Dimension Reduction in Text Classification with Support Vector Machines

Hyunsoo Kim, Peg Howland, and Haesun Park *

Department of Computer Science and Engineering,
University of Minnesota,
200 Union Street S.E., 4-192 EE/CS Building,
Minneapolis, MN 55455, U.S.A.

March 17, 2003

Abstract

Support vector machines (SVMs) have been recognized as one of the most successful classification methods for many applications including text classification. Even though the learning ability and computational complexity of training in support vector machines may be independent of the dimension of the feature space, reducing computational complexity is an essential issue to efficiently handle a large number of terms in practical applications of text classification. In this paper, we adopt novel dimension reduction methods to reduce the dimension of the document vectors dramatically. We also introduce decision functions for the centroid-based classification algorithm and support vector classifiers to handle the classification problem where a document may belong to multiple classes. Our substantial experimental results show that with several dimension reduction methods that are designed particularly for clustered data, higher efficiency for both training and testing can be achieved without sacrificing prediction accuracy of text classification even when the dimension of the input space is significantly reduced.

1 Introduction

Text classification is a supervised learning task for assigning text documents to pre-defined classes of documents. It is used to find valuable information from a huge collection of text documents available in digital libraries, knowledge databases, the world wide web (WWW), and company-wide intranets, to name a few.

*Corresponding author: Haesun Park, E-mail: hpark@cs.umn.edu, Phone: 612-625-0041, Fax: 612-625-0572.

This work was supported in part by the National Science Foundation grant CCR-0204109. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation (NSF).

Several characteristics have been observed in vector space based methods for text classification [19, 20], including the high dimensionality of the input space, sparsity of document vectors, linear separability in most text classification problems, and the belief that few features are irrelevant. It has been conjectured that an aggressive dimension reduction may result in a significant loss of information, and therefore, result in poor classification results [12].

Assume that training data $(\mathbf{x}_i, y_i)$ with $y_i \in \{-1, +1\}$ for $1 \le i \le n$ are given. The dual formulation of soft margin support vector machines (SVMs) with a kernel function K and control parameter C is

$$\begin{aligned} \max_{\alpha_i} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j), \\ s.t. \quad & \sum_{i=1}^{n} \alpha_i y_i = 0, \;\; 0 \le \alpha_i \le C, \;\; i = 1, \dots, n. \end{aligned} \tag{1}$$

The kernel function

$$K(\mathbf{x}_i, \mathbf{x}_j) =< \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) >,$$

where $<, >$ denotes an inner product between two vectors, is introduced to handle nonlinear separable cases without any explicit knowledge of the feature mapping ϕ. The formulation (1) shows that the computational complexity of SVM training depends on the number of training data samples which is denoted as n in (1). The dimension of the feature space does not influence the computational complexity of training or testing due to the use of the kernel function.

However, an often neglected fact is that the computational complexity of training depends on the *dimension of the input space*. This is clear when we consider some typical kernel functions such as the linear kernel

$$K(\mathbf{x}, \mathbf{x}_i) =< \mathbf{x}, \mathbf{x}_i >,$$

the polynomial kernel

$$K(\mathbf{x}, \mathbf{x}_i) = [< \mathbf{x}, \mathbf{x}_i > + \beta]^d,$$

where d is the degree of the polynomial, and the Gaussian RBF (radial basis function) kernel

$$K(\mathbf{x}, \mathbf{x}_i) = \exp(-\gamma \|\mathbf{x} - \mathbf{x}_i\|^2),$$

where γ is a parameter to control. The evaluation of the kernel function depends on the dimension of the input data, since the kernel functions contain the inner product of two input vectors for the polynomial kernel or the distance of two vectors for the Gaussian RBF kernel. Let α_i^* denote the optimal solution for (1). The optimal separating hyperplane $f(\mathbf{x}, \alpha^*, b)$ also requires evaluation of the kernel function since

$$f(\mathbf{x}, \alpha^*, b) = \sum_{i \in SV} y_i \alpha_i K(\mathbf{x_i}, \mathbf{x}) + b$$

where SV denotes the set of support vectors,

$$b = -\frac{min_{y_i=1} < w^*, \phi(\mathbf{x_i}) > + \max_{y_i=-1} < w^*, \phi(\mathbf{x_i}) >}{2}$$

and

$$w^* = \sum_{i=1}^{l} y_i {\alpha_i}^* \phi(\mathbf{x_i}).$$

Therefore, more efficient testing as well as training is expected from dimension reduction.

For k-nearest neighbor classification (kNN), the distances of vector pairs need to be computed when finding k nearest neighbors. Therefore, one can significantly reduce computational complexity by dimension reduction.

In the next section, we review Latent Semantic Indexing (LSI), which uses the truncated singular value decomposition (SVD) as a low-rank approximation of A. Although the truncated SVD provides the closest approximation to A in Frobenius or L_2 norm, LSI ignores the cluster structure while reducing the dimension of the data. In contrast, in Section 3, we review several dimension reduction methods that are especially effective for classification of clustered data: two methods based on centroids [15, 11], and one method which is a generalization of linear discriminant analysis (LDA) using the generalized singular value decomposition (GSVD) [10]. With dimension reduction, computational complexity can be dramatically reduced for all classifiers including support vector machines and k-nearest neighbor classification. In many document data sets, documents can be assigned to more than one cluster upon classification. To handle this problem more effectively, we introduce a threshold based extension of several classification algorithms in Section 4. Our numerical experiments illustrate that the cluster-preserving dimension reduction algorithms we employ reduce the data dimension without any significant loss of information. In fact, in many cases, they seem to have the effect of noise reduction, since prediction accuracy becomes better after dimension reduction when compared to that in the original high dimensional input space.

