On Applying Linear Discriminant Analysis for Multi-labeled Problems *

Cheong Hee Park * Moonhwi Lee

Dept. of Computer Science and Engineering
Chungnam National University
220 Gung-dong, Yuseong-gu
Daejeon, 305-763, Korea

Abstract

Linear discriminant analysis (LDA) is one of the most popular dimension reduction methods, but it is originally focused on a single-labeled problem. In this paper, we derive the formulation for applying LDA for a multi-labeled problem. We also propose a generalized LDA algorithm which is effective in a high dimensional multi-labeled problem. Experimental results demonstrate that by considering multi-labeled structure, LDA can achieve computational efficiency and also improve classification performances.

Key words: Dimension Reduction, Linear Discriminant Analysis, Multi-labeled Problems, Text categorization

* This work was supported by the Korea Research Foundation Grant funded by the Korean Government(MOEHRD, Basic Research Promotion Fund)(KRF-2006-331-D00510)

* Tel: 82-42-821-6651, Fax: 82-42-822-4997

Email addresses: cheonghee@cnu.ac.kr (Cheong Hee Park), moone81@cnu.ac.kr (Moonhwi Lee).

Preprint submitted to Elsevier 29 November 2007
1 Introduction

While traditional classification problem assumes that a data sample belongs to only one class among the predefined classes, a multi-labeled problem can arise in real situation where a data sample is associated with multiple class labels. For example, in text categorization documents can be classified to multiple categories of topics (Lewis et al., 2004). In bioinformatics, each gene is associated with a set of functional classes (Pavlidis et al., 2001). In a multi-labeled problem, the main task is to output a set of class labels associated with a new unseen data sample. One common way to deal with a multi-labeled problem is to transform it to several binary problems. For each class, a binary problem is constructed where data samples belonging to the class compose the positive set and the remaining data makes up the negative set. Outputs from each binary classifier are combined to produce a set of class labels for new data (Schapire and Singer, 2000; Elisseeff and Weston, 2002; Zhang and Zhou, 2005; Godbole and Sarawagi, 2004).

When the data dimensionality is high, learning a classifier in a high dimensional space can be difficult, known as the curse of dimensionality. It has been shown that performing dimension reduction as a preprocessing step can improve classification performances greatly, especially for high dimensional data such as text documents, microarray, or face images (Torkkola, 2001; Belhumeur et al., 1997; Nguyen and Rocke, 2002). By extracting a small number of most optimal features for an intended main task, original data is transformed to a low dimensional space where learning process can be performed more efficiently. Among several statistical dimension reduction methods, Linear discriminant analysis (LDA) performs dimension reduction to maximize class separability in the reduced dimensional space (Fukunaga, 1990). Due to this objective criterion, LDA can be most optimal
for classification tasks. In a multi-labeled problem, LDA can be applied independently for each binary problem as in a traditional single-labeled problem. However, the classes may not be independent from each other and dimension reduction applied independently to each binary problem would accelerate distortion of relations among classes. Hence it would be desirable to apply dimension reduction based on data samples and their associated class labels all together.

In this paper, we first derive formulations for applying LDA in a multi-labeled problem, especially focusing on high dimensional data. Since LDA was originally developed for a single-labeled problem, the objective function should be interpreted suitably when it is applied for a multi-labeled problem. We also propose a computationally efficient LDA algorithm for a multi-labeled problem with a small sample size. The proposed method can save computational costs and memory requirements by utilizing QR-decomposition. And through the generation of semi-artificial data samples, it overcomes problems occurred due to a small sample size and multiple class labels.

The paper is organized as follows. In Section 2, brief reviews for a multi-labeled problem and generalized LDA algorithms are given. In Section 3, some problems which need to be considered in applying generalized LDA algorithms for multi-labeled data are discussed and motivations for a new approach to overcome those problems are described. In Section 4, we propose an efficient dimension reduction method which is more effective in multi-labeled data sets. Experimental results in Section 5 compare performances of dimension reduction methods under various conditions. Conclusions follow in Section 6.
2 Multi-labeled Classification and Linear Discriminant Analysis

2.1 Multi-labeled Classification

Let $X = \{x_1, x_2, \cdots, x_k\}$ be a set of data samples, and a data sample is represented as a vector in a $m$-dimensional space such as $x_i = [x_{1i}, \cdots, x_{mi}]^T$. The notation $T$ denotes the transpose of a vector or a matrix. We also assume that data samples can have one or more class labels assigned to them among the predefined $r$ classes. Let $Y = \{y_1, y_2, \cdots, y_k\}$ be the set of class labels for the data samples in $X$. Each $y_i = [y_{i1}, \cdots, y_{ir}]$ is a vector denoting class labels associated with $x_i$ such that $y_{ij} = 1$ if $x_i$ belongs to the class $j$, and $y_{ij} = 0$ if $x_i$ does not belong to the class $j$.

