Feature Space Restructuring for SVMs with Application to Text Categorization

Hiroya Takamura and Yuji Matsumoto
Department of Information Technology
Nara Institute of Science and Technology
8516-9, Takayama, Ikoma, 630-0101 Japan
{hiroya-t,matsu}@is.aist-nara.ac.jp

Abstract

In this paper, we propose a new method of text categorization based on feature space restructuring for SVMs. In our method, independent components of document vectors are extracted using ICA and concatenated with the original vectors. This restructuring makes it possible for SVMs to focus on the latent semantic space without losing information given by the original feature space. Using this method, we achieved high performance in text categorization both with small number and large numbers of labeled data.

1 Introduction

The task of text categorization has been extensively studied in Natural Language Processing. Most successful works rely on a large number of classified data. However, it is hard to collect classified data, so considering real applications, text categorization must be realized even with a small number of labeled data. Several methods to realize it have been proposed so far (Nigam et al, 2000), but they need to be further developed. For that purpose, we have to take advantage of invaluable information offered by the property of unlabeled data. In this paper, we propose a new categorization method based on Support Vector Machines (SVMs) (Vapnik, 1995) and Independent Component Analysis (ICA) (Herault and Jutten, 1986; Bell and Sejnowski, 1995). SVM is gaining popularity as a classifier with high performance, and ICA is one of the most prospective algorithms in the field of signal processing, which extracts independent components from mixed signals.

SVM has been applied in many applications such as Image Processing and Natural Language Processing. The idea to apply SVM for text categorization was first introduced in (Joachims, 1998). However, when the number of labeled data are small, SVM often fails to produce a good result, although several efforts against this problem have been made. There are two strategies for improving performance in the case of a limited number of data. One is to modify the learning algorithm itself (Joachims, 1999a; Glenn and Mangasarian, 2001), and the other is to process training data (Weston et al, 2000), including the selection of features. In this paper, we focus on the latter, especially on *feature space restructuring*. For processing training data, Principal Component Analysis (PCA) is often adopted in classifiers such as k-Nearest Neighbor method (Mitchell, 1997). But the conventional dimension-reduction methods fail for SVM as shown by experiments in Section 6. Unlike the conventional ones, our approach uses the components obtained with ICA to augment the dimension of the feature space.

ICA is built on the assumptions that the sources are independent of each other and that the signals observed at multiple-points are linear mixtures of the sources. While the theoretical aspects of ICA are being studied, its possibility to applications is often pointed out as in (Bell and Sejnowski, 1997). The idea of using ICA for text clustering is adopted in several works such as in (Isbell and Viola, 1998). In those works, vector representation model is adopted (i.e. each text is represented as a vector with the word-frequencies as the elements). It is reported however that the independent components do not always correspond to the desired classes, but represent some kind of characteristics of texts (Kolenda et al, 2000). In (Kaban and Girolami, 2000), they showed that the number of potential components were larger than that of human-annotated classes. These facts imply that it is not easy to apply ICA directly

for text classification.

Taking these observations into consideration, we take the following strategy: first we perform ICA on input document vectors, and second, create the *restructured* information by concatenating the reduced vectors (i.e. the values of the independent components) and the original feature vectors.

PCA is an alternative restructuring method. So we conducted experiments using SVM with various input vectors: original feature vectors, reduced feature vectors and restructured feature vectors (reduction and restructuring are performed by PCA and ICA). For comparison, we conducted experiments using Transductive SVM (TSVM) (Joachims, 1999a) as well, which is designed for the case of a small number of labeled data.

Using the proposed method (SVM with ICA), we obtain better results than ordinary SVM and TSVM, with both small and large numbers of labeled data.

