Automated Text Categorization Using Support Vector Machine

James Tin-Yau Kwok
Email:jamesk@comp.hkbu.edu.hk

Department of Computer Science
Hong Kong Baptist University
Hong Kong

ABSTRACT

In this paper, we study the use of support vector machine in text categorization. Unlike other machine learning techniques, it allows easy incorporation of new documents into an existing trained system. Moreover, dimension reduction, which is usually imperative, now becomes optional. Thus, SVM adapts efficiently in dynamic environments that require frequent additions to the document collection. Empirical results on the Reuters-22173 collection are also discussed.

1. Introduction

The increasingly widespread use of information services made possible by the Internet and World Wide Web (WWW) has led to the so-called *information overloading* problem. Today, millions of online documents on every topic are easily accessible via the Internet. As the available information increases, the inability of people to assimilate and profitably utilize such large amounts of information becomes more and more apparent. Developing user-friendly, automatic tools for efficient as well as effective retrieval of relevant information thus has great demand in the Internet world.

The most successful paradigm to organize a large mass of information is by categorizing the different documents according to their topics. Recently, various machine learning techniques have been attempted to automate text categorization. These approaches are usually based on the *vector space model*, in which documents are represented by sparse vectors, with one component for each unique word extracted from the document. Typically, the document vector is very high-dimensional, at least in the thousands for large collections. This is a major stumbling block in applying many machine learning methods. Existing techniques rely heavily on dimension reduction as a preprocessing step, which are sometimes quite *ad hoc* and can be computational expensive. Another concern is that, in view of the rapid explosion in the amount of online information, these automatic text categorization tools should be able to handle a steady influx of new information. However, many existing machine learning and dimension reduction techniques require complete re-training whenever a larger collection of documents becomes available. In this paper, we investigate a different approach of integrating both dimension reduction and classification, by applying the technique of *support vector machine* (SVM).

The rest of this paper is organized as follows. In Section 2, we will first review some representative machine learning approaches that have been used in text categorization. Section 3 gives a brief introduction to the SVM. Section 4 discusses the application of SVM to text categorization. Experimental results on the Reuters-22173 corpus will be presented in Section 5, followed by some concluding remarks in the last section.

2. Representative Machine Learning Approaches for Text Categorization

Rule-Based Approach This is one of the earliest approaches attempted. A well-known example is the CONSTRUE system [6]. Recently, automatic rule induction techniques have also been used to avoid the laborious construction of the rule base. Dimension reduction is always necessary in order to reduce the search space. One of the main advantages of these rule-based approaches is that the resulting rules are usually readily interpretable by humans.

k-Nearest Neighbor (k-NN) Classifier Variants of the k-NN classifiers have shown very good performance in text categorization [16]. However, on categorizing a new document, a large amount of computational power is required for calculating its similarity with every training document.

Linear Models A number of linear models have been used [10, 16]. They are usually based on linear regression, linear discriminant analysis, logistic regression, iterative learning algorithms or the Rocchio algorithm in traditional information retrieval. Although the full set of terms can sometimes be used [10], in general there are computational problems (such as very slow convergence or very large matrix) and the resultant linear model is also prone to over-fitting. Thus, dimension reduction is usually required.

Neural Networks Because the decision boundary in a text categorization problem may not be linear, nonlinear classifiers such as artificial neural networks may produce better results than linear models. To alleviate the problem of high dimensionality (and hence a large network size), Wiener *et al.* [14] used both weight elimination and early stopping in addition to dimension reduction. However, experimental results suggest that the nonlinearity in neural networks does not yield substantial gain over linear models.

Probabilistic Models. They are widely used in many text categorization experiments. Examples include the naive Bayes classifier [7] and the hierarchical Bayesian classifiers [8]. A commonly used simplifying, though unrealistic, assumption is that words in the training set are probabilistically independent of one another. Moreover, dimension reduction is always necessary. Furthermore, it is not obvious whether these classifiers have any intrinsic model complexity control mechanism, and so over-fitting may occur.

Most of the above techniques, with the exception of the k-NN classifiers, work best on static corpuses. In dynamic environments that require frequent additions to the document collection, incorporating the new documents into an existing trained system is difficult.

