Efficient Multi-way Text Categorization via Generalized Discriminant Analysis

Tao Li
Computer Science Dept.
University of Rochester
Rochester, NY 14627-0226
taoli@cs.rochester.edu

Shenghuo Zhu*
Computer Science Dept.
University of Rochester
Rochester, NY 14627-0226
zsh@cs.rochester.edu

Mitsunori Ogihara
Computer Science Dept.
University of Rochester
Rochester, NY 14627-0226
ogihara@cs.rochester.edu

ABSTRACT

Text categorization is an important research area and has been receiving much attention due to the growth of the on-line information and of Internet. Automated text categorization is generally cast as a multi-class classification problem. Much of previous work focused on binary document classification problems. Support vector machines (SVMs) excel in binary classification, but the elegant theory behind large-margin hyperplane cannot be easily extended to multi-class text classification. In addition, the training time and scaling are also important concerns. On the other hand, other techniques naturally extensible to handle multi-class classification are generally not as accurate as SVM. This paper presents a simple and efficient solution to multi-class text categorization. Classification problems are first formulated as optimization via discriminant analysis. Text categorization is then cast as the problem of finding coordinate transformations that reflects the inherent similarity from the data. While most of the previous approaches decompose a multi-class classification problem into multiple independent binary classification tasks, the proposed approach enables direct multi-class classification. By using Generalized Singular Value Decomposition (GSVD), a coordinate transformation that reflects the inherent class structure indicated by the generalized singular values is identified. Extensive experiments demonstrate the efficiency and effectiveness of the proposed approach.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; I.2 [**Artificial Intelligence**]: Learning; I.5 [**Pattern Recognition**]: Applications

General Terms

Algorithms, Measurement, Performance, Experimentation, Verification

*The current affiliation: Amazon.com, Inc.

CIKM'03, November 3–8, 2003, New Orleans, Louisiana, USA.

Keywords

Multi-class Text Categorization, GSVD, Discriminant Analysis

1. INTRODUCTION

With the ever-increasing growth of the on-line information and the permeation of Internet into daily life, methods that assist users in organizing large volumes of documents are in huge demand. In particular, automatic text categorization has been extensively studied recently. This categorization problem is usually viewed as supervised learning, where the gaol is to assign predefined category labels to unlabeled documents based on the likelihood inferred from the training set of labeled documents. Numerous approaches have been applied, including Bayesian probabilistic approaches [20, 31], nearest neighbor [22, 19], neural networks [33], decision trees [2], inductive rule learning [4, 9], support vector machines [18, 14], Maximum Entropy [26], boosting [28], and linear discriminate projection [3] (see [34] for comparative studies of text categorization methods).

Although document collections are likely to contain many different categories, most of the previous work was focused on binary document classification. One of the most effective binary classification techniques is the support vector machines (SVMs) [32]. It has been demonstrated that the method performs superbly in binary discriminative text classification [18, 34]. SVMs are accurate and robust, and can quickly adapt to test instances. However, the elegant theory behind the use of large-margin hyperplanes cannot be easily extended to multi-class text categorization problems. A number of techniques for reducing multi-class problems to binary problems have been proposed, including one-versus-the-rest method, pairwise comparison [16] and error-correcting output coding [8, 1]. In these approaches, the original problems are decomposed into a collection of binary problems, where the assertions of the binary classifiers are integrated to produce the final output. In practice, which reduction method is best suited is problem-dependent, so it is a non-trivial task to select the decomposition method. Indeed, each reduction method has its own merits and limitations [1]. In addition, regardless of specific details, these reduction techniques do not appear to be well suited for text categorization tasks with a large number of categories, because training of a single, binary SVM requires $O(n^{\alpha})$ time for $1.7 \leq \alpha \leq 2.1$ where n is the number of training data [17]. Thus, having to train many classifiers has a significant impact on the overall training time. Also, the use of multiple classifiers slows down prediction. Thus, despite its elegance and superiority, the use of SVM may not be best suited for multi-class document classification. However, there do not appear to exist many alternatives, since many other techniques that can be naturally extended to handle multi-class classification problems,

such as neural networks and decision trees, are not so accurate as SVMs [34, 35].

In statistics pattern recognition literature, discriminant analysis approaches are well known to be able to learn discriminative feature transformations (see, e.g., [12]). For example, Fisher discriminant analysis [10] finds a discriminative feature transformation as eigenvectors associated with the largest eigenvalues of matrix $T = \hat{\Sigma}_w^{-1}\hat{\Sigma}_b$, where $\hat{\Sigma}_w$ is the intra-class covariance matrix and $\hat{\Sigma}_b$ is the inter-class covariance matrix[1]. Intuitively, T captures not only compactness of individual classes but separations among them. Thus, eigenvectors corresponding to the largest eigenvalues of T are likely to constitute a discriminative feature transform. However, for text categorization, $\hat{\Sigma}_w$ is usually singular owing to the large number of terms. Simply removing the null space of $\hat{\Sigma}_w$ would eliminate important discriminant information when the projections of $\hat{\Sigma}_b$ along those directions are not zeros [12]. This issue has stymied attempts to use traditional discriminant approaches in document analysis.

In this paper we resolve this problem. We extend discriminant analysis and present a simple, efficient, but effective solution to text categorization. We propose a new optimization criterion for classification and cast text categorization as the problem of finding transformations to reflect the inherent similarity from the data. In this framework, given a document of unknown class membership, we compare the distance of the new document to the centroid of each category in the transformed space and assign it to the class having the smallest distance to it. We call this method Generalized Discriminant Analysis (GDA), since it uses generalized singular value decomposition to optimize transformation. We show that the transformation derived using GDA is equivalent to optimization via the trace or determinant ratios.

