Text Categorization Using Transductive Boosting

Hirotoshi Taira[1] and Masahiko Haruno[2]

[1] NTT Communication Science Laboratories
2-4, Hikaridai, Seika-cho, Soraku-gun
Kyoto 619-0237, Japan
taira@cslab.kecl.ntt.co.jp

[2] Advanced Telecommunications Research Institute International
2-2, Hikaridai, Seika-cho, Soraku-gun
Kyoto 619-0288, Japan
mharuno@isd.atr.co.jp

Abstract. In natural language tasks like text categorization, we usually have an enormous amount of unlabeled data in addition to a small amount of labeled data. We present here a transductive boosting method for text categorization in order to make use of the large amount of unlabeled data efficiently. Our experiments show that the transductive method outperforms conventional boosting techniques that employ only labeled data.

1 Introduction

We can easily access vast quantities of online information with the rapid growth of the Internet and the increase in computing power. However, the more information we can access, the more difficult it becomes to get necessary and sufficient information. Automatic *text categorization* has attracted a lot of attention among researchers and companies as a measure to extract the necessary and sufficient information efficiently.

The approach of machine learning has recently become popular for text categorization since it can create text classifiers with a high accuracy and without difficulty even if the target text is large and updated frequently. In this approach, inductive methods have played a central role in discriminating unlabeled test data with classifiers constructed from a priori labeled training data. For instance, k-nearest-neighbor [17], Rocchio [12], decision trees [7], Naive-Bayes [7], and SVM [5,1,15] have been applied to text categorization and have achieved remarkable success.

However, the inductive approach can not guarantee a high enough accuracy when there is a great difference between the training and test data distributions. The problem becomes extremely serious if the amount of training data is small. This is often the case under many practical conditions such as the classification of online Internet texts. Therefore, it is reasonable to utilize the unlabeled test data distribution for training as well as the distribution of a small number of labeled training data. Nigam et al. proposed an EM-based method with Naive-Bayes to take account of the distribution of test data under the situation of a small amount of training data [11]. Although the method shows substantial improvements over the performance of the standard Naive Bayes classifier, one of its limitations is the nature that it sometimes obtains a local optimum.

In contrast, Joachims adapted a transductive method to Support Vector Machines (SVM) [6] and obtained significant improvements in the classification performance. Transduction is a general learning framework that minimizes classification errors only for the test data, while induction tries to minimize classification errors for both the training and test data [16]. Transductive SVM (TSVM) achieves a high performance by assuming that the portion of unlabeled examples to be classified into the positive class is determined by the ratio of positive and negative examples in the training data. Accordingly, the possibility of a performance decrease remains under different but typical conditions when the ratio of positive and negative examples in the training data is very different from that in the test data, e.g., a classifier learned by articles in 1995, classifies articles related to the Internet in 2000.

Like SVM, AdaBoost [3,2] is an alternative large margin classifier recently noted for its high generalization ability in NLP applications [4,13]. It produces highly accurate classification rules by combining a number of weak hypotheses, each of which is only moderately accurate. The advantage of AdaBoost over SVM is that we can choose any classifier suitable for our own classification applications. Since the original AdaBoost is an inductive learning method, we propose here a novel transductive boosting algorithm to cope with a small number of training data; in particular, the condition where the ratio of positive and negative examples greatly differs between the training and test data. Our experimental results demonstrate that the proposed algorithm not only outperforms SVM and AdaBoost but is also comparative and sometimes superior to TSVM. The advantage of the method is significant when the number of training data is small and the ratio of positive examples to negative ones in the training data is different from that in the test data. These results confirm that the usefulness of the transductive approach is not limited to SVM but is also effective for a variety of learning methods.

The remainder of the paper is organized as follows. The next section introduces AdaBoost and its global error analysis. After briefly describing how boosting can be regarded as a gradient descent method in a function space, we show how a transductive method can be adapted to AdaBoost. We then report our experimental results on a text categorization task using Japanese newspaper articles and discuss these results. The last section concludes the paper.