Throughout the paper, we will assume that the document set is represented in a term-by-document $m \times n$ matrix $A = (a_{ij})$, in which each column represents a document, and each entry a_{ij} represents the weighted frequency of term i in document j [1, 2]. The clustering of data is assumed to be performed previously.

2 Low-Rank Approximation Using Latent Semantic Indexing

LSI is based on the assumption that there is some underlying latent semantic structure in the term-document matrix that is corrupted by the wide variety of words used in documents and queries. This is referred to as the problem of polysemy and synonymy [6]. The basic idea is that if two document vectors represent the same topic, they will share many associating words with a keyword, and they will have very close semantic structures after dimension reduction via SVD. Thus LSI/SVD breaks the original relationship of the data into linearly independent components [6], where the original term vectors are represented by left singular vectors and document vectors by right singular vectors. That is, if $l \leq \mathrm{rank}(A)$, then

$$A \approx U_l \Sigma_l V_l^T$$

where the columns of U_l are the leading l left singular vectors, Σ_l is a diagonal matrix with the l largest singular values in nonincreasing order along its diagonal, and the columns of V_l are the leading l right singular vectors. Then $\Sigma_l V_l^T$ is the reduced dimensional representation of A, or equivalently, a new document $\mathbf{q} \in \mathbb{R}^{m \times 1}$ can be projected to the l-dimensional space as $\hat{\mathbf{q}} = U_l^T \mathbf{q}$.

This low-rank approximation has been widely applied in information retrieval [2]. Since the complete orthogonal decomposition (ULV or URV) has computational advantages over the SVD, such as easier updating [21, 22, 23] and downdating [16], dimension reduction by low-rank orthogonal decompositions has also been exploited [3]. However, LSI ignores the cluster structure while reducing the dimension. In addition, since there is no theoretical optimum value for the reduced dimension, potentially expensive experimenta-

Algorithm 1 : Centroid algorithm for Dimension Reduction

Given a data set $A \in \mathbb{R}^{m\times n}$ with p clusters and a vector $\mathbf{q} \in \mathbb{R}^{m\times 1}$, this algorithms computes a p dimensional representation $\hat{\mathbf{q}} \in \mathbb{R}^{p\times 1}$ of $\mathbf{q}$.

1. Compute the centroid $\mathbf{c_i}$ of the ith cluster, $1 \leq i \leq p$
2. Set $C = \begin{bmatrix}\mathbf{c_1} & \mathbf{c_2} & \cdots & \mathbf{c_p}\end{bmatrix}$
3. Solve $\min_{\hat{\mathbf{q}}} \|C\hat{\mathbf{q}} - \mathbf{q}\|_2$

Algorithm 2 : Orthogonal Centroid algorithm for Dimension Reduction

Given a data set $A \in \mathbb{R}^{m\times n}$ with p clusters and a vector $\mathbf{q} \in \mathbb{R}^{m\times 1}$, it computes a p dimensional representation $\hat{\mathbf{q}}$ of $\mathbf{q}$.

1. Compute the centroid $\mathbf{c_i}$ of the ith cluster, $1 \leq i \leq p$
2. Set $C = \begin{bmatrix}\mathbf{c_1} & \mathbf{c_2} & \cdots & \mathbf{c_p}\end{bmatrix}$
3. Compute the reduced QR decomposition of C, which is $C = Q_pR$
4. $\hat{\mathbf{q}} = Q_p^T\mathbf{q}$

tion may be required. As we report in Section 5, classification results after LSI vary depending upon the reduced dimension, classification method, and similarity measure employed.

3 Dimension Reduction Algorithms for Clustered Data

To achieve higher efficiency in manipulating data represented in a high dimensional space, it is often necessary to reduce the dimension *dramatically*. In this section, several dimension reduction methods that preserve the cluster structure are reviewed. Each method attempts to choose a projection to the reduced dimension that will capture the cluster structure of the data collection as well as possible.

3.1 Centroid based Algorithms for Dimension Reduction of Clustered Data

Suppose we are given a data matrix A whose columns are grouped into p clusters. Instead of treating each column of the matrix A equally regardless of its membership in a specific cluster as in LSI/SVD [6, 2, 1], we want to find a lower dimensional representation Y of A so that the p clusters are preserved in Y. Given a term-document matrix, the problem is to find a transformation that maps each document vector in the m dimensional space to a vector in the l dimensional space for some $l < m$. For this, either the dimension reducing transformation $G^T \in \mathbb{R}^{l\times m}$ is computed explicitly or the problem is formulated as a rank reducing approximation where the given matrix A is to be decomposed into two matrices B and Y. That is,

$$A \approx BY \tag{2}$$

where $B \in \mathbb{R}^{m\times l}$ with rank(B) = l and $Y \in \mathbb{R}^{l\times n}$ with rank(Y) = l. The matrix B accounts for the dimension reducing transformation. If this matrix B is already determined, the matrix Y can be computed by solving the least squares problem [8, 11, 15]

$$\min_{B,Y} \|BY - A\|_F. \tag{3}$$

Any given document $\mathbf{q} \in \mathbb{R}^{m\times 1}$ can be transformed to the lower dimensional space by solving the minimization problem

$$\min_{\hat{\mathbf{q}}\in\mathbb{R}^{l\times 1}} \|B\hat{\mathbf{q}} - \mathbf{q}\|_2. \tag{4}$$

