Available online at www.sciencedirect.com
SCIENCE DIRECT®
Pattern Recognition Letters 25 (2004) 1051–1057
Pattern Recognition Letters
www.elsevier.com/locate/patrec

SVM-based feature selection of latent semantic features

K. Shima *, M. Todoriki, A. Suzuki

Department of Quantum Engineering and Systems Science, University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 1138656, Japan

Received 16 July 2003; received in revised form 28 November 2003
Available online 15 April 2004

Abstract

Latent Semantic Indexing (LSI) is an effective method to extract features that captures underlying latent semantic structure in the word usage across documents. However, subspace selected by this method may not be the most appropriate one to classify documents, since it orders extracted features according to their variances, not the classification power. We propose to apply feature ordering method based on support vector machines in order to select LSI-features that is suited for classification. Experimental results suggest that the method improves classification performance with considerably more compact representation.

Keywords: Support Vector Machines; Text categorization; Latent Semantic Indexing; Feature selection

1. Introduction

Text categorization is a task of classifying text documents into predefined set of categories based on their content. Documents are usually represented using a vector space model (Salton and McGill, 1983), where each word appeared in documents are treated as features. However, using this simple representation, documents that are actually similar in their content may not be considered as similar if they do not share same words. Latent Semantic Indexing (LSI) (Deerwester et al., 1990) tries to overcome such problem using a statistical technique. It applies Singular Value Decomposition (SVD) to term-document matrix, and extracts linear combinations of original features based on word co-occurrence. Lower-dimensional representation using a subset of extracted features is known to capture underlying latent semantic structure in the word usage across documents. LSI has proven to be effective in information retrieval (Deerwester et al., 1990; Dumais, 1995), document clustering (Schütze and Silverstein, 1997).

One of the crucial issue in applying LSI is the large computational cost. Even if a term-document matrix is sparse, truncated term-document matrix is no longer sparse and requires a large amount of computation time for training and testing. LSI orders features according to their variances, and selects a subspace spanned by features having largest variances. However, for classification tasks,

* Corresponding author. Tel.: +81-35-841-8501; fax: +81-33-818-3455.
E-mail address: shima@lyman.q.t.u-tokyo.ac.jp (K. Shima).

0167-8655/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2004.03.002

the most important property that must be preserved is the classification power. There is no theoretical justification that features having largest variance is informative for classification. For example, in the face recognition domain, it has been empirically shown that superior performance is achieved when features having three largest eigenvalues are not used (Pentland et al., 1993). It can be expected that more compact LSI subspace can be obtained by selecting discriminative LSI features.

Support Vector Machine (SVM) (Vapnik, 1998; Cortes and Vapnik, 1995) is a state-of-the-art classification algorithm that is shown to be particularly successful in text categorization (Joachims, 1998; Dumais et al., 1998). Although SVM can deal with high-dimensional data effectively, it has been reported that classification performance of SVM can be improved by applying LSI as a preprocessing step (Kwok, 1998). Recently, it was recognized that SVM can be used for feature selection (Guyon et al., 2002; Brank et al., 2002). We propose to apply SVM-based feature selection method in order to select LSI-features that is suited for classification. We investigate the effectiveness of the proposed method by testing on a real-world text dataset. Result is compared against the normal LSI.

2. Latent Semantic Indexing

Latent Semantic Indexing (Deerwester et al., 1990) tries to find a lower-dimensional subspace that takes into account association between terms and documents. The method applies SVD to estimate such subspace. Let $\boldsymbol{X}$ be a term-document matrix represented by a vector space model, $\boldsymbol{X}$ can be decomposed into the product of three matrices:

$$\boldsymbol{X} = \boldsymbol{TSD}^{\mathrm{T}}$$

where $\boldsymbol{T}, \boldsymbol{D}$ are orthonormal matrices and $\boldsymbol{S}$ is a diagonal matrix whose diagonal components corresponds to singular values ordered in decreasing order. LSI approximates term-document matrix using bases that correspond to largest singular values. Let $\hat{\boldsymbol{S}}$ be a matrix whose all but k-largest diagonal elements in $\boldsymbol{S}$ were set to zero, term-document matrix can be approximated as

$$\hat{\boldsymbol{X}} = \boldsymbol{T}\hat{\boldsymbol{S}}\boldsymbol{D}^{\mathrm{T}}$$