2 Boosting

2.1 AdaBoost Algorithm

Boosting became popular for practical use after Freund and Schapire proposed the AdaBoost algorithm [2]. We briefly introduce AdaBoost here.

(step 1) Given m training samples $(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_m, y_m)$ with feature vectors $\boldsymbol{x}_1, ..., \boldsymbol{x}_m$ and classification classes $y_1, ..., y_m$ ($+1$ for positive, -1 for negative).
Initialize the weights for training data $D_1(i) = \frac{1}{m}$, where $i = 1, ..., m$
(step 2) For $t = 1, ..., T$, repeat (step 3)-(step 5).
(step 3) A weak learner learns the training data under weight D_t, and we get weak hypothesis $h_t(\boldsymbol{x})$, which outputs $+1$ for a positive evaluation for $\boldsymbol{x} = \boldsymbol{x}_i$; -1 for a negative evaluation.
(step 4) Calculate parameter α_t based on $\alpha_t = \frac{1}{2}\ln(\frac{1-\epsilon_t}{\epsilon_t})$, where ϵ_t denotes weighted error rates calculated based on $\epsilon_t = \sum_{i:h_t(\boldsymbol{x}_i)\neq y_i} D_t(i)$.
(step 5) Update every weight of the training data based on the following,

$$D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t y_i h_t(\boldsymbol{x}_i))}{Z_t}$$

where Z_t, the normalized factor for $\sum_{i=1}^{m} D_{t+1}$, equals 1.
(step 6) Return the final hypothesis, merging weak hypotheses linearly as,

$$H(\boldsymbol{x}) = \sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x}) \quad .$$

AdaBoost produces a weak hypothesis for a round and updates weights for the training data T times ($t = 1, ..., T$). As we can see based on (step 5), weight $D_t(i)$ is multiplied by $\exp(-\alpha_t)$ when data i is classified correctly by the weak hypothesis (that is, when $h_t(\boldsymbol{x}_i) = y_i$); it is multiplied by $\exp(\alpha_t)$ when data i is classified incorrectly (that is, when $h_t(\boldsymbol{x}_i) \neq y_i$). When error rates ϵ_t are less than 50%, parameter α_t takes a positive value based on (step 4). The weights of the data learned incorrectly are multiplied by a number larger than one and the learners learn weak hypotheses to focus on their data in the next round. Finally, in (step 6), the learners combine all of the weak learners weighted by parameter α_t, and the final classifier $H(\boldsymbol{x})$ is obtained.

Schapire et al. introduced the idea of "margin" and analyzed global errors of AdaBoost (that is, classification errors for unlabeled test data) [14]. The margin for training data in boosting is determined as $y\sum_t \alpha_t h_t(\boldsymbol{x})/\sum_t \alpha_t$. When α_t is normalized as $\sum_t \alpha_t = 1$, the margin equals $yH(\boldsymbol{x})$. If we can take larger margins, the global errors become smaller [14].

3 Transductive Methods and Text Categorization

3.1 The Transductive Method Used in TSVM

We illustrate the transductive SVM (TSVM) and the novel transductive boosting in Figure 1. The circles, crosses, and triangles denote positive training data, negative training data, and unlabeled test data, respectively. TSVM produces separated hyperplanes by finding the positive examples closest to the negative side and the negative examples closest to the positive side. The "margin" of

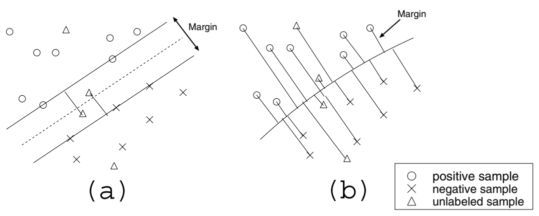

Fig. 1. TSVM (a) and Transductive Boosting (b).

TSVM is defined as the distance between two separated hyperplanes. TSVM chooses separated hyperplanes such that they maximize the margins while allowing classification errors below some fixed rate.