2 Support Vector Machines

2.1 Brief Overview of Support Vector Machines

Support Vector Machine (SVM) is one of the large-margin classifiers (Smola et al, 2000). Given a set of pairs,

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_n, y_n) \quad (1)$$
$$\forall i,\ \mathbf{x}_i \in \mathbf{R}^d, y_i \in \{-1, 1\}$$

of a feature vector and a label, SVM constructs a separating hyperplane with the largest margin (the distance between the hyperplane and the vectors, see Figure 1):

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b. \quad (2)$$

Finding the largest margin is equivalent to minimizing the norm $\|\mathbf{w}\|$, which is expressed as:

$$\begin{aligned} \min . \quad & \tfrac{1}{2}\|\mathbf{w}\|^2, \\ \text{s.t.} \quad & \forall i,\ y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geq 0. \end{aligned} \quad (3)$$

This is realized by solving the quadratic program (dual problem of (3)):

$$\begin{aligned} \max . \quad & \textstyle\sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \\ \text{s.t.} \quad & \textstyle\sum_i \alpha_i y_i = 0, \\ & \forall i,\ \alpha_i \geq 0, \end{aligned} \quad (4)$$

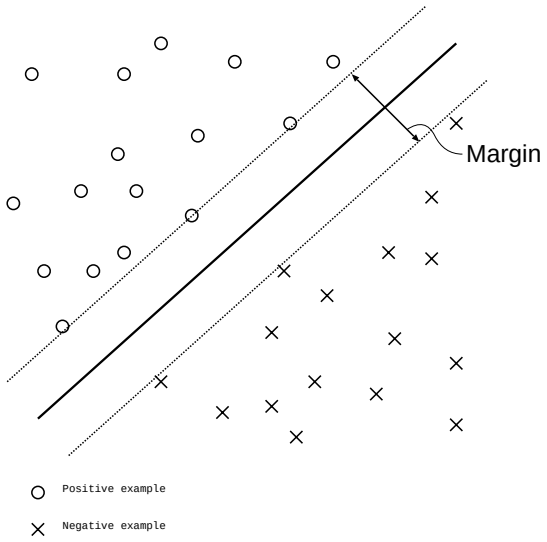

Figure 1: Support Vector Machine
(the solid line corresponds to the optimal hyperplane).

where α_i's are Lagrange multipliers. Using the α_i's that maximize (4), $\mathbf{w}$ is expressed as

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i. \quad (5)$$

Substituting (5) into (2), we obtain

$$f(\mathbf{x}) = \sum_i \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b. \quad (6)$$

Unlabeled data are classified according to the signs of (6).

2.2 Kernel Method

SVM is a linear classifier and its separating ability is limited. To compensate this limitation, Kernel Method is usually combined with SVM (Vapnik, 1995).

In Kernel Method, the dot-product in (4) and (6) is replaced by a more general inner-product $K(\mathbf{x}_i, \mathbf{x})$, called the kernel function. Polynomial kernel $(\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d$ $(d \in \mathbf{N}_+)$ and RBF kernel $\exp\{-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma^2\}$ are often used. Using kernel method means that feature vectors are mapped into a (higher dimensional) Hilbert space and linearly separated there. This mapping structure makes non-linear separation possible, although SVM is basically a linear classifier.

Another advantage of kernel method is that although it deals with a high dimensional (possibly infinite) Hilbert space, there is no need to compute high dimensional vectors explicitly. Only the general inner-products of two vectors are needed. This leads to a relatively small computational overhead.

2.3 Transductive SVMs

The Transductive Support Vector Machine (TSVM) is introduced in (Joachims, 1999a), which is one realization of *transductive learning* in (Vapnik, 1995). It is designed for the classification with a small number of labeled data. Its algorithm is approximately as follows:

1. construct a hyperplane using labeled data in the same way as the ordinary SVMs.
2. classify the unlabeled (test) data according to the current hyperplane.
3. select the pair of a positively classified sample and a negatively classified sample that are nearest to the hyperplane.
4. exchange the labels of those samples, if the margin gets larger by exchanging them.
5. terminate if a stopping-criterion is satisfied. Otherwise, go back to step 2.

This is one way to search for the largest margin, permitting the relabeling of test data that have already been labeled by the classifier in the previous iteration.