3. Support Vector Machine

Although there has been continuous improvement in dimension reduction techniques, a more promising approach for use in text categorization is to integrate dimension reduction and classification. The *support vector machine* (SVM) [3, 13] is a new machine learning technique along this line[1]. SVM is for two-group classification problems. Let the training set D be $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, with each input $\mathbf{x}_i \in \Re^m$ and the output label $y_i \in \{\pm 1\}$. The SVM performs a mapping ϕ from the input space $\Re^m$ to a "feature" space $\Re^n$. First consider the case when the data is linearly separable in $\Re^n$, i.e., there exists a vector $\mathbf{w} \in \Re^n$ and a scalar $b \in \Re$ such that

$$\mathbf{w}^T\phi(\mathbf{x}_i) + b \geq 1, \quad \text{if } y_i = 1, \tag{1}$$

$$\mathbf{w}^T\phi(\mathbf{x}_i) + b \leq -1, \quad \text{if } y_i = -1, \tag{2}$$

for all elements in the training set. The SVM constructs a hyperplane $(\mathbf{w}^T\phi(\mathbf{x}) + b)$ for which the separation between the positive and negative examples is maximized (Figure 1). Note that a hyperplane in $\Re^n$ may correspond to a nonlinear decision surface in $\Re^m$. It can be shown [3] that the $\mathbf{w}$ for this "optimal" hyperplane can be found by minimizing $\|\mathbf{w}\|^2$, and the resultant solution can be written as a linear combination of $\phi(\mathbf{x})$'s, i.e., $\mathbf{w} = \sum_{i=1}^N \alpha_i y_i \phi(\mathbf{x}_i)$ for some $\alpha_i \geq 0$. This vector of α_i's, $\mathbf{\Lambda} = (\alpha_i, \ldots, \alpha_N)$, can be found by solving the following quadratic programming (QP) problem:

$$\text{maximize } W(\mathbf{\Lambda}) = \mathbf{\Lambda}^T 1 - \frac{1}{2}\mathbf{\Lambda}^T Q \mathbf{\Lambda}, \tag{3}$$

with respect to $\mathbf{\Lambda}$, under the constraints $\mathbf{\Lambda} \geq 0$ and $\mathbf{\Lambda}^T\mathbf{Y} = 0$, where $\mathbf{Y}^T = (y_1, \ldots, y_N)$ and Q is a symmetric matrix with elements $Q_{ij} = y_i y_j \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$. To obtain Q_{ij}, one can find a *kernel* $K(\cdot,\cdot)$ such that $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$, and Q_{ij} is then computed as $y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$. For example, the kernel for a polynomial classifier of degree d is $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T\mathbf{x}_j + 1)^d$. Moreover, notice that Q is always positive semi-definite and so there is no local optima in (3). For those training examples along the margins of the decision boundary, the corresponding α_i's are greater than zero (by the Kuhn-Tucker theorem), and these examples are called the *support vectors*.

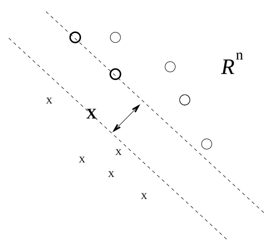

Figure 1: An example of separable data. Patterns from the two classes are represented by $\times$ and $\circ$ respectively. The extracted support vectors are shown in bold.

During testing, for a test vector $\mathbf{x} \in \Re^m$, we first compute

$$s = \mathbf{w}^T\phi(\mathbf{x}) + b = \sum_{i=1}^N \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b. \tag{4}$$

[1]Besides classification and pattern recognition problems, SVM has also been successfully applied to problems like regression estimation and time series prediction.

With a user-defined threshold s_0, the class label for $\mathbf{x}$ is assigned by the following rule:

$$\text{label} = \begin{cases} 1 & s > s_0, \\ -1 & \text{otherwise.} \end{cases} \tag{5}$$

When the training set is not separable in $\Re^n$, the SVM algorithm tries to minimize $\|\mathbf{w}\|$ while at the same time separating the set with a minimum number of errors. This is done by minimizing

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^N \xi_i, \tag{6}$$

where $\xi_i \geq 0$ satisfy the constraints:

$$\mathbf{w}^T\phi(\mathbf{x}_i) + b \geq 1 - \xi_i, \quad \text{if } y_i = 1, \tag{7}$$

$$\mathbf{w}^T\phi(\mathbf{x}_i) + b \leq -1 + \xi_i, \quad \text{if } y_i = -1. \tag{8}$$

The user-defined parameter C balances contributions from the first and second terms. Again, minimization of (6) can be transformed to a QP problem: maximize (3) under the constraints $0 \leq \mathbf{\Lambda} \leq C\mathbf{1}$ and $\mathbf{\Lambda}^T\mathbf{Y} = 0$.