TSVM first produces a classifier using only the training data by SVM. All of the test data are given temporary classes by the classifier. The classifiers, after that, are iteratively constructed by focusing only on the temporary labeled data. As the figure shows, if they can find a pair of positive and negative examples near the classification boundary such that an exchange of their temporary classes decreases the classification errors, they exchange their classes and re-learn the classifier by SVM. They repeat the exchange of classes and re-learning until there is no pair of test data for which labels have to be exchanged, and they finally obtain a hyperplane fitting the distribution of the test data.

On the other hand, the margin in boosting is the sum of the distances between every training data and optimal classification bound, such as the right side of Figure 1. Boosting tries to maximize the average of all margins. Our transductive method labels the one most reliable example at every round as described in the following sections.

3.2 Explanation of Boosting Using Gradient Descent Methods

Recently, it has become clear that boosting can be regarded as an algorithm that chooses a weak hypothesis in the direction of the gradient descent of cost functions in a function space [9].

Mason et al. stated that the AdaBoost algorithm corresponds to an algorithm that minimizes a cost function of the $\exp(-M)$ type in MarginBoost, where M is the margin. The cost function of the AdaBoost algorithm is

$$Cost(H(\boldsymbol{x})) = \frac{1}{m}\sum_{i=1}^{m} \exp(-y_i H(\boldsymbol{x}_i)).$$

The cost is taken as an average of margins for a power function measure. As a result, a larger cost is evaluated for classification errors for example. Minimizing the value of this cost function corresponds to maximizing the margins and also corresponds to minimizing the global errors as mentioned in the previous section.

Let us consider a class $\mathcal{H}$ of weak classifiers $h : X \to \{+1, -1\}$ (where X is the space of feature vectors). $lin(\mathcal{H})$ is the set of all linear combinations of the functions in $\mathcal{H}$. The inner product is defined by $< F, G > \overset{\text{def}}{=} \frac{1}{m}\sum_{i=1}^{m} F(\boldsymbol{x}_i)G(\boldsymbol{x}_i)$ for all $F, G \in lin(\mathcal{H})$. We define the inner product space $(\mathcal{X}, <, >)$ using the inner product, where $\mathcal{X}$ is a linear space of functions that contains $lin(\mathcal{H})$, and $<, >$ stands for the inner product. Now suppose we have a function $H \in lin(\mathcal{H})$ and we wish to find a new $h \in lin(\mathcal{H})$ to add to H so that $Cost(H + \epsilon h)$ decreases for some small value of ϵ. We define the functional derivative of the cost function of H as

$$\nabla Cost(H)(\boldsymbol{x}) \overset{\text{def}}{=} \left.\frac{\partial Cost(H + \alpha 1_x)}{\partial \alpha}\right|_{\alpha=0}$$

where $1_{\boldsymbol{x}}$ is the indicator function of $\boldsymbol{x}$. The cost function can be expanded to the first order in ϵ,

$$Cost(H + \epsilon h) = Cost(H) + \epsilon < \nabla Cost(H), h > .$$

Here, we can use the gradient descent method. That is, the greatest reduction in cost will occur for the h maximizing $- < \nabla Cost(H), h >$. After all, we wish to find h_{t+1}, α_{t+1} to minimize

$$\sum_{i=1}^{m} Cost(y_i H_t(\boldsymbol{x}_i) + y_i \alpha_{t+1} h_{t+1}(\boldsymbol{x}_i))$$

in the function space to get the weak hypothesis h_{t+1} at round t.

3.3 The Transductive Boosting Method

Let us consider the minimization of the cost function mentioned in the former section in the framework of transductive methods. The cost function, including n test examples $\boldsymbol{x}_{m+1}, ..., \boldsymbol{x}_{m+n}$, is described as

$$Cost(H(\boldsymbol{x})) = \frac{1}{m+n}\{\sum_{i=1}^{m} \exp(-y_i H(\boldsymbol{x}_i)) + \sum_{j=m+1}^{m+n} \exp(-y_j^* H(\boldsymbol{x}_j))\}$$

where y_j^* is a temporary class label for $\boldsymbol{x}_j$. y_j^* is unknown, and the initial value of y_j^* is stored with 0. This algorithm aims to label +1(positive) or −1(negative) correctly for y_j^*. In the early rounds, the accuracy of the classifiers combining linearly weak learners is low because the learning is not sufficient unlike in SVM. If we label the classes for y_j^* based on these classifiers, incorrect labels are labeled to the data at a high ratio. Then, if boosting is performed with this large amount of wrongly labeled test data, an incorrect gradient descent is obtained, and moreover, the accuracy of the final classifier is low. We should perform labeling for the test data at a high accuracy.