3 Independent Component Analysis

Independent Component Analysis (ICA) is a method by which source signals are extracted from mixed signals. It is based on the assumptions that the sources $\mathbf{s} \in \mathbf{R}^m$ are statistically independent of each other and that the observed signals $\mathbf{x} \in \mathbf{R}^n$ are linear mixtures of the sources:

$$\mathbf{x} = A\mathbf{s}. \tag{7}$$

Here the matrix A is called a *mixing matrix*. We observe $\mathbf{x}$ as a time series and estimate both A and $\mathbf{s} = (s_1, \cdots, s_m)$. So our purpose here is to find a demixing matrix W such that $s_1, \cdots, s_m$ are as independent of each other as possible:

$$\mathbf{s} = W\mathbf{x}. \tag{8}$$

The computation proceeds by way of descent learning with an objective function indicating independence. There are several criteria of independence and their learning rules, among which we take here Infomax approach (Bell and Sejnowski, 1995), but with natural gradient (Amari, 1998). Its learning rule is

$$\delta W = (I + (I - 2g(W\mathbf{x}))(W\mathbf{x})^t)W, \quad where,\ g(u) = 1/(1+\exp(-u)). \tag{9}$$

4 Text Categorization Enhanced with Feature Space Restructuring

As in most previous works, we adopt Vector Space Model (Salton and McGill, 1983) for representing documents. In this framework, each document $\mathbf{d}$ is represented as a vector $(f_1, \cdots, f_d)$ with word-frequencies as its elements.

4.1 Feature Space Restructuring

First we reduce the dimension of document vectors using PCA or ICA. As for PCA, we follow the previous work described in , e.g., (Deerwester et al, 1990). In (Isbell and Viola, 1998), they use ICA for dimension reduction and obtain a good result in Information Retrieval. At the first step of our method, where the reduced vectors are obtained, we follow their method. In this framework, each document $\mathbf{d}$ is considered as a linear mixture of sources $\mathbf{s}$ representing *topics*. Each word plays a role of "microphone" and receives a word-frequency in the document as a mixed signal at each time unit. This formulation is represented by the equation:

$$\mathbf{d} = A\mathbf{s}, \tag{10}$$

where A is a mixing matrix. Although both A and $\mathbf{s}$ are unknown, they can be obtained using the independence assumption. The source signals $\mathbf{s}$ are considered as a reduced expression of this document. In the case of PCA, the restructuring is processed in the same way. The only difference is that independent components correspond to principal components for the PCA case.

After computing a reduced vector $\mathbf{s}$ with PCA or ICA, we concatenate the original vector $\mathbf{d}$ and the reduced vector $\mathbf{s}$:

$$\hat{\mathbf{d}} = \begin{bmatrix} \mathbf{d} \\ \mathbf{s} \end{bmatrix}. \tag{11}$$

This transformation means that we do not rely only on the reduced information, but make use of both the reduced and the original information, that is, the *restructured information*.

4.2 Text Categorization

Regarding $\hat{\mathbf{d}}$ as the input feature vector of a document, we use SVM for categorization.

Since SVMs are binary classifiers themselves, so we take here the *one-versus-rest* method to apply them for multi-class classification tasks.

5 Theoretical Perspective

5.1 Validation as a Kernel Function

The proposed feature restructuring method can be considered as the use of a certain kernel for the pre-restructured feature space. We give an explanation for the linear case. Given two vectors, $\mathbf{d}_1$ and $\mathbf{d}_2$, the kernel function K in the restructured space is expressed as,

$$\begin{aligned} K(\hat{\mathbf{d}_1}, \hat{\mathbf{d}_2}) &= \hat{\mathbf{d}}_1^t \hat{\mathbf{d}}_2 \\ &= \mathbf{d}_1^t \mathbf{d}_2 + \mathbf{s}_1^t \mathbf{s}_2 \\ &= \mathbf{d}_1^t \mathbf{d}_2 + \mathbf{d}_1^t A^t A \mathbf{d}_2. \end{aligned} \quad (12)$$

Considering the fact that each of two terms above is a kernel and that the sum of two kernels is also a kernel (Vapnik, 1995), the proposed restructuring is equivalent to using a certain kernel in the pre-restructured space.

5.2 Interpretation of Feature Space Restructuring

The expression (12) shows that weights are put on the latent semantic indices determined by ICA and PCA respectively. The criterion of meaningfulness depends on which of ICA and PCA is used. Note that weighting is different from reducing. In the dimension-reduction methods, only the *latent semantic space* is considered, but in our method, the original feature space still directly influences the classification result.

This property of our method makes it possible to focus on the information given by the latent semantic space, without losing information given by the original feature space.