Resulted matrix $\hat{\boldsymbol{X}}$ is an optimal approximation of $\boldsymbol{X}$ in terms of mean-square error. LSI assumes that there is an underlying latent semantic structure in the word usage across documents, and the basic idea is that such structure can be uncovered by dropping small singular values in $\boldsymbol{S}$. It has been empirically shown that LSI improves information retrieval performance (Deerwester et al., 1990; Dumais, 1995), and its effectiveness has been explained theoretically by means of the Bayesian regression model (Story, 1996).

3. Support Vector Machines

Support Vector Machine (Vapnik, 1998; Cortes and Vapnik, 1995) is a state-of-the-art classification algorithm that is known to be successfull in a wide variety of applications. High generalization ability of the method makes it particularly suited for high dimensional data such as text. Indeed, it has been shown that SVM outperformed most of the other classification algorithms in text categorization tasks (Joachims, 1998; Dumais et al., 1998).

Since many of the text datasets are linearly separable (Joachims, 1998), we will only consider linear SVMs. The basic idea of SVM is to maximize the margin between classes, which can be formulated as the following convex quadratic programming problem:

maximize

$$\sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j$$

subject to

$$0 \leqslant \alpha_i \leqslant C (i = 1, \ldots, m), \sum_{i=1}^{m} \alpha_i y_i = 0$$

where $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\}$ is a training set in $\boldsymbol{R}^d$ space, $\{y_1, \ldots, y_m\} \in \{-1, 1\}$ is class label data, and α_i ($\geqslant 0$) are Lagrange multipliers. C is a parameter that assigns penalty cost to misclassification of samples. By solving the above optimization prob-

lem, the form of decision function can be derived as

$$f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b$$

where

$$\boldsymbol{w} = \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i$$

and b is a bias term. Only vectors corresponding to nonzero α_i contribute to decision function, and are called support vectors.

4. SVM-based feature selection of latent semantic features

High generalization ability of SVM is based on the idea of maximizing the margin. Inverse-square of margin is given by

$$\|\boldsymbol{w}\|^2 = \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j$$

Feature selection methods that use the above quantity as a criterion have been proposed by several authors (Guyon et al., 2002; Brank et al., 2002). For linear SVMs, it is possible to decompose the above quantity into sum of terms corresponding to original features:

$$\|\boldsymbol{w}\|^2 = \sum_{k=1}^{d} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j x_{ki} x_{kj}$$

where x_{ki} is the kth component of $\boldsymbol{x}_i$. Therefore, contribution of kth feature to the inverse-square-margin can be given by

$$w_k^2 = \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j x_{ki} x_{kj}$$

Importance of a features were evaluated according to their values of w_k^2, and features having w_k^2 close to zero can be discarded without deteriorating classification performance. Guyon et al. (2002) employed iterative procedure: starting with all features, a feature having smallest w_k^2 was repeatedly set aside and w_k^2 was evaluated using parameter values of a classifier trained at each iteration. However, we took a simpler approach. Classifier was trained only once and w_k^2 was evaluated using parameter values computed from that initial classifier.

