Document Representation for One-Class SVM

Xiaoyun Wu, Rohini Srihari, and Zhaohui Zheng

University at Buffalo, Buffalo NY 14260, USA

Abstract. Previous studies have shown that one-class SVM is a rather weak learning method for text categorization problems. This paper points out that the poor performance observed before is largely due to the fact that the standard term weighting schemes are inadequate for one-class SVMs. We propose several representation modifications, and demonstrate empirically that, with the proposed document representation, the performance of one-class SVM, although trained on only small portion of positive examples, can reach up to 95% of that of two-class SVM trained on the whole labeled dataset.

1 Introduction

Like most multi-labeled classification problems, text categorization problems are usually converted to binary classification problems in "one-versus-rest" fashion, where examples that belong to the category of interest are labeled as positive, and the others as negative. In a number of recent empirical studies [3, 15], Support Vector Machines (SVMs) have been shown to be among the most effective methods for such binary text categorization problems. Applying one-class SVM [11] on text categorization, which uses only the positive examples in the training phase, is worthy exploring for the following three reasons:

First, since negative examples are from many different categories, they are generally not as representative. It is logical to hypothesize that the resulting binary problems can be characterized mostly by their positive examples. And ideally, the classifier learned from positive examples should perform reasonably close to the classifier learned from fully labeled dataset. To the best of our knowledge, there is no empirical evidence for such conjecture. Applying one-class SVM on text categorization is one way of testing such hypothesis.

Second, the problem of learning with positive examples commonly arises in many real-world applications, particularly in information retrieval domain. For example, to learn a user's preference, pages in his bookmarks are readily available as positive examples, but it will be difficult to come up with enough representative negative examples. Effective learning methods that rely on only positive examples thus are of great practical interest.

Third, one characteristic for these binary problems is the skewness in the dataset, since frequently there are only a small number of positive examples but a very large number of negative examples. The time complexity of typical SVM training methods like sequential minimal optimization (SMO) is super-linear in the number of examples in the dataset [9]. The training of one-class SVMs

J.-F. Boulicaut et al. (Eds.): ECML 2004, LNAI 3201, pp. 489–500, 2004.

should be much more efficient than that of two-class SVMs since a large number of negative examples are ignored. In the case where there is a stringent constrain on the training time, highly effective one-class SVM can serve as an alternative to its two-class courter-part.

A previous study [7] showed empirically that performance of one-class SVMs is nowhere near that of two-class SVMs. In this paper, we reveal that standard document representation is inappropriate for one-class SVMs. We propose three modifications of document representation, including removing negative features, scaling dimensions and length normalization. We further demonstrate that the category statistics needed to tailor the document representation for one-class SVMs can be reliably estimated from both the fully labeled dataset and also the datasets with only positive and unlabeled examples. Experiments show that one-class SVM with the proposed representation modifications is effective for text categorization problems.

The rest of the paper is structured as follows. Section 2 gives the background of one-class SVMs, the problems of using the standard document representation with one-class SVMs and their respective fixes are detailed in section 3. Section 4 lists the proof that the same modification is valid in learning with positive and unlabeled examples. Section 5 provides the empirical results. We conclude in section 6 with some discussion of the results.

2 Introduction to One-Class SVMs

One-class SVMs [11] are closely related to the so-called minimum volume estimators which try to find a small region containing most of the positive examples. The goal is to find the boundary function that can be used for discrimination purposes. Avoiding the density estimation problem, this region estimation approach is in line with Vapnik's principle of never solving a problem which is more general than the one that actually needs to be solved. Similar to the two-class SVMs, regularization is also used to balance the training errors
plexity for better generalization. Since text categorization problems are generally considered as linearly separable even in original feature space [15], we will focus on hyperplane-based one-class SVMs and use only dot product (linear) kernel in this paper.

Hyperplane-based one-class SVMs seeks a hyperplane that pushes positive examples away from origin as much as possible without leaving too many positive examples behind. Given a set of positive examples: $S = (x_1, \cdots, x_m)$, the hyperplane (w, ρ) is given by solving the following primal quadratic optimization problem:

$$\begin{aligned} \text{minimize}: & \frac{1}{2}\langle w \cdot w \rangle - \rho + \frac{1}{\nu m}\sum_{i=1}^{m}\xi_i \\ \text{subject to}: & \langle w \cdot x_i \rangle \geq \rho - \xi_i \\ & \xi_i \geq 0, i = 1, \cdots, m \end{aligned}$$

Here, w is the weight vector for decision hyperplane, ρ is the functional distance from origin to hyperplane (w, ρ). The slack ξ_i is defined by how far a data point fails to stay away from origin with respect to boundary hyperplane (w, ρ). Like their two-class counter-part [11], there are two goals sought by its objective function: larger geometric distance defined by $\rho/||w||$ and smaller training errors approximated by $\sum \xi_i$.