Unlike other machine learning methods, SVM does not depend explicitly on the dimensionality of the problem. Experimentally, SVM has achieved superior performance on a number of fairly high-dimensional data sets. For example, it outperforms radial basis networks on recognizing the US postal service database of handwritten digits (using 16×16 bitmap as input) [12]. Promising results are also reported in face detection [11], where the input dimensionality is 283.

4. Using SVM in Text Categorization

The success of SVM lies in its automatic capacity tuning by minimizing $\|\mathbf{w}\|$, extraction of a small number of support vectors from the training data that are relevant for the classification, and also framing the computational problem as a QP problem with no local optima. Moreover, motivated by its superior performance on the data sets mentioned above, we here apply SVM to text categorization, a problem of very high dimensionality. As the document topics are not mutually exclusive, text categorization is usually analyzed as a series of dichotomous classification problems, i.e., whether the document belongs to a particular topic or not. In Section 4.1, we discuss the suitability of using SVM in dynamic environments. Section 4.2 considers combining SVM with latent semantic indexing to further improve its performance, and Section 4.3 discusses the possibility of using SVM to identify mis-labeled documents.

4.1. Ability to Incorporate New Documents

In the formulation of the SVM, the number of variables in the QP problem is equal to the number of data points. Moreover, the quadratic form is completely dense. These make the optimization problem very challenging when the data set is large, as the memory requirement grows with the square of the size of data set. *Chunking* [3] aims at reducing the computational and storage complexities by breaking this QP problem into smaller sub-problems. The training corpus is divided into a number of *chunks*, support vectors are extracted from each of these, and are later combined together. In this paper, we use the decomposition algorithm in [11]. This algorithm has the advantage of guaranteeing global optimality, in the sense that the final solution obtained is the same as if the original optimization problem had not been decomposed.

Besides allowing SVM to scale-up to large training corpus, chunking also makes SVM very suitable for dynamic environments. When new documents arrive, they can be easily incorporated into the existing set of support vectors (extracted from the original corpus) using the above decomposition algorithm. The property of global optimality guarantees that the quality of the final solution is as good as we had all the documents processed together. This is in sharp contrast to most traditional machine learning methods mentioned in Section 2, with the exception of the k-NN classifiers. Though k-NN classifier does not require any training phase when new documents arrive, its large storage and computational requirements may soon become infeasible as the corpus becomes very large. On the other hand, SVM only needs to store the support vectors and to operate on them during testing. Experimentally, the number of support vectors only increases sub-linearly with the number of training patterns [5]. Further speed-up is also possible [2].

4.2. Combining with Latent Semantic Indexing

In the formulation of the SVM, one always tries to minimize $\|\mathbf{w}\|$. This tends to shrink the weights associated with any spurious features, and thus can be seen as a "soft" version of feature selection. However, this may not perfectly remove all the irrelevant features. Moreover, textual documents has the intrinsic problem of *synonymy*, i.e., there are usually many ways to refer to the same concept in natural language. This synonymy problem is not even addressed in the general framework of SVM.

One method to alleviate these problems is to combine SVM with *latent semantic indexing* (LSI) [4]. By analyzing the correlational structure of terms in the document collection, it seeks to linearly transform the set of document vectors to a lower-dimensional space. It also helps to alleviate the problem of synonymy in textual documents. We apply LSI on the set of support vectors, and then run SVM again on the resultant LSI representation. This makes sense when the data is separable, as then all the support vectors lie on the margins of the decision boundary (Figure 1). They contain all the essential features to shape and define the boundary, and are thus more important than patterns further away. Alternatively, one may view the SVM as a pre-processing step to first extract a "crucial" subset of documents on which LSI is performed, similar to the idea of using *local regions* in [14]. Besides improving the quality of the resultant LSI components as compared to performing LSI on the whole corpus, the computational demand is also significantly reduced as the set of support vectors is usually a small fraction of the corpus. Compared to [14], our approach is more disciplined in that both the size of this subset and its membership are determined by the learning process through SVM.