In a multi-labeled problem, for a new unseen data sample the classification task is to output a set of class labels associated with it. Instead of making a hard decision whether a data sample belongs to a class or not, multi-labeled classification can produce confidence level at which a data sample is assigned to each class so that $r$ class labels can be ordered according to their confidence levels. By setting a threshold to accept the class label, ranking based classification can be transformed to hard-decision-making classification. Also note that a single-label problem is a special case of a multi-label problem, in which each data sample is to have only one positive class label. From the multi-labeled data, binary problems are constructed by the positive samples and the negative samples for each class. For the constructed binary problems, single-label classification methods such as Support vector machines, kernel methods, $k$-nearest neighbor classifier were applied (Elisseeff and Weston, 2002; Zhang and Zhou, 2005; Godbole and Sarawagi, 2004). A maximum entropy based method was also developed which explored correlations among classes (Zhu et al., 2005).
Classification performance in a multi-labeled problem can be evaluated by several measures. Among them, we introduce one-error and f1-measure. We refer to the papers (Schapire and Singer, 2000; Luo and Zincir-Heywood, 2005) for more details about evaluation measures. One-error is used for multi-labeled classification which produces only one positive class label. For a ranking based classifier, the class label with the highest rank is only considered in one-error measurement. Let us assume that $T$ is a collection of new unseen data samples. For each $x$ in $T$, $t_x$ denotes the set of the true class labels of $x$ and $p_x$ is the predicted class label. One-error measures the probability that the predicted class label is not one of the true class labels as follows.

$$\text{one-error} = \frac{1}{|T|} \left| \{ x \in T | p_x \notin t_x \} \right|,$$

(1)

where $| \cdot |$ means the cardinality of the set. One-error is actually same as prediction error in a single-labeled problem.

The f1-measure was originally used in information retrieval. For binary classification, the precision ($p$), recall ($r$) and f1-measure are defined such as

$$p = \frac{TP}{TP + FP}, \quad r = \frac{TP}{TP + FN}, \quad f1 = \frac{2pr}{p + r},$$

(2)

where TP represents the number of positive samples which are predicted as positive, FP is the number of negative samples which are predicted as positive, and FN is the number of positive samples which are predicted as negative. In a multi-labeled problem, f1 value averaged over all binary classification is macro-averaged f1 measure. On the other hand, when TP, FP, FN are first summed over all the binary classification respectively and then f1 is calculated based on them, it is called micro-averaged f1 measure. In our experiments, we use one-error and micro-averaged f1
measure to evaluate performances.

2.2 Linear Discriminant Analysis (LDA)

LDA utilizes the between-class scatter and within-class scatter as a means to measure class separability. When the distance between classes is maximal and the scatterness within classes is minimal, it is an ideal clustering structure for classification. LDA finds projection to a low dimensional space that maximizes class separability.

We first review LDA in a single-labeled problem (Fukunaga, 1990). Let us represent the data set as

$$A = \{a^1_1, \cdots, a^1_{n_1}, \cdots, a^r_1, \cdots, a^r_{n_r}\},$$  \hspace{1cm} (3)

where \{a^i_1, \cdots, a^i_{n_i}\} are the data samples belonging to the class \(i\) and the total number of data is \(n = n_1 + \cdots + n_r\). The between-class scatter matrix \(S_b\), the within-class scatter matrix \(S_w\), and the total scatter matrix \(S_t\) are defined as

$$S_b = r \sum_{i=1}^{r} n_i (c_i - c)(c_i - c)^T, \hspace{1cm} S_w = \sum_{i=1}^{r} n_i \sum_{j=1}^{n_i} (a^i_j - c_i)(a^i_j - c_i)^T, \hspace{1cm} S_t = \sum_{i=1}^{r} \sum_{j=1}^{n_i} (a^i_j - c)(a^i_j - c)^T,$$  \hspace{1cm} (4)

using the class centroids \(c_i = \frac{1}{n_i} \sum_{j=1}^{n_i} a^i_j\) and the global centroid \(c = \frac{1}{n} \sum_{i=1}^{r} \sum_{j=1}^{n_i} a^i_j\). The trace which is defined as the sum of the diagonal components of a matrix (Golub and Loan, 1996) gives measures for the between-class scatter and the within-class scatter such as

$$\text{trace}(S_b) = \sum_{i=1}^{r} n_i \|c_i - c\|^2, \hspace{1cm} \text{trace}(S_w) = \sum_{i=1}^{r} \sum_{j=1}^{n_i} \|a^i_j - c_i\|^2.$$  \hspace{1cm} (5)

One of optimization criteria in LDA is to find a linear transformation \(G^T\) which
maximizes

\[ J = \text{trace}((G^T S_w G)^{-1}(G^T S_b G)), \]

(6)

where \( G^T S_i G \) for \( i = b, w \) is the scatter matrix in the transformed space by \( G^T \). It is well known (Fukunaga, 1990) that \( J \) in (6) is maximized when the columns of \( G \) are composed of the eigenvectors \( g \) corresponding to the \( r - 1 \) largest eigenvalues \( \lambda \) of

\[ S_w^{-1} S_b g = \lambda g. \]

(7)