GDA has several favorable properties: First, it is simple and can be programed in a few lines in MATLAB. Second, it is efficient. (Most of our experiments only took several seconds.) Third, the algorithm does not involve parameter tuning. Finally, and probably the most importantly, it is very accurate. We have conducted extensive experiments on various datasets to evaluate its performance. The rest of the paper is organized as follows: Section 2 reviews the related work on text categorization. Section 3 introduces our new criterion for discriminant analysis. Section 4 introduces the basics of generalized singular value decomposition and gives the solution of the optimization problem. Section 5 shows that the transformation derived using GDA can also be obtained by optimizing the trace or determinant ratios. Section 6 presents some illustrating examples. Section 7 shows experimental results. Finally, Section 8 provides conclusions and discussions.

2. RELATED WORK

Text categorization algorithms can be roughly classified into two types: those algorithms that can be naturally extended to handle multi-class cases and those require decomposition into binary classification problems. The first consists of such algorithms as Naive Bayes [22, 19], neural networks [25, 33], K-Nearest Neighbors [22, 19], Maximum Entropy [26] and decision trees. Naive Bayes uses the joint distributions of words and categorizes to estimate the probabilities that an input document belongs to each document class and then selects the most probable class. K-Nearest Neighbor finds the k nearest neighbors among training documents and uses the categories of the k neighbors to determine the category of the test document. The underlying principle of maximum entropy is that without external knowledge, uniform distribution should be preferred. Based on this principle, it estimate the conditional distribution of the class label given a document.

The reduction techniques that are used by the second group include one-versus-the-rest method [29], error-correcting output coding [8], pairwise comparison [16], and multi-class objective functions, where the first two have been applied to text categorization [34, 13].

In the one-versus-the-rest method a classifier separating between from a class and the rest is trained for each class. Multi-class classification is carried out by integrating prediction of these individual classifiers with a strategy for resolving conflicts. The method is sometimes criticizes for solving asymmetric problems in a symmetrical manner and for not considering correlations between classes.

Error-correcting output coding (ECOC) [8] partitions the original set of classes into two sets in many different ways. A binary classifier is trained for each partition. The partitions are carefully chosen so that the outputs of these classifiers assign a unique binary codeword for each class (with a large Hamming distance between any pair of them). The class of an input with unknown class membership is chosen by computing the outputs of the classifiers on that input and then finding the class with the codeword closest to the output codeword.

Although SVMs are considered to be very effective in binary classification, its large training costs may make it unsuitable for multi-class classification with a large number of classes if the above decomposition techniques are applied. Also, the lack of a clear winner among the above techniques makes the reduction task complicated. Our GDA directly deals with multi-class classification and does not require reduction to binary classification problems.

Other techniques for text categorization exist. Godbole et al. [14] propose a new multi-class classification technique that exploits the accuracy of SVMs and the speed of Naive Bayes. It uses a Naive Bayes classifier to compute a confusion matrix quickly. Then it uses this matrix to reduce both the number and the complexity of binary SVMs to be built. Chakrabarti et al. [3] propose a fast text classification technique that uses multiple linear projections. It first projects training instances to low-dimensional space and then builds decision tree classifiers on the projected spaces. Fragoudis et al. [11] propose a new algorithm that targets both feature and instance selection for text categorization.

In summary, as pointed out in [34, 26], there is no obvious winner in multi-class classification techniques. For practical problems, the choice of approach will have to be made depending on the constraints, e.g., the desired accuracy level, the time available, and the nature of the problem.

3. NEW CRITERION FOR DISCRIMINANT ANALYSIS

3.1 Classification as Discrimination

Suppose the dataset D has m instances, $d_1, \ldots, d_m$, having p features each. Then D can be viewed as a subset of R^p as well as a member of $R^{m\times p}$. Suppose D has L classes, $D_1, \ldots, D_L$ having $m_1, \ldots, m_L$ instances, respectively, where $m = \sum_{i=1}^{L} m_i$. For each i, $1 \le i \le L$, let J_i be the set of all j, $1 \le j \le m$, such that the j-th instance belongs to the i-th class, and let $c^{(i)}$ be the centroid of the i-th class, *i.e.*, the component-wise average of the m_i vectors in the

[1]This is equivalent to using eigenvectors associated with the smallest eigenvalues of matrix $T = \hat{\Sigma}_b^{-1}\hat{\Sigma}_w$. It indicates that traditional discriminant analysis requires the non-singularity of at least one covariance matrix. Since the rank of $\hat{\Sigma}_w$ is usually greater than that of $\hat{\Sigma}_b$, we will base our discussion on the eigenvalue-decomposition of $T = \hat{\Sigma}_w^{-1}\hat{\Sigma}_b$.

class. Let c be the centroid of the entire dataset. The *intra-class scatter matrix* of D, $\hat{\Sigma}_w$, is defined by

$$\hat{\Sigma}_w = \sum_{i=1}^{L} \sum_{j \in J_i} (d_j - c^{(i)})^T (d_j - c^{(i)}),$$

and its *inter-class scatter matrix*, $\hat{\Sigma}_b$, is defined by

$$\hat{\Sigma}_b = \sum_{i=1}^{L} \sum_{j \in J_i} (d_j - c)^T (d_j - c).$$

Let A_w be the $m \times p$ matrix constructed by stacking $D_1 - (e^{(1)})^T c^{(1)}$, ..., $D_L - (e^{(L)})^T c^{(L)})$ and let A_b be the $p \times m$ matrix whose columns are, from left to right, $\sqrt{m_1}(c^{(1)} - c)^T, \cdots, \sqrt{m_L}(c^{(L)} - c)^T$. Then

$$\hat{\Sigma}_w = A_w A_w^T \text{ and } \hat{\Sigma}_b = A_b A_b^T.$$

Although there are ways (such as Kernel tricks [24]) for utilizing non-linear transformation, we will focus on linear transformation. Given a linear transformation Φ, the covariance matrices in the transformed space are