Parameter $\nu \in (0, 1]$ is used to control the trade-off between these two possibly conflicting goals. It is an upper bound on the fraction of training margin errors and lower bound on the fraction of support vectors. With probability approaching 1, asymptotically, ν equals to both the fraction of support vectors and fraction of training margin errors. A training margin error occurs $\xi_i > 0$. We choose to work with this ν formulation because it is intuitive to pick the parameter ν due to its property.

Due to the nonsmoothness introduced by ξ in primal cost function, it is a common practice to solve the dual problem to find the coefficients αs. Using dot product as kernel, both one-class and two-class SVM training result in a linear decision function in the same form of:

$$f(x) = sign(\langle w \cdot x \rangle - \rho) = sign(\sum w^k x^k - \rho) \tag{1}$$

Where the kth component of the weight vector w is defined as:

$$w^k = \sum \alpha_i y_i x_i^k, \ \alpha_i \geq 0 \tag{2}$$

While $y_i \in (-1, +1)$ for two-class SVM, we have $y_i \in (+1)$ for one-class SVM since there are only positive examples available for one-class SVM training.

3 Document Representation Issues

Most text categorization studies use the term weighting scheme that is developed for information retrieval applications [10]. Such representation includes three components: term frequency, document frequency, and normalization component. Let x_i^k denote the k component for ith document x_i, we have $x_i^k \geq 0$ for the standard term weighting scheme. However, the following analysis holds for $x_i^k \leq 0$.

The document representation based on term weighting scheme reportedly works well with two-class SVM classifiers [3]. However, due to the lack of negative examples, such standard representation does not bring out the full potential of one-class SVMs. Our discussion on document representation in this section is based on a fully labeled dataset with both positive and negative examples. Document representation issues for partially labeled dataset with positive and unlabeled examples are addressed

3.1 Using Positive Features Only

In feature selection research, it is commonly known that there are two types of features: positive and negative features [16]. Positive features are positively

correlated with the category of interest; occurrences of such features in a document basically add more support for the document belonging to the category. Similarly, occurrences of negative features in a document should decrease the probability of the document belonging to the category. When working with linear decision functions, one typically expects that weights for positive features stay positive and weights for negative features stay negative. This is because a positive weight for a negative feature is deemed to degrade the performance of the linear classifier.

From expression (2), for one-class SVM, one easily has $w^k \geq 0$ since $y_i = +1$. So if there is a negative feature contained in one of the support vectors, its influence on decision hyperplane will be positive instead of negative. This suggests that, for one-class SVMs, the document representation should only use positive features. Inclusion of negative features will only degrade the performance of the learned classier.

Table 1. Contingency table for feature-category correlation, where category absent implies either not-in-class or lack the label information

	category present	category absent
feature present	a	b
feature absent	c	d

In this paper, we use correlation coefficient (CC) to determine whether a feature is positively correlated with the category of interest. It is first used as a feature selection measure in [8] and is defined as:

$$(ad - bc)\sqrt{N}/\sqrt{(a+b)(a+c)(b+d)(c+d)}$$

where a, b, c, d are defined in table 1, and $N = a+b+c+d$. The sign of a feature can thus be easily decided based on the feature-category correlation contingency table if one works on fully labeled dataset.

It is the common knowledge that negative features are potentially useful for discrimination purpose. However, from expression (2), one needs $\sum_{y_i=-1} \alpha_i x_i^k > \sum_{y_i=+1} \alpha_i x_i^k$ to have $w^k \leq 0$. This suggests that importance of negative features is mainly characterized by negative examples. It is difficult to model negative features with one-class SVMs since its training involves only positive examples. Luckily, as we will reveal later, the categories are largely characterized by their corresponding positive features.

3.2 Relative Importance of Features

With both positive and negative examples, two-class SVMs are generally capable of determining the importance of each feature. However, with only positive examples, one-class SVMs lack some of the information needed to determine the importance of features. To see this, assume that there are two positive features,

f_1 and f_2, and these two features always occur together in the positive dataset. But feature f_1 occurs a lot more in negative examples. Although there is evidence for two-class SVMs to give feature f_2 more weight, for one-class SVMs, there is no reason to treat them differently.