In the Centroid dimension reduction algorithm (see Algorithm 1), the ith column of B is the centroid vector of the ith cluster, which is the average of the data items in the ith cluster, for $1 \le i \le p$. This matrix is called the centroid matrix. Then, any vector $\mathbf{q} \in \mathbb{R}^{m\times 1}$ can be represented in the p dimensional space as $\hat{\mathbf{q}}$, the solution of the least squares problem (4), where B is the centroid matrix. In the Orthogonal Centroid algorithm (see Algorithm 2), the p dimensional representation of a data vector $\mathbf{q} \in \mathbb{R}^{m\times 1}$ is given as $\hat{\mathbf{q}} = Q_p^T\mathbf{q}$ where Q_p is an orthonormal basis for the centroid matrix obtained from its orthogonal decomposition.

Latent Semantic Indexing that utilizes the SVD (LSI/SVD) can be viewed as a variation of the model (2) with $B = U_l$ [15], where $U_l\Sigma_l V_l^T$ is the rank l truncated SVD of A. Then $\hat{\mathbf{q}} = U_l^T\mathbf{q}$ is obtained by solving the least squares problem

$$\min_{\hat{\mathbf{q}}\in R^{l\times 1}} \|B\hat{\mathbf{q}} - \mathbf{q}\|_2 = \min_{\hat{\mathbf{q}}\in R^{l\times 1}} \|U_l\hat{\mathbf{q}} - \mathbf{q}\|_2. \tag{5}$$

The centroid based dimension reduction algorithms are computationally less costly than LSI/SVD. They are also more effective when the data are already clustered. Although the centroid based schemes can be applied only when the data are linearly separable, they are suitable for text classification problems, since text data is usually linearly separable in the original dimensional space [12].

3.2 Generalized Discriminant Analysis based on the Generalized Singular Value Decomposition

Recently, a new algorithm has been developed for cluster-preserving dimension reduction based on the generalized singular value decomposition (GSVD) [10]. This algorithm generalizes classical discriminant analysis, by extending its application to very high-dimensional data such as that encountered in text classification.

Classical discriminant analysis [7, 24] preserves cluster structure by maximizing the scatter between clusters while minimizing the scatter within clusters. For this purpose, the within-cluster scatter matrix S_w and the between-cluster scatter matrix S_b are defined. If we let N_i be the set of column indices that belong to the cluster i, n_i be the number of columns in cluster i, and $\mathbf{c}$ be the global centroid, then

$$S_w = \sum_{i=1}^{p}\sum_{j\in N_i}(\mathbf{a}_j - \mathbf{c}_i)(\mathbf{a}_j - \mathbf{c}_i)^T,$$

Algorithm 3 LDA/GSVD

Given a data matrix $A \in \mathbb{R}^{m\times n}$ with p clusters, it computes the columns of the matrix $G \in \mathbb{R}^{m\times(p-1)}$, which preserves the cluster structure in the reduced dimensional space, and it also computes the $p-1$ dimensional representation Y of A.

1. Compute $H_b \in \mathbb{R}^{m\times p}$ and $H_w \in \mathbb{R}^{m\times n}$ from A according to Eqns. (7) and (6), respectively.
2. Compute the complete orthogonal decomposition of $H = (H_b, H_w)^T \in \mathbb{R}^{(p+n)\times m}$, which is
$$P^T HQ = \begin{pmatrix} R & 0 \\ 0 & 0 \end{pmatrix}.$$
3. Let $t = \mathrm{rank}(H)$.
4. Compute W from the SVD of $P(1:p, 1:t)$, which is $U^T P(1:p,1:t)W = \Sigma_A$.
5. Compute the first $p-1$ columns of
$$X = Q\begin{pmatrix} R^{-1}W & 0 \\ 0 & I \end{pmatrix},$$
and assign them to G.
6. $Y = G^T A$

and

$$\begin{aligned} S_b &= \sum_{i=1}^{p}\sum_{j\in N_i}(\mathbf{c}_i - \mathbf{c})(\mathbf{c}_i - \mathbf{c})^T \\ &= \sum_{i=1}^{p} n_i(\mathbf{c_i} - \mathbf{c})(\mathbf{c_i} - \mathbf{c})^T. \end{aligned}$$

Since

$$trace(S_w) = \sum_{i=1}^{p}\sum_{j\in N_i}\|\mathbf{a}_j - \mathbf{c}_i\|_2^2$$

measures the closeness within the clusters, and

$$trace(S_b) = \sum_{i=1}^{p}\sum_{j\in N_i}\|\mathbf{c}_i - \mathbf{c}\|_2^2$$

measures the remoteness between the clusters, the goal is to minimize the former while maximizing the latter in the reduced dimensional space. Once again letting $G^T \in \mathbb{R}^{l\times m}$ denote the transformation that maps a column of A in the m dimensional space to a vector in the l dimensional space, the goal can be expressed as the simultaneous minimization of $\mathrm{trace}(G^T S_w G)$ and maximization of $\mathrm{trace}(G^T S_b G)$.