Therefore, in every round, we label the class for only the most reliable test data. We label this class supposing that the ratio of positive and negative examples is the same as the ratio of positive and negative examples in the training data. For the test data, we add (step 2) and (step 7) to the AdaBoost algorithm in section 2.1 as follows.

(step 1) Given m training samples $(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_m, y_m)$ with feature vectors $\boldsymbol{x}_1, ..., \boldsymbol{x}_m$ and classification classes $y_1, ..., y_m$ ($+1$ for positive, -1 for negative).
Initialize the weights for training data $D_1(i) = \frac{1}{m}$, where $i = 1, ..., m$.

(step 2) Given n test samples $(\boldsymbol{x}_{m+1}, y^*_{m+1}), ..., (\boldsymbol{x}_{m+n}, y^*_{m+n})$ with feature vectors $\boldsymbol{x}_{m+1}, ..., \boldsymbol{x}_{m+n}$ and classification classes $y^*_{m+1}, ..., y^*_{m+n}$ whose initial values are 0.
Initialize the weights for test data $D_1(j) = 0 \quad (j = m+1, ..., m+n)$.

(step 3) For $t = 1, ..., T$, repeat (step 4)-(step 7)

(step 4) A weak learner learns labeled data (that is, $y_i \neq 0$) under weight D_t, and we get weak hypothesis $h_t(\boldsymbol{x})$, which outputs $+1$ for a positive evaluation for $\boldsymbol{x} = \boldsymbol{x}_i$; -1 for a negative evaluation.

(step 5) Calculate parameter α_t based on $\alpha_t = \frac{1}{2}\ln(\frac{1-\epsilon_t}{\epsilon_t})$, where ϵ_t denotes weighted error rates calculated based on $\epsilon_t = \sum_{i:h_t(\boldsymbol{x}_i)\neq y_i} D_t(i)$.

(step 6) Update every weight of the training data based on the following,

$$D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t y_i h_t(\boldsymbol{x}_i))}{Z_t}$$

where Z_t, the normalized factor for $\sum_{i=1}^{m} D_{t+1}$, equals 1.

(step 7) Let m^+ be the number of training data with a positive class, $n_{labeled}$ be the number of test data with an already labeled class, and $n^+_{labeled}$ be the number of test data with an already labeled positive class.

(i) If $n_{labeled} = 0$ or $m^+/m \geq n^+_{labeled}/n_{labeled}$, then
we choose the test sample j that maximizes

$$H(\boldsymbol{x}_j) = \sum_{k=1}^{t} \alpha_k h_k(\boldsymbol{x}_j)$$

in the test data such that $y_j = 0$, and give $y_j = +1$ and $D_{t+1}(j) = \epsilon$ (ϵ is a small value, for example, $\epsilon = 0.01$). Then, we update the weight of the data already labeled as follows,

$$D_{t+1}(i) = \frac{D_t(i)}{Z'_t}.$$

Here, Z'_t is the normalizing factor such that the sum of the data without j equals $1 - \epsilon$.

(ii) If $n_{labeled} \neq 0$ and $m^+/m < n^+_{labeled}/n_{labeled}$, then we choose the test sample j that minimizes

$$H(\boldsymbol{x}_j) = \sum_{k=1}^{t} \alpha_k h_k(\boldsymbol{x}_j)$$

in the test data such that $y_j = 0$, and give $y_j = -1$ and $D_{t+1}(j) = \epsilon$. Then, we update the weight of the data already labeled as follows,

$$D_{t+1}(i) = \frac{D_t(i)}{Z'_t}.$$

Here, Z'_t is the normalizing factor such that the sum of the data without j equals $1 - \epsilon$.

