Available online at www.sciencedirect.com
ScienceDirect
Computational Statistics & Data Analysis 52 (2007) 423–437
COMPUTATIONAL STATISTICS & DATA ANALYSIS
www.elsevier.com/locate/csda

Improving implementation of linear discriminant analysis for the high dimension/small sample size problem

Jurjen Duintjer Tebbens[a,*], Pavel Schlesinger[b]

[a] *Institute of Computer Science, Czech Academy of Sciences, Pod Vodarenskou vezi 2, 182 07 Prague 8, Czech Republic*
[b] *Institute of Formal and Applied Linguistics, Charles University, Malostranske namesti 25, 118 00 Prague 1, Czech Republic*

Available online 15 February 2007

Abstract

Classification based on Fisher's linear discriminant analysis (FLDA) is challenging when the number of variables largely exceeds the number of given samples. The original FLDA needs to be carefully modified and with high dimensionality implementation issues like reduction of storage costs are of crucial importance. Methods are reviewed for the high dimension/small sample size problem and the one closest, in some sense, to the classical regular approach is chosen. The implementation of this method with regard to computational and storage costs and numerical stability is improved. This is achieved through combining a variety of known and new implementation strategies. Experiments demonstrate the superiority, with respect to both overall costs and classification rates, of the resulting algorithm compared with other methods.

Keywords: Linear discriminant analysis; Numerical aspects of FLDA; Small sample size problem; Dimension reduction; Sparsity

1. Introduction

Fisher's linear discriminant analysis (FLDA) takes as one of the basic and first methods a prominent place in supervised classification tasks. Even in the presence of more advanced and sophisticated classification techniques and today's necessity to handle high dimensional data, FLDA has not left the minds of researchers. In this paper we address FLDA for the case where the number of variables largely exceeds the number of objects. In the literature this case has several names; in the pattern recognition community one mostly calls it the "small sample size" problem (see, e.g. Chen et al., 2000; Howland et al., 2006), in more general statistical literature like Hastie and Tibshirani (2003) we rather find the expression "$p > n$" or even "$p \gg n$" problem. To emphasize that the problem lies in the combination of many variables and few samples, we will here use "high dimension/small sample size" problem. Many classification strategies like nearest neighbor or support vector machines can be used to solve high dimension/small sample size problems. In this paper we restrict ourselves to FLDA-based approaches with emphasis on computational aspects. Generally, choosing the proper classification method is a state-of-the-art problem of analysis practise which we consider out of the scope of this paper. Also, we are aware of the general theoretical questions on applicability common to all methods for the high dimension/small sample size case that cope with singularity of covariance matrices. For these issues we refer the interested reader in the first place to Friedman (1989) and also to Hoffbeck and Landgrebe (1996) and

* Corresponding author.
E-mail addresses: tebbens@cs.cas.cz (J. Duintjer Tebbens), schlesinger@ufal.mff.cuni.cz (P. Schlesinger).

0167-9473/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.csda.2007.02.001

Bensmail and Celeux (1996) for *regularized discriminant analysis* and related strategies to solve the classical tasks of linear and quadratic discriminant analysis in the high dimension/small sample size case. For FLDA in particular we refer to the paper by Krzanowski et al. (1995).

Assuming we have decided to use an FLDA-based approach (a good reason may be FLDA's relative simplicity), we will first compare various criteria used to adapt FLDA to the high dimension/small sample size problem. Then the main part of the paper addresses implementation of the chosen criterion with emphasis on computational and storage costs. With high dimensional data, these issues are of crucial importance for the efficiency of the whole process; improved implementation may change a seemingly uncomputable problem to a perfectly solvable one. In addition, numerical stability plays an important role in the high dimension/small sample size case. We will propose an algorithm that exploits all advantageous implementation strategies we know of and we add some new ones. Our experiments show that, when thus implemented, FLDA has the potential to solve classification tasks with very high dimensional data.

In the remainder of this section we briefly recall original FLDA. Section 2 compares some of the best-known modifications of FLDA for the high dimension/small sample size problem. It concludes with the choice of the one we consider closest to the original FLDA. In Section 3 we present a very detailed description of our improved implementation. Numerical examples comparing it with other implementations are given in Section 4.

1.1. Classical FLDA

Consider a classification task with g groups, $g \geqslant 2$, and assume that n training objects (x_i, y_i) with $x_i \in \mathbb{R}^p$ and $y_i \in \{1, \ldots, g\}$ are available. Using the mean vector $\bar{x} = (1/n)\sum_{i=1}^{n} x_i$ and denoting by N_j the index set of objects in group j, by n_j the size of group j and by $\bar{x}_j = (1/n_j)\sum_{i \in N_j} x_i$ the corresponding group's mean vector, the *between-* and *within-group covariance matrix* $\mathbf{B}$ and $\mathbf{W}$, respectively, are defined by

$$\mathbf{B} = \frac{1}{g-1} \sum_{j=1}^{g} n_j (\bar{x}_j - \bar{x})(\bar{x}_j - \bar{x})^{\mathrm{T}}, \tag{1}$$

$$\mathbf{W} = \frac{1}{n-g} \sum_{j=1}^{g} \sum_{i \in N_j} (x_i - \bar{x}_j)(x_i - \bar{x}_j)^{\mathrm{T}}. \tag{2}$$

The rank of $\mathbf{B}$ is at most $\min(g-1, p)$, the rank of $\mathbf{W}$ is at most $\min(n, p)$.

Let us assume for the moment that $p < n$. Then *Fisher's criterion* (see, e.g. Duda et al., 2000; Ripley, 1996, originally Fisher, 1936) is to find, subsequently, at most $g-1$ transformation vectors c that have maximal separation ratio by solving the maximization problem

$$\max_{c \in \mathbb{R}^p,\, c \neq 0} \frac{c^{\mathrm{T}} \mathbf{B} c}{c^{\mathrm{T}} \mathbf{W} c}. \tag{3}$$

It can be translated to finding the largest eigenpairs of the generalized eigenproblem

$$(\mathbf{B} - \lambda \mathbf{W})c = 0, \tag{4}$$

which, in turn, can be transformed to a standard eigenproblem, for example $(\mathbf{W}^{-1}\mathbf{B} - \lambda \mathbf{I})c = 0$. Then the FLDA-reduced space of dimension i, $i < g$, is spanned by the eigenvectors corresponding to the i largest eigenvalues. They are ordered decreasingly according to the eigenvalues and are orthogonal to each other (see, e.g. Guo et al., 2003). Many applications just aim at dimension reduction and stop after mapping onto the FLDA-reduced space. In the original classification process, the simplest and most frequent way to classify is by assigning to the group j of the transformed group mean vector $(c_1, \ldots, c_i)^{\mathrm{T}} \bar{x}_j$ which is closest in the L_2-norm.

2. Fisher's criterion for the high dimension/small sample size problem

2.1. The $p > n$ case

When $p > n$ the covariance matrix $\mathbf{W}$ from (2) is singular. This makes the classical FLDA process we describe above hard to perform. The main problem is solving the generalized eigenproblem (4). We cannot transform it to a standard

eigenproblem anymore. This paper addresses cases where even $p \gg n$. Then the problem will be challenging to solve also with respect to storage and computational costs. These numerical aspects will be considered in Section 3.

With singular covariance matrices, the generalized eigenproblem can be ill-posed itself. We recall some facts from linear algebra to explain this (see, e.g. Bai et al., 2000). Eigenvectors c of (4) satisfy $\mathbf{B}c = \lambda \mathbf{W}c$, for some value λ. If c lies in the null space of $\mathbf{B}$ but not of $\mathbf{W}$, then λ is a zero eigenvalue. On the other hand, if c lies in the null space of $\mathbf{W}$ but not of $\mathbf{B}$, then we say λ is an infinite eigenvalue. If c does not lie in the null space of $\mathbf{B}$ and neither in the null space of $\mathbf{W}$, then λ must be finite and nonzero. If c lies in the common null space of $\mathbf{B}$ and $\mathbf{W}$, any value λ is an eigenvalue! In fact, in this case corresponding eigenvectors are not even defined (Bai et al., 2000). The presence of a common null space will make solving the eigenproblem (4) very challenging. For example, it is well known that the QZ-algorithm (Moler and Stewart, 1973) may solve generalized eigenproblems with singular matrices but suffers from numerical instability precisely with a common null space. Unfortunately, the covariance matrices $\mathbf{W}$ and $\mathbf{B}$ must have a common null space as soon as $n + g - 1 < p$.