In text categorization, classes to be predicted are sometimes characterized by local information such as the occurrence of a certain word, but sometimes dominated by global information such as the total frequency of a certain group of words. Considering this situation and the above property of our method, it is not surprising that out method gives a good result.

6 Experiments

To evaluate the proposed method, we conducted several experiments.

The data used here is the Reuters-21578 dataset. The most frequent 6 categories are extracted from the training-set of the corpus. This leaves 4872 documents (see Table 1). Some part of them is used as training data and the rest is used as test data. Only the words occurring more than twice are used. Both stemming and stop-word removal are performed. For computation, we used SVM-*light* (Joachims, 1999b).

We conducted two kinds of experiments. The first one focuses on evaluating the performance of the proposed method for each category, with a fixed number of labeled data (Section 6.1). The second one is conducted to show that the proposed method gives a good result also when the number of labeled data increases (Section 6.2).

The results are evaluated by F-measures. To evaluate the performance across categories, we computed Micro-average and Macro-average (Yang, 1999) of F-measures. Micro-average is obtained by first computing precision and recall for all categories and then using them to compute the F-measure. Macro-average is computed by first calculating F-measures for each category and then averaging them. Micro-average tends to be dominated by large-sized categories, and Macro-average by small-sized ones.

The kernel function used here is a linear kernel. The number of independent or principal components extracted by ICA or PCA is set to 50.

6.1 Performance with a Fixed Number of Data

In this experiment, we treated 100, 500, 1000 and 2000 samples as labeled respectively and kept the other 4772, 4372, 3872 and 2872 samples unlabeled. The experiment was conducted 10 times for each sample-size repeatedly with randomly selected labeled samples and their average values are computed. The result is shown in Tables 2, 3, 4 and 5. In the row of "Method",

Table 1: Documents used in Experiments

category	number of documents
earn	2673
acq	1435
trade	225
crude	223
money-fx	176
interest	140

combinations of restructuring methods are written. "Original" means the data of original document vectors. "PCA" and "ICA" mean the data of only reduced vectors, respectively. "Original+PCA" and "Original+ICA" are the restructured data explained in Section 4.

The proposed method yields a high F-measure in all the categories for 1000 and 2000 labeled data and in most categories for 100 and 500 labeled data. The last two rows of Tables 2, 3, 4 and 5 show that both Micro-average and Macro-average are the highest for the proposed method. This means that the proposed method performs well both for large-sized categories (e.g., earn) and small-sized categories (e.g., interest), regardless with the number of labeled data.

6.2 Performance for the Increase of the Labeled Data

To investigate how each method behaves when the number of labeled data increases, we conducted this experiment. The number of labeled data ranges from 100 to 2000. The results are shown in Figure 2 and Figure 3. "PCA" gives a good score only with a small number of data and "Original" gives a good score only with a large number of data. In contrast to them, the proposed method produces high performance both with small and large numbers of data.