When using this method in the LSI-framework, $\|\boldsymbol{w}\|^2$ needs to be decomposed into terms corresponding to LSI features. Let $\boldsymbol{t}_k$ be kth left singular vector of $\boldsymbol{T}$, contribution of $\boldsymbol{t}_k$ to the inverse-square-margin can be given by

$$(\boldsymbol{t}_k^{\mathrm{T}}\boldsymbol{w})^2 = \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (\boldsymbol{t}_k^{\mathrm{T}}\boldsymbol{x}_i)(\boldsymbol{t}_k^{\mathrm{T}}\boldsymbol{x}_j) \quad (1)$$

In order to consider both classification power and data representation, we used a harmonic mean of $(\boldsymbol{t}_k^{\mathrm{T}}\boldsymbol{w})^2$ and variance as a feature evaluation measure. Let v_k be a proportion of variance to the total sum of variances, and u_k be a proportion of $(\boldsymbol{t}_k^{\mathrm{T}}\boldsymbol{w})^2$ to $\|\boldsymbol{w}\|^2$, their harmonic mean q_k can be given by

$$q_k = \frac{2v_k u_k}{v_k + u_k}$$

q_k takes large values only if both v_k and u_k takes large values. We sort columns of $\boldsymbol{T}, \boldsymbol{D}$ and the diagonal components of $\boldsymbol{S}$ according to their values of q_k in decreasing order. Let $\boldsymbol{T}', \boldsymbol{D}', \boldsymbol{S}'$ be such matrices after sorting, alternative LSI representation $\hat{\boldsymbol{X}}'$ can be derived as

$$\hat{\boldsymbol{X}}' = \boldsymbol{T}'\hat{\boldsymbol{S}}'\boldsymbol{D}'^{\mathrm{T}}$$

where $\hat{\boldsymbol{S}}'$ is a matrix whose all but first k-diagonal elements in $\boldsymbol{S}'$ were set to zero. We will refer to this form of LSI representation as SVM-LSI throughout this paper.

5. Experimental results

5.1. Experimental setup

In order to evaluate the performance of the proposed method, we conducted an experiment on Reuters-21578 dataset. [1] It contains 21,578 news articles taken from Reuters newswire in 1987. Articles were manually assigned to categories such

[1] Reuters-21578 Corpus available from: http://www.daviddlewis.com/resources/testcollections/reuters21578/. Preprocessed version obtained from: http://www.cs.technion.ac.il/~ronb/thesis.html

Table 1
Top 10 Reuters category and the number of articles contained in the training, test, and the whole set

Class #	Name	Train	Test	Total
1	earn	2709	1044	3753
2	acq	1488	643	2131
3	money-fx	460	141	601
4	grain	394	134	528
5	crude	349	161	510
6	trade	337	112	449
7	interest	289	100	389
8	ship	191	85	276
9	wheat	198	66	264
10	corn	159	48	207

as "earn (earnings)", "acq (corporate acquisitions)", "grain", and "interest", and multiple category assignments were allowed. We used the popular "ModApté" split to divide the dataset into a training and a test set. This split contains 7063 training and 2742 test set, all of which at least one category is assigned. Out of 114 possible categories, we focused on classifying top 10 frequent categories. Table 1 shows the names of such categories and the number of articles contained in training, test and the whole set, respectively. Since SVM is basically a two-class classifier, we considered 10 classification problems of classifying if articles belong to a category or not.

Texts were represented using a vector space model, where documents are represented by feature vector of terms (Salton and McGill, 1983). Data was preprocessed using the *rainbow* program, a part of the *Bow* toolkit. [2] Preprocessing consisted of transforming to lower-cases, removing stop words (524 words), applying the Porter stemmer (Porter, 1980). Then, frequency of each term was counted, and terms appeared less than three times were not included as features. As a result, 6801 distinct words were used as features. Obtained term-document matrix was then transformed to TF-IDF weight that is normalized by document lengths.

[2] For more information, see: Andrew Kachites McCallum, Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering, http://www.cs.cmu.edu/~mccallum/bow

SVD was applied to this term-document matrix. LSI basis vectors that corresponds to 1000 largest singular values were computed due to the limitation of memory. Subspace spanned by these basis vectors accounts for approximately 77% of total variances in the original space. $\boldsymbol{T}', \boldsymbol{D}', \boldsymbol{S}'$ are obtained by sorting these 1000 features according to values of q_k.