When S_w is nonsingular, this simultaneous optimization is commonly approximated by maximizing

$$J_1(G) = \mathrm{trace}((G^T S_w G)^{-1}(G^T S_b G)).$$

It is well known that the global maximum is achieved when the columns of G are the eigenvectors of $S_w^{-1} S_b$ that correspond to the l largest eigenvalues [7, 24]. In fact, when the reduced dimension $l \geq p-1$, $\mathrm{trace}(S_w^{-1} S_b)$ is exactly preserved upon dimension reduction, and equals $\lambda_1 + \cdots + \lambda_{p-1}$, where each $\lambda_i \geq 0$. If we express S_w and S_b in terms of

$$H_w = [\mathbf{a}_1 - \mathbf{c}_1, \mathbf{a}_2 - \mathbf{c}_1, \ldots, \mathbf{a}_n - \mathbf{c}_p] \in \mathbb{R}^{m \times n} \tag{6}$$

and

$$H_b = [\sqrt{n_1}(\mathbf{c}_1 - \mathbf{c}), \ldots, \sqrt{n_p}(\mathbf{c}_p - \mathbf{c})] \in \mathbb{R}^{m \times p}, \tag{7}$$

then $S_w = H_w H_w^T$ and $S_b = H_b H_b^T$. As the product of an $m \times n$ matrix with an $n \times m$ matrix, S_w will be singular when the number of terms m exceeds the number of documents n. In that case, the J_1 criterion is undefined and classical discriminant analysis fails. However, if we rewrite the eigenvalue problem $S_w^{-1} S_b \mathbf{x}_i = \lambda_i \mathbf{x}_i$ as

$$\beta_i^2 H_b H_b^T \mathbf{x}_i = \alpha_i^2 H_w H_w^T \mathbf{x}_i,$$

it can be solved by the GSVD.

The resulting algorithm, called LDA/GSVD, is summarized in Algorithm 3. It follows the construction of the Paige and Saunders [14] proof, but only computes the necessary part of the GSVD. The most expensive step of LDA/GSVD is the complete orthogonal decomposition of the composite H matrix. When $\max(p, n) \ll m$, the SVD of $H = [H_b^T, H_w^T] \in \mathbb{R}^{(p+n) \times m}$ can be computed by first computing the reduced QR decomposition $H^T = K_1 T$, and then computing the SVD of $T \in \mathbb{R}^{(p+n) \times (p+n)}$ as

$$T = Z \begin{pmatrix} R & 0 \\ 0 & 0 \end{pmatrix} P^T.$$

This gives

$$H = T^T K_1^T = P \begin{pmatrix} R & 0 \\ 0 & 0 \end{pmatrix} Z^T K_1^T,$$

where the $p+n$ columns of $K_1 Z$ are orthonormal. This is adequate because LDA/GSVD only requires the first $t \leq p+n$ columns of Q. Since $T \in \mathbb{R}^{(p+n) \times (p+n)}$ is a much smaller matrix than $H \in \mathbb{R}^{(p+n) \times m}$ and T is upper triangular, the required memory is remarkably reduced.

4 Classification Methods

For testing the effect of dimension reduction in text classification, three different methods were used: centroid-based classification, k-nearest neighbor (kNN), and support vector machines (SVMs). Each classification method is modified by introducing some threshold values to perform classification correctly when a document has a membership in multiple classes. In this section, we briefly review the above three classification methods and discuss their modifications.

Algorithm 4 : Centroid-based Classification

Given a data matrix A with p clusters and p corresponding centroids, $\mathbf{c_i}$, $1 \leq i \leq p$, and a vector $\mathbf{q} \in \mathbb{R}^{m\times 1}$, it finds the index j of the cluster in which the vector $\mathbf{q}$ belongs.

- find the index j such that $sim(\mathbf{q}, \mathbf{c_j})$, $1 \leq i \leq p$, is minimum (or maximum), where $sim(\mathbf{q}, \mathbf{c_j})$ is the similarity measure between $\mathbf{q}$ and $\mathbf{c_j}$. (For example, $sim(\mathbf{q}, \mathbf{c_j}) = \|\mathbf{q} - \mathbf{c_j}\|_2$ using the L_2 norm, and we take the index with the minimum value. Using the cosine measure,

$$sim(\mathbf{q}, \mathbf{c_j}) = cos(\mathbf{q}, \mathbf{c_j}) = \frac{\mathbf{q}^T\mathbf{c_j}}{\|\mathbf{q}\|_2\|\mathbf{c_j}\|_2},$$

  and we take the index with the maximum value.)

4.1 Centroid-based Classification

Centroid-based classification, summarized in Algorithm 4, is one of the simplest classification methods. A test document is assigned to a class that has the most similar centroid. Using the cosine similarity measure, we can classify a test document $\mathbf{q}$ by computing

$$arg \max_{1\leq i\leq p} \frac{\mathbf{q}^T\mathbf{c}_i}{\|\mathbf{q}\|_2\|\mathbf{c}_i\|_2} \tag{8}$$

where $\mathbf{c}_i$ is the centroid of the ith cluster of the training data. When dimension reduction is performed by the Centroid algorithm, the centroids of the full space become the columns $\mathbf{e_i} \in \mathbb{R}^{p\times 1}$ of the identity matrix. Then the decision rule becomes

$$arg \max_{1\leq i\leq p} \frac{\hat{\mathbf{q}}^T\mathbf{e_i}}{\|\hat{\mathbf{q}}\|_2\|\mathbf{e_i}\|_2}, \tag{9}$$

where $\hat{\mathbf{q}}$ is the reduced dimensional representation of the document $\mathbf{q}$. This shows that classification can be performed by simply finding the index i of the vector $\hat{\mathbf{q}}$ with the largest component. Although centroid-based classification has the advantage that the computation involved is extremely simple when cosine measure is used, it has a disadvantage that it can only represent the classifier which passes through the origin. We can also find the centroid that is closest to $\mathbf{q}$ using the L_2 norm similarity measure.