(step 8) Return the final hypothesis, merging weak hypotheses linearly as,

$$H(\boldsymbol{x}) = \sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x}) \quad .$$

(step 2) is for initializing the labels and weights of the test samples. (step 7) is for performing labeling such that the ratio of positive and negative test samples always equals that of the training data. A small value is given for the weight to test samples selected at this step, because the reliability of the labels is lower than that of the training samples.

Taking these steps can produce classifiers minimizing the value of the cost function because we can select data to lower the probability of labeling y_j^* wrongly and can choose data to maximize $y_j^* H(\boldsymbol{x}_j)$ at a round in the hill-climbing way with the term of $\exp(-y_j^* H(\boldsymbol{x}_j))$ in the cost function (Figure 2).

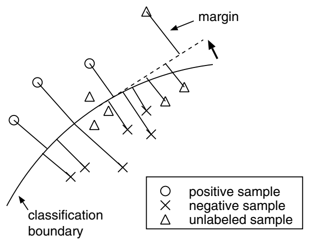

Fig. 2. The effect of unlabeled samples.

As another option of our algorithm, we can use the algorithm after weak learners are produced using only labeled data several times. We can also choose to label several test examples per round by executing (step 7) several times per round, although the algorithm would need more iterations than the number of test examples.

4 Experimental Results

4.1 Experimental Settings

Our experiments were conducted using the RWCP corpus, which contains 30,207 newspaper articles taken from the Mainichi Shinbun Newspaper published in 1994 [8]. Each article was assigned multiple UDC (Universal Decimal Classification) codes, each of which represented a category of the articles. UDC is a hierarchical classification system and has about 60,000 main categories. The text collection has a total of 97,095 categories among which there are 14,407 different categories, i.e., 3.2 categories per article.

In the remainder of this paper, we focus on the ten categories that appeared most often in the corpus: [1] sports, criminal law, government, education, traffic, military affairs, international relations, communications, theater, and agriculture. We made binary classifiers for each of the categories whether a sample belonged to the category or not. The total number of articles used for both the training and the test was 1,000. Table 1 summarizes the numbers of training and test articles in each category. These articles were word-segmented and part

Table 1. RWCP corpus for training and test.

Category	training articles	test articles
sports	161	147
criminal law	156	148
government	135	142
educational system	110	124
traffic	112	103
military affairs	110	118
international relations	96	97
communications	76	83
theater	86	95
agriculture	78	72

of speech tagged by the Japanese morphological analyzing system Chasen [10]. This process generated 20,490 different words. Throughout our experiments, 1000 words with high mutual information were used as the input feature space because they were keywords sufficient enough to characterize the classes.

The mutual infomation (MI) between a word t and a category c is defined as follows,

$$MI(t,c) = \sum_{t \in \{0,1\}} \sum_{c} P(t,c) \log \frac{P(t,c)}{P(t)P(c)}.$$

MI becomes large when the occurrence of t is biased to one side between category c and other categories. Consequently, the words with high mutual information in category c can be considered as keywords in the category.

The iteration T was 1000 in all of our experiments. The value of 0.01 was used for ϵ in transductive AdaBoost. The value for each feature was a binary value, which indicated whether the word appeared in a document or not. This binary value was employed to study the pure effects of each word.

4.2 The Evaluation Method

The F-measure was used for the evaluation measure. For every classification, we can calculate

[1] The results for other categories were very similar to these 10 categories.

$a =$ (the number of data the classifier evaluates positive for positive data),
$b =$ (the number of data the classifier evaluates positive for negative data),
$c =$ (the number of data the classifier evaluates negative for positive data).
Then, we can calculate the precision (P) and recall (R) as

$$P = \frac{a}{a+b}, \quad R = \frac{a}{a+c}.$$

By combining the precision and recall, the F-measure is defined as follows:

$$F = \frac{1+\beta^2}{\frac{1}{P} + \beta^2 \frac{1}{R}}.$$

The F-measure varies between 0 and 1. The larger the F-measure becomes, the higher the classification accuracy gets. β is a weight parameter and we set $\beta = 1$.