This, however, may not work when the data is not separable, as then the support vectors may not be lying on the margin of the decision boundary. Applicability of LSI under such circumstances is still under investigation.

4.3. Mis-labeled Documents

Human errors may be introduced in the manual process of categorizing documents in the training set, and SVM can be used to help locating these errors. From the theory of constrained optimization, each Lagrangian multiplier indicates how sensitive the objective function (3) is to changes in the corresponding constraints ((1), (2), (7) or (8)). SVM is very sensitive to atypical patterns that are close to the decision boundary and patterns associated with very large Lagrangian multipliers are usually atypical (i.e., mis-labeled documents). Moreover, the robustness of the solution – and potentially its generalization performance – can be increased by omitting these support vectors with large Lagrangian multipliers [5].

5. Simulation

5.1. Setup

We use the 14704/6746 division of the Reuters-22173 collection[2]. This is a set of 21,450 Reuters newswire stories from the year 1987. These stories have been manually indexed using 135 financial topic categories, to support document routing and retrieval by Reuters customers. All stories dated April 7, 1987 and earlier went into a set of 14,704 training documents, and all stories from April 8, 1987 or later went into a test set of 6,746 documents. We drop those topics that have no positive training examples. Then we perform inflectional stemming, stop word removal, conversion to lower case, and elimination of words that appeared in fewer than three documents. Documents are finally coded using the traditional $tf \cdot idf$ (term frequency – inverse document frequency) representation:

$$\frac{f_i(w_j)}{\sqrt{\sum_{w_j \in D_i} f_i^2(w_j)}} \log(\frac{N}{N(w_j)}),$$

where $f_i(w_j)$ is the frequency of word w_j in document D_i, N is the total number of documents, and $N(w_j)$ the number of documents containing w_j. As in [1], we use local dictionaries for each topic. Only single words found in documents on the given topic are entered in the local dictionary. However, unlike [1] where they used the 80–100 most frequently occurring words in the local dictionaries as features, here no feature selection step is taken. The number of features in the local dictionaries is thus very large, with an average of 12433.6.

In the experiment, we use SVM of order 1 polynomial kernel, which corresponds to a hyperplane in the $tf \cdot idf$ space. Training patterns are processed in chunks, with a maximum size of 1500. As mentioned in Section 4.1, one may regard this SVM as processing the training corpus in an incremental manner. Training of the SVM proceeds by first assuming that the data for that particular topic is separable. If failed, then we proceed to the non-separable case. The value of C in (6) is always set to 10.

Performance is measured by *recall* and *precision*[3]. Different combinations of recall and precision are obtained in SVM by altering the threshold s_0 in (5). The point at which recall equals precision is the *break-even* point. It is used as a single summarizing measure for comparison of results. Because the text categorization problem is analyzed as a series of dichotomous classification problems, we use both *micro-averaging*

[2]The Reuters-22173 collection is available by anonymous ftp from /pub/reuters1 on ciir-ftp.cs.umass.edu. Arrangements for access were made by David Lewis.

[3]Recall is the percentage of total documents for the given topic that are correctly classified. Precision is the percentage of predicted documents for the given topic that are correctly classified.

and *macro-averaging* to evaluate overall performance across the entire set of topics[4].

One can show that all the support vectors, with the exception of those whose Lagrangian multipliers equal to the value of C in the non-separable case, attain ± 1 for the value of s given in (4) [11]. This property enables the SVM outputs to be comparable across all categories. They thus can be easily combined together during micro-averaging, without resorting to more *ad hoc* schemes like proportional assignment [9].

5.2. Results

Most of the topics in the Reuters data set turn out to be linearly separable in the $tf \cdot idf$ space. Those that are not separable are listed in Table 1. We suspect that many other text categorization data sets are also linearly separable. If this is the case, then it may well explain the success and popularity of linear techniques in text categorization. It also agrees with the observation that nonlinear classifiers usually cannot outperform linear classifiers [14].