Clearly, for one-class SVMs to work better, the relative importance of features, usually measured by feature selection metric, has to be embedded in document representation. We propose in this paper to scale each dimension based on correlation coefficient. The basic idea behind this proposed solution is simple: we want training process pay more attention to these important features. In another word, cost rooted from the less important feature should be penalized less.

Scaling features using feature weight is equivalent to modifying the similarity measure, or equivalently, distance metric, kernel function. Appropriate scaling can bring all positive examples closer thus make training discriminant model easier. For example, it is shown that performance of Nested Generalized Exampler(NGE) can be greatly improved when the Mutual Information (MI) is used to reweight features [12]. In general, it is hard to justify picking one feature selection metric over another theoretically. We choose correlation coefficient based on empirical evidence, since preliminary experiments show that it is more effective comparing to other feature selection metric such as χ^2 and information gain.

3.3 Document Length Normalization Issue

Document length normalization is important for one-class SVM since only positive features are correctly modeled. To see this, assume we are to determine the class label for a long document d_l which contains multiple copies of a not-in-class document d_s (thus $\langle w \cdot d_s \rangle < \rho$). For one-class SVM, we always have $\langle w \cdot d_s \rangle \geq 0$. So, the long document d_l with $\lceil \rho / d_s \rceil$ copies of d_s will be considered as in-class since one have $\langle w \cdot d_l \rangle \geq \rho$. Here, $\lceil \cdot \rceil$ denote the ceil function. For two-class SVM, since both positive and negative features are modeled, typically we will have $\langle w \cdot d_s \rangle \leq 0$, which makes $\langle w \cdot d_d \rangle \leq 0$ as expected.

To address this issue, we propose to apply the cosine normalization in test phase. Although the above analysis also holds when such normalization is applied in training phase, preliminary results show that normalization in training phase hurts the performance of one-class SVM. This is because length normalization in training stage makes values of positive features depend on values of negative features, while the negative features are not correctly modeled by one-class SVM.

Normalization in test phase will only make the learned threshold value unusable. However, it is not a big issue since a separated thresholding component is needed for one-class SVM anyway. To see this, note that the default threshold returned by one-class SVMs training tends to be too high as it touches all the positive support vectors.

4 Learning with Positive and Unlabeled Examples

Except for the document length normalization, the modifications we proposed in the last section are based on the assumption that we have access to the fully labeled dataset. We now show that the category statistics needed for proposed document representation modification can also be estimated from datasets with positive and unlabeled examples.

Learning with Positive and Unlabeled examples (LPU) itself is interesting research topic for both theoretical [2, 5] and practical [14, 5, 4, 6] reasons. Formally, learning with positive and unlabeled examples can be modeled as follows: positive examples are randomly labeled positive with probability $1 - \beta$, and are left unlabeled with probability β. With this model, if we label all unlabeled examples as negative, we will never make an error on a negative example but will label positive examples as negative with probability β. In practice, β is generally unknown, effective solutions to this problem thus should not depend on the knowledge of β.

We have following proposition that characterizes the expectation of the quality of correlation coefficient estimated from positive and unlabeled examples.

Proposition 1 *Let a^*, b^*, c^* and d^* be corresponding entries in the contingency table for feature-category correlation with positive and unlabeled data, where positive examples are left unlabeled with probability β. Assume that feature occurrences are independent of the labeling process. We have, first, the sign of a feature is then expected to be the same as that of the expression $a^*d^* - b^*c^*$. Second, let CC^* be the correlation coefficient defined on table 1, the ratio between CC and CC^* is expected to be a constant that is feature independent.*

Proof sketch. For first part, from the independent assumption, we have $E(a^*) = a - \beta a$, $E(b^*) = b - \beta a$, $E(c^*) = c - \beta c$ and $E(d^*) = d - \beta a$. Here $E(\cdot)$ denotes the expected value of a random variable. It is not difficult to see that $E(a^*d^* - b^*c^*) = (ad - bc)(1 - \beta)$. When $\beta < 1$, $a^*d^* - b^*c^*$ and $(ad - bc)$ is thus expected to have the same sign. For second part, using the expected value for a^*, b^*, c^*, d^* as before, it is not difficult to see:

$$E(CC^*/CC) = \sqrt{(b+d)(1-\beta)/(b+d+\beta a+\beta c)}$$

Note that both $a + c$ and $b + d$ are fixed for all features for each category, and β is also feature independent.