Now suppose data samples can have more than one class label. Let \( X = \{x_1, x_2, \cdots, x_k\} \) and \( Y = \{y_1, y_2, \cdots, y_k\} \) denote data samples and their class label vectors respectively as in section 2.1. In computing the objective function \( J \) in (6), a data sample with multiple class labels should contribute to all classes for which it is a positive sample. We compose the data set \( A \) in (3) by taking all data samples positive to each class. Hence a data sample with multiple class labels appears several times in \( A \) and the total number \( n \equiv n_1 + \cdots + n_r \) can be greater than the number of the original data samples. LDA is applied to the composed set \( A \) as in a single-labeled problem. Let us call this approach LDA-ALL. If a data sample belongs to two classes, what role does it play in separating two classes and minimizing scatter within each class at the same time? A data sample belonging to multiple classes may cause contradiction between the maximization of the between-class distance and the minimization of the within-class scatter, since it is placed in the overlapping area of multiple classes. But, on the contrary it can make the maximization of the between-class distance and the minimization of the within-class scatter get balanced.
The other way to apply LDA for multi-labeled data is to transform it to multiple binary problems and perform LDA for each binary problem. We call this approach LDA-BIN. In this setting, dimension reduction should be performed as many times as the number of classes and it does not reflect correlations among classes. In our experiments, we demonstrate that the classification performance is better when multiple class labels are considered all together as in LDA-ALL rather than when it is transformed to binary problems.

Dimension reduction is most effective in high dimensional data as in text categorization. Often high dimensional data is closely related with undersampled problems, where the number of data samples is smaller than the data dimension. In the classical LDA, $S_w$ is assumed to be nonsingular and the problem becomes computing eigenvectors of $S_w^{-1}S_b$ as in (7). However, in undersampled problems, all of the scatter matrices become singular and the classical LDA is difficult to apply. In order to make LDA applicable for undersampled problems, several methods have been proposed (Friedman, 1989; Yu and Yang, 2001; Chen et al., 2000; Yang and Yang, 2003; Howland and Park, 2004; Zheng et al., 2004). In the next section, we discuss applying generalized LDA algorithms for multi-labeled high dimensional data.

3 On applying LDA for multi-labeled problems

In undersampled problems, the minimization of the within-class scatter can be accomplished by using the null space of the within-class scatter matrix $S_w$, since the projection by the vectors in the null space of $S_w$ makes the within-class scatter zero. It has been shown that a linear transformation based on the null space of $S_w$ can improve classification performance greatly (Chen et al., 2000; Yang and Yang,
2003; Howland and Park, 2004; Zheng et al., 2004). On the other hand, the maximization of \( \text{trace}(G^T S_b G) \) suggests that the column vectors of \( G \) should come from the range space of \( S_b \), since for any \( g \in \text{null}(S_b) \)

\[
\text{trace}(g^T S_b g) = \text{trace} \left( \sum_{i=1}^{r} n_i g^T (c_i - c) (c_i - c)^T g \right) = \sum_{i=1}^{r} n_i \| g^T c_i - g^T c \|^2 = 0, \tag{8}
\]

and therefore all the class centroids become equal to the global centroid in the projected space.

Generalization of LDA for undersampled problems can be characterized by the two-step process of the minimization of the within-class scatter and the maximization of the between-class scatter where one of them is performed after the other. In the next sections, based on which one in two steps is first applied, we explore the applicability of generalized LDA algorithms for a multi-labeled problem in LDA-ALL approach.

3.1 Maximizing the between-class distance in the first stage

The method by Yu and Yang (Yu and Yang, 2001) first transforms the original space by using a basis of range \((S_b)\). Then in the transformed space the minimization of the within-class scatter is performed by the eigenvectors corresponding to the smallest eigenvalues of the within-class scatter matrix. This method is called Direct LDA (or DLDA). The computation in DLDA can be very efficient by taking advantage of the singular value decomposition (SVD) for the smaller matrix \( H_b^T H_b \in \mathbb{R}^{r \times r} \) instead of \( S_b = H_b H_b^T \in \mathbb{R}^{m \times m} \), where \( H_b = [\sqrt{n_1}(c_1 - c), \cdots, \sqrt{n_r}(c_r - c)] \in \mathbb{R}^{m \times r} \) for the class centroids \( c_i \) and the global centroid \( c \). Our experiments show that in LDA-ALL approach, DLDA obtains competitive performances while maintaining low computational complexities.
3.2 Minimizing the within-class scatter in the first stage

Unlike DLDA, the methods in (Chen et al., 2000; Yang and Yang, 2003; Howland and Park, 2004; Zheng et al., 2004) utilize the null space of $S_w$, more specifically $\text{null}(S_w) \cap \text{range}(S_b)$, in the first stage, and the maximization of the between-class distance is pursued further in the second stage. Now we discuss some difficulties with using the null space of $S_w$ in a multi-labeled problem. When a vector from the null space of $S_w$ is used for the projection, it should be also in the range space of $S_t$. If it belongs to null($S_t$) then it also belongs to null($S_b$), which is undesirable as described in (8). Since range($S_w$) is the orthogonal complement of null($S_w$),

$$\mathbb{R}^m = \text{range}(S_w) \oplus \text{null}(S_w),$$

where $\oplus$ denotes the direct sum of the vector spaces range($S_w$) and null($S_w$) (Kolman and Hill, 2005). Hence we have

$$\dim(\text{range}(S_t)) = \dim(\text{range}(S_t) \cap \text{range}(S_w)) + \dim(\text{range}(S_t) \cap \text{null}(S_w)). \quad (9)$$