4.3 Relation between the Number of Training Data and the Accuracy

First, when the ratio of positive to negative examples in the training data was the same as that in the test data, we carried out experiments on text classification using the transductive AdaBoost algorithm with a one-depth decision tree as a weak learner. For comparison, we also performed experiments using the standard AdaBoost with a one-depth decision tree (BoosTexter) [13], SVM, and TSVM. We increased the number of training data from 75 to 1000, and we classified the 1000 test data. Figure 3 shows the results (F-measure) of the average among ten categories.

The accuracy increases significantly using the transductive method considering the distribution of 1000 test data. In particular, when there is a small number of training data, the growth of the classification accuracy becomes dramatically large. When the number of training data is 75, the F-measure for boosting is 0.438 and that for transductive boosting is 0.569; the difference is 0.131. The accuracy of the classification using only 75 training data in the transductive method almost equals the accuracy of that using 200 training data in the inductive method. This indicates that the transductive method is useful for improving the classification accuracy for a small number of training examples. Compared with SVM and TSVM with a linear kernel function, the classification using transductive boosting is slightly weaker than TSVM but exceeds SVM. The increase in the accuracy from the boosting to the transductive boosting decreases, while the classification accuracy increases monotonically without 1000 training data in the boosting. This might be because when the number of training data is 1000, the distribution of the test data is similar to that of the training data.

We show the details of the results for every category in Table 2 and Table 3. The bold face numbers denote the best accuracies for every category. The categories with high classification accuracies sometimes show decreases using the transductive method, e.g., 0.750 to 0.723 for criminal law using 1000 training data. However, the categories of education system, military affairs, international relations, and communications, which have low classification accuracies using the inductive method, show dramatically increases using the transductive method.

Table 2. F-measure for the number of training data (Transductive Boosting).

Category \ # of training data	75	100	200	500	750	1000
sports	0.642	0.726	0.766	0.875	0.901	**0.903**
criminal law	0.600	0.571	0.663	0.656	0.743	**0.750**
government	0.723	0.560	0.622	0.689	**0.727**	0.722
educational system	0.459	0.624	0.661	0.675	0.762	**0.778**
traffic	0.495	0.493	0.500	0.638	0.680	**0.698**
military affairs	0.507	0.561	0.688	0.748	0.754	**0.781**
international relations	0.429	0.396	0.363	0.558	0.508	**0.560**
communications	0.493	0.523	0.641	0.612	**0.703**	0.692
theater	0.583	0.749	0.756	0.795	0.857	**0.862**
agriculture	0.750	0.817	0.831	0.805	0.761	**0.853**
avg.	0.569	0.602	0.649	0.705	0.740	**0.760**

Table 3. F-measure for the number of training data (Boosting).

Category \ # of training data	75	100	200	500	750	1000
sports	0.675	0.681	0.826	0.867	0.891	**0.912**
criminal law	0.561	0.402	0.649	0.664	0.681	**0.723**
government	0.607	0.524	0.580	0.683	**0.692**	0.670
educational system	0.287	0.525	0.563	0.646	0.667	**0.714**
traffic	0.514	0.510	0.493	0.647	**0.658**	0.579
military affairs	0.216	0.321	0.550	**0.728**	0.686	0.628
international relations	0.324	0.317	0.233	0.428	**0.490**	0.329
communications	0.119	0.220	0.528	**0.576**	0.561	0.559
theater	0.385	0.645	0.767	0.693	0.800	**0.813**
agriculture	0.690	0.643	0.734	0.855	**0.864**	0.850
avg.	0.438	0.479	0.592	0.679	**0.699**	0.678

Let us move on to the second experimental condition. We changed the ratio of positive to negative examples in the training data from 1:1 to 1:4 (the original data distribution depicted in Figure 3 is 1:9). The total number of training examples was changed from 20 to 150. The results (the averages of ten categories) are shown in Figure 4. The performance of the transductive boosting is almost the same as that of TSVM and sometimes outperforms TSVM when the training and test distributions are significantly distinct, for example, $N_p : N_n = 1 : 1$ and 150 training samples. This indicates that the good performance of TSVM is largely dependent on the ratio of positive and negative examples.