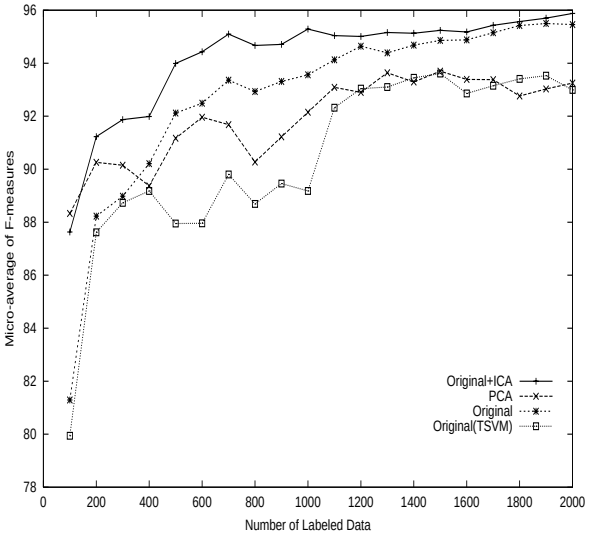

Figure 2: Micro-average

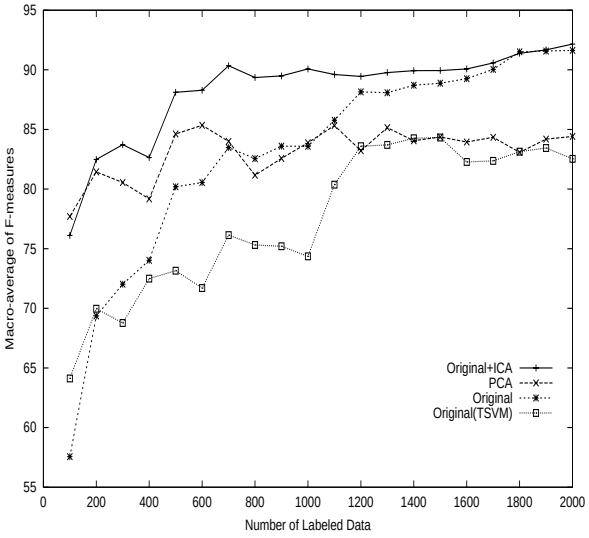

Figure 3: Macro-average

7 Conclusions

We proposed a new method of feature space restructuring for SVM. In our method, independent components are extracted using ICA and concatenated with the original vectors. Using this new vectors in the restructured space, we achieved high performance both with small and large numbers of labeled data.

The proposed method can be applied also to other machine learning algorithms provided that they are robust against noise and can handle a high-dimensional feature space. From this point of view, it is expected that the proposed method is useful for *kernel-based* methods, to which SVM belongs.

As a future work, we need to find a way to decide the number of independent components to be extracted. In this paper, we set the number to 50 in an ad-hoc way. However, the appropriate number must be predicted based on a theo-

Table 2: F-Measures (100 Labeled Data)

Method	Original	Original(TSVM)	PCA	ICA	Original+PCA	Original+ICA
earn	92.96	84.00	91.13	86.60	**92.97**	92.88
acq	85.88	81.42	85.67	80.86	85.91	**87.48**
trade	36.52	65.59	**72.41**	72.28	36.68	70.73
crude	65.69	70.90	79.75	80.67	65.93	**82.87**
money-fx	32.46	45.01	52.69	**54.37**	32.47	48.62
interest	51.30	52.69	64.44	63.48	51.30	**64.84**
microaverage	83.63	79.48	85.98	82.14	83.66	**87.40**
macroaverage	60.80	66.60	74.34	73.04	60.87	**74.56**

Table 3: F-Measures (500 Labeled Data)

Method	Original	Original(TSVM)	PCA	ICA	Original+PCA	Original+ICA
earn	96.49	93.97	94.38	93.45	96.49	**96.70**
acq	93.23	91.57	89.18	87.45	93.22	**93.41**
trade	86.31	80.81	87.42	86.58	86.37	**91.70**
crude	83.33	79.78	81.36	78.28	83.43	**87.12**
money-fx	62.94	64.88	72.83	73.45	63.17	**73.99**
interest	59.31	52.02	**73.37**	72.18	59.31	70.41
microaverage	92.17	89.75	90.54	89.33	92.19	**93.48**
macroaverage	80.26	77.17	83.09	81.89	80.34	**85.55**

Table 4: F-Measures (1000 Labeled Data)

Method	Original	Original(TSVM)	PCA	ICA	Original+PCA	Original+ICA
earn	97.15	95.52	96.07	95.53	97.15	**97.26**
acq	94.60	93.77	92.18	91.44	94.60	**94.84**
trade	91.19	86.11	87.13	86.87	91.23	**93.25**
crude	87.99	80.03	80.93	78.75	87.99	**89.41**
money-fx	73.68	68.85	72.96	72.68	69.96	**80.99**
interest	75.34	57.26	72.83	68.25	75.34	**79.27**
microaverage	94.23	91.79	92.31	91.54	94.09	**94.90**
macroaverage	86.65	80.25	83.68	82.25	86.04	**89.17**

Table 5: F-Measures (2000 Labeled Data)

Method	Original	Original(TSVM)	PCA	ICA	Original+PCA	Original+ICA
earn	97.48	95.92	97.18	97.12	97.48	**97.55**
acq	95.39	94.39	94.78	94.80	95.39	**95.65**
trade	93.81	86.33	88.61	85.28	93.81	**95.90**
crude	89.88	80.35	82.63	78.56	89.88	**90.25**
money-fx	77.44	70.60	74.84	70.69	77.49	**81.56**
interest	82.71	62.15	73.99	68.46	82.76	**83.02**
microaverage	95.19	92.43	93.93	93.26	95.20	**95.58**
macroaverage	89.45	81.62	85.33	82.48	89.47	**90.65**

retical reason. Toward this problem, theories of model selection such as Minimum Description Length (Rissanen, 1987) or Akaike Information Criterion (Akaike, 1974) could be a good theoretical basis.