The original form of centroid-based classification finds the nearest centroid and assigns the corresponding class as the predicted class. To allow an assignment of any document to multiple classes, we introduce the decision rule for centroid-based classification as

$$y(\mathbf{x}, j) = \text{sign}\{sim(\mathbf{x}, \mathbf{c}_j) - \theta_j^c\}, \tag{10}$$

where where $y(\mathbf{x}, j) \in \{+1, -1\}$ is the classification for document $\mathbf{x}$ with respect to class j (if $y > 0$ then the class is j, else the class is not j); $sim(\mathbf{x}, \mathbf{c}_j)$ is the similarity between the test document $\mathbf{x}$ and the centroid vector $\mathbf{c}_j$ for the class j; and θ_j^c is the class specific threshold for the binary decision in centroid-based classification.

Algorithm 5 : k Nearest Neighbor (kNN) Classification

Given a data matrix $A = [\mathbf{a_1}, \ldots, \mathbf{a_n}]$ with p clusters and a vector $\mathbf{q} \in \mathbb{R}^{m\times 1}$, it finds the cluster in which the vector $\mathbf{q}$ belongs.

1. From the similarity measure $sim(\mathbf{q}, \mathbf{a_j})$ for $1 \le j \le n$, find the k nearest neighbors of $\mathbf{q}$.
2. Among these k vectors, count the number belonging to each cluster.
3. Assign $\mathbf{q}$ to the cluster with the greatest count in the previous step.

4.2 k-Nearest Neighbor Classification

The kNN algorithm, summarized in Algorithm 5, is one of the most commonly used classification algorithms.

To correctly predict the membership of a document which belongs to multiple classes, we used the following modified decision rule for kNN [28]:

$$y(\mathbf{x}, j) = \text{sign}\{\sum_{\mathbf{d}_i \in kNN} sim(\mathbf{x}, \mathbf{d}_i) y(\mathbf{d}_i, j) - \theta_j^{kNN}\} \quad (11)$$

where kNN is the set of k nearest neighbors for document $\mathbf{x}$; $y(\mathbf{d}_i, j) \in \{+1, -1\}$ is the classification for document $\mathbf{d}_i$ with respect to class j (if $y > 0$ then the class is j, else the class is not j); $sim(\mathbf{x}, \mathbf{d}_i)$ is the similarity between the test document $\mathbf{x}$ and the training document $\mathbf{d}_i$; and θ_j^{kNN} is the class specific threshold for kNN classification.

4.3 Support Vector Machines

The optimal separating hyperplane of the one-vs-rest binary classifier can be obtained by conventional SVMs. We introduce the following decision rule for support vector machines as

$$y(\mathbf{x}, j) = \text{sign}\{\sum_{\mathbf{x}_i \in SV} \alpha_i y_i K(\mathbf{x}, \mathbf{x_i}) + b - \theta_j^{SVM}\}, \quad (12)$$

where $y(\mathbf{x}, j) \in \{+1, -1\}$ is the classification for document $\mathbf{x}$ with respect to class j, SV is the set of support vectors, b is a bias, and θ_j^{SVM} is the class specific threshold for the binary decision. This threshold is set so that a new document $\mathbf{x}$ must not be classified to belong to class j when it is located very close to the optimal separating hyperplane, i.e. when the decision is made with a low reliability.

We used the linear kernel $K = < \mathbf{x}, \mathbf{x}_i >$, the polynomial kernel $K = [< \mathbf{x}, \mathbf{x}_i > +1]^d$, where d is the degree of the polynomial, and the Gaussian RBF (radial basis function) kernel $K = \exp(-\gamma \|\mathbf{x} - \mathbf{x}_i\|^2)$, where γ is a parameter that controls the width of the Gaussian function.

5 Experimental Results

Prediction results are compared for the test documents in the full space without any dimension reduction as well as those in the reduced space obtained by LSI/SVD, Centroid, Orthogonal Centroid, and LDA/GSVD

classification	The rank-l approximation of LSI/SVD								
methods	l=5	l=100	l=200	l=300	l=500	l=1000	l=1246	l=1247	Full
centroid (L_2)	71.5	82.4	83.2	84.0	84.3	84.8	84.8	84.8	84.8
centroid (Cosine)	78.8	86.8	86.9	87.4	87.7	88.0	88.0	88.0	88.0
5nn (L_2)	75.9	69.4	55.1	47.2	66.3	76.8	79.3	79.3	79.3
15nn (L_2)	76.5	69.0	53.4	49.1	77.0	74.8	83.4	83.4	83.4
30nn (L_2)	77.4	65.0	48.2	57.4	81.4	72.4	83.9	83.9	83.9
5nn (Cosine)	76.6	81.4	79.3	79.5	79.6	79.1	77.8	77.8	77.8
15nn (Cosine)	79.0	83.3	82.7	83.2	82.7	82.5	82.3	82.3	82.3
30nn (Cosine)	79.0	83.4	84.2	84.2	84.3	83.9	83.5	83.5	83.5
SVM	79.4	88.3	89.0	88.0	89.3	88.6	89.0	89.0	88.9

Table 1: Text classification accuracy (%) using centroid-based classification, k-nearest neighbor classification, and SVMs, with LSI/SVD dimension reduction on the MEDLINE data set. The Euclidian norm (L_2) and the cosine similarity measure (Cosine) were used for the centroid-based and kNN classification.

dimension reduction methods. For SVMs, we optimized the regulation parameter C, polynomial degree d for the polynomial kernel, and γ for the Gaussian RBF (radial basis function) kernel for each full and dimension reduced data set.