The proposition states that, on average, CC^* is a good replacement of CC since the ratio between them is expected to stay constant for each feature. The probability of such statement to hold, however, depends on both the dataset and feature. In general, the more positive examples labeled and the higher frequent of a feature, the higher the probability for these statements to stay true. Moreover, less frequent word tend to have smaller impact on text categorization applications, as noted in [13].

Since both the quality of the representation modifications based on CC^* and the training for the one-class SVMs are independent of the percentage of the

positive examples left unlabeled, the performance of the one-class SVM with proposed representation modifications is arguably also independent of the percentage of the positive examples left unlabeled. This is a highly desired property for obvious reasons. For example, to get an exact value for β for quality control purposes, one has to label the entire dataset. Note however, more positive examples are beneficial for both the estimating of CC^* and the training of one-class SVM.

5 Experiments

We conduct all our experiments on the standard text categorization dataset: Reuters-21578 compiled by David Lewis from Reuters newswire. The ModApte split we used has 90 categories. After removing all numbers, stop words and low frequency terms, there are about 10,000 unique
selection is done. Words occurring in the title are simply counted three times. Baseline document representation is $log(1 + tf)$, where tf is the term frequency defined by the number of occurrences for that term. We use libSVM [1] to train both one-class and two-class SVMs in this paper. To compare the performance of linear classifiers based on the orientation of their decision hyperplane, and to stay comparable with [15], we use both the micro-average F1 and macro-average F1 over Break-Even-Point(BEP) as performance measurement.

Experiments are organized in two different parts. In the next subsection, we examine effectiveness of the three representation improvements on the fully labeled datasets. In the subsection that follows, the effectiveness of the improvements on the positive and unlabeled dataset is studied.

5.1 Effectiveness of Representation Improvements

To test the effectiveness of the proposed modifications to document representation, we run both one-class SVMs (oc) and two-class SVMs (tc) on baseline document representation. We then modify the document representation for one-class SVMs by incorporating the following representation changes one at a time: removing negative features based on the sign of correlation coefficient (p), scaling dimensions based on the magnitude of correlation coefficient(s), and also normalization (m). Note the correlation coefficient is computed here based on the contingency table 1. To make our results comparable with previous reported results, we report results on both the first 10 most frequent categories and all 90 categories. From table 2, our results on all 90 categories with two-class SVMs are comparable with that of [15], and our results on first 10 most frequent categories with one-class SVMs are comparable with that of [7]. Table 2 suggest that these three modifications for document representation can provide significant performance improvements for one-class SVM. Using all three modification together, on all 90 categories, the performance gap between one-class and two-class is reduced from 0.364 to 0.048 measured in micro-average F1 that is an

Table 2. Performance of different document representation methods on one-class and two-class SVMs, measured in micro-average F1(miF1) and macro-average F1(maF1) over BEP on both all 90 categories and first 10 most frequent categories on the Reuters-25718 dataset. Here, oc(tc) corresponds to one(two) class SVM

	90 categories		10 categories	
	miF1	maF1	miF1	maF1
oc	0.516	0.293	0.583	0.460
oc.pos	0.599	0.340	0.676	0.538
oc.scale	0.745	0.641	0.767	0.710
oc.norm	0.715	0.493	0.784	0.681
oc.pos.scale	0.763	0.651	0.792	0.755
oc.norm.scale	0.834	0.685	0.872	0.810
oc.norm.pos	0.750	0.499	0.823	0.722
oc.norm.pos.scale	0.835	0.686	0.873	0.817
tc	0.880	0.679	0.924	0.858
tc.scale	0.880	0.679	0.924	0.858
tc.pos	0.840	0.650	0.878	0.829
tc.norm	0.866	0.662	0.917	0.844

87% reduction. At same time, macro-average F1 is reduced from 0.386 to -0.006, which suggests that one-class SVMs are more effective on the rare categories.

It is interesting that with appropriate representation, using only positive examples can result in a performance that is close to 95% of that of using both positive and negative examples. It suggests that binary text categorization problems reduced from multi-label problems in one-versus-rest fashion are mostly characterized by its positive examples. Moreover, if one works on positive features only, one-class SVMs with proposed document representation (oc.m.p.s) is as effective as two-class SVMs (tc.p). This provides the empirical evidence that the importance of positive features can mostly be modeled by positive examples. For two-class SVMs, the performance difference between using all features and using only positive features is rather small, again 5% difference. This is rather surprising, but it suggests that, for the purpose of discrimination, the additional information embedded in negative features is really small.