As the ratio of data samples which are not independent or belong to multiple classes increases, the dimension of range($S_t$) $\cap$ null($S_w$) becomes lower, possibly to zero. It is because the subtraction of class centroids from each data sample makes rank reduction in range($S_w$) less than in range($S_t$). In order to visualize it, using a real data set we tested effects of multi-labeled data on the space range($S_t$) $\cap$ null($S_w$). From the largest eight classes in the Reuter-21578 text dataset, independent 286 documents were chosen to construct a base set. Adding documents with multiple class labels to a base set, dim(range($S_t$)) and dim(range($S_t$) $\cap$ range($S_w$)) were computed. As shown in Figure 2, as data samples with multiple class labels are added to the base set more and more, the dimension of range($S_t$) $\cap$ range($S_w$) gets
to close to the dimension of range($S_t$), and therefore the dimension of range($S_t$) $\cap$ null($S_w$) goes down to near zero. It implies that any algorithms utilizing the null space of $S_w$ will suffer from the shrinking of the space range($S_t$) $\cap$ null($S_w$).

3.3 Motivation for a new algorithm in multi-labeled problems

Most of generalized LDA algorithms mainly rely on the Singular value decomposition (SVD) in order to compute eigenvectors. But, the computational complexities and memory requirements for the SVD can be very demanding, especially for high dimensional data. Zheng et al.’s method called GSLDA (Zheng et al., 2004) obtains computational efficiency by using QR-decomposition which is cheaper than the SVD. However GSLDA assumes that the given data samples are independent. Hence it can not work well in LDA-ALL approach. Even in LDA-BIN setting, it performs poorly when data samples are nearly dependent as will be demonstrated in our experiments.
In the next section, we propose a LDA algorithm which utilizes QR decomposition and overcomes the shrinking of \( \text{range}(S_t) \cap \text{null}(S_w) \) in multi-labeled problems. Before going into technical details of our new algorithm, more intuitive and illustrating motivation can be given as follows. In order to make simplified visualization, let us represent two orthogonal spaces \( \text{range}(S_w) \) and \( \text{null}(S_w) \) by corn areas around two perpendicular axes as in Figure 2. We should note that Figure 2 is only symbolic visualization to show the idea and it can not be mathematically correct drawing. The figure (a) illustrates that the vector \( x \) in \( \text{range}(S_t) \) is expressed as the sum of the orthogonal projection of \( x \) onto the range space of \( S_w \) and the orthogonal projection of \( x \) onto the null space of \( S_w \), which means

\[
x = \text{proj}_{\text{range}(S_w)}x + \text{proj}_{\text{null}(S_w)}x.
\]

The figure (b) symbolically shows three vectors \( x_i, i = 1, 2, 3 \), in \( \text{range}(S_t) \) whose projected components to \( \text{null}(S_w) \) are zeros, in other words,

\[
\text{proj}_{\text{null}(S_w)}x_i = x_i - \text{proj}_{\text{range}(S_w)}x_i = 0.
\]

This illustrates the shrinking of \( \text{range}(S_t) \cap \text{null}(S_w) \) which results from the dimension of \( \text{range}(S_t) \) being close to the dimension of \( \text{range}(S_t) \cap \text{range}(S_w) \). Now in order to prevent the shrinking of \( \text{range}(S_t) \cap \text{null}(S_w) \), we compute the components projected to \( \text{null}(S_w) \) by using any \( x_j \) in the place of \( x_i \) such as

\[
x_i - \text{proj}_{\text{range}(S_w)}x_j, \text{ for any } j.
\]

The figure (c) shows the components derived by (10). The extended space of \( \text{range}(S_t) \cap \text{null}(S_w) \) is further proceeded to the next step for between-class scatter maximization.
Supervised learning is based on the expectation that new unseen data would come from the same distribution as training data. The vector from $\text{range}(S_t) \cap \text{null}(S_w)$ is ideal projective direction for training data, but it is too optimistic to expect that new data would perfectly fit to the data model based on small training data. Over-fitting to the training data leads to generalization errors. Eq. (10) gives the effects of generating artificial data which can slightly deviate from $\text{range}(S_t) \cap \text{null}(S_w)$ and providing abundant input source to the next step for the maximization of the
between-class distance. Extending range\((S_t) \cap \text{null}(S_w)\) by the utilization of the training data can reduce the generalization errors caused by small sample size and efficient computation can be obtained by using QR decomposition which is cheaper than SVD.

4 An efficient LDA algorithm for multi-labeled problems

The scatter matrices in (4) can be computed as a product of the smaller matrices as follows:

\[
S_t = H_t H_t^T \in \mathbb{R}^{m \times m}, \quad S_w = H_w H_w^T \in \mathbb{R}^{m \times m}, \quad S_b = H_b H_b^T \in \mathbb{R}^{m \times m},
\]

where

\[
H_t = [a_1^1 - c, \ldots, a_{n_1}^1 - c, \ldots, a_1^r - c, \ldots, a_{n_r}^r - c] \in \mathbb{R}^{m \times n},
\]

\[
H_w = [a_1^1 - c_1, \ldots, a_{n_1}^1 - c_1, \ldots, a_1^r - c_r, \ldots, a_{n_r}^r - c_r] \in \mathbb{R}^{m \times n},
\]

\[
H_b = [\sqrt{n_1}(c_1 - c), \ldots, \sqrt{n_r}(c_r - c)] \in \mathbb{R}^{m \times r}.
\]