The first data set that we used was a subset of the MEDLINE database with 5 classes. Each class has 500 documents. The set was divided into 1250 training documents and 1250 test documents. After stemming and stoplist removal, the training set contains 22095 distinct terms. For this data, each document belongs to only one class, and we used the original form of the three classification algorithms without introducing the threshold.

The second data set was the "ModApte" split of the Reuter-21578 text collection. We only used 90 classes for which there is at least one training and one test example in each class. It contains 7769 training documents and 3019 test documents. The training set contains 11941 distinct terms after preprocessing with stoplist removal and stemming. The Reuter data set contains documents that belong to multiple classes, so the classification methods utilize thresholds.

We used a standard weight factor for each word stem :

$$\phi_i(\mathbf{x}) = \frac{tf_i \log(idf_i)}{\kappa}, \tag{13}$$

where tf_i is the number of occurrences of term i in document $\mathbf{x}$, $idf_i = n/d$ is the ratio between the total number of documents n and the number of documents d containing the term, and κ is the normalization constant that makes $\|\phi\|_2 = 1$.

Table 1 reports text classification accuracy for the MEDLINE data set using LSI/SVD with a range of values for the reduced dimension. The smallest reduced dimension, $l = 5$, is also included in subsequent tables in order to compare with centroid-based and LDA/GSVD methods, which reduce the dimension to 5 and 4, respectively. Since the training set has the nearly-full rank of 1246, we include the reduced dimensions 1246 and 1247 at the high end of the range. For a training set of size 1250, the reduced dimension $l = 300$ is generous. However, we observe that kNN classification with L_2 norm similarity produces poor classification results for l values from 100 to 500. This is consistent with the common belief that cosine similarity performs

kernel		Dimension reduction methods				
	Full	LSI/SVD	Centroid	Orthogonal Centroid	LDA/ GSVD4	LDA/ GSVD5
	22095×1250	5×1250	5×1250	5×1250	4×1250	5×1250
linear (C=1.0)	88.3	68.4	88.7	85.5	86.3	86.4
linear (C=10.0)	88.9	74.5	88.4	87.9	86.6	86.6
linear (C=50.0)	88.9	76.9	87.8	88.6	87.0	87.0
linear^{opt}	88.9	77.4	88.7	88.7	87.1	87.2
polynomial(d=2)	88.2	76.7	88.8	88.7	87.1	87.1
polynomial(d=3)	88.0	76.7	89.0	88.6	87.2	87.2
polynomial(d=4)	87.2	76.7	89.0	88.6	87.2	87.1
polynomial(d=5)	86.7	76.8	88.7	88.6	87.1	87.1
polynomial^{opt}	88.2	76.8	89.0	88.7	87.2	87.2
RBF ($\gamma = 0.5$)	88.2	78.1	88.8	88.6	87.0	87.4
RBF ($\gamma = 1.0$)	87.5	78.4	89.0	88.6	87.1	87.1
RBF ($\gamma = 1.5$)	86.7	79.4	89.1	88.7	87.0	87.1
RBF^{opt}	88.7	79.4	89.1	88.7	87.1	87.4

Table 2: Text classification accuracy (%) with different kernels in SVMs with and without dimension reduction on the MEDLINE data set. The regulation parameter C for linear^{opt}, polynomial, and RBF and γ for RBF^{opt} were optimized by numerical experiments. Dimension of each training term-document matrix is shown. LDA/GSVD4 and LDA/GSVD5 represents the results from LDA/GSVD where the reduced dimensions are 4 and 5, respectively.

better with unnormalized text data. Also, classification accuracy using 5nn lags that for higher values of k, suggesting that k=5 is too small for classes of size 250. It is noteworthy that even with LSI, which makes no attempt to preserve the cluster structure upon dimension reduction, SVM classification achieves very consistent classification results for reduced dimensions of 100 or greater, and the SVM accuracy exceeds that of the other classification methods.

Table 2 shows text classification accuracy (%) with different kernels in SVMs, with and without dimension reduction on the MEDLINE data set. It shows that the prediction results in the reduced dimension are similar to those in the original full dimensional space, while achieving a significant reduction in time and space complexity. In the reduced space obtained by the Orthogonal Centroid dimension reduction algorithm, the classification accuracy is insensitive to the choice of the kernel. Thus, we can choose the linear kernel in this case instead of the computationally more expensive polynomial or RBF kernel. Table 3 shows classification accuracy obtained by all three classification methods – centroid-based, knn with three different values of k, and the optimal result from SVM – for each dimension reduced data set and the full space. For the LDA/GSVD dimension reduction method, the classification accuracy with cosine similarity measure is lower with the centroid-based classification as well as with the knn, while the results with L_2 norm are the best. With LDA/GSVD, documents from the same class in the full dimensional space tend to be transformed to a very tight cluster or even to a single point in the reduced space, since the LDA/GSVD algorithm tends to minimize the trace of the within cluster scatter. This seems to make it difficult for SVMs to find a binary classifier with low generalization error. Table 4 shows text classification accuracy for the 5 classes using

classification		Dimension reduction methods				
methods	Full	LSI/SVD	Centroid	Orthogonal Centroid	LDA/ GSVD4	LDA/ GSVD5
	22095×1250	5×1250	5×1250	5×1250	4×1250	5×1250
centroid (L_2)	84.8	71.5	87.5	84.8	88.9	88.9
centroid (Cosine)	88.0	78.8	87.5	88.0	83.9	83.8
5nn (L_2)	79.3	75.9	88.4	88.6	81.5	84.7
15nn (L_2)	83.4	76.5	88.3	88.2	89.0	89.0
30nn (L_2)	83.9	77.4	88.6	88.6	89.0	89.0
5nn (Cosine)	77.8	76.6	88.4	88.1	83.8	84.0
15nn (Cosine)	82.3	79.0	87.8	88.3	83.9	84.3
30nn (Cosine)	83.5	79.0	88.3	88.4	83.9	84.1
SVM	88.9	79.4	89.1	88.7	87.2	87.4