Apart from the difficulties of solving (4), with a singular covariance matrix $\mathbf{W}$ Fisher's criterion itself to some extent loses its meaning: Transformation vectors c in the null space of $\mathbf{W}$ would lead to division by zero in (3). Here we focus on interpretation and modification of Fisher's criterion (3) in the $p \gg n$ case. Papers that address these issues include Chen et al. (2000), Cheng et al. (1992), Hong and Yang (1991), Howland et al. (2006), Li et al. (1999), Krzanowski et al. (1995) and Yang et al. (2000). We will briefly review and compare some of the most popular methods described in these papers. This will motivate our choice of the one modified criterion whose implementation we address afterwards.

2.2. Perturbation methods

One type of methods tries to transform (4) to a standard eigenproblem by overcoming the singularity of $\mathbf{W}$. A way to achieve this is by perturbation of the singular values of $\mathbf{W}$. More precisely, let $\mathbf{W} = \mathbf{Q}\mathbf{S}\mathbf{Q}^{\mathrm{T}}$ be the singular value decomposition (SVD) of $\mathbf{W}$ (because $\mathbf{W}$ is symmetric it coincides with a spectral decomposition). Then the matrix of singular values $\mathbf{S}$ is replaced by $\mathbf{S} + \mathbf{D}$ where $\mathbf{D}$ is a diagonal matrix of small norm such that $\mathbf{S} + \mathbf{D}$ is nonsingular. Several choices of $\mathbf{D}$ are described in Cheng et al. (1992), Hong and Yang (1991), and Krzanowski et al. (1995). When $\tilde{\mathbf{W}}$ is the nonsingular matrix obtained by this kind of perturbation, then these methods determine the FLDA-vectors c by transforming the eigenproblem $(\mathbf{B} - \lambda\tilde{\mathbf{W}})c = 0$ to a standard eigenproblem. Working with the perturbed matrix $\tilde{\mathbf{W}}$ implies solving the *modified* criterion

$$\max_{c \in \mathbb{R}^p,\, c \neq 0} \frac{c^{\mathrm{T}}\mathbf{B}c}{c^{\mathrm{T}}\tilde{\mathbf{W}}c}. \tag{5}$$

Apart from the fact that it is not clear whether this method manages to solve Fisher's original criterion (3), it has the disadvantage that an optimal choice of the perturbation matrix $\mathbf{D}$ has to be determined, for example by cross-validation. While computing the spectral decomposition $\mathbf{Q}\mathbf{S}\mathbf{Q}^{\mathrm{T}}$ of $\mathbf{W}$, the method asks for solving a symmetric p-dimensional eigenproblem with computational costs of order $\mathcal{O}(p^3)$ and storage costs of order $\mathcal{O}(p^2)$.

2.3. Methods exploiting the Moore–Penrose pseudo-inverse

A different way to obtain a standard eigenproblem results from considering the truncated SVD of $\mathbf{W}$. This method is mentioned, among others, in Cheng et al. (1992), Hong and Yang (1991), and Krzanowski et al. (1995), and is implemented in the statistical software `R`-environment (R Development Core Team, 2005) by the `lda`-function (see also Ripley, 1996; Venables and Ripley, 2002). Let the SVD of $\mathbf{W}$ be

$$\mathbf{W} = \mathbf{Q}\,\mathrm{diag}(s_1, \ldots, s_p)\mathbf{Q}^{\mathrm{T}}, \tag{6}$$

and let $|s_i| \leqslant \varepsilon$ for $i > r$ and some small tolerance $\varepsilon > 0$. Then if $\mathbf{Q}_r$ consists of the first r columns of $\mathbf{Q}$ and $\Lambda_r = \mathrm{diag}(s_1, \ldots, s_r)$, the truncated SVD of $\mathbf{W}$ is $\tilde{\mathbf{W}} = \mathbf{Q}_r\Lambda_r\mathbf{Q}_r^{\mathrm{T}}$. Instead of solving (4), these methods try to transform

$$(\mathbf{B} - \lambda\mathbf{Q}_r\,\Lambda_r\mathbf{Q}_r^{\mathrm{T}})c = 0 \tag{7}$$

to a standard eigenproblem by multiplication with the Moore–Penrose pseudo-inverse $\mathbf{Q}_r \Lambda_r^{-1}\mathbf{Q}_r^{\mathrm{T}}$ of $\tilde{\mathbf{W}}$. They solve, for example, the symmetric eigenproblem

$$(\Lambda_r^{-1/2}\mathbf{Q}_r^{\mathrm{T}}\mathbf{B}\mathbf{Q}_r\Lambda_r^{-1/2} - \lambda\mathbf{I})c^* = 0, \tag{8}$$

where the desired eigenvectors c are obtained from

$$c = \mathbf{Q}_r\Lambda_r^{-1/2}c^* \tag{9}$$

and $\Lambda_r^{-1/2} = \mathrm{diag}(1/\sqrt{s_1}, \ldots, 1/\sqrt{s_r})$. The eigenproblem (8) is in general *not* equivalent to (7) because $\mathbf{Q}_r\mathbf{Q}_r^{\mathrm{T}} \neq \mathbf{I}$. Instead, the eigenvectors c obtained from solving (8) and (9) satisfy the equality $(\Lambda_r^{-1/2}\mathbf{Q}_r^{\mathrm{T}}\mathbf{B}\mathbf{Q}_r\mathbf{Q}_r^{\mathrm{T}} - \lambda\Lambda_r^{1/2}\mathbf{Q}_r^{\mathrm{T}})c = 0$, hence by multiplying with $\mathbf{Q}_r\Lambda_r^{1/2}$ we have

$$(\mathbf{Q}_r\mathbf{Q}_r^{\mathrm{T}}\mathbf{B}\mathbf{Q}_r\mathbf{Q}_r^{\mathrm{T}} - \lambda\mathbf{Q}_r\Lambda_r\mathbf{Q}_r^{\mathrm{T}})c = 0,$$

and one maximizes

$$\frac{c^{\mathrm{T}}\mathbf{Q}_r\mathbf{Q}_r^{\mathrm{T}}\mathbf{B}\mathbf{Q}_r\mathbf{Q}_r^{\mathrm{T}}c}{c^{\mathrm{T}}\mathbf{Q}_r\Lambda_r\mathbf{Q}_r^{\mathrm{T}}c}.$$

Here again, we do not know to what extent the original problem (3) is maximized. All we can do is measure the quality of the vectors c as approximate eigenvectors for the original eigenproblem (4).