In order to obtain an orthonormal basis of range\((H_w)\), we apply QR-decomposition with column pivoting (Golub and Loan, 1996) for \(H_w \in \mathbb{R}^{m \times n}\) such as

\[
H_w \Pi_1 = Q_1 R_1,
\]

where \(\Pi_1\) is a column permutation in \(H_w\). When \(t = \text{rank}(H_w)\), \(Q_1 \in \mathbb{R}^{m \times t}\) has orthonormal columns and \(R_1 \in \mathbb{R}^{t \times n}\) has zeros below the main diagonal. The columns of \(Q_1\) make an orthonormal basis of range\((H_w)\), which is also an orthonormal basis of range\((S_w)\). Note that \(\text{null}(S_w)\) is the orthogonal complement of range\((S_w)\). Hence any vector \(x\) is uniquely expressed as the sum of the orthogonal projection of \(x\) onto the range space of \(S_w\) and the orthogonal projection of \(x\) onto
the null space of $S_w$ (Kolman and Hill, 2005). For $1 \leq i \leq r$ and $1 \leq j \leq n_t,$

$$a_j^i - c = \text{proj}_{\text{range}(S_w)}(a_j^i - c) + \text{proj}_{\text{null}(S_w)}(a_j^i - c),$$

(13)

where $\text{proj}_W(x)$ denotes the orthogonal projection of $x$ onto the space $W.$ Since the columns of $Q_1$ are an orthonormal basis for $\text{range}(S_w),$ $Q_1Q_1^T$ is the orthogonal projection onto $\text{range}(S_w)$ (Golub and Loan, 1996). Hence from (13), we have

$$\text{proj}_{\text{null}(S_w)}(a_j^i - c) = (a_j^i - c) - \text{proj}_{\text{range}(S_w)}(a_j^i - c)$$

$$= (a_j^i - c) - Q_1Q_1^T(a_j^i - c) \in \text{range}(S_t) \cap \text{null}(S_w).$$

Let

$$X = H_t - Q_1Q_1^T H_t.$$  

(14)

Then QR-decomposition with column pivoting for $X \in \mathbb{R}^{m \times n}$ gives an orthonormal basis $Q_2$ of $\text{range}(S_t) \cap \text{null}(S_w)$ as

$$X \Pi_2 = Q_2 R_2.$$  

(15)

In the above process, QR-decomposition can save computational complexities and memory requirement greatly compared with the SVD. However, there still exist some problems related with multi-labeled data. As shown in Fig 2, as the number of data samples with multiple class labels gets increased, the dimension of $\text{range}(S_t) \cap \text{null}(S_w)$ becomes zero. Therefore the above process cannot completely resolve the shrinking of $\text{range}(S_t) \cap \text{null}(S_w)$ caused by multi-labeled data. Now we propose a method to overcome the problem through the generation of semi-artificial data.

Let $P$ be any column permutation matrix on $H_t.$ Instead of $X = H_t - Q_1Q_1^T H_t$ in (14), we shuffle the columns of $H_t$ in the second term by using a permutation
matrix $P$:

$$
\hat{X} = H_t - Q_1 Q_1^T H_t P.
$$

(16)

It means that instead of

$$
\text{proj}_{\text{null}(S_w)}(a_j^i - c) = (a_j^i - c) - Q_1 Q_1^T (a_j^i - c) \in \text{range}(S_t) \cap \text{null}(S_w),
$$

we create artificial data samples

$$
\hat{x} \equiv (a_j^i - c) - Q_1 Q_1^T (a_l^s - c) \in \text{range}(S_t)
$$

for some $1 \leq l \leq r$ and $1 \leq s \leq n_l$. The generated data may deviate from the space $\text{range}(S_t) \cap \text{null}(S_w)$. However, this deviation from $\text{range}(S_t) \cap \text{null}(S_w)$ can prevent the shrinking of $\text{range}(S_t) \cap \text{null}(S_w)$ occurred by multi-labeled data. In this sense, we call $\hat{X}$ in 16 as the complement to $\text{range}(S_t) \cap \text{null}(S_w)$. Now from QR-decomposition with column pivoting for $\hat{X}$,

$$
\hat{X} \hat{\Pi}_2 = \hat{Q}_2 \hat{R}_2,
$$

(17)

where $\hat{\Pi}_2$ is a column permutation of $\hat{X}$, we obtain the complement to a basis of $\text{range}(S_t) \cap \text{null}(S_w)$, $\hat{Q}_2$.

In the projected space by $\hat{Q}_2$, the maximization of the between-class scatter is pursued. Let the SVD of $\hat{Q}_2^T H_b$ be

$$
\hat{Q}_2^T H_b = U_2 \Sigma_2 V_2^T.
$$

Then

$$
\hat{Q}_2^T S_b \hat{Q}_2 = U_2 (\Sigma_2 \Sigma_2^T) U_2^T
$$
Algorithm 1 LDA for multi-labeled problems using QR-decomposition

For a given data set \( A = \{ a_{ij} | 1 \leq i \leq r, 1 \leq j \leq n_i \} \), this algorithm computes a transformation matrix \( G \) by which dimension reduction for multi-labeled data is performed.