$$(A_b\Phi)^T (A_b\Phi) = \Phi^T A_b^T A_b \Phi = \Phi^T \hat{\Sigma}_b \Phi$$

and

$$(A_w\Phi)^T (A_w\Phi) = \Phi^T A_w^T A_w \Phi = \Phi^T \hat{\Sigma}_w \Phi.$$

Fisher's linear discriminant analysis discriminates inter-class distance and intra-class distance by using their corresponding covariance matrices. The optimal projection can be obtained by solving the generalized eigenvalue problem:

$$\hat{\Sigma}_b \Phi = \lambda \hat{\Sigma}_w \Phi \tag{1}$$

If $\hat{\Sigma}_w$ is nonsingular, Φ is given by the eigenvectors of matrix $\hat{\Sigma}_w^{-1}\hat{\Sigma}_b$. As we already pointed out, the approach fails if $\hat{\Sigma}_w$ is singular which is often the case in document classification[2]. Usually, this problem is overcome by using a nonsingular intermediate space of $\hat{\Sigma}_w$ obtained by removing the null space of $\hat{\Sigma}_w$ and then computing eigenvectors. However, the removal of the null space of $\hat{\Sigma}_w$ possibly eliminates some useful information because some of the most discriminant dimensions may be lost by the removal. In fact, the null space of $\hat{\Sigma}_w$ is guaranteed to contain useful discriminant information when the projections of $\hat{\Sigma}_b$ are not zeros along those directions. Thus, simple removal of the null space of $\hat{\Sigma}_w$ is not an effective resolution [12].

Once the transformation Φ has been determined, classification is performed in the transformed space based on a distance metrics, such as Euclidean distance

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_i (x_i - y_i)^2}$$

and cosine measure

$$d(\mathbf{x}, \mathbf{y}) = 1 - \frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2}\sqrt{\sum_i y_i^2}}.$$

A new instance, $\mathbf{z}$, it is classified to

$$\operatorname*{argmin}_k d(\mathbf{z}\Phi, \bar{\mathbf{x}}_k\Phi), \tag{2}$$

where $\bar{\mathbf{x}}_k$ is the centroid of k-th class.

[2]In fact, $\hat{\Sigma}_w$ is nonsingular only if there are $p + L$ samples. This is usually impractical.

3.2 The New Criterion

We propose the use of the following criterion for discriminating inter-class and intra-class distances by inter-class and intra-class covariance matrices:

$$\min_{\Phi} \{ \|A_b\Phi - I_n\|_F^2 + \|A_w\Phi\|_F^2 \}, \tag{3}$$

where $\|X\|_F$ is the Frobenius norm of the matrix X, *i.e.*, $\sqrt{\sum_{i,j} x_{ij}^2}$. The criterion does not involve the inverse of the intra-class matrix and is similar to Tikhonov regularization of least squares problems. Intuitively, the first term of (3) is used to minimize the difference between the projection of $\bar{\mathbf{x}}_i - \bar{\mathbf{x}}$ in a new space and the i-th unit vector of the new space. The second term is used to minimize the intra-class covariance.

The equation (3) can be rewritten as

$$\min_{\Phi} \left\| \begin{bmatrix} A_w \\ A_b \end{bmatrix} \Phi - \begin{bmatrix} 0 \\ I_n \end{bmatrix} \right\|_F^2, \tag{4}$$

and this is a least squares problem with the solution

$$(A_w^T A_w + A_b^T A_b)\Phi = A_b^T. \tag{5}$$

4. GENERALIZED SINGULAR VALUE DECOMPOSITION

Here we will show how to use GSVD to compute efficiently the solution to the optimization problem formulated in Section 3 and show that the solution thus obtained is stable.

4.1 The Basics of GSVD

Singular value decomposition (SVD) is a process of decomposing a rectangular matrix into three other matrices of a very special form. It can be viewed as a technique for deriving a set of uncorrelated indexing variables or factors [6]. A *Generalized Singular Value Decomposition* (GSVD) is an SVD of a sequence of matrices. GSVD has played a significant role in signal processing and in signal identification and has been widely used in such problems as source separation, stochastic realization and generalized Gauss-Markov estimation.

The diagonal form of GSVD, shown below, was first introduced in [21].

THEOREM 1. **(GSVD Diagonal Form [21])** *If $A \in R^{m \times p}$, $B \in R^{n \times p}$, and $rank(A^T, B^T) = k$, then there exist two orthogonal matrices, $U \in R^{m \times m}$ and $V \in R^{n \times n}$, and a non-singular matrix, $\Theta \in R^{p \times p}$, such that*

$$\begin{bmatrix} U^T & 0 \\ 0 & V^T \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} X = \begin{bmatrix} C \\ S \end{bmatrix} \begin{bmatrix} I_k & 0 \end{bmatrix} \tag{6}$$

where C and S are nonnegative diagonal and of dimension $m \times k$ and $n \times k$, respectively, $1 \geq S_{11} \geq \cdots S_{\min(n,k),\min(n,k)} \geq 0$, and $C^T C + S^T S = I_k$.

The *generalized singular values* are defined to be the component-wise ratios of the diagonal entries of the two diagonal matrices. In signal processing, A is often the signal matrix and B is the noise matrix, in which case the generalized singular values are referred to as signal-noise ratios.

4.2 Stable Solution

By plugging the GSVD matrices of A_w and A_b in (5), we have $\Phi = X \begin{bmatrix} \mathbf{I_k} \\ \mathbf{0} \end{bmatrix} S^T V^T$. Since V is orthogonal, we can drop it without

changing the squared distance. So, we have

$$\Phi = X \begin{bmatrix} \mathbf{I_k} \\ \mathbf{0} \end{bmatrix} S^T. \quad (7)$$

This derivation of Φ holds even if $\hat{\Sigma}_w$ is singular. Thus, by using GSVD to solve the new criterion, we can avoid removing null space, thereby keeping all the useful information. The degree of linear independence of the original data, $\text{rank}(A_w^T, A_b^T)$, is equal to k, Since $\Phi \in R^{p\times k}$, $\text{rank}((A_w\Phi)^T, (A_b\Phi)^T)$, the degree of linear independence in the transformed space, is at most k.