Proposition 2.1. *Let us assume that $s_i = 0$ for all $i > r$ in the SVD* (6) *of* $\mathbf{W}$ *and let the last $p - r$ columns of* $\mathbf{Q}$, *corresponding to zero singular values*, *be denoted by* $\mathbf{Q}_z$. *Then the eigenpairs* $\{\lambda, c\}$ *defined through* (8) *and* (9) *satisfy*, *in the Euclidean norm*,

$$\|\mathbf{B}c - \lambda\mathbf{W}c\| = \|\mathbf{Q}_z^{\mathrm{T}}\mathbf{B}c\|.$$

Proof. We have

$$\mathbf{B}c - \lambda\mathbf{W}c = \mathbf{B}c - \lambda\mathbf{Q}_r\Lambda_r\mathbf{Q}_r^{\mathrm{T}}c = \mathbf{B}\mathbf{Q}_r\Lambda_r^{-1/2}c^* - \lambda\mathbf{Q}_r\Lambda_r^{1/2}c^*.$$

As $(\mathbf{Q}_r, \mathbf{Q}_z)$ is an orthonormal matrix,

$$\begin{aligned}
\|\mathbf{B}c - \lambda\mathbf{W}c\| &= \left\| \begin{pmatrix} \mathbf{Q}_r^{\mathrm{T}} \\ \mathbf{Q}_z^{\mathrm{T}} \end{pmatrix} (\mathbf{B}\mathbf{Q}_r\Lambda_r^{-1/2}c^* - \lambda\mathbf{Q}_r\Lambda_r^{1/2}c^*) \right\| \\
&= \left\| \begin{pmatrix} \Lambda_r^{1/2}\Lambda_r^{-1/2}\mathbf{Q}_r^{\mathrm{T}} \\ \mathbf{Q}_z^{\mathrm{T}} \end{pmatrix} (\mathbf{B}\mathbf{Q}_r\Lambda_r^{-1/2}c^* - \lambda\mathbf{Q}_r\Lambda_r^{1/2}c^*) \right\| \\
&= \left\| \begin{pmatrix} \Lambda_r^{1/2}(\Lambda_r^{-1/2}\mathbf{Q}_r^{\mathrm{T}}\mathbf{B}\mathbf{Q}_r\Lambda_r^{-1/2}c^* - \lambda c^*) \\ \mathbf{Q}_z^{\mathrm{T}}\mathbf{B}\mathbf{Q}_r\Lambda_r^{-1/2}c^* \end{pmatrix} \right\| = \left\| \begin{pmatrix} 0 \\ \mathbf{Q}_z^{\mathrm{T}}\mathbf{B}c \end{pmatrix} \right\|. \qquad \square
\end{aligned}$$

A similar result can easily be proven for perturbation methods. The previous proposition shows that methods exploiting the pseudo-inverse of $\mathbf{W}$ offer no room for improving the computed FLDA-vectors, their quality is fully determined by $\|\mathbf{Q}_z^{\mathrm{T}}\mathbf{B}c\|$. But they have the advantage they do not need to determine optimal perturbation parameters (only the truncation parameter ε is needed). Note that in (8) we solve an eigenproblem of dimension r, which is in general significantly less than p. However, the whole method asks for an initial p-dimensional spectral decomposition of $\mathbf{W}$ with computational costs of order $\mathcal{O}(p^3)$ and storage costs of order $\mathcal{O}(p^2)$.

2.4. A method based on the GSVD

Both types of methods we have described so far suffer from potential deterioration of the original eigenproblem (4) and hence Fisher's original criterion (3). A method that does not modify eigenproblem (4) is the LDA/GSVD (generalized singular value decomposition) method from Howland and Park (2004), Howland et al. (2003,2006) and Kim et al. (2005). It extracts the eigenvectors of (4) needed for FLDA. This is achieved by using the GSVD

(Paige and Saunders, 1981; Golub and van Loan, 1996). Leaving aside the details, with a numerically stable algorithm for the GSVD we can find diagonal matrices with nonnegative entries $\mathbf{S}_\alpha = \mathrm{diag}(\alpha_1, \ldots, \alpha_t)$ and $\mathbf{S}_\beta = \mathrm{diag}(\beta_1, \ldots, \beta_t)$ and a nonsingular matrix $\mathbf{C} \in \mathbb{R}^{p\times p}$ such that

$$\mathbf{B} = \mathbf{C}^{-\mathrm{T}} \begin{pmatrix} \mathbf{S}_\alpha & 0 \\ 0 & 0 \end{pmatrix} \mathbf{C}^{-1}, \quad \mathbf{W} = \mathbf{C}^{-\mathrm{T}} \begin{pmatrix} \mathbf{S}_\beta & 0 \\ 0 & 0 \end{pmatrix} \mathbf{C}^{-1},$$

with $\mathbf{S}_\alpha + \mathbf{S}_\beta = \mathbf{I}_t$ and $t \leqslant n + g$. This implies the first t columns c_i of $\mathbf{C}$ are eigenvectors for (4) and satisfy

$$\beta_i \mathbf{B} c_i = \alpha_i \mathbf{W} c_i.$$

If β_i is zero, the eigenvectors lie in the null space of $\mathbf{W}$ but not in the null space of $\mathbf{B}$. Then the within-group variance $c_i^{\mathrm{T}} \mathbf{W} c_i$ is zero and hence minimal. For this reason, the LDA/GSVD method chooses these vectors as the leading FLDA-transformation vectors. The remaining ones are chosen according to the ratio α_i/β_i; those for which

$$\frac{\alpha_i}{\beta_i} = \frac{c_i^{\mathrm{T}} \mathbf{B} c_i}{c_i^{\mathrm{T}} \mathbf{W} c_i}$$

is largest are chosen first. This corresponds to Fisher's original criterion (3). The last $p - t$ columns of $\mathbf{C}$ span the common null space of $\mathbf{B}$ and $\mathbf{W}$. In the common null space both between-group and within-group variance are zero. Therefore, no vectors from this space are used. The LDA/GSVD method can be implemented attractively by exploiting the special structure of the covariance matrices (we explain this in Section 3). This causes computational costs to be reduced to $\mathcal{O}(pn^2) + \mathcal{O}(n^3)$ and storage costs to $\mathcal{O}(p(n+g))$. In addition, the method offers a mathematical framework that helps in better understanding the high dimension/small sample size problem, see, e.g. Howland et al. (2003).

2.5. The null space method

We see that in the LDA/GSVD method the criterion (3) is modified by separating vectors for which $c_i^{\mathrm{T}} \mathbf{W} c_i$ is zero from those that yield a finite ratio $(c_i^{\mathrm{T}} \mathbf{B} c_i)/(c_i^{\mathrm{T}} \mathbf{W} c_i)$. The classical null space method (see, e.g. Chen et al., 2000 or the so-called zero-variance discrimination method in Krzanowski et al., 1995) fully concentrates on the null space of $\mathbf{W}$. As in LDA/GSVD, this is motivated by the fact that in this space within-group variance is minimal. The null space method simply modifies (3) as

$$\max_{c\in\mathbb{R}^p,\ \mathbf{W}c=0} c^{\mathrm{T}} \mathbf{B} c. \tag{10}$$

Of course, we maximize over vectors c with unit norm. The criterion leads to a standard eigenproblem in the null space of $\mathbf{W}$. It should be noticed that this null space has large dimension if $p \gg n$. As the rank of $\mathbf{W}$ is at most n, the null space has dimension at least $p - n$, which is just a little less than p. Therefore, this method finds, in addition to the spectral decomposition of $\mathbf{W}$, another large spectral decomposition and may be very time-consuming. Dominating computational costs are of order $\mathcal{O}(p^3)$; storage costs are of order $\mathcal{O}(p^2)$.

2.6. An intuitively reasonable criterion

Of all methods we discussed, Fisher's original idea to minimize within-group variance and maximize between-group variance seems best realized by the criterion (10). However, the null space method may choose vectors from the common null space where $c^{\mathrm{T}} \mathbf{B} c$ is zero as well: We look for $g - 1$ transformation vectors in total and the number of vectors in the null space of $\mathbf{W}$ that are not in the common null space can be less than $g - 1$. We avoid this by using a combined criterion that can be described as follows.

Transformation vectors from the null space of $\mathbf{W}$ give the "maximal" ratio $c^{\mathrm{T}} \mathbf{B} c / c^{\mathrm{T}} \mathbf{W} c = \infty$. As in the previous two methods, we choose them as leading transformation vectors because their within-group variance $c^{\mathrm{T}} \mathbf{W} c = 0$ is minimal. We order them according to their between-group variance, i.e. we use the criterion from the null space method,

$$\max_{c\in\mathbb{R}^p,\ \mathbf{W}c=0} c^{\mathrm{T}} \mathbf{B} c. \tag{11}$$

However, transformation vectors for which the maximum in (11) is zero are not interesting anymore; their between-group variance is minimal, hence they do not contribute to discrimination. Therefore, we select with criterion (11) only transformation vectors with nonzero between-group variance. If this does not yield enough (mutually orthogonal) transformation vectors, we leave the null space of $\mathbf{W}$ and select the next transformation vectors in the complement of the null space of $\mathbf{W}$. Here of course, the ratio $c^{\mathrm{T}}\mathbf{B}c/c^{\mathrm{T}}\mathbf{W}c$ is always finite and we can use the original criterion

$$\max_{c\in\mathbb{R}^p,\,\mathbf{W}c\neq 0} \frac{c^{\mathrm{T}}\mathbf{B}c}{c^{\mathrm{T}}\mathbf{W}c}. \tag{12}$$

The intuitively reasonable, combined criterion (11–12), which follows logically from the previously considered criteria, has been proposed, for example, in Yang and Yang (2003). We believe it reproduces Fisher's original idea best and we will use it in our implementation too. Our experiments seem to indicate that it leads to at least as powerful discrimination as other criteria.