Table 3: Text classification accuracy (%) using centroid-based classification, k-nearest neighbor classification, and SVMs, with and without dimension reduction on the MEDLINE data set. The Euclidian norm (L_2) and the cosine similarity measure (Cosine) were used for centroid-based and kNN classification.

SVMs with and without dimension reduction methods on the MEDLINE data set. The colon cancer and oral cancer documents were relatively hard to classify correctly.

The REUTERS data set has many documents that are classified to more than 2 classes, while no document is classified to belong to more than one class in the MEDLINE data set. While we could handle relatively large matrices using a sparse matrix representation and sparse QR decomposition in the Centroid and Orthogonal Centroid dimension reduction methods, results for the LDA/GSVD dimension reduction method are not reported, since we ran out of memory while running the GSVD algorithm. For this data set, we built a series of threshold-based classifiers, optimizing the thresholds to capture the multiple class membership. All class specific thresholds (θ_j^{kNN}, θ_j^c, θ_j^{SVM}) are determined by numerical experiments. Though we obtained precision/recall break even points by optimizing the thresholds, we reported the values of the

class		Dimension reduction				
	Full	LSI/SVD	Centroid	Orthogonal Centroid	LDA/ GSVD4	LDA/ GSVD5
	22095×1250	5×1250	5×1250	5×1250	4×1250	5×1250
heart attack	92.4	83.2	94.8	93.6	91.6	90.0
colon cancer	85.6	62.8	84.0	84.8	82.8	84.0
glycemic	95.6	96.8	97.6	97.6	94.8	95.6
oral cancer	81.6	65.6	75.6	75.2	79.6	80.4
tooth decay	89.2	88.4	93.6	92.4	86.8	86.8
microavg	88.9	79.4	89.1	88.7	87.2	87.4

Table 4: Text classification accuracy (%) of the 5 classes and the microaveraged performance over all 5 classes on the MEDLINE data set. All results are from SVMs using optimal kernels.

classification methods	Full	Dimension reduction Centroid	Orthogonal Centroid
centroid(L_2)	78.89	73.32	78.00
centroid(Cosine)	80.45	74.79	80.46
15nn	78.65	81.70	85.51
30nn	80.21	81.94	86.19
45nn	80.29	81.01	84.79
SVM	87.11	84.54	87.03

Table 5: Comparison of micro-averaged F_1 scores for 3 different classification methods with and without dimension reduction on the REUTERS data set. The Euclidian norm (L_2) and the cosine similarity measure (Cosine) were used for the centroid-based classification. The cosine similarity measure was used for the kNN classification. The dimension of the full training term-document matrix is 11941×9579 and that of the reduced matrix is 90×9579.

F_1 measure [25] which is defined as

$$F_1 = \frac{2rp}{r+p}, \tag{14}$$

where r is recall and p is precision for a binary classification. Table 5 shows that the effectiveness of classification was preserved for the Orthogonal Centroid dimension reduction algorithm, while it became worse for the Centroid dimension reduction algorithm. This is due to a property of the Centroid algorithm that the centroids of the full space are projected to the columns of the identity matrix in the reduced space. This orthogonality between the centroids may make it difficult to represent the multiclass membership of a document by separating closely related classes after dimension reduction. The pattern of prediction measure F_1 for each class is also preserved in the Orthogonal Centroid in Table 6. The macro-averaged F_1 and micro-averaged F_1 for the 10 most frequent classes are also presented.

6 Discussion

In this paper, we applied three methods, Centroid, Orthogonal Centroid, and LDA/GSVD, which are designed for reducing the dimension of clustered data. For comparison, we also applied LSI/SVD, which does not attempt to preserve cluster structure upon dimension reduction. We tested the effectiveness in classification with dimension reduction using three different classification algorithms, SVMs, kNN, and centroid-based classification. For the three cluster-preserving methods, the results show surprisingly high prediction accuracy , which is essentially the same as in the original full space, even with very dramatic dimension reduction. They justify dimension reduction as a worthwhile preprocessing stage for achieving high efficiency and effectiveness. Especially for kNN classification, the savings in computational complexity in classification after dimension reduction are significant. In the case of SVM the savings are also clear, since the distance between two pairs of input data points need to be computed repeatedly with and without the use of the kernel function, and the vectors become significantly shorter with dimension reduction.