We now state a theorem that shows that the solution is stable.

THEOREM 2. **(GSVD relative perturbation bound [7])** *Suppose A and B be matrices with the same number of columns and B is of full column rank. Let $A = A_1D_1$ and $B = B_1D_2$ such that D_1 and D_2 have full rank. Let $E = E_1D_1$ and $F = F_1D_2$ be perturbations of A and B, respectively, such that for all x there exist some $\eta_1, \eta_2 < 1$ for which it holds that*

$$||E_1x||_2 \le \eta_1||A_1x||_2, \quad ||F_1x||_2 \le \eta_2||B_1x||_2.$$

Let σ_i and $\tilde{\sigma}_i$ be the i-th generalized singular value of (A,B) and that of $(A+E, B+F)$, respectively. Then either $\sigma_i = \tilde{\sigma}_i = 0$ or

$$\frac{|\sigma_i - \tilde{\sigma}_i|}{\sigma_i} \le \frac{\eta_1 + \eta_2}{1 - \eta_2}.$$

The above theorem gives a bound on the relative error of the generalized eigenvalues (C_{ii} and S_{ii}) if the difference between the estimated covariance matrices and the genuine covariance matrices is small. This guarantees that the relative error of Φ is bounded by the relative error of estimated intra- and inter-class covariance matrices.

GSVD also brings some favorable features, which might improve accuracy. In particular, computation of the cross products $A_b^TA_b$ and $A_w^TA_w$, which causes roundoff errors, is not required.

4.3 The GDA Algorithm

The pseudo codes of the training and prediction procedures are described as follows:

Algorithm 1 Training procedure Φ = *Train* (x**'s)**

Input: the training data x_i's
Output: the transformation Φ;
begin
1. Construct the matrices A_w and A_b;
2. Perform GSVD on the matrix pair;
3. Obtain Φ as described in equation 7.
4. **Return** Φ;

end

Algorithm 2 Prediction Procedure T = *Predict* (Φ, **x)**

Input: the transformation Φ generated by the training procedure; and a new instance x;
Output: the label T of the new instance;
begin
1. Perform Prediction as in equation 2;
2. **Return** T;

end

5. CONNECTIONS

Here we show that the above transformation derived using our new criterion can also be obtained by optimizing the trace or determinant ratios.

5.1 Optimizing the determinant ratio

Fisher's criterion is to maximize the ratio of the determinant of the inter-class scatter matrix of the projected samples to the determinant of the intra-class scatter matrix of the projected samples:

$$J(\Phi) = \frac{|\Phi^T\hat{\Sigma}_b\Phi|}{|\Phi^T\hat{\Sigma}_w\Phi|}. \quad (8)$$

One way to overcome the requirements of non-singularity of Fisher's criterion is looking for solutions that simultaneously maximize $|\Phi^T\hat{\Sigma}_b\Phi|$ minimize $|\Phi^T\hat{\Sigma}_w\Phi|$. Using GSVD, A_b and A_w are decomposed as $A_w = UC[\mathbf{I_k}\ \mathbf{0}]\mathbf{X^{-1}}$ and $A_b = VS[\mathbf{I_k}\ \mathbf{0}]\mathbf{X^{-1}}$. To maximize $J(\Phi)$, $|\Phi^T\hat{\Sigma}_b\Phi|$ should be increased while decreasing $|\Phi^T\hat{\Sigma}_w\Phi|$. Let $C' = C[\mathbf{I_k}\ \mathbf{0}]$ and $S' = S[\mathbf{I_k}\ \mathbf{0}]$. Then we have $\hat{\Sigma}_b = A_b^TA_b = XS'^2X^{-1}$ and $\hat{\Sigma}_w = A_w^TA_w = XC'^2X^{-1}$. This implies

$$\begin{aligned} |\Phi^T\hat{\Sigma}_b\Phi| &= |\Phi^TXS'^2X^{-1}\Phi| \\ &= (|S'X^{-1}\Phi|)^2 \text{ and} \\ |\Phi^T\hat{\Sigma}_w\Phi| &= |\Phi^TXC'^2X^{-1}\Phi| \\ &= (|C'X^{-1}\Phi|)^2. \end{aligned}$$

Thus, the matrix Φ satisfying $X^{-1}\Phi = \begin{bmatrix} \mathbf{I_k} \\ \mathbf{0} \end{bmatrix}$ would simultaneously maximize $|\Phi^T\hat{\Sigma}_b\Phi|$ and minimize $|\Phi^T\hat{\Sigma}_w\Phi|$ (since the diagonal of S is decreasing). So, we have $\Phi = X \begin{bmatrix} \mathbf{I_k} \\ \mathbf{0} \end{bmatrix}$. In the case where we must weight the transformation with the generalized singular, $\Phi = X \begin{bmatrix} \mathbf{I_k} \\ \mathbf{0} \end{bmatrix} S^T$ is the transformation we want.