5 Conclusion

We have proposed a transductive boosting method for text classification problems. We carried out experiments in which we varied the number of training

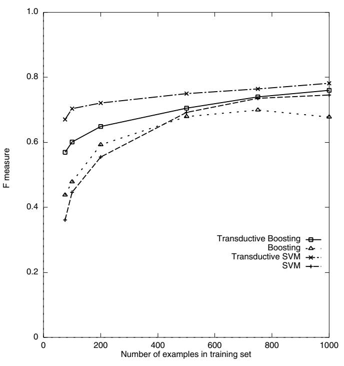

Fig. 3. F-measure and the number of training data.

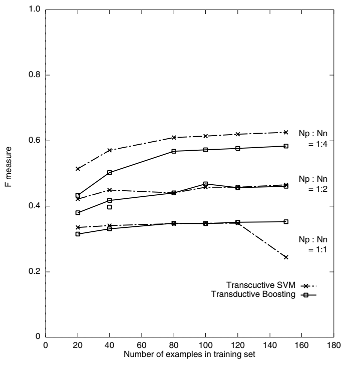

Fig. 4. F-measure and the ratio of positive and negative examples in the training data.

data, and compared the transductive method to the standard AdaBoost, SVM, and TSVM. The results indicated that the transductive boosting method can improve the performance of text categorization in situations where we have an enormous number of unlabeled data in addition to a small number of labeled training data. When the ratio of positive to negative examples in the training data differs from that in the test data, the advantage of our transductive boosting method in terms of performance is significant. This suggests that our transductive boosting method might be appropriate particularly when we do not know the ratio of positive and negative examples in the test data. Overall, our results show that the transductive approach is effective for a variety of learning methods and potentially promising for other applications.

References

1. S. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In *Proc. of 7th International Conference on Information and Knowledge Management*, 1998.
2. Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.
3. Y. Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2):256–285, 1995.
4. M. Haruno, S. Shirai, and Y. Ooyama. Using decision trees to construct a practical parser. *Machine Learning*, 34:131–149, 1999.
5. T. Joachims. Text categorization with support vector machines. In *Proc. of European Conference on Machine Learning(ECML)*, 1998.
6. T. Joachims. Transductive inference for text classification using support vector machines. In *Proc. of the 16th International Conference on Machine Learning (ICML'99)*, 1999.

7. D.D. Lewis and M. Ringuette. A comparison of two learning algorithms for text categorization. In *Proc. of Third Annual Symposium on Document Analysis and Information Retrieval*, pages 81–93, 1994.
8. Mainichi. *CD Mainichi Shinbun 94*. Nichigai Associates Co., 1995.
9. L. Mason, J. Baxter, P. Bartlett, and M. Frean. Boosting algorithms as gradient descent. In *Proc. of Neural Information Processing Systems 1999 (NIPS-99)*, 1999.
10. Y. Matsumoto, A Kitauchi, T. Yamashita, Y. Hirano, O. Imaichi, and T. Imamura. *Japanese Morphological Analysis System Chasen Manual*, 1997. NAIST Technical Report NAIST-IS-TR97007.
11. K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39:103–134, 2000.
12. G. Salton (Ed.). *The Smart Retrieval System-experiments in Automatic Document Processing*. Prentice-Hall, 1971.
13. R. E. Schapire and Y. Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39:135–168, 2000.
14. R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):1651–1686, 1998.
15. H. Taira and M. Haruno. Feature selection in SVM text categorization. In *Proc. of the 16th National Conference on Artificial Intelligence (AAAI-99)*, pages 480–486, 1999.
16. V. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, 1998.
17. Y. Yang. Expert network: Effective and efficient learning from human decisions in text categorization and retrieval. In *Proc. of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 13–22, 1994.