3. Efficient implementation

The main focus of this paper is efficient implementation of (11) and (12). We have tried to combine as many clever strategies that are known as possible in order to minimize the overall costs and reduce numerical instability. In addition, we introduce some new ideas that make the algorithm even faster. We begin with two commonly used tools: Writing $\mathbf{B}$ and $\mathbf{W}$ as products of rectangular matrices and elimination of the common null space.

3.1. Exploiting the special structure of covariance matrices

The within-group and between-group covariance matrices $\mathbf{W}$ and $\mathbf{B}$ are both full matrices of dimension p. In many applications p is just too large to be able to store $2p^2$ matrix entries. For example, when $p = 10\,000$, which is realistic among others in modern document classification tasks, then $\mathbf{B}$ and $\mathbf{W}$ take already 1.6 GB to be stored in double precision arithmetic. Also, computations with these large matrices are rather expensive. In order to work efficiently with covariance matrices one commonly takes advantage of the fact that they can be written as a product of one and the same rectangular matrix. Let $\mathbf{X} \in \mathbb{R}^{n\times p}$ be the sample matrix whose ith row contains the ith training object and let $\mathbf{1_n} = (1, 1, \ldots, 1)^{\mathrm{T}} \in \mathbb{R}^n$. Furthermore, let $\mathbf{M} \in \mathbb{R}^{g\times p}$ be the group mean matrix whose jth row contains $\bar{x}_j^{\mathrm{T}}$ and let $\mathbf{G} \in \mathbb{R}^{n\times g}$ be the group coding matrix. If the ith object belongs to group j, $\mathbf{G}_{i,j} = 1$ and $\mathbf{G}_{i,k} = 0$ for $k \neq j$. Then (see, e.g. Venables and Ripley, 2002),

$$\mathbf{W} = \frac{1}{n-g}\sum_{j=1}^{g}\sum_{i\in N_j}(x_i - \bar{x}_j)(x_i - \bar{x}_j)^{\mathrm{T}} = \frac{(\mathbf{X} - \mathbf{GM})^{\mathrm{T}}(\mathbf{X} - \mathbf{GM})}{n-g}. \tag{13}$$

The matrix $\mathbf{B}$ can be written as

$$\mathbf{B} = \frac{1}{g-1}\sum_{j=1}^{g} n_j(\bar{x}_j - \bar{x})(\bar{x}_j - \bar{x})^{\mathrm{T}} = \frac{(\mathbf{GM} - \mathbf{1_n}\bar{x}^{\mathrm{T}})^{\mathrm{T}}(\mathbf{GM} - \mathbf{1_n}\bar{x}^{\mathrm{T}})}{g-1} \tag{14}$$

and also as (see, e.g. Kim et al., 2005)

$$\mathbf{B} = \frac{(\tilde{\mathbf{D}}(\mathbf{M} - \mathbf{1}_g\bar{x}^{\mathrm{T}}))^{\mathrm{T}}\tilde{\mathbf{D}}(\mathbf{M} - \mathbf{1}_g\bar{x}^{\mathrm{T}})}{g-1}, \tag{15}$$

where $\tilde{\mathbf{D}} = \mathrm{diag}(1/n_1, \ldots, 1/n_g)$.

In all cases a covariance matrix is decomposed into two rectangular matrices that are each others transposed. The number of rows of the right rectangles n (or even g for (15)) is by assumption much smaller than the number of columns p. It is advantageous to store only one rectangular part and replace computations with the covariance matrices by computations with their rectangles. A very simple example is the product $z = \mathbf{W}v$ of $\mathbf{W}$ with a vector v. It can be

computed by first forming the n-dimensional vector $y = (\mathbf{X} - \mathbf{GM})v$ and then putting

$$z = \frac{(\mathbf{X} - \mathbf{GM})^{\mathrm{T}} y}{n - g}.$$

As the direct product $z = \mathbf{W}v$ costs $2p^2$ floating point operations and the small products cost $2pn$ operations each, computational costs are reduced as soon as $n < p/2$. We consider efficient multiplications more in detail in Section 3.4. If we manage to restrict all computations needed with $\mathbf{B}$ and $\mathbf{W}$ to their rectangular factors in similar ways, we avoid storing $\mathbf{B}$ and $\mathbf{W}$. This has been successfully accomplished in the LDA/GSVD method and in the lda()-function implemented in the R-environment (R Development Core Team, 2005). Our implementation will also take advantage of the special structure of the covariance matrices.

3.2. Elimination of the common null space

A technique that has many advantages in FLDA-based computations is elimination of the common null space of $\mathbf{B}$ and $\mathbf{W}$. It is justified by the fact that vectors c in the common null space do not contribute to discrimination because $c^{\mathrm{T}}\mathbf{B}c = 0 = c^{\mathrm{T}}\mathbf{W}c$ (see, e.g. Yang and Yang, 2003). The common null space can be eliminated very efficiently by considering the *total* covariance matrix. This matrix is defined as

$$\mathbf{T} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})^{\mathrm{T}} = \frac{1}{n-1}(\mathbf{X}^{\mathrm{T}} - \bar{x}\mathbf{1}_{\mathbf{n}}^{\mathrm{T}})(\mathbf{X} - \mathbf{1}_{\mathbf{n}}\bar{x}^{\mathrm{T}}). \tag{16}$$

The following relation between the covariance matrices holds:

$$(n-1)\mathbf{T} = (g-1)\mathbf{B} + (n-g)\mathbf{W},$$

see, e.g. Howland et al. (2003). If we drop the denominators in (13), (14) and (16), the relation translates to

$$\mathbf{T} = \mathbf{B} + \mathbf{W}. \tag{17}$$

Discarding the denominators has no influence on the computations we perform. Hence from now on we consider unscaled covariance matrices for simplicity. As a consequence of (17) we have the following well-known lemma (Yang and Yang, 2003). For completeness we also give its proof.

Lemma 1. *The common null space of* $\mathbf{B}$ *and* $\mathbf{W}$ *is the null space of* $\mathbf{T}$.

Proof. A vector $v \in \mathbb{R}^p$ lies in the null space of $\mathbf{T}$ if and only if $v^{\mathrm{T}}\mathbf{T}v = 0$. This is readily seen from

$$v^{\mathrm{T}}\mathbf{T}v = 0 \;\Rightarrow\; v^{\mathrm{T}}(\mathbf{X}^{\mathrm{T}} - \bar{x}\mathbf{1}_{\mathbf{n}}^{\mathrm{T}})(\mathbf{X} - \mathbf{1}_{\mathbf{n}}\bar{x}^{\mathrm{T}})v = 0 \;\Rightarrow\; \|(\mathbf{X} - \mathbf{1}_{\mathbf{n}}\bar{x}^{\mathrm{T}})v\|^2 = 0,$$

the other direction is trivial. The same holds for $\mathbf{W}$ and $\mathbf{B}$ because they can be written as (13) and (14), respectively (here without the denominators). With (17) and the fact that $\mathbf{W}$ and $\mathbf{B}$ are positive semi-definite we have

$$v^{\mathrm{T}}\mathbf{T}v = 0 \;\Leftrightarrow\; v^{\mathrm{T}}(\mathbf{B} + \mathbf{W})\,v = 0 \;\Leftrightarrow\; v^{\mathrm{T}}\mathbf{B}v = 0 \quad \text{and} \quad v^{\mathrm{T}}\mathbf{W}v = 0. \qquad \square$$

In other words, the complement of the common null space of $\mathbf{B}$ and $\mathbf{W}$ is spanned by the eigenvectors of $\mathbf{T}$ which correspond to nonzero eigenvalues of $\mathbf{T}$. We show below that these eigenvectors can be computed inexpensively. Note that restriction to the eigenvectors of nonzero eigenvalues of $\mathbf{T}$ is nothing but performing a classical PCA as a preprocessing step and including all principal components explaining the full 100% of total variability (Yang and Yang, 2003).