While the feature scaling can greatly improve the performance of one-class SVM(oc vs. oc.s), their influence on two-class SVM is not observable at all(tc vs tc.s). This suggests that two-class SVM has all the information needed to learn the importance of each feature and one-class SVM does not. Note that the identical performance of two-class SVM measured in average F1 before and after feature scaling is misleading, because stronger h as AUC (Area Under Curve)reveals that scaling does provide some marginal improvement for two-class SVM. Both one-class and two-class SVM training return a different set of support vectors after the feature scaling. Furthermore, the number of support vectors decreases noticeably in the scaled feature space for both one-class and two-class SVM training. For example, at $\nu = 0.01$, for category "earn", feature

scaling reduces the number of positive/negative support vectors from 260/436 to 169/263 for two-class SVM training and from 99/0 to 44/0 for one-class SVM training. Due to the high dimensionality of the problem, it is difficult to understand exactly how the scaling influences the training process. But the fewer support vectors seems to suggest that data is easier to separate after feature scaling.

All the three document representations provide some meaningful performance gains when used alone, with scaling as the most effective factor and negative-feature removing the least. Furthermore, it seems that the discrimination power contained in the negative-feature removing is mostly contained in scaling. This is because scaling can greatly reduce e features, since the absolute value of feature selection metric such as correlation coefficient is often much smaller than that of positive features, as noted [16].

There is no known close-form time-complexity analysis for SVM training as it depends on the dataset, termination criteria and parameter choices. To get a rough idea, we timed libSVM java implementation using a Pentium-M 1.3Ghz Linux PC with 512M memory. Including the feature scoring for all 10 categories, it takes 90 minutes for two-class SVM training, and 2 minutes for one-class SVM. This is mostly because one-class SVM training only uses positive examples.

5.2 Learning from Positive and Unlabeled Examples

To test whether the performance of one-class with proposed modifications depends on β, the fraction of positive examples left unlabeled, we intentionally hide the label for 19, 36, 51, 64, 75, 84, 91, 96, and 99 percent of randomly selected positive examples. One-class SVMs are trained on the remaining positive examples. For the document representation, we removed negative features, scale features based on magnitude of correlation co t computed from the contingency table 1. Document representation is normalized in the test phase.

Figure 1 reports both the micro and macro F1 (based on BEP) for first 10 most frequent categories. Notice that the performance of one-class stays virtually constant until there is only 4% ($\beta = 0.96$) positive examples left labeled. The significant performance drop of both micro and macro F1 at $\beta = 0.99$ is understandable, as 8 out of 10 categories are left with less than 6 positive examples. We believe that the performance of one-class SVMs on positive and unlabeled dataset depends only on the number of positive examples used, not the fraction of positive examples used. To test this argument, we also report F1 (over BEP) on the most frequent category "earn", which has 27 positive examples at $\beta = 0.99$. From figure 1, the performance of the category "earn" stays almost untouched even when only 1% of positive examples are used in the training. The results thus conform elegantly to the analysis we had in section 5.

We are able to compare the result directly with biased two-class SVM approach in [6] since we are using the same 10 categories from the same dataset. Note that when β increased from 0.3 to 0.7, the micro F1 stayed around 0.87 for one-class SVMs, the macro F1 dropped from 0.856 to 0.785 for biased wo-class

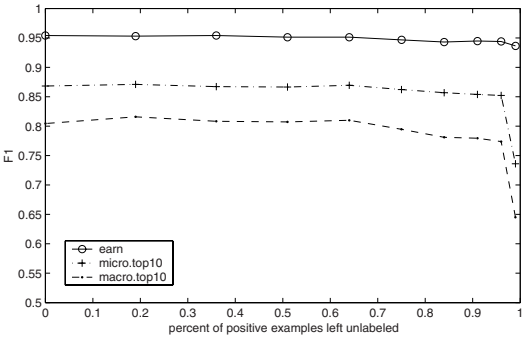

Fig. 1. performance of one-class SVMs versus β, the fraction of positive examples left unlabeled

Table 3. Performance of one-class and two-class SVMs on different β value on 10 most frequent categories on Reuters-25718, measured in micro-average F1.

β	0.0	0.3	0.7
oc.n.p.s	0.873	0.867	0.881
tc	0.924	0.856	0.785

SVMs. It appears from this direct comparison on macro F1 that one-class SVM is a more effective method, particularly when β is approaching to 1.0.

6 Conclusion and Discussion

In this paper, we identify the "incompatibility" between the standard document representation and one-class SVMs. We propose several modifications to document representation that use the correlation coefficient, which can be estimated from not only the fully labeled dataset but also the dataset with positive and unlabeled examples. The experiments show that the proposed representation modifications can greatly improve the performance of one-class SVMs.