1. Compute QR-decomposition with column pivoting for \( H_w \):

\[
H_w \Pi_1 = Q_1 R_1.
\]

\( \Pi_1 \) is a column permutation and the columns of \( Q_1 \) are an orthonormal basis of \( \text{range}(H_w) = \text{range}(S_w) \).

2. Let \( P \) be any column permutation on \( H_t \).

3. Compute QR-decomposition with column pivoting for \( \hat{X} \equiv H_t - Q_1 Q_1^T H_t P \):

\[
\hat{X} \hat{\Pi}_2 = \hat{Q}_2 \hat{R}_2.
\]

\( \hat{\Pi}_2 \) is a permutation matrix and \( \hat{Q}_2 \) gives the complement to an orthonormal basis of \( \text{range}(S_t) \cap \text{null}(S_w) \).

4. Compute the SVD of \( \hat{Q}_2^T H_b \):

\[
\hat{Q}_2^T H_b = U_2 \Sigma_2 V_2^T.
\]

Let \( U_{21} \) be composed of the columns of \( U_2 \) corresponding to nonzero diagonal components of \( \Sigma_2 \).

5. \( G^T = (\hat{Q}_2 U_{21})^T \) gives a transformation for dimension reduction.

is the SVD of \( \hat{Q}_2^T S_b \hat{Q}_2 \). When \( U_{21} \) consists of the columns of \( U_2 \) corresponding to nonzero diagonal components of \( \Sigma_2 \), the projection by \( U_{21} \) realizes the maximization of the between-class scatter. Hence \( G = \hat{Q}_2 U_{21} \) gives a transformation matrix for dimension reduction. Algorithm 1 summarizes the proposed method.
When the size of the matrix $A$ is $m \times n$, the time complexity of $O(mn^2)$ is sufficient for QR-decomposition with column pivoting, while the SVD requires $O(m^2n)$ for obtaining a basis of the null space of $A$ (Golub and Loan, 1996). The efficiency of Algorithm 1 is achieved by replacing the SVD with QR decomposition of two times, which produces significant savings especially when the data dimension $m$ is much bigger than the number of data $n$.

5 Experimental Results

5.1 Data sets

We performed experiments to test performances of the proposed algorithm in multi-labeled problems of text categorization. The first data set is from Reuters corpus volume I which is a text categorization test collection of about 800,000 stories from Reuters newswire. We used a publically available version of it, RCV1-v2 (Lewis et al., 2004). As in (Yu et al., 2005), choosing a small part of data and picking up topics with more than 50 documents and words occurred in more than 5 documents, we composed a data set of 3307 documents with 5104 terms over 42 topics. A document belongs to 2.93 topics on average. One third of the data was used as a training set and two thirds was for a test set, and this splitting was randomly repeated ten times.

The second and third data sets were composed from Reuter-21578 \(^1\) which has been a very popular benchmark data set in text mining. Similar as in the first data set, taking topics with more than 50 documents and words occurred in more than

\(^1\) http://www.research.att.com/~lewis
<table>
<thead>
<tr>
<th>Data set</th>
<th>#docu</th>
<th>#term</th>
<th>#topic</th>
<th>avg. topics per docu</th>
</tr>
</thead>
<tbody>
<tr>
<td>data 1</td>
<td>3307</td>
<td>5104</td>
<td>42</td>
<td>2.93</td>
</tr>
<tr>
<td>data 2</td>
<td>6537</td>
<td>4347</td>
<td>23</td>
<td>1.17</td>
</tr>
<tr>
<td>data 3</td>
<td>2189</td>
<td>2993</td>
<td>21</td>
<td>1.71</td>
</tr>
</tbody>
</table>

Table 1
Data description

5 documents, we composed the second data set with 6537 documents and 4347 terms over 23 topics. In this data set, a document belongs to 1.17 topics on average. In order to make another data set with greater portion of multi-labeled data, some of single-labeled documents were randomly removed and the third data set was composed by 2189 documents and 2993 terms over 21 topics. In the third data set, a document belongs to 1.71 topics on average. Table 1 summarizes the data sets composed.

5.2 Choosing a permutation matrix

For Algorithm 1, we tested the following three approaches in choosing a permutation $P$.

(1) Use any random permutation.

(2) Use random permutations within each class.

(3) Use the permutation matrix $\Pi_1$ obtained by QR-decomposition of $H_w$ in (12).
While random permutation in the approach (1) makes almost all data samples permuted, using the permutation matrix $\Pi_1$ obtained by QR-decomposition of $H_w$ in (12) caused permutations of data samples of about 65 percent in some case of our experiments. However, three approaches did not make significant differences of classification performance in the experiments conducted. A reason why the permutation matrix $\Pi_1$ in (12) works like any random permutation can be explained as follows. The permutation matrix $\Pi_1$ moves independent columns in $H_w$ to the front producing an orthogonal basis of the range space of $H_w$, and dependent columns are moved to the back or remain in the original back positions. The columns of $H_t$ and $H_w$ in (11) are ordered similarly, and therefore the columns remaining in the back without permutation would be dependent on others and do not make great effects in generating semi-artificial data. In all experiments performed below, we report the results obtained from the third approach of using $\Pi_1$ obtained by QR-decomposition of $H_w$ in (12).