If the total covariance matrix $\mathbf{T}$ has rank q where $q \leqslant n$, this preprocessing reduces the original p-dimensional problem to the dimension q. As we assume $p \gg n$, the benefit is considerable. Another important advantage of elimination of the common null space is that it enhances numerical stability of algorithms for generalized eigenproblems, see for example Parlett (1998).

The eigenvectors of $\mathbf{T}$ corresponding to nonzero eigenvalues can be computed very efficiently with the following lemma.

Lemma 2. *Let $\mathbf{Z} \in \mathbb{R}^{n\times p}$ with $n < p$, let the diagonal matrix $\mathbf{D}_1$ contain the nonzero eigenvalues of $\mathbf{Z}\mathbf{Z}^{\mathrm{T}} \in \mathbb{R}^{n\times n}$ and let the columns of $\mathbf{V}_1$ contain the corresponding eigenvectors. Then the normalized eigenvectors for nonzero eigenvalues of $\mathbf{Z}^{\mathrm{T}}\mathbf{Z} \in \mathbb{R}^{p\times p}$ are given by the columns of $\mathbf{Z}^{\mathrm{T}}\mathbf{V}_1\mathbf{D}_1^{-1/2}$.*

Proof. See Johnson and Wichern (1998). □

This lemma is widely used in PCA computations. It says we can extract eigenvectors of the p-dimensional matrix $\mathbf{T} = (\mathbf{X}^{\mathrm{T}} - \bar{x}\mathbf{1}_{\mathbf{n}}^{\mathrm{T}})(\mathbf{X} - \mathbf{1}_{\mathbf{n}}\bar{x}^{\mathrm{T}})$ by forming the n-dimensional spectral decomposition

$$(\mathbf{X} - \mathbf{1}_{\mathbf{n}}\bar{x}^{\mathrm{T}})(\mathbf{X}^{\mathrm{T}} - \bar{x}\mathbf{1}_{\mathbf{n}}^{\mathrm{T}}) = \mathbf{V}\mathbf{D}\mathbf{V}^{\mathrm{T}}, \tag{18}$$

where $\mathbf{D}$ is a diagonal matrix containing the eigenvalues in decreasing order. Let it have the form

$$\mathbf{D} = \begin{pmatrix} \mathbf{D}_1 & 0 \\ 0 & 0 \end{pmatrix},$$

where $\mathbf{D}_1 \in \mathbb{R}^{q\times q}$ is nonsingular. If we collect in $\mathbf{V}_1$ the eigenvectors of $\mathbf{V}$ corresponding to the nonzero eigenvalues then the complement of the null space of $\mathbf{T}$ is spanned by the q orthonormal columns of

$$(\mathbf{X}^{\mathrm{T}} - \bar{x}\mathbf{1}_{\mathbf{n}}^{\mathrm{T}})\mathbf{V}_1\mathbf{D}_1^{-1/2}. \tag{19}$$

Note that the only computation depending on p needed to find the complement of the common null space (19) is multiplication with $(\mathbf{X}^{\mathrm{T}} - \bar{x}\mathbf{1}_{\mathbf{n}}^{\mathrm{T}})$. In Section 3.4 we show how to circumvent the high costs of this multiplication.

3.3. Efficient computations in the complement of the common null space

We denote the projections of the matrices $\mathbf{B}$, $\mathbf{W}$ and $\mathbf{T}$ onto the complement of the common null space, which is spanned by the columns of (19), by $\overline{\mathbf{B}}$, $\overline{\mathbf{W}}$ and $\overline{\mathbf{T}}$, respectively. We now show how we solve (4) in the complement of the common null space. To facilitate computations we use the following simple lemma from linear algebra (Bai et al., 2000).

Lemma 3. *Any generalized eigenvector c satisfying $\mathbf{Y}c = \mu(\mathbf{Y} + \mathbf{Z})c$ for some eigenvalue $\mu \in \mathbb{R}$ satisfies $\mathbf{Y}c = (\mu/(1-\mu))\mathbf{Z}c$, where the corresponding eigenvalue is infinite if $\mu = 1$.*

Hence with $\overline{\mathbf{T}} = \overline{\mathbf{B}} + \overline{\mathbf{W}}$, any eigenvector c with

$$\overline{\mathbf{B}}c = \mu\overline{\mathbf{T}}c \tag{20}$$

satisfies

$$\overline{\mathbf{B}}c = \lambda\overline{\mathbf{W}}c, \tag{21}$$

where $\lambda = \mu/(1-\mu)$. This means that the eigenvectors of (20) are the same as those of (21). As we need in FLDA only the eigenvectors, we can solve (20) instead of (21), provided we select the eigenvectors correctly. Infinite eigenvalues of (21) take the value 1 in (20) and finite eigenvalues change to eigenvalues that are smaller than 1.

Using (20) instead of (21) has been proposed among others in Cheng et al. (1992), and Hong and Yang (1991) in order to modify Fisher's criterion. We are here interested in two important implementational advantages which to our knowledge the literature is not fully aware of. The first one is that $\overline{\mathbf{T}}$ is nonsingular because it is the restriction of $\mathbf{T}$ to the complement of its own null space. Hence (20) can be transformed to a standard eigenproblem. The second advantage is that (20) takes a particularly simple form.

Lemma 4. *The projection $\overline{\mathbf{T}}$ of $\mathbf{T}$ to the complement of the common null space is the nonsingular diagonal matrix $\mathbf{D}_1$.*

Proof. Using (19), we have

$$\begin{aligned}\overline{\mathbf{T}} &= ((\mathbf{X}^{\mathrm{T}} - \bar{x}\mathbf{1}_{\mathbf{n}}^{\mathrm{T}})\mathbf{V}_1\mathbf{D}_1^{-1/2})^{\mathrm{T}}\mathbf{T}((\mathbf{X}^{\mathrm{T}} - \bar{x}\mathbf{1}_{\mathbf{n}}^{\mathrm{T}})\mathbf{V}_1\mathbf{D}_1^{-1/2}) \\ &= \mathbf{D}_1^{-1/2}\mathbf{V}_1^{\mathrm{T}}(\mathbf{X} - \mathbf{1}_{\mathbf{n}}\bar{x}^{\mathrm{T}})(\mathbf{X}^{\mathrm{T}} - \bar{x}\mathbf{1}_{\mathbf{n}}^{\mathrm{T}})(\mathbf{X} - \mathbf{1}_{\mathbf{n}}\bar{x}^{\mathrm{T}})(\mathbf{X}^{\mathrm{T}} - \bar{x}\mathbf{1}_{\mathbf{n}}^{\mathrm{T}})\mathbf{V}_1\mathbf{D}_1^{-1/2}.\end{aligned}$$

From (18) we obtain $(\mathbf{X} - \mathbf{1}_{\mathbf{n}}\bar{x}^{\mathrm{T}})(\mathbf{X}^{\mathrm{T}} - \bar{x}\mathbf{1}_{\mathbf{n}}^{\mathrm{T}})\mathbf{V}_1 = \mathbf{V}_1\mathbf{D}_1$ and $\overline{\mathbf{T}}$ simplifies to the diagonal matrix $\mathbf{D}_1$. □