5.2 Optimizing the trace ratio

The same transformation can also be obtained by optimizing the trace ratio. Using GSVD, we have

$$\begin{aligned} \text{trace}(\Phi^T\hat{\Sigma}_b\Phi) &= \text{trace}(S'S'^TX^{-1}\Phi\Phi^TX^{-T}) \\ &= \text{trace}(S'S'^TGG^T) \\ &= \sum_{i=1}^{k} S_{ii}^2 g_{ii} \text{ and} \\ \text{trace}(\Phi^T\hat{\Sigma}_w\Phi) &= \text{trace}(C'C'^TX^{-1}\Phi\Phi^TX^{-T}) \\ &= \text{trace}(C'C'^TGG^T) \\ &= \sum_{i=1}^{k} C_{ii}^2 g_{ii}, \end{aligned}$$

where $G = X^{-1}\Phi$ and g_{ii} is the ii-th term of G. Since $C^TC + S^TS = I_k$, we have

$$\begin{aligned} \text{trace}(\Phi^T\hat{\Sigma}_b\Phi) + \text{trace}(\Phi^T\hat{\Sigma}_w\Phi) &= \sum_{i=1}^{k} S_{ii}^2 g_{ii} + \sum_{i=1}^{k} C_{ii}^2 g_{ii} \\ &= \sum_{i=1}^{k} g_{ii}. \end{aligned}$$

If we force that $\text{trace}(\Phi^T\hat{\Sigma}_b\Phi) = 1$, the optimization is formulated as minimization of $\text{trace}(\Phi^T\hat{\Sigma}_w\Phi) = \sum_{i=1}^{k} g_{ii} - 1$. Here g_{ii}'s are diagonal elements of a positive semi-definite matrix, so they are nonnegative. Also, for all i, $g_{ii} = 0$ implies that for all j

$g_{ij} = g_{ji} = 0$. Note that GG^T is a $p \times p$ matrix. Since only the first k diagonal entries, $\{g_{ii}\}_{i=1}^{k}$, appear in the formula for $\text{trace}(\Phi^T \hat{\Sigma}_w \Phi) = \sum_{i=1}^{k} g_{ii} - 1$, the quantities of other $m-k$ diagonal entries do not affect the optimization. Thus, we may set all of these to 0, thereby obtaining $\Phi = X \begin{bmatrix} \mathbf{I_k} \\ \mathbf{0} \end{bmatrix}$. In the case when we want to weight the transformation with the generalized singular values, we obtain $\Phi = X \begin{bmatrix} \mathbf{I_k} \\ \mathbf{0} \end{bmatrix} S^T$.

6. TEXT CLASSIFICATION VIA GDA: EXAMPLES

A well-known transformation method in information retrieval is Latent Semantic Indexing (LSI) [6], which applies Singular Value Decomposition (SVD) to the document-term matrix and computes eigenvectors having largest eigenvalues as the directions related to the dominant combinations of the terms occurring in the dataset (*latent semantics*). A transformation matrix constructed from these eigenvectors projects a document onto the latent semantic space. Although LSI has been proven extremely useful in information retrieval, it is not optimal for text categorization because LSI is completely unsupervised. In other words, LSI deals with the data without paying any particular attention to the underlying class structure. It only aims at optimally transforming the original data into a lower dimensional space with respect to the mean squared error, which has nothing to do with the discrimination of the different classes. Our GDA approach possesses advantages of both discriminant analysis and of latent semantic analysis. By explicitly taking the intra-class and inter-class covariance matrices into the optimization criterion, GDA deals directly with discrimination between classes. Furthermore, by employing GSVD to solve the optimization problem, GDA tries to identify the latent concepts indicated by the generalized singular values.

To illustrate how well GDA can perform, we present here two examples. In the first example, we compare GDA against LDA and LSI. Figure 1 shows a small dataset consisting of nine phrases in three topics: user interaction, graph theory, and distributed systems.

No.	Class	Phrase
1	1	Human interface for user response
2	1	A survey of user opinion of computer system response time
3	1	Relation of user-perceived response time to error measurement
4	2	The generation of random, binary, unordered trees
5	2	The intersection graph of paths in trees
6	2	Graph Minors IV: Widths of trees and well-quasi-ordering
7	3	A survey of distributed shared memory system
8	3	RADAR: A multi-user distributed system
9	3	Management interface tools for distributed computer system

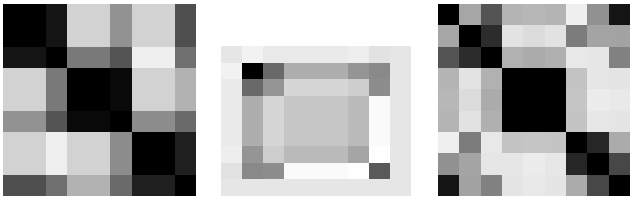
Figure 1: Nine example sentences

After removing words (terms) that occurs only once, we have the document-term matrix as shown in Figure 2.

The first and second samples in each class are used for training. GDA, LDA, and LSI are run on the training data to obtain transformation matrices. Figure 3 shows the plot of the distances/similarities between document pairs in the transformed space using each of the three methods.

word\No.	1	2	3	4	5	6	7	8	9
a		1					1	1	
computer		1							1
distributed							1	1	1
for	1								1
graph					1	1			
interface	1								1
of		2	1	1	1	1	1		
response	1	1	1						
survey		1					1		
system		1					1	1	1
the				1	1				
time		1	1						
trees				1	1	1			
user	1	1	1					1	

Figure 2: Document-term Matrix



(a) GDA (b) LDA (c) LSI

Figure 3: Pairwise document similarity via GDA, LDA, and LSI. The darker the close is the more similar the documents are. GDA is a clear winner.

The second example illustrates differences between GDA and LSI. Distinction among three newsgroups in 20NG are attempted by selecting from each newsgroup twenty training and twenty for testing. Figure 4 shows plots of the the sixty testing articles using the two dominant directions as the axes. GDA has clear separation while the LSI plot shows an L-shaped concentration of the data points. The confusion matrices of these methods are shown in Table 1. GDA clearly performed better than LSI.

	prediction		
actual	1	2	3
1	20	0	0
2	0	19	1
3	0	0	0

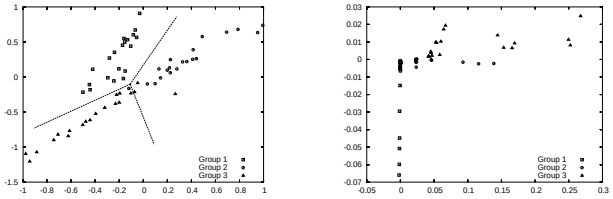
	prediction		
actual	1	2	3
1	20	0	0
2	0	3	17
3	7	5	8

Table 1: The confusion matrices. Left: GDA. Right: LSI.