For evaluating LSI features, training and testing SVM, we used SVM-light (Joachims, 1999). For linear SVMs, the only parameter that needs to be specified beforehand is the constant C. When a dataset is separable, performance is rather insensitive to the choice of C as long as its value is sufficiently large (Drucker et al., 1999). Since most datasets in text categorization domain are linearly separable, we simply used a fixed value of $C = 1000$ throughout the experiment. It can be expected that the performance may improve by tuning the value of C. Linear SVM ($C = 1000$) was first trained in order to derive SVM-LSI feature matrix $\boldsymbol{T}'$. Then, data vectors were projected onto a SVM-LSI subspace $\hat{\boldsymbol{X}}'$, and an another linear SVM ($C = 1000$) was trained on $\hat{\boldsymbol{X}}'$. Finally, vectors in a test set is projected onto a SVM-LSI subspace. Let $\boldsymbol{X}_t$ be a test set and $\hat{\boldsymbol{I}}$ be a diagonal matrix whose first k diagonal elements are set to 1 and the rest are set to 0, the trained linear SVM is tested against $\boldsymbol{T}'\hat{\boldsymbol{I}}\boldsymbol{T}'^{\mathrm{T}}\boldsymbol{X}_t$.

Most of the classification performance measures uses the following four quantities:

- *TP*. The number of documents correctly classified to a class.
- *TN*. The number of documents correctly rejected from a class.
- *FP*. The number of documents incorrectly rejected from a class.
- *FN*. The number of documents incorrectly classified to a class.

For datasets having only small fraction of positive samples, it is not appropriate to use accuracy ($\frac{TP+TN}{TP+TN+FP+FN}$) as a classification performance measure. In such cases, a trivial regector classifier that classifies all samples as not belonging to a class achieves high accuracy, but it is easy to see such classifiers are useless. As a classification perfor-

mance measure, we used break-even-point (BEP), which is popularly used in text categorization (Joachims, 1998; Dumais et al., 1998). Let precision be a quantity given by $\frac{TP}{TP+FP}$ and recall be a quantity given by $\frac{TP}{TP+FN}$, BEP is a value when precision and recall are tuned to be equal by varying threshold of decision function.

5.2. Results

We compared feature selection performance of normal LSI and SVM-LSI. Fig. 1 shows the result for `earn` and `money-fx` class. As the number of features was reduced from 1000, BEP value increased at some intermediate dimensions. Table 2 summarizes results for all classes. Maximum BEP values and the number of features that achieved the maxima are listed. Except for few classes that already achieved high BEP using all features (classes 1, 2, 4), benefit of applying LSI was evident. There were not so much difference in maximum BEP values between LSI and SVM-LSI. However, the number of features that achieves the maxima were significantly smaller for SVM-LSI. This is a great benefit over LSI, since the crucial problem in applying LSI is the high computational