Hence we do not need to compute $\overline{\mathbf{T}}$ at all. For $\overline{\mathbf{B}}$, we have

$$\overline{\mathbf{B}} = ((\mathbf{X}^{\mathrm{T}} - \bar{x}\mathbf{1}_{\mathbf{n}}^{\mathrm{T}})\mathbf{V}_1\mathbf{D}_1^{-1/2})^{\mathrm{T}}\mathbf{B}((\mathbf{X}^{\mathrm{T}} - \bar{x}\mathbf{1}_{\mathbf{n}}^{\mathrm{T}})\mathbf{V}_1\mathbf{D}_1^{-1/2}).$$

As in (14), we can write $\overline{\mathbf{B}}$ as $\overline{\mathbf{B}} = \mathbf{B}_1^{\mathrm{T}}\mathbf{B}_1$ where

$$\mathbf{B}_1 = (\mathbf{GM} - \mathbf{1}_{\mathbf{n}}\bar{x}^{\mathrm{T}})(\mathbf{X}^{\mathrm{T}} - \bar{x}\mathbf{1}_{\mathbf{n}}^{\mathrm{T}})\mathbf{V}_1\mathbf{D}_1^{-1/2}.$$

Thus (20) takes the form

$$(\mathbf{B}_1^{\mathrm{T}}\mathbf{B}_1 - \mu\mathbf{D}_1)c = 0. \tag{22}$$

In our implementation we will transform (22) to the symmetric standard eigenproblem

$$(\mathbf{D}_1^{-1/2}\mathbf{B}_1^{\mathrm{T}}\mathbf{B}_1\mathbf{D}_1^{-1/2})c^* = \mu c^*, \quad c = \mathbf{D}_1^{-1/2}c^*. \tag{23}$$

We emphasize that transformation to this standard eigenproblem is possible because we base our computations on (20) instead of (21). Note that we are not interested in all q eigenpairs but only in the $g - 1$ leading ones.

The next and last step is solving the maximization problems (11) and (12) from Section 2.6 in the complement of the common null space. In contrast with other implementations (see, e.g. Yang and Yang, 2003) we do not solve these maximization problems separately but we extract all the needed vectors from (22). This makes the implementation faster and simpler. As explained by Lemma 3, eigenvalues $\mu = 1$ for (20) are infinite eigenvalues λ for (21). Hence the corresponding eigenvectors lie in the null space of $\overline{\mathbf{W}}$ and we will use them to solve the first part (11) of our criterion

$$\max_{c\in\mathbb{R}^p,\, \overline{\mathbf{W}}c=0} c^{\mathrm{T}}\overline{\mathbf{B}}c. \tag{24}$$

The computed eigenvectors for $\mu = 1$ necessarily form a basis for this null space because of the following lemma.

Lemma 5. *The null space of* $\overline{\mathbf{W}}$ *has dimension at most* $g - 1$.

Proof. Assume the dimension of the null space of $\overline{\mathbf{W}}$ is larger than $g - 1$. Then there exists at least one vector v in this null space with $v^{\mathrm{T}}\overline{\mathbf{B}}v = 0$ because the rank of $\overline{\mathbf{B}}$ is at most $g - 1$. Hence v lies in the common null space, which is a contradiction to the definition of the null space of $\overline{\mathbf{W}}$. □

To solve the maximization problem (24) correctly we need an *orthogonal* basis of the null space of $\overline{\mathbf{W}}$. Let us collect computed eigenvectors for $\mu = 1$ in a matrix $\mathbf{V}_2$. Then we propose to compute the reduced QR-decomposition

$$\mathbf{V}_2 = \mathbf{QR},$$

i.e. $\mathbf{Q}$ is orthogonal and rectangular, $\mathbf{R}$ is upper triangular and square with dimension equal to the number of columns of $\mathbf{V}_2$. This QR-decomposition is very cheap because $\mathbf{V}_2$ has few columns (namely, less than g). The columns of $\mathbf{Q}$ form an orthogonal basis of the null space of $\overline{\mathbf{W}}$ and we compute the eigenvectors $\tilde{c}$ of $\mathbf{Q}^{\mathrm{T}}\mathbf{B}_1^{\mathrm{T}}\mathbf{B}_1\mathbf{Q}$. Then the (ordered) vectors $\mathbf{Q}\tilde{c}$ maximize $c^{\mathrm{T}}\overline{\mathbf{B}}c$ subject to $\overline{\mathbf{W}}c = 0$.

For the second part (12) of our criterion, we consider the remaining eigenvalues from (22). They satisfy $\mu < 1$ and are finite eigenvalues λ for (21). The corresponding eigenvectors lie in the complement of the null space of $\overline{\mathbf{W}}$ and maximize Fisher's original criterion

$$\frac{c^{\mathrm{T}}\overline{\mathbf{B}}c}{c^{\mathrm{T}}\overline{\mathbf{W}}c}. \tag{25}$$

in the complement of the common null space.

If we store first the vectors obtained from (24) and then those from (25) in a matrix $\mathbf{C}$, we return to the original p-dimensional space by multiplying $\mathbf{C}$ with $(\mathbf{X} - \mathbf{1_n}\bar{x}^T)^T\mathbf{V}_1\mathbf{D}_1^{-1/2}$ from (19) and thus obtain the final FLDA-transformation vectors. The overall algorithm has the following form.

Algorithm 1. A fast algorithm to solve the FLDA-based criterion (11)–(12).

(1) Compute the spectral decomposition of $(\mathbf{X} - \mathbf{1_n}\bar{x}^T)(\mathbf{X}^T - \bar{x}\mathbf{1_n}^T)$; store the nonzero eigenvalues in the diagonal matrix $\mathbf{D}_1$ and the corresponding eigenvectors in $\mathbf{V}_1$.
(2) Compute $\mathbf{B}_1 = ((\mathbf{GM} - \mathbf{1_n}\bar{x}^T)(\mathbf{X}^T - \bar{x}\mathbf{1_n}^T))\mathbf{V}_1\mathbf{D}_1^{-1}$.
(3) Compute the eigenvectors of the $g-1$ largest eigenvalues of $\mathbf{B}_1^T\mathbf{B}_1$ and multiply them with $\mathbf{D}_1^{-1/2}$.
(4) If any, collect the eigenvectors for the eigenvalue 1 in $\mathbf{V}_2$ and
  (a) compute the reduced QR-decomposition $\mathbf{V}_2 = \mathbf{QR}$;
  (b) compute the eigenvectors of $\mathbf{Q}^T\mathbf{D}_1^{1/2}\mathbf{B}_1^T\mathbf{B}_1\mathbf{D}_1^{1/2}\mathbf{Q}$;
  (c) multiply them with $\mathbf{Q}$ and substitute the eigenvectors for the eigenvalue 1 from step (3) with these vectors.
(5) Multiply the vectors obtained from step (3) and possibly step (4) with $(\mathbf{X}^T - \bar{x}\mathbf{1_n}^T)\mathbf{V}_1\mathbf{D}_1^{-1/2}$ and normalize them.

3.4. Remarks on the algorithm

Here we address two more issues that may accelerate the overall algorithm: Clever multiplication of matrices and the usage of so-called sparse methods to solve the eigenproblems. We consider here the possibilities offered by MATLAB (MathWorks, Inc., 1984–2005). Sufficient experience with LAPACK (Anderson et al., 2000) and similar packages to implement our algorithm seems unrealistic for the average statistician. On the other hand, the relatively simple programming language of MATLAB has an advantage over the R-environment (R Development Core Team, 2005) that is important in the context of FLDA-based classification: Working with sparse matrices is very well-integrated. Storage of matrices in sparse format is possible without loading special packages and, more important, MATLAB contains so-called sparse algorithms for eigenvalue computations with sparse matrices. Such algorithms are not yet available in the R-environment.