7. EXPERIMENTS

7.1 The Datasets

For our experiments we used a variety of datasets, most of which are frequently used in the information retrieval research. The range of the number of classes is from four to 105 and the range of the number of documents is from 476 to 20,000, which seem varied



(a) GDA (b) LSI

Figure 4: Document plots. The three groups are separated significantly better with GDA than with LSI.

enough to obtain good insights as to how GDA performs. Table 2 summarizes the characteristics of the datasets.

20Newsgroups The 20Newsgroups (20NG) dataset contains approximately 20,000 articles evenly divided among 20 Usenet newsgroups. The raw text size is 26MB. All words were stemmed using a porter stemming program, all HTML tags were skipped, and all header fields except subject and organization of the posted article were ignored.

WebKB The WebKB dataset[3] contains Web pages collected from university computer science departments. There are approximately 8,300 documents in the set and they are divided into seven categories: student, faculty, staff, course, project, department, and other. The raw text size of the dataset is 27MB. Among the seven categories, student, faculty, course, and project are the four most populous. The subset consisting only of these categories is also used here, which is called **WebKB4**. In neither of the datasets, we used stemming or stop lists.

Industry Sector The Industry Section dataset[4] is based on the data made available by Market Guide, Inc. (www.marketguide.com). The set consists of company homepages that are categorized in a hierarchy of industry sectors, but we disregarded the hierarchy. There were 9,637 documents in the dataset, which were divided into 105 classes. We tokened the documents by skipping all MIME and HTML headers and using a standard stop list. We did not perform stemming.

Reuters The Reuters-21578 Text Categorization Test Collection contains documents collected from the Reuters newswire in 1987. It is a standard text categorization benchmark and contains 135 categories. We used its subsets: one consisting of the ten most frequent categories, which we call **Reuters-Top10**, and the other consisting of documents associated with a single topic, which we call **Reuters-2**. Reuters-2 had approximately 9,000 documents and 50 categories.

TDT2 TDT2 is the NIST Topic Detection and Tracking text corpus version 3.2 released in December 6, 1999 [30]. This corpus contains news data collected daily from nine news sources in two languages (American English and Mandarin Chinese), over a period of six months (January–June in 1998). We used only the English news texts, which were collected from New York Times Newswire Service, Associated Press Worldstream Service, Cable News Network, Voice of America, American Broadcasting Company, and Public Radio International. The documents were manually annotated using 96 target topics. We selected the documents having annotated topics and removed the brief texts. The resulting dataset contained 7,980 documents.

K-dataset This dataset was obtained from the WebACE project [15]. It contained 2,340 documents consisting of news articles from Reuters News Service made available on the Web in October 1997. These documents were divided into 20 classes. They were processed by eliminating stop words and HTML tags, stemming the remaining words using Porter's suffix-stripping algorithm.

CSTR This is the dataset of the abstracts of technical reports published in the Department of Computer Science at the University of Rochester between 1991 and 2002[5]. The dataset contained 476 abstracts, which were divided into four research areas: Symbolic-AI, Spatial-AI, Systems, and Theory. We processed the abstracts by removing stop words and applying stemming operations on the remaining words.

Datasets	# documents	# class
20NG	20,000	20
WebKB4	4,199	4
WebKB	8,280	7
Industry Sector	9,637	105
Reuters-Top10	2,900	10
Reuters-2	9,000	50
CSTR	476	4
K-dataset	2,340	20
TDT2	7,980	96

Table 2: Data Sets Descriptions

7.2 Data Preprocessing

In all experiments, we randomly chose 70% of the documents for training and assigned the rest for testing. It is suggested in [35] that information gain is effective for term removal and it can remove up to 90% or more of the unique terms without performance degrade. So, we first selected the top 1,000 words by information gain with class labels. The feature selection is done with the Rainbow package [23].

Here we use classification accuracy for evaluation. Different measures, such as precision-recall graphs and F_1 measure [34], have been used in the literature. However, since the datasets used in our experiments are relatively balanced and single-labeled, and our goal in text categorization is to achieve low misclassification rates and high separation between different classes on a test set, we thought that accuracy is the best measure of performance. All of our experiments were carried out on a P4 2GHz machine with 512M memory running Linux 2.4.9-31.

7.3 Experimental Results

Now we present and discuss the experimental results. Here we compare GDA against Naive Bayes (NB for short), K-Nearest Neighbor (KNN for short), Maximum Entropy (ME for short), LDA, and SVM on the same datasets with the same training and testing data. Recall that the first three of the methods we compare against are commonly-used direct methods for multi-class classification (in the sense that they do not require reduction to binary classification problems). For experiments involving SVM we used SVMTorch [5][6], which uses the one-versus-the-rest decomposition.

Table 3 and Figure 5 show performance comparisons. GDA outperformed all the other five methods on 20NG, WebKB4, WebKB and Industry Sector. SVM performed the best on Reuters-2,

[3]Both 20NG and WebKB are available at http://www-2.cs.cmu.edu/afs/cs/project/theo-11/www/wwkb.

[4]Available at http://www.cs.cmu.edu/ TextLearning/datasets.html

[5]The TRs are available at http://www.cs.rochester.edu/trs.

[6]Download-able at http://old-www.idiap.ch/learning/SVMTorch.html.

K-dataset, and TDT2. GDA outperformed LDA on all the experiments, and the improvement was significant (more than 10%) when the sample size was relatively small (in the case of CSTR, Reuters-Top10, and K-dataset).