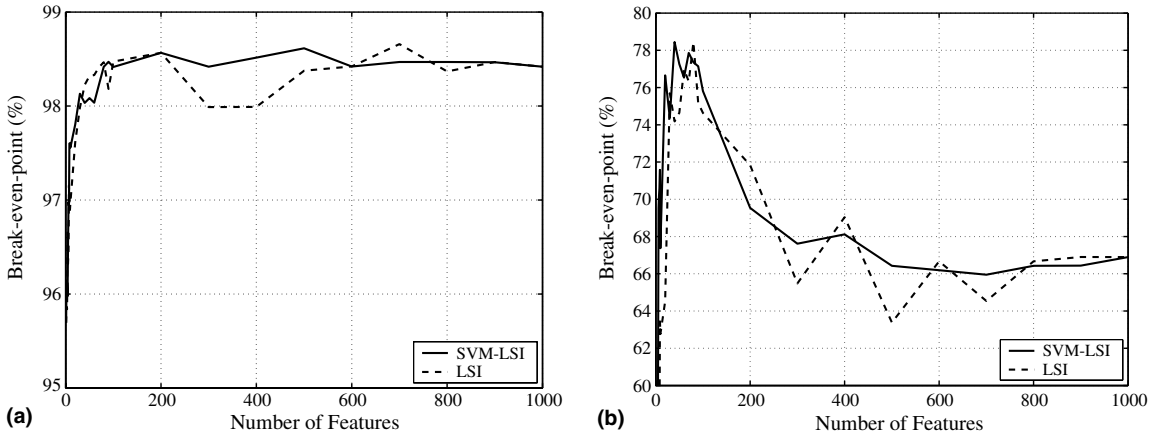

Fig. 1. Performance comparison of feature selection between normal LSI and SVM-LSI for: (a) `earn` and (b) `money-fx` class.

Table 2
Summary of results for all Reuters-21578 dataset

Class #	Max. BEP (%)		Number of features		BEP (all)
	LSI	SVM-LSI	LSI	SVM-LSI	
1	***98.7***	98.6	700	***500***	98.6
2	95.5	***95.8***	70	***20***	95.6
3	78.4	78.4	80	***40***	73.3
4	***88.1***	88.0	600	***300***	89.6
5	85.2	***86.3***	1000	***400***	85.9
6	80.2	***81.2***	60	***20***	74.6
7	***79.6***	78.6	30	***10***	71.4
8	81.6	***85.4***	60	***40***	81.4
9	82.4	***84.0***	***60***	70	82.4
10	***84.5***	82.1	500	500	83.3

In second and third column, maximum break-even-point (BEP) were compared between normal LSI and SVM-LSI. Bold italic numbers indicate that BEP was higher for that method compared to the other method. Fourth and fifth columns are the number of features that achieved maximum BEP for LSI and SVM-LSI, respectively. Bold italic numbers indicate that the number of features were lower. Rightmost column is BEP using all of the original features.

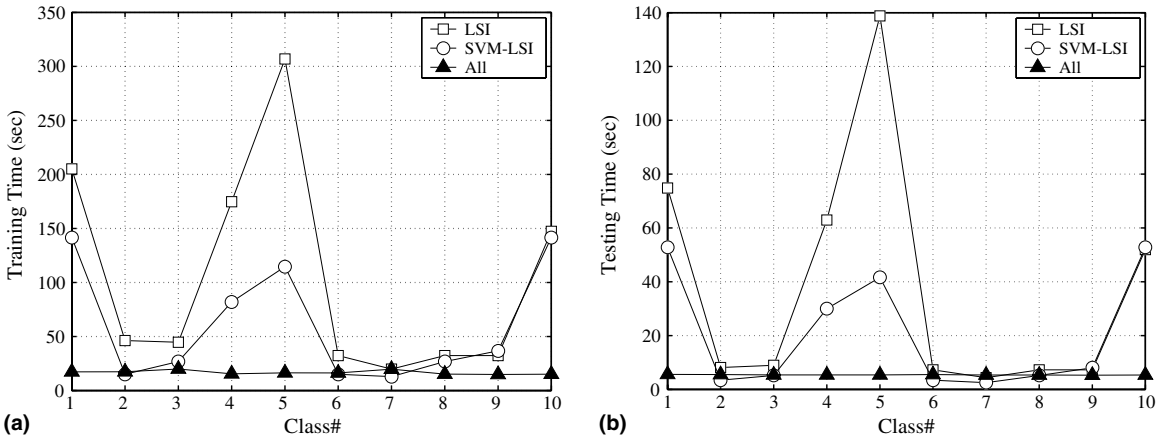

Fig. 2. Computation time for each class: (a) training time and (b) testing time.

cost required for training and testing. Even if original term-document matrix is sparse, truncated term-document matrix generated by LSI is no longer sparse, thus it requires great amount of computation time. Actual computation time required for training and testing SVM are shown in Fig. 2. Both LSI and SVM-LSI required considerably more training and testing time than using all features for classes 1, 4, 5, 10. However, training and testing time for SVM-LSI were comparable for class 2, 3, 6, 7, in which maximum BEPs were improved by the method. These results suggests that SVM-based feature selection method is an effective method to improve classification performance while requiring significantly smaller training and testing time than LSI.