MATLAB basically offers two functions to solve eigenproblems numerically: The command eig uses a so-called direct method and eigs uses a sparse method. At the end of the computation, a direct method has found all eigenpairs, but no eigenpairs are available during the process. Computational costs are of order m^3 if m is the dimension of the eigenproblem. Direct methods are backward stable, i.e. the computed pairs are exact eigenpairs for a different yet close eigenproblem. Accuracy of an eigenvector is endangered only when the corresponding eigenvalue lies close to other eigenvalues. Sparse methods, on the other hand, are advantageous if multiplication of vectors with the involved matrix is inexpensive and if we need only a few eigenvalues and eigenvectors. They compute one eigenvalue at a time and can be stopped after a predefined number of eigenpairs has been found. Computational costs depend on the sparsity of the matrix and the number of eigenvalues that is needed. They are not backward stable and convergence of computed eigenpairs to the wanted eigenpairs is not guaranteed.

In our algorithm we solve eigenproblems in steps (1), (3) and (4b). The first and third one need all eigenpairs, hence we recommend to use the eig command to solve them. In step (3) we need the leading $g-1$ eigenpairs of a q-dimensional problem. If the dimension q of the common null space is clearly larger than $g-1$, using a sparse method with eigs is in general much faster than using eig. In our implementation we used a sparse method in step (3).

To further reduce computational and storage costs we propose to perform the multiplications of $p \times n$ matrices in Algorithm 1 as follows. In step (1) we recommend to directly form the sum $\mathbf{X}\mathbf{X}^T - \mathbf{X}\bar{x}\mathbf{1_n}^T - \mathbf{1_n}\bar{x}^T\mathbf{X}^T + \|\bar{x}\|^2\mathbf{1_n}\mathbf{1_n}^T$. If we would form the factor $(\mathbf{X} - \mathbf{1_n}\bar{x}^T)$ we would create an additional $p \times n$ matrix to be stored; in our case we store only matrices of dimension n. Because $\bar{x} = \mathbf{X}^T\mathbf{1_n}/n$ we need to compute only $\tilde{\mathbf{X}} \equiv \mathbf{X}\mathbf{X}^T$ and the vector $\tilde{\mathbf{X}}_1 \equiv \tilde{\mathbf{X}}\mathbf{1_n}$; then the wanted sum is

$$\tilde{\mathbf{X}} - (\tilde{\mathbf{X}}_1\mathbf{1_n}^T)/n - (\mathbf{1_n}\tilde{\mathbf{X}}_1^T)/n + \|\bar{x}\|^2\mathbf{1}_n\mathbf{1}_n^T. \tag{26}$$

It is easy to see that $\|\bar{x}\|^2$ can be computed as $\mathbf{1_n}^T\tilde{\mathbf{X}}\mathbf{1_n}$ divided by n^2. Computational costs are dominated by the computation of $\tilde{\mathbf{X}}$. Although in general this computation involves $2n^2p$ operations (Golub and van Loan, 1996), here it may be significantly reduced because in many applications (protein fold prediction, text document classification, etc.)

the sample matrix $\mathbf{X}$ has many zero entries. If we store it as a sparse matrix and the number of nonzero entries in row i is denoted by $\mathtt{nnz}_i$, then forming the matrix vector product $\mathbf{X}v$ for some vector $v \in \mathbb{R}^p$ costs $2\sum_{i=1}^{n} \mathtt{nnz}_i$ operations in MATLAB. With $\mathtt{nnz} = \sum_{i=1}^{n} \mathtt{nnz}_i$ the total costs of computing $\tilde{\mathbf{X}}$ are $2\mathtt{nnz}n$ at most and can be significantly less than $2n^2p$.

Similarly, it is important to compute $\mathbf{B}_1$ efficiently in step (2). Again, if we would form $(\mathbf{X} - \mathbf{1_n}\bar{x}^{\mathrm{T}})^{\mathrm{T}}\mathbf{V}_1\mathbf{D}_1^{-1/2}$ we would create an additional $p \times n$ matrix to be stored. This is avoided by computing first the product

$$(\mathbf{GM} - \mathbf{1_n}\bar{x}^{\mathrm{T}})(\mathbf{X}^{\mathrm{T}} - \bar{x}\mathbf{1_n^{\mathrm{T}}}) = \mathbf{GMX}^{\mathrm{T}} - \mathbf{GM}\bar{x}\mathbf{1_n^{\mathrm{T}}} - (\mathbf{1_n}\tilde{\mathbf{X}}_1^{\mathrm{T}})/n + \|\bar{x}\|^2\mathbf{1}_n\mathbf{1}_n^{\mathrm{T}},$$

which is a sum of $n \times n$ matrices. Note that the last two terms have been computed already in step (1) in (26). Moreover, the group coding matrix $\mathbf{G}$ is always sparse and we can write $\mathbf{M}$ as $\mathbf{M} = \tilde{\mathbf{D}}\mathbf{G}^{\mathrm{T}}\mathbf{X}$ with the diagonal matrix $\tilde{\mathbf{D}}$ from (15). With $\mathbf{M}_1 \equiv \tilde{\mathbf{D}}\mathbf{G}^{\mathrm{T}}\tilde{\mathbf{X}}$ we have

$$(\mathbf{GM} - \mathbf{1_n}\bar{x}^{\mathrm{T}})(\mathbf{X}^{\mathrm{T}} - \bar{x}\mathbf{1_n^{\mathrm{T}}}) = \mathbf{G}(\mathbf{M}_1(\mathbf{I} - \mathbf{1_n}\mathbf{1_n^{\mathrm{T}}}/n)) - (\mathbf{1_n}\tilde{\mathbf{X}}_1^{\mathrm{T}})/n + \|\bar{x}\|^2\mathbf{1}_n\mathbf{1}_n^{\mathrm{T}}.$$

Finally, we need in step (5) the product $(\mathbf{X}^{\mathrm{T}} - \bar{x}\mathbf{1_n^{\mathrm{T}}})\mathbf{V}_1\mathbf{D}_1^{-1/2}$. For the same reasons as in step (2), it is best computed as

$$\mathbf{X}^{\mathrm{T}}\left(\mathbf{V}_1\mathbf{D}_1^{-1/2} - \mathbf{1}_n\left(\frac{\mathbf{1}_n^{\mathrm{T}}\mathbf{V}_1\mathbf{D}_1^{-1/2}}{n}\right)\right).$$

All together, we see that we do not even need to store the rectangular factors of the covariance matrices. It suffices to store $\mathbf{X}$, $\tilde{\mathbf{X}}$, $\tilde{\mathbf{X}}_1$, $\mathbf{G}$ and $\mathbf{M}_1$.

3.5. Concluding remarks

We conclude this section with a brief summary of the influence on overall costs of the techniques discussed here. In our algorithm, computational costs are dominated by the products with $p \times n$ matrices and the eigenproblem in step (1). In general the products ask for $\mathcal{O}(pn^2)$ operations. In many cases, however, the data matrix is sparse and the costs will be of order $\mathcal{O}(\mathtt{nnz}n)$, where $\mathtt{nnz}$ is the number of nonzero entries in $\mathbf{X}$. The eigenproblem in step (1) needs $\mathcal{O}(n^3)$ operations. We have to store only one $p \times n$ matrix, namely $\mathbf{X}$. The final FLDA-transformation vectors can be stored in the first columns of $\mathbf{X}$. Hence memory requirements are of order $\mathcal{O}(pn)$.

The methods from Section 2 can partially profit from our implementation strategies too. This would give the following rough costs for the individual methods. Perturbation methods can take advantage of the structure of covariance matrices and of a sparse method to find the $g-1$ largest eigenvalues. For dense sample matrices this gives a complexity of order $\mathcal{O}(p^2n)$; storage costs are of order $\mathcal{O}(p^2)$. Methods exploiting the Moore–Penrose pseudo-inverse can be implemented with elimination of the common null space, making usage of the special structure of covariance matrices and a sparse method for the eigenproblem in the complement of the null space of $\overline{\mathbf{W}}$. This gives main computational costs of order $\mathcal{O}(pn^2)$ for a dense sample matrix and storage costs of order $\mathcal{O}(pn)$. We used the optimized implementation of the LDA/GSVD method (Kim et al., 2005), with costs mentioned in Section 2.4. Finally, the null space method with taking advantage of the structure of covariance matrices can be implemented with order $\mathcal{O}(p^2n)$ computational costs and order $\mathcal{O}(p(p-r))$ storage costs ($p-r$ is the dimension of the null space of $\mathbf{W}$).