5.3 Performance analysis of LDA algorithms in multi-labeled problems

Several LDA algorithms including LDA (Yu and Yang, 2001), GSLDA (Zheng et al., 2004), PCA-LDA (Yang and Yang, 2003) and the proposed Algorithm 1 were applied. Dimension reduction is performed in two ways, LDA-ALL and LDA-BIN, as explained in Section 2.2. In both cases, a nearest neighbor classifier and microaveraged f1-measure were used for performance evaluation. Table 2 and 3 summarize the experimental results. The mean and standard deviations of f1 values obtained from 10 times splitting are shown.

In both tables, DLDA and the proposed Algorithm 1 demonstrate the best perfor-
1-NN in the original space

<table>
<thead>
<tr>
<th>Method</th>
<th>LDA-ALL</th>
<th>LDA-BIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using range($S_b$)</td>
<td>DLDA</td>
<td>0.737 (0.011)</td>
</tr>
<tr>
<td>Methods</td>
<td>GSLDA</td>
<td>0.449 (0.016)</td>
</tr>
<tr>
<td>using null($S_w$)</td>
<td>PCA-LDA</td>
<td>0.657 (0.010)</td>
</tr>
<tr>
<td>Algorithm 1</td>
<td>0.747 (0.010)</td>
<td>0.685 (0.007)</td>
</tr>
</tbody>
</table>

Table 2
Microaveraged f1-measures in the first data set.

<table>
<thead>
<tr>
<th></th>
<th>Data 2</th>
<th>Data 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-NN</td>
<td>0.793 (0.005)</td>
<td>0.722 (0.012)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>LDA-ALL</th>
<th>LDA-BIN</th>
<th>LDA-ALL</th>
<th>LDA-BIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLDA</td>
<td>0.851 (0.005)</td>
<td>0.802 (0.006)</td>
<td>0.799 (0.010)</td>
<td>0.738 (0.005)</td>
</tr>
<tr>
<td>GSLDA</td>
<td>0.613 (0.018)</td>
<td>0.490 (0.086)</td>
<td>0.527 (0.019)</td>
<td>0.684 (0.065)</td>
</tr>
<tr>
<td>PCA-LDA</td>
<td>0.737 (0.012)</td>
<td>0.725 (0.033)</td>
<td>0.726 (0.016)</td>
<td>0.767 (0.031)</td>
</tr>
<tr>
<td>Algorithm 1</td>
<td>0.864 (0.005)</td>
<td>0.802 (0.005)</td>
<td>0.803 (0.010)</td>
<td>0.738 (0.004)</td>
</tr>
</tbody>
</table>

Table 3
Microaveraged f1-measures in the second and third data sets.

...performance in LDA-ALL, compared with other methods utilizing the null space of the within-class scatter matrix. In LDA-BIN, since the data set in Table 2 has 42 topic classes, dimension reduction process was repeated 42 times for the constructed bi-
nary problems. While it makes the time complexity of LDA-BIN worse than LDA-ALL, the performance of LDA-BIN was also not good as in LDA-ALL. However, in both cases the dimension reduction by DLDA and Algorithm 1 improved the performances compared with those by a nearest neighbor classification in the original space, as shown in the first rows of Table 2 and 3. By dimension reduction, training a classifier is performed in very low dimensional space instead of high dimensional original space. Therefore it can save computational costs and also circumvent the curse of dimensionality.

5.4 Performance comparison of several multi-label classification algorithms

Next we compare the proposed Algorithm 1 with other multi-label classification algorithms. Support vector machine (SVM) is a well known state-of-the-art binary classification algorithm. In a multi-labeled problem, SVM can be applied to binary problems which are constructed from multi-labeled data in the one-vs-rest manner. Elisseeff et al. proposed a SVM-based multi-label classification algorithm, rank-SVM, and compared it with the binary SVM approach (Elisseeff and Weston, 2002). ML-KNN is a multi-label learning algorithm which combines a k-nearest neighbor (kNN) classifier with maximum a posterior (MAP) principle and its comparable performances with other multi-label classification algorithms were shown in (Zhang and Zhou, 2005).

Table 4 reports the comparison results of those algorithms by precision, recall and microaveraged f1 measures in three data sets. "1NN/LDA" and "ML-kNN/LDA" denote that each classification algorithm was applied after the preprocessing of dimension reduction. Since Algorithm 1 demonstrated competitive performances in the tests of the previous section, in this experiment we applied Algorithm 1 for
<table>
<thead>
<tr>
<th>Method</th>
<th>data 1</th>
<th>data 2</th>
<th>data 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>p</td>
<td>r</td>
<td>f1</td>
</tr>
<tr>
<td>1NN</td>
<td>0.64</td>
<td>0.64</td>
<td>0.64</td>
</tr>
<tr>
<td>1NN/LDA</td>
<td>0.79</td>
<td>0.71</td>
<td>0.75</td>
</tr>
<tr>
<td>bin. SVM</td>
<td>0.93</td>
<td>0.59</td>
<td>0.72</td>
</tr>
<tr>
<td>rank-SVM</td>
<td>0.76</td>
<td>0.59</td>
<td>0.66</td>
</tr>
<tr>
<td>ML-kNN(9)</td>
<td>0.83</td>
<td>0.57</td>
<td>0.68</td>
</tr>
<tr>
<td>ML-kNN(5)/LDA</td>
<td>0.82</td>
<td>0.70</td>
<td>0.76</td>
</tr>
</tbody>
</table>

