of the hierarchy and the numbers in parentheses refer to the number of categories. “Best” and “Worst” refer to the best and the lowest F-scores in each level of a hierarchy, respectively. “Iter” in PEBL indicates the number of iterations until the number of negative documents is zero in the convergence procedure. Similarly, “Iter” in the MCDC indicates the number of iterations until no unlabeled documents are judged to be either positive or negative. As can be seen clearly from Table 1, the results with MCDC were better than those obtained by PEBL in each level of the hierarchy. Similarly, the results were bet-

δ	SV	Ec	Err	Correct		
				Prec	Rec	F
0.3	227,547	54,943	79,329	.693	.649	.670
0.7	141,087	34,944	42,385	.712	.673	.692

Table 2: Miss-classified data correction results

ter than those of biased-SVM except for the fourth level, “C1511”(Annual results). The average numbers of iterations with MCDC and PEBL were 8 and 19 times, respectively. In biased-SVM, it is necessary to run SVM many times, as we searched “*c*” and “*j*”. In contrast, MCDC does not require such parameter tuning.

3.3 Correction of misclassified candidates

Our goal is to achieve classification accuracy from only positive documents and unlabeled data as high as that from labeled positive and negative data. We thus applied a miss-classified data detection and correction technique for the classification results obtained by SVM. Therefore, it is important to examine the accuracy of miss-classified correction. Table 2 shows detection and correction performance against all categories. “SV” shows the total number of SVs in 88 categories in all iterations. “Ec” refers to the total number of extracted error candidates. “Err” denotes the number of documents classified incorrectly by SVM and added to the training data, *i.e.*, the number of documents that should be assigned correctly by the correction procedure. “Prec” and “Rec” show the precision and recall of correction, respectively.

Table 2 shows that precision was better than recall with both δ values, as the precision obtained by γ value = 0.3 and 0.7 were 4.4% and 3.9% improvement against recall values, respectively. These observations indicated that the error candidates extracted by our method were appropriately

corrected. In contrast, there were still other documents that were miss-classified but not extracted as error candidates. We extracted error candidates using the results of SVM and NB classifiers. Ensemble of other techniques such as boosting and kNN for further efficacy gains seems promising to try with our method.

4 Conclusion

The research described in this paper involved text classification using positive and unlabeled data. Miss-classified data detection and correction technique was incorporated in the existing classification technique. The results using the 1996 Reuters corpora showed that the method was comparable to the current state-of-the-art biased-SVM method as the F-score obtained by our method was 0.627 and biased-SVM was 0.614. Future work will include feature reduction and investigation of other classification algorithms to obtain further advantages in efficiency and efficacy in manipulating real-world large corpora.

References

- S. Abney, R. E. Schapire, and Y. Singer. 1999. Boosting Applied to Tagging and PP Attachment. In *Proc. of the Joint SIGDAT Conference on EMNLP and Very Large Corpora*, pages 38–45.
- L. Akoglu, H. Tong, J. Vreeken, and C. Faloutsos. 2013. Fast and Reliable Anomaly Detection in Categorical Data. In *Proc. of the CIKM*, pages 415–424.
- A. Blum, J. Lafferty, M. Rwebangira, and R. Reddy. 2001. Learning from Labeled and Unlabeled Data using Graph Mincuts. In *Proc. of the 18th ICML*, pages 19–26.
- A. Boyd, M. Dickinson, and D. Meurers. 2008. On Detecting Errors in Dependency Treebanks. *Research on Language and Computation*, 6(2):113–137.
- R. G. Cabrera, M. M. Gomez, P. Rosso, and L. V. Pineda. 2009. Using the Web as Corpus for Self-Training Text Categorization. *Information Retrieval*, 12(3):400–415.
- M. Dickinson and W. D. Meurers. 2005. Detecting Errors in Discontinuous Structural Annotation. In *Proc. of the ACL'05*, pages 322–329.
- C. Elkan and K. Noto. 2008. Learning Classifiers from Only Positive and Unlabeled Data. In *Proc. of the KDD'08*, pages 213–220.
- E. Eskin. 2000. Detecting Errors within a Corpus using Anomaly Detection. In *Proc. of the 6th ANLP Conference and the 1st Meeting of the NAACL*, pages 148–153.
- C. H. Ho, M. H. Tsai, and C. J. Lin. 2011. Active Learning and Experimental Design with SVMs. In *Proc. of the JMLR Workshop on Active Learning and Experimental Design*, pages 71–84.
- T. Joachims. 1998. SVM Light Support Vector Machine. In *Dept. of Computer Science Cornell University*.
- B. Liu, Y. Dai, X. Li, W. S. Lee, and P. S. Yu. 2003. Building Text Classifiers using Positive and Unlabeled Examples. In *Proc. of the ICDM'03*, pages 179–188.
- A. K. McCallum. 2001. Multi-label Text Classification with a Mixture Model Trained by EM. In *Revised Version of Paper Appearing in AAAI'99 Workshop on Text Learning*, pages 135–168.
- Reuters. 2000. *Reuters Corpus Volume1 English Language*. 1996-08-20 to 1997-08-19 Release Date 2000-11-03 Format Version 1.
- N. Roy and A. K. McCallum. 2001. Toward Optimal Active Learning through Sampling Estimation of Error Reduction. In *Proc. of the 18th ICML*, pages 441–448.
- H. Schmid. 1995. Improvements in Part-of-Speech Tagging with an Application to German. In *Proc. of the EACL SIGDAT Workshop*, pages 47–50.
- H. Yu, H. Han, and K. C-C. Chang. 2002. PEBL: Positive Example based Learning for Web Page Classification using SVM. In *Proc. of the ACM Special Interest Group on Knowledge Discovery and Data Mining*, pages 239–248.