Figure 2 shows the micro-averaged recall-precision curves. SVM1 refers to the basic SVM. SVM2 refers to SVM with the combined use of LSI (Section 4.2) and removal of large Lagrangian multipliers (Section 4.3). For comparison, we also plot the performance of a k-NN classifier [15], with k set to 30 (as in [15]). Table 2 shows the micro-averaged and macro-averaged break-even points for SVM, k-NN classifier and five other categorization methods on this collection as reported in [9]. As can be seen, just plain vanilla SVM does a very good job, despite the fact that there is no feature selection performed in our experiment. Adding the refinements in Section 4.2 and Section 4.3 further boost the performance. The k-NN classifier, with no prior feature selection, also attain comparable performance to those methods reported in [9]. However, in general k-NN classifiers are susceptible to the *curse of dimensionality*, a situation in which many irrelevant attributes are present. This explains its relatively inferior performance when compared to SVM.

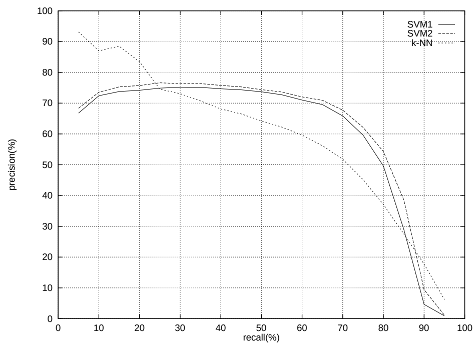

Figure 2: Micro-averaged recall-precision curves. Refer to the main text for the labels.

Figure 3 shows the distribution of the number of support vectors before performing LSI[5]. As can be seen, all of them are a very small fraction of the size of the training corpus. This leads to significant computational saving as compared to performing LSI on the whole corpus.

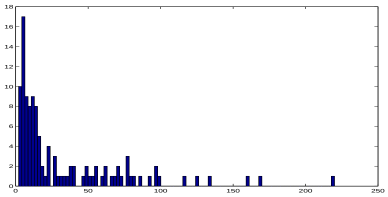

Figure 3: Histogram showing the distribution of the number of support vectors.

We also illustrate an example in which SVM can be used to detect mis-labeled documents. In examining the topic fuel, documents 8893 and 18274 have exceedingly large Lagrangian multipliers. Document 8893 (Figure 4) has the topic label crude assigned, whereas document 18274 (Figure 5) has the labels crude, gas and fuel. However, as can be seen in Figures 4 and 5, the contents of the two documents are basically the same, and thus the missing labels for document 8893 are likely to be caused by mistake.

EIA SAYS DISTILLATE, GAS STOCKS OFF IN WEEK

WASHINGTON, March 11 - Distillate fuel stocks held in primary storage fell by 8.8 mln barrels in the week ended March six to 119.6 mln barrels, the Energy Information Administration (EIA) said.

In its weekly petroleum status report, the Department of Energy agency said gasoline stocks were off 500,000 barrels in the week to 251.0 mln barrels and refinery crude oil stocks fell 1.2 mln barrels to 331.8 mln.

The EIA said residual fuel stocks fell 1.5 mln barrels to 36.4 mln barrels and crude oil stocks in the Strategic Petroleum Reserve (SPR) rose 200,000 barrels to 516.7 mln.

The total of all crude, refined product and SPR stocks fell 10.3 mln barrels to 1,564.8, it said.

Figure 4: Reuters document 8893.

EIA SAYS DISTILLATE, GAS STOCKS OFF IN WEEK

WASHINGTON, March 4 - Distillate fuel stocks held in primary storage fell by 3.4 mln barrels in the week ended Feb 27 to 128.4 mln barrels, the Energy Information Administration (EIA) said.

In its weekly petroleum status report, the Department of Energy agency said gasoline stocks were off 100,000 barrels in the week to 251.5 mln barrels and refinery crude oil stocks were up 3.2 mln barrels to 333.0 mln.

The EIA said residual fuel stocks fell 2.2 mln barrels to 37.9 mln barrels and crude oil stocks in the Strategic Petroleum Reserve (SPR) were up 700,000 barrels to 516.5 mln.

The total of all crude, refined product and SPR stocks were unchanged at 1,575.1 mln barrels, it said.

Figure 5: Reuters document 18274.

6. Conclusion

The integration of dimension reduction and classification by SVM has a number of advantages in applications to text categorization. Firstly, unlike other machine learning techniques that have been used, it can easily incorporate new

[4]In micro-averaging, the performance tables for each of the topics are added and the overall recall and precision are computed. In macro-averaging, performance measures are computed separately for each topic, and the mean of the resulting performance values is taken.

[5]The graph does not include statistics for the categories acq and earn, which contain over 1000 support vectors (though this is still small compared to the size of the whole training corpus). We do not perform LSI on these two categories.

Table 1: List of topics that are not separable in the $tf \cdot idf$ space.

bop	bypass	cbond	corn	corp-news	crude	dlr	dmk	ebond	gas	gbond
grain	hog	interest	iran	iraq	livestock	loan	maize	money-fx	money-supply	
ship	soybean	stg	sugar	tbill	tbond	trade	wheat	wool	yen	

Table 2: Break-even points for SVM, k-NN and five other categorization methods as reported in [9].

	break-even points micro(%)	break-even points macro(%)
SVM2	68.6	62.5
SVM1	67.6	58.1
k-NN	59.8	51.7
proportional assignment (unscaled)	65	43
proportional assignment (scaled)	62	41
proportional thresholding (unscaled)	59	46
proportional thresholding (scaled)	61	44
k-per-doc	55	42

documents. Moreover, dimension reduction, which can be computationally intensive for large training corpus and may even result in a loss of information, is no longer imperative. Thus, SVM adapts efficiently in dynamic environments that require frequent additions to the document collection. Besides, although the text categorization problem is formulated as a series of dichotomous classification problems, SVM outputs from each of these problems are comparable across categories. On the computational side, QP codes are reliable and robust, and are more efficient than nonlinear optimization codes as required by neural networks.

We believe that SVM is very useful to the problem of text categorization, and to information retrieval at large. The experiment here serves as a promising start for the use SVM in this domain. Of course, the use of SVM does not preclude the use of dimension reduction, and possibly better results can be obtained by using them as pre-processing. Moreover, to further improve the text categorization performance, one may also experiment with other document representations, or to organize the document set into a hierarchical structure [8], or to utilize additional structural information in the newswires by, for example, giving more emphasis to words occurring in the headline.

References

[1] C. Apte and F. Damerau. Automated learning of decision rules for text categorization. *ACM Transcations on Information Systems*, 12(3):233–251, July 1994.

[2] C.J.C. Burges. Simplified support vector decision rules. In *International Conference on Machine Learning*, 1996.

[3] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.

[4] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, September 1990.

[5] I. Guyon, N. Matic, and V. Vapnik. Discovering informative patterns and data cleaning. In U.M. Fayyad, G.P. Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 181–204. MIT, 1996.

[6] P.J. Hayes, P.M. Anderson, I.B. Nirenburg, and L.M. Schmandt. TCS: A shell for context-based text categorization. In *Sixth Conference on Artificial Intelligence Applications*, May 1990.

[7] T. Joachims. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. Technical Report CMU-CS-96-118, Carnegie Mellon University, 1996.

[8] D. Koller and M. Sahami. Hierarchically classifying documents using very few words. In *Proceedings of the 14th International Conference on Machine Learning*, Nashville, Tennessee, July 1997.

[9] D.D. Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In *SIGIR Forum*, pages 37–50, Copenhagen, Denmark, June 1992.

[10] D.D. Lewis, R.E. Schapire, J.P. Callan, and R. Papka. Training algorithms for linear text classifiers. In *Proceedings of the Annual International ACM-SIGIR Conference*, pages 298–306, Zurich, August 1996.

[11] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: an application to face detection. In *Proceedings of Computer Vision and Pattern Recognition*, Puerto Rico, June 1997.

[12] B. Scholkopf, K. Sung, C. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik. Comparing support vector machines with Gaussian kernels to radial basis function classifiers. A.I. Memo 1599, MIT, December 1996.

[13] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, 1995.

[14] E. Wiener, J.O. Perdersen, and A.S. Weigend. A neural network approach to topic spotting. In *Proceedings of Fourth Annual Symposium on Document Analysis and Information Retrieval*, pages 317–332, Las Vegas, April 1995.

[15] Y. Yang. Expert network: Effective and efficient learning from human decisions in text categorization and retrieval. In *Proceedings of the Annual International ACM-SIGIR Conference*, pages 13–22, Dublin, Ireland, July 1994.

[16] Y. Yang. An evaluation of statistical approaches to text categorization. CMU-CS 97-127, Carnegie Mellon University, 1997.