We have also introduced threshold based classifiers for centroid-based classification and SVMs in order

class	Full	Dimension reduction Centroid	Orthogonal Centroid
earn	98.25	97.49	96.60
acq	95.57	95.45	94.94
money-fx	75.78	77.97	79.44
grain	92.88	86.62	92.26
crude	88.11	86.49	87.70
trade	75.32	75.11	77.25
interest	77.99	78.13	83.21
ship	84.09	85.71	88.00
wheat	84.14	81.94	84.06
corn	87.27	74.78	89.47
microavg (top 10)	92.21	91.32	92.21
avg (top 10)	85.94	83.96	87.32
microavg(all)	87.11	84.54	87.03

Table 6: F_1 scores of the 10 most frequent classes and micro-averaged performance over all 90 classes on the REUTERS data set. All results are from SVMs using optimal kernels. The dimension of the full training term-document matrix is 11941$\times$9579 and that of the reduced matrix is 90$\times$9579.

to capture the overlap structure between closely related classes. Prediction results with the Centroid dimension reduction method became better compared to those from the full space for the completely disjoint MEDLINE data set, but became worse in the REUTERS data set. Since the Centroid dimension reduction method maps the centroids to unit vectors $\mathbf{e}_i$ which are orthogonal to each other, it is helpful for the disjoint data set, but not for a data set which contains documents belonging multiple classes. We observed that prediction accuracy with the Orthogonal Centroid dimension reduction algorithm was preserved for SVMs as well as with centroid-based classification. The Orthogonal Centroid dimension reduction method maximizes the between cluster relationship using the relatively inexpensive reduced QR decomposition, compared to LDA/GSVD which also considers the within cluster relationship but requires a more expensive rank revealing decomposition such as the singular value decomposition.

The better prediction accuracy using SVMs is due to low generalization error by maximizing the margin, and the capability to handle non-linearity by kernel choice. Although most classes of the Reuters-21578 data set are linearly separable [12], there seems to be some level of non-linearity. For non-linearly separable data, SVMs with appropriate nonlinear kernel functions would work as a better classifier. Another way to handle non-linearly separable data is to apply nonlinear extensions of the dimension reduction methods, including those presented in [17, 18]. All of the dimension reduction methods presented here can also be applied to visualize the higher dimensional structure by reducing the dimension to 2- or 3-dimensional space.

7 Acknowledgements

The authors would like to thank University of Minnesota Supercomputing Institute (MSI) for providing the computing facilities.

References

[1] M. W. Berry, Z. Drmac, and E. R. Jessup. Matrices, vector spaces, and information retrieval. *SIAM Review*, 41:335–362, 1999.

[2] M. W. Berry, S. T. Dumais, and G. W. O'Brien. Using linear algebra for intelligent information retrieval. *SIAM Review*, 37:573–595, 1995.

[3] M. W. Berry and R. D. Fierro. Low-rank orthogonal decompositions for information retrieval applications. *Numerical linear algebra with applications*, 3(4):301–327, 1996.

[4] Å. Björck. *Numerical Methods for Least Square Problems*. SIAM, Philadelphia, PA, 1996.

[5] N. Cristianini and J. Shawe-Taylor. *Support Vecor Machines and other kernel-based learning methods*. Cambride University Press, 2000.

[6] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *J. of the Society for information Science*, 41:391-407, 1990.

[7] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Second ed., Academic Press, 1990.

[8] G. H. Golub and C. F. Van Loan. *Matrix Computations, third edition*. Johns Hopkins University Press, Baltimore, 1996.

[9] M. Heiler. *Optimization Criteria and Learning Algorithms for Large Margin Classifiers*. Diploma Thesis, University of Mannheim., 2002.

[10] P. Howland, M. Jeon, and H. Park. Cluster structure preserving dimension reduction based on the generalized singular value decomposition, 2001. *SIAM J. Matrix Anal. Appl*, to appear.

[11] M. Jeon, H. Park, and J. B. Rosen. Dimensional reduction based on centroids and least squares for efficient processing of text data. In *Proceedings for the first SIAM international workshop on text mining*. Chicago, IL, 2001.

[12] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the European Conference on Machine Learning*, pages 137–142, Berlin, 1998. Springer.

[13] H. Lodhi, N. Cristianini, J. Shawe-Taylor, and C. Watkins. Text classification using string kernels. *Advances in Neural Information Processing Systems*, 13:563–569, 2000.

[14] C. C. Paige and M. A. Saunders, *Towards a generalized singular value decomposition*, SIAM J. Numer. Anal., 18 (1981), pp. 398–405.

[15] H. Park, M. Jeon, and J. B. Rosen. Lower dimensional representation of text data based on centroids and least squares, *BIT*, to appear.

[16] H. Park and L. Eldén. Downdating the rank-revealing URV decomposition. *SIAM J. Matrix Anal. Appl*, 16 (1995), pp. 138-155.

[17] C. Park and H. Park. Nonlinear feature extraction based on centroids and kernel functions. Technical report 02-041, Department of Computer Science and Engineering, University of Minnesota, 2002.

[18] C. Park and H. Park. Kernel discriminant analysis based on the generalized singular value decomposition, in preparation.

[19] G. Salton, *The SMART Retrieval System*, Prentice Hall, 1971.

[20] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill, 1983.

[21] G. W. Stewart. An updating algorithm for subspace tracking. *IEEE Transactions on Signal Processing*, 40:1535–1541, 1992.

[22] G. W. Stewart. Updating URV decompositions in parallel. *Parallel Computing*, 20(2):151–172, 1994.

[23] M. Stewart and P. Van Dooren. Updating a generalized URV decomposition. *SIAM J. Matrix Anal. Appl.*, 22(2):479–500, 2000.

[24] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, 1999.

[25] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, 1979.

[26] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995.

[27] V. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, New York, 1998.

[28] Y. Yang and X. Liu. A re-examination of text categorization methods. In *22nd Annual International SIGIR*, pages 42–49, Berkley, August 1999.