On 20NG, the performance of GDA s 95.03%, which is approximately 10% higher than that of NB, 6% higher than that of ME, and 4% higher than that of SVM. On the WebKB4 dataset, GDA beats NB by approximately 5%, and both ME and SVM by approximately 2%. On the WebKB dataset, GDA beats NB by approximately 16% and ME by 6%. The performance of GDA is about 8% higher than that of NB and by 6% than that of ME on the Industry Sector. The results with GDA and with SVM are almost the same on WebKB, Industry Sector, Reuters-Top10, and CSTR. On Reuters-2, K-dataset, and TDT2, SVM performs slightly better than GDA by 3%. ME achieves the best results on the CSTR dataset while NB is the winner on Reuters-top10 in terms of performance On CSTR, the performance of GDA is 2% lower than that of NB and 4% lower than that of ME. On Reuters-Top10, GDA is beaten by NB by approximately 1%. In total, the performance of GDA is always either the winner or very close to the winner: it is ranked the first four times, ranked the second three times, and ranked the third in the remaining two. Although there is no single winner over all datasets, GDA seems to outperform the rest on most counts. We can say that GDA is a viable, competitive algorithm in text categorization.

Datasets	**GDA**	NB	KNN	ME	LDA	SVM
20NG	**95.03**	85.60	50.70	89.06	93.90	91.07
WebKB4	**94.01**	85.13	37.29	91.93	90.72	92.04
WebKB	**79.02**	61.01	44.81	71.30	77.35	78.89
Industry Sector	**66.57**	56.32	39.48	58.84	66.49	65.96
Reuters-Top10	81.98	**83.33**	74.07	81.65	71.46	81.13
Reuters-2	89.82	87.88	73.22	88.56	88.65	**92.43**
CSTR	88.50	90.85	82.53	**92.39**	68.29	88.71
K-dataset	88.44	86.14	58.26	86.19	77.69	**91.90**
TDT2	90.54	91.59	86.63	89.18	88.41	**93.85**

Table 3: Performance comparisons. For KNN we set k to 30.

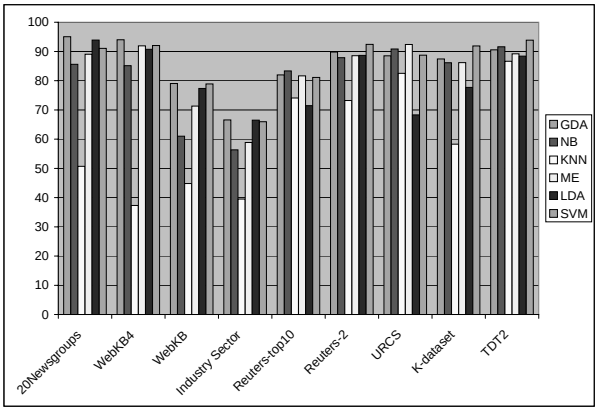


Figure 5: Performance Comparison

GDA is also very efficient and most experiments are done in several seconds. Table 4 summarizes the running time for all the experiments of GDA and SVM. Figure 6 and Figure 7 present the comparisons of training and prediction time respectively. The time saving of GDA is very obvious. In summary, these experiments have shown that GDA provides an alternate choice for fast and efficient text categorization.

	GDA	GDA	SVM	SVM
Datasets	Training	Prediction	Training	Prediction
20NG	171.80	6.86	270.20	64.28
WebKB4	63.4	0.20	114.67	54.72
WebKB	94.64	0.43	1108.17	103.03
Industry Sector	88.23	6.45	423.54	79.82
Reuters-Top10	61.23	0.15	94.28	18.65
Reuters-2	96.19	1.13	566.53	85.10
CSTR	3.65	0.02	7.50	2.77
K-dataset	62.88	0.18	84.56	47.70
TDT2	21.69	5.14	89.91	26.76

Table 4: Time Table in seconds.

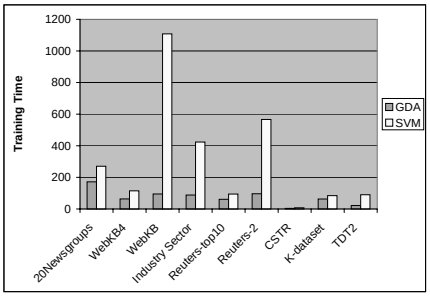


Figure 6: Training Time Comparisons

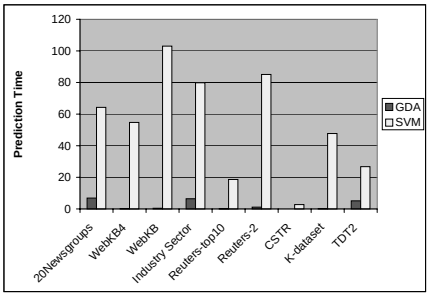


Figure 7: Prediction Time Comparisons

8. DISCUSSIONS AND CONCLUSIONS

In this paper, we presented GDA, a simple, efficient, and yet accurate, direct approach to multi-class text categorization. GDA utilizes GSVD to transform the original data into a new space, which could reflect the inherent similarities between classes based on a new optimization criterion. Extensive experiments clearly demonstrate its efficiency and effectiveness.

Interestingly enough, although traditional discriminant approaches have been successfully applied in pattern recognition, little work has been reported on document analysis. As we mentioned earlier, this is partly because the intra-class covariance matrix is usually singular for document-term data and hence restrict the usage of discriminant. Our new criterion avoids the problem while still preserving the discriminative power of the covariance matrix.