As explained in Section 4, two terms $\mathbf{d}_1^t\mathbf{d}_2$ and $\mathbf{d}_1^t A^t A\mathbf{d}_2$ are simply concatenated in our method. But either of these terms can be multiplied with a certain constant. This means that either of the original space and the Latent Semantic Space can be weighted. Searching for the best weighting scheme is one of the future works.

Acknowledgment

We would like to thank Thomas Kolenda (Technical University of Denmark) for helping us with the code.

References

Akaike, H. 1974. A New Look at the Statistical Model Identification. *IEEE Trans. Autom. Control*, vol. AC-19, pp. 716–723.

Amari, S. 1998. Natural Gradient Works Efficiently in Learning. *Neural Computation*, vol. 10-2, pp. 251–276.

Bell, A. J. and Sejnowski, T. J. 1995. An Information Maximization Approach to Blind Separation and Blind Deconvolution. *Neural Computation*, 7, 1129–1159.

Bell, A. J. and Sejnowski, T. J. 1997. The 'Independent Components' of Natural Scenes are Edge Filters. *Vision Research*, 37(23), pp. 3327–3338.

Deerwester, S., Dumais, T., Landauer, T., Furnas, W. and Harshman, A. 1990. Indexing by Latent Semantic Analysis. *Journal of the Society for Information Science*, 41(6), pp. 391–497.

Glenn, F. and Mangasarian, O. 2001. Semi-Supervised Support Vector Machines for Unlabeled Data Classification. *Optimization Methods and Software*, pp. 1–14.

Herault, J. and Jutten, J. 1986. Space or Time Adaptive Signal Processing by Neural Network Models. *Neural networks for computing: AIP conference proceedings* 151, pp. 206–211.

Isbell, C. and Viola. P. 1998. Restructuring Sparse High Dimensional Data for Effective Retrieval. *Advances in Neural Information Processing Systems*, volume 11.

Joachims, T. 1998. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. *Proceedings of the European Conference on Machine Learning*, pp. 137–142.

Joachims, T. 1999a. Transductive Inference for Text Classification using Support Vector Machines. *Machine Learning – Proc. 16th Int'l Conf. (ICML '99)*, pp. 200–209.

Joachims, T. 1999b. Making large-Scale SVM Learning Practical. *Advances in Kernel Methods - Support Vector Learning*, pp. 169–184.

Kaban, A. and Girolami, M. 2000. Unsupervised Topic Separation and Keyword Identification in Document Collections: A Projection Approach *Technical Report.*

Kolenda, T, Hansen, L., K. and Sigurdsson, S. 2000. Indepedent Components in Text . *Advances in Independent Component Analysis*, Springer-Verlag, pp. 235–256.

Mitchell, T. 1997. *Machine Learning*, McGraw Hill.

Nigam, K., McCallum, A., Thrun, S. and Mitchell, T. 2000. Text Classification from Labeled and Unlabeled Documents using EM. *Machine Learning*, 39(2/3). pp. 103–134.

Rissanen, J. 1987. Stochastic Complexity. *Journal of Royal Statistical Society*, Series B, 49(3), pp. 223–239.

Salton, G. and McGill, M. J. 1983. *Introduction to Modern Information Retrieval.* McGraw-Hill Book Company, New York.

Smola, A., Bartlett, P., Schölkopf, B. and Schurmans, D. 2000. *Advances in Large Margin Classifiers.* MIT Press

Vapnik, V. 1995. *The Nature of Statistical Learning Theory.* Springer.

Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T. and Vapnik, V. 2000. Feature Selection for SVMs. *In Advances in Neural Information Processing Systems*, volume 13.

Yang, Y. An Evaluation of Statistical Approaches to Text Categorization. *Information Retrieval*, volume 1, 1-2, pp. 69–90.