Table 4
Performance comparison. (p: precision, r: recall, f1: microaveraged f1-measure.)

dimension reduction. For ML-kNN, the number of neighbors, k, was tested with 5, 7, and 9, and the best results among them are reported, although the difference was very little. Linear SVM was used for binary SVM and rank SVM while default value for the regularization parameter \( C \) was set to 1. Polynomial kernels with various degrees were also tried for rank-SVM, but the performances were not good as with linear SVM. It is worth to note that rank-SVM was implemented in Matlab following the pseudo-code of the algorithm in (Elisseeff and Weston, 2001).

As shown in Table 4, 1NN/LDA and ML-kNN/LDA obtained superior or comparable performances in all data sets. In particular, compared with 1NN or ML-kNN without dimension reduction, dimension reduction by Algorithm1 improved classification performances greatly. For rank-SVM, it produces high precision in most of cases, but recall measure was low compared with other methods, resulting in degraded f1-values. The binary SVM approach showed competent performances,
but we note that binary SVM learning needs to be performed in high dimensional space as many times as the number of classes, which costs high computational complexities. On the other hand, dimension reduction by Algorithm 1 is performed only once and the classification process is conducted in the low dimensional space instead of the original high dimensional space.

### 5.5 Performance in a highly correlated class label space

In order to test the performance of the proposed algorithm under the situation of high correlation in class label space, we composed an extreme case. Three classes were chosen from data 1, whose class distribution is shown in Figure 3. One third of the data was used as a training set and two thirds was for a test set, and this splitting was randomly repeated ten times. Table 5 shows microaveraged f1-measure as well as f1-measures in each class. For binary SVM, the construction of the binary problem by class 3-vs-rest was not possible, since all the data samples belong to class 3 and the negative set is empty. In that case, f1 measurement was computed by assuming all test data samples to belong to class 3. The results in Table 5 show that the proposed algorithm can be effective in a case of highly correlated class labels.
### Table 5
Performance in a highly correlated class label space.

<table>
<thead>
<tr>
<th>Method</th>
<th>f1 in each class</th>
<th>microaveraged</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1NN</td>
<td>0.752</td>
<td>0.779</td>
</tr>
<tr>
<td>1NN/LDA</td>
<td>0.832</td>
<td>0.877</td>
</tr>
<tr>
<td>binary SVM</td>
<td>0.855</td>
<td>0.893</td>
</tr>
<tr>
<td>rank-SVM</td>
<td>0.036</td>
<td>0.473</td>
</tr>
<tr>
<td>ML-kNN(9)</td>
<td>0.794</td>
<td>0.854</td>
</tr>
<tr>
<td>ML-kNN(7)/LDA</td>
<td>0.825</td>
<td>0.874</td>
</tr>
</tbody>
</table>

5.6 Performances on Single-labeled Problems

We also tested how the proposed method works for independent or nearly dependent data in a single-labeled problems. In this experiment, we used four text data sets \(^2\) which do not contain multi-labeled data. Each text data set was randomly split to the training and test set with the ratio of 1 : 1 and it is repeated ten times. One-error in (1) was used as a performance measure. In order to measure data independence, for each data set

\[
I = \frac{\text{rank}(S_t)}{\text{total number of data} - 1}
\]  

\(^2\) The data sets were downloaded and preprocessed from http://www-users.cs.umn.edu/~karypis/cluto/download.html.
was computed. The range of $I$ is $[0, 1]$, and if the data is independent, then $I$ equals to 1. Figure 4 (a) compares the $I$ values for each data set and Figure 4 (b) plots one-errors. In the data sets re0 and re1 whose $I$ values indicate data dependence, Algorithm 1 achieves superior performances to other methods. It indicates that the proposed method can be effective for dependent data sets in a single-labeled prob-
lem as well.

6 Conclusion

Linear discriminant analysis is one of the most popular dimension reduction methods, but some ambiguities and difficulties arise in applying it for a multi-labeled problem. In this paper, we derived the formulation for applying LDA for multi-labeled problems. We also proposed a generalization of LDA which can be most effective for high dimensional multi-labeled data. The proposed method utilizes QR-decomposition which is cheaper than SVD. By the permutation of data it obtains effects of generating semi-artificial data which can prevent shrinking of the null space of $S_w$ occurred due to multiple labels.

Instead of applying LDA for each binary problem transformed from a multi-labeled problem, in LDA-ALL approach LDA is only once applied to the whole data reflecting correlations among classes. Hence the computational cost is saved by avoiding the repeated processes of dimension reduction. Our experiments demonstrate that the classification performance can be improved greatly by dimension reduction with LDA-ALL approach where multiple class labels are considered all together.

References


system which can solve the small sample size problem. pattern recognition 33, 1713–1726.