4. Experiments

4.1. Data description

In this section we test our algorithm on two data sets where the number of variables largely exceeds the sample size. We compare it with the methods described in Section 2. All methods were implemented with the most advantageous choice of strategies as described in Section 3.5.

The first data set is taken from the gene expression data studied in Tibshirani et al. (2002), available as `Khan` data in the `pamr` package for the `R`-environment. It consists of $n = 63$ measurements of $p = 2308$ genes belonging to $g = 4$ groups. We divided the objects by choosing randomly from every group, one half as training and one half as test set. This gave a training sample matrix of dimension 32×2308. For these data the sample matrix is dense.

The MEDLINE data (see http://www.ncbi.nlm.nih.gov/PubMed) has been used several times, among others, in the context of dimension reduction with the LDA/GSVD method (see, e.g. Howland et al., 2003; Kim et al., 2005). We use here the same data as in Kim et al. (2005), which is available at http://www-users.cs.umn.edu/~hpark/data.html. It studies the classification of documents into five groups. All groups are represented homogenously, e.g. there are 500 documents of each group. After applying a preprocessing technique we obtain $p = 22\,095$ distinct terms as explanatory variables. The corresponding object vectors have a large number of zero-entries and resulting sample matrices are sparse. We use a training set and test set with the same number of examples $n = 1250$; the number of nonzeroes of the $1250 \times 22\,095$ training sample matrix is 99 765.

Note that for both data sets the cited publications are freely available and give, among others, information on the performance of non-LDA-based methods. Thus the performance of our generalization of FLDA to the high dimension/small sample size problem can be compared with other classification methods like shrunken centroids, support vector machine, nearest neighbor methods, etc.

4.2. Results

We are here primarily interested in the costs of individual methods and in how successful they are in satisfying Fisher's criterion. We therefore compare overall time costs (measured at a server with 2 Dual Core AMD Opteron™ Processor 275 at 2191 MHz with 10 179 288 kB of memory) and the obtained between- and within-group covariance matrices. Secondarily, we add the rates of successful classification of the test data set. We used the classical and most current classification based on assigning to the class of the nearest transformed class centroid in the L_2-norm.

4.3. Gene expression data

Table 1 displays the timings of the methods we described in Section 2 and our method for the gene expression data. In the perturbation method we perturbed with the matrix $\varepsilon\mathbf{I}$ where $\varepsilon = 10^{-5}$. MP denotes the method based on the Moore–Penrose pseudo-inverse and GSVD the method from Section 2.4. In the Moore–Penrose method there was a clear gap between nonzero singular values and singular values zero to machine precision, hence the choice of a truncation parameter was trivial. Alg1 denotes our algorithm implemented as described in Section 3.4.

As we have here $p = 2308 \gg n = 63$, the acceleration with restriction to the q-dimensional complement of the common null space ($q < n$) is remarkable. The perturbation and null space methods are slow because they do not allow such a restriction. In the first case we compute $g - 1 = 3$ leading p-dimensional eigenvectors, in the second case we need at least $p - n$ eigenvectors to span the null space of $\mathbf{W}$. This explains the inferior performance of the latter method.

In Tables 2 and 3 we show the traces of between- and within-group covariance matrices from the individual methods. The null space of $\mathbf{W}$ has dimension larger than 3. Hence all methods find LDA-transformation vectors in this null space, except for the Moore–Penrose method which is defined on the complement of the null space. As for the traces of the between-group covariance matrices, we see that the criterion (11) is fully satisfied only by the perturbation and

Table 1
Gene expression data: overall computational time (in seconds) for the methods from Section 2 and our algorithm

Perturbation	MP	GSVD	Null space	Alg1
2.9	0.025	0.024	8.6	0.024

Table 2
Gene expression data: traces of between-group covariance matrices ($c^{\mathrm{T}}Bc$) achieved by the methods of Section 2 and our algorithm with growing number of transformation vectors (dimension)

Dimension	Perturbation	MP	GSVD	Null space	Alg1
1	794	183	602	794	794
2	1405	331	1148	1405	1405
3	1829	391	1715	1829	1829

Table 3
Gene expression data: traces of within-group covariance matrices ($c^T W c$) achieved by the methods of Section 2 and our algorithm with growing number of transformation vectors (dimension)

Dimension	Perturbation	MP	GSVD	Null space	Alg1
1	0	15	0	0	0
2	0	32	0	0	0
3	0	43	0	0	0

Table 4
Gene expression data: successful classification rates with L_2-norm similarity

Dimension	Perturbation (%)	MP (%)	GSVD (%)	Null space (%)	Alg1 (%)
1	74.2	51.6	51.6	74.2	74.2
2	93.6	77.4	96.8	93.6	93.6
3	96.8	83.9	96.8	96.8	96.8

null space methods and our method. The Moore–Penrose method particularly clearly fails to maximize between-group variance. In the LDA/GSVD method the failure is much less pronounced.

Table 4 displays the successful classification rates obtained with the individual methods. They correspond more or less to the relation between the traces of Tables 2 and 3. This shows that Fisher's idea to minimize within-group variance and maximize between-group variance makes sense for classifying these data.

4.4. Medline data

The dimensions for the Medline data are much larger than for the previous data as here $p = 22\,095$ and $n = 1250$. This allows us to demonstrate the benefits of our implementation compared with other fast methods that eliminate the common null space. However, we were not able to execute the perturbation and null space methods: With about $\mathcal{O}(p^2)$ storage costs we ran out of memory. For the remaining methods the overall costs, expressed by their timings, are displayed in Table 5. The table also addresses implementations of our algorithm which do not make use of the acceleration techniques from Section 3.4. The third column contains the timing for our algorithm with a direct method for the eigenproblem in step (3) and the fourth column with the product of step (2) formed as $\mathbf{B}_1 = (\mathbf{GM} - \mathbf{1_n}\bar{x}^T)((\mathbf{X}^T - \bar{x}\mathbf{1_n^T})\mathbf{V}_1\mathbf{D}_1^{-1})$. Clearly, the contribution from the issues from Section 3.4 to the high speed of our algorithm is considerable.

To explain the relatively high costs of the LDA/GSVD method we must realize that the sample matrix $\mathbf{X}$ is in this problem sparse. The LDA/GSVD method cannot profit from the sparsity; it needs the full orthogonal matrix Q of a QR-decomposition (Kim et al., 2005), giving computational costs of order $\mathcal{O}(pn^2)$. The other two methods project onto the complement of the common null space and exploit the sparsity of $\mathbf{X}$ which yields main computational costs of order $\mathtt{nnz}n$ where $\mathtt{nnz} = 99\,765$, see Section 3.4. The Moore–Penrose method is slower because it needs a full spectral decomposition of $\overline{\mathbf{W}}$ in the complement of the common null space, which has dimension $q = 1245$ for the given data.

The performance of the methods concerning approximation of Fisher's criteria can be taken from Tables 6 and 7. By definition, the Moore–Penrose method does not look for eigenvectors in the null space of $\overline{\mathbf{W}}$. This prevents the method from maximizing between-group variance in this example. The two other methods, on the contrary, first detect the two-dimensional null space and then proceed to its complement. The main difference between LDA/GSVD and our algorithm can be observed in Table 6: In the first dimension the value of $c^T B c$ is maximized only by our algorithm whereas LDA/GSVD takes "any" proper vector from the null space without taking into account $c^T B c$. However, at the second dimension (after adding the last vector from the null space of $\overline{\mathbf{W}}$) the trace of $c^T B c$ has been corrected. Table 8 displays the successful classification rates for the considered methods.

Table 5
Medline data: overall computational time (in seconds) for the Moore–Penrose method (MP), the LDA/GSVD method (GSVD) and three variants of our algorithm

MP	GSVD	Alg1, direct method	Alg1, slow product	Alg1
81	150.5	60.5	71.5	33

Table 6
Medline data: traces of between-group covariance matrices ($c^{\mathrm{T}}Bc$) achieved by the Moore–Penrose