Another big barrier to application of discriminant analysis in document classification is its large computation cost. As we know, traditional discriminant analysis requires a large amount of computation on matrix inversion, SVD, and eigenvalue-analysis. The costs of these operations are extremely large in document analysis because the matrices have thousands of dimension. Our approach makes use of effective feature selection via information gain, with which we can remove up to 90% or more of the unique terms without significant performance degrade [35]. One of our future plans is to explore how the performance correlates with different feature selection methods and the number of words selected. There are also other possible extensions such as using random projection to reduce the dimensionality before applying discriminant analysis [27].

Acknowledgments

This work is supported in part by NSF grants EIA-0080124, DUE-9980943, and EIA-0205061, and NIH grant P30-AG18254.

9. REFERENCES

[1] Allwein, E. L., Schapire, R. E., & Singer, Y. (2000). Reducing multiclass to binary: A unifying approach for margin classifiers. *ICML-00* (pp. 9–16).

[2] Apte, C., Damerau, F., & Weiss, S. (1998). Text mining with decision rules and decision trees. *Proceedings of the Workshop with Conference on Automated Learning and Discovery: Learning from text and the Web*.

[3] Chakrabarti, S., Roy, S., & Soundalgekar, M. V. (2002). Fast and accurate text classification via multiple linear discriminant projections. *Proceedings of the 28th International Conference on Very Large Databases* (pp. 658–669).

[4] Cohen, W. W., & Singer, Y. (1996). Context-sensitive learning methods for text categorization. *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information* (pp. 307–315).

[5] Collobert, R., & Bengio, S. (2001). SVMTorch: Support vector machines for large-scale regression problems. *Journal of Machine Learning Research*, *1*, 143–160.

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

[7] Demmel, J., & Veselić, K. (1992). Jacobi's method is more accurate than QP. *SIAM Journal on Matrix Analysis and Applications*, *13*, 10–19.

[8] Dietterich, T. G., & Bakiri, G. (1995). Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, *2*, 263–286.

[9] Dumais, S., Platt, J., Heckerman, D., & Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. *CIKM-98* (pp. 148–155).

[10] Fisher, R. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, *7*, 179–188.

[11] Fragoudis, D., Meretakis, D., & Likothanassis, S. (2002). Integrating feature and instance selection for text classification. *SIGKDD-02* (pp. 501–506).

[12] Fukunaga, K. (1990). *Introduction to statistical pattern recognition*. Academic Press.

[13] Ghani, R. (2000). Using error-correcting codes for text classification. *ICML-00* (pp. 303–310).

[14] Godbole, S., Sarawagi, S., & Chakrabarti, S. (2002). Scaling multi-class support vector machine using inter-class confusion. *SIGKDD-02* (pp. 513–518).

[15] Han, E.-H., Boley, D., Gini, M., Gross, R., Hastings, K., Karypis, G., Kumar, V., Mobasher, B., & Moore, J. (1998). WebACE: A web agent for document categorization and exploration. *Agents-98* (pp. 408–415).

[16] Hastie, T., & Tibshirani, R. (1998). Classification by pairwise coupling. *Advances in Neural Information Processing Systems*. The MIT Press.

[17] Joachims, T. (1998). Making large-scale support vector machine learning practical. In *Advances in kernel methods: Support vector machines*.

[18] Joachims, T. (2001). A statistical learning model of text classification with support vector machines. *SIGIR-01* (pp. 128–136).

[19] Lam, W., & Ho., C. (1998). Using a generalized instance set for automatic text categorization. *SIGIR-98* (pp. 81–89).

[20] Lewis, D. D. (1998). Naive (Bayes) at forty: The independence assumption in information retrieval. *ECML-98*.

[21] Loan, C. V. (1976). Generalizing the singular value decomposition. *SIAM J. Num. Anal.*, *13*, 76–83.

[22] Masand, B., Linoff, G., & Waltz., D. (1992). Classifying news stories using memory based reasoning. *SIGIR-92* (pp. 59–64).

[23] McCallum, A. K. (1996). Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. http://www.cs.cmu.edu/ mccallum/bow.

[24] Mika, S., Rätsch, G., Weston, J., Schölkopf, B., & Müller, K.-R. (1999). Fisher discriminant analysis with kernels. *Neural Networks for Signal Processing IX* (pp. 41–48). IEEE.

[25] Ng, H. T., Goh, W. B., & Low, K. L. (1997). Feature selection, perceptron learning, and a usability case study for text categorization. *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information* (pp. 67–73).

[26] Nigam, K., Lafferty, J., & McCallum, A. (1999). Using maximum entropy for text classification. *In IJCAI-99 Workshop on Machine Learning for Information Filtering* (pp. 61–67).

[27] Papadimitriou, C. H., Tamaki, H., Raghavan, P., & Vempala, S. (1998). Latent semantic indexing: A probabilistic analysis. *Proceedings of the Symposium on Principles of Database Systems* (pp. 159–168).

[28] Schapire, R. E., & Singer, Y. (2000). Boostexter: A boosting-based system for text categorization. *Machine Learning*, *39*, 135–168.

[29] Scholkopf, B., & J.Smola, A. (2002). *Learning with kernels*. MIT Press.

[30] TDT2 (1998). Nist topic detection and tracking corpus. http://www.nist.gove/speech/tests/tdt/tdt98/index.htm.

[31] Tzeras, K., & Hartmann, S. (1993). Automatic indexing based on Bayesian inference networks. *SIGIR-93* (pp. 22–34).

[32] Vapnik, V. N. (1998). *Statistical learning theory*. Wiley, New York.

[33] Wiener, E. D., Pedersen, J. O., & Weigend, A. S. (1995). A neural network approach to topic spotting. *4th Annual Symposium on Document Analysis and Information Retrieval* (pp. 317–332).

[34] Yang, Y., & Liu, X. (1999). A re-examination of text categorization methods. *SIGIR-99* (pp. 42–49).

[35] Yang, Y., & Pederson, J. O. (1997). A comparative study on feature selection in text categorization. *ICML-97* (pp. 412–420).