6. Conclusion

In this paper, we have argued that the variance-based feature ordering that is commonly used in LSI is not necessarily appropriate for classification tasks. We proposed to apply SVM-based feature selection method in order to identify LSI-subspace that is suited for classification. Experimental results showed that the proposed method achieved high classification accuracies as seen in LSI with less training and testing time. The proposed method provides a mean to enhance the effectiveness of LSI in text categorization tasks.

References

Brank, J., Grobelnik, M., Milic-Frayling, N., Mladenic, D., 2002. Feature selection using support vector machines. In: Proc. Third Internat. Conf. on Data Mining Methods and Databases for Eng. Finance Other Fields, pp. 25–27.

Cortes, C., Vapnik, V., 1995. Support vector networks. Mach. Learning 20, 273–297.

Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., 1990. Indexing by latent semantic analysis. J. Amer. Soc. Inform. Sci. 41, 391–407.

Drucker, H., Wu, D., Vapnik, V.N., 1999. Support vector machines for spam categorization. IEEE Trans. Neural Networks 10, 1048–1054.

Dumais, S.T., 1995. Using LSI for information filtering. In: Harman, D., (Eds.), The Third Text REtrieval Conference (TREC3). National Institute of Standards and Technology Special Publications 500-215, pp. 219-230.

Dumais S., Platt, J., Heckerman, D., Sahami, M., 1998. Inductive learning algorithm and representation for text categorization. In: Proc. 7th Internat. Conf. on Inform. Retrieval Knowledge Manage. (ACM-CIKM-98), pp. 148–155.

Guyon, I., Weston, J., Barnhill, S., 2002. Gene selection for cancer classification using support vector machines. Mach. Learning 46, 389–422.

Joachims, T., 1998. Text categorization with support vector machines: Learning with many relevant features. In: Proc. 10th Eur. Conf. on Machine Learning, pp. 137–142.

Joachims, T., 1999. Making large-scale SVM learning practical. In: Schölkopf, B., Burges, C., Smola, A. (Eds.), Advances in Kernel Methods–Support Vector Learning. MIT Press, Cambridge, MA, pp. 169–184.

Tin-Yau Kwok, J., 1998. Automated text categorization using support vector machine. In: Proc. 5th Internat. Conf. on Neural Inform. Process. (ICONIP'98), pp. 347–351.

Porter, M.F., 1980. An algorithm for suffix stripping. Program 14, 130–137.

Pentland, A., Starner, T., Etcoff, N., Masoiu, N., Oliyide, O., Turk, M., 1993. Experiments with eigenfaces, In: Proc. Looking at People Workshop Internat. Jnt. Conf. Artificial Intell.

Salton, G., McGill, M.J., 1983. An Introduction to Modern Information Retrieval. McGraw-Hill, New York.

Schütze, H., Silverstein, C., 1997. Projections for efficient document clustering, In: Proc. SIGIR'97, pp. 74–81.

Story, R.E., 1996. An explanation of the effectiveness of latent semantic indexing by means of a Bayesian regression model. Inform. Process. Manage. 32, 329–344.

Tin-Yau Kwok, J., 1998. Automated text categorization using support vector machine. In: Proc. 5th Internat. Conf. on Neural Inform. Process. (ICONIP'98), pp. 347–351.

Vapnik, V., 1998. Statistical Learning Theory. John Wiley & Sons, New York.