As a case study on text categorization, this paper provides quantitative empirical evidence that for binary classification problems verted from multi-label problem, the category is mostly characterized by positive examples. Furthermore, we also reveal through experiment that the nature of category is mostly embedded in the feature space spanned by positive features. It is thus interesting to see whether these trends exist in other binary classification problems that are converted from multi-label problems.

In practice, it is usually difficult to obtain labeled data, but unlabeled data is often abundantly available. This means that learning with positive and unlabeled examples is a much more realistic problem when β approaches to 1.0 than when it approaches to 0.0. Most previous studies assume at least some of unlabeled

examples as negative examples so that they can use standard two-class learning methods. The probability for such assumption to hold is thus negatively correlated with β. These previous methods usually perform reasonably well when β is small and the performance degrades with an increasing β [5, 6]. However, the performance of one-class SVM with proposed document representation is independent on β. While not impressive when β is small, it is very competitive when β is large. This makes one-class SVM a very useful method in many real-world applications.

Although the example application used in this paper is text categorization problems, the only requirement for the proposed representation modifications is the sparseness of the data representation, which makes computing the correlation coefficient possible. In general, we believe that one-class SVM with the proposed modification can be directly used with any sparse data with appropriate data preprocessing.

Acknowledgments

We thank Dr. Zhixin Shi, Dr. Ajay Shekhawat and anonymous reviewers for their value comments.

References

1. C. Chang and C. Lin. LIBSVM: a library for support vector machines (version 2.3), 2001.
2. François Denis. PAC learning from positive statistical queries. In *Algorithmic Learning Theory, 9th International Conference, ALT '98, Otzenhausen, Germany, October 1998, Proceedings*, volume 1501, pages 112–126. Springer, 1998.
3. Thorsten Joachims. *Learning To Classify Text Using Support Vector Machines.* Kluwer Academic Publishers, Boston, 2002.
4. Wee Sun Lee and Bing Liu. Learning with positive and unlabeled examples using weighted logistic regression. In *Proceedings of ICML-03, 20th International Conference on Machine Learning.* ACM Press, US, 2003.
5. B. Liu, W. Lee, P. Yu, and X. Li. Partially supervised classification of text documents. In *Proc. 19th Intl. Conf. on Machine Learning*, Sydney, Australia, July 2002.
6. Bing Liu, Yang Dai, Xiaoli Li, Wee Sun Lee, and Philip S. Yu. Building text classifiers using positive and unlabeled examples. In *Proceedings of Third IEEE International Conference on Data Mining*, Melbourne, Florida, 2003.
7. Larry M. Manevitz and Malik Yousef. One-class svms for document classification. *Journal of Machine Learning Research*, 2:139–154, 2001.
8. Hwee T. Ng, Wei B. Goh, and Kok L. Low. Feature selection, perceptron learning, and a usability case study for text categorization. In Nicholas J. Belkin, A. Desai Narasimhalu, and Peter Willett, editors, *Proceedings of SIGIR-97, 20th ACM International Conference on Research and Development in Information Retrieval*, pages 67–73, Philadelphia, US, 1997. ACM Press, New York, US.
9. J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Scholkopf, C. Burges, and A. Smola, editors, *Advances in kernel methods - support vector learning.* MIT Press, 1998.

10. Gerald Salton and Buckley C. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.
11. Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization,Optimization and Beyond*. The MIT Press, Cambridge, Massachusetts, 2002.
12. Dietrich Wettschereck and Thomas G. Dietterich. An experimental comparison of the nearest-neighbor and nearest-hyperrectangle algorithms. *Machine Learning*, 19(1):5–27, 1995.
13. Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In *Proc. 14th International Conference on Machine Learning*, pages 412–420. Morgan Kaufmann, 1997.
14. Hwanjo Yu, Jiawei Han, and K. C-C. Pebl: Positive example-based learning for web page classification using svm. In *Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery in Databases (KDD02)*, pages 239–248, New York, 2002.
15. Tong Zhang and Frank J. Oles. Text categorization based on regularized linear classification methods. *Information Retrieval*, 4(1):5–31, 2001.
16. Zhaohui Zheng, Xiaoyun Wu, and Rohini Srihari. Feature selection for text categorization on imbalanced datasets. *KDD Exploration, Special issue on Learning from Imbalanced Datasets(to appear)*, 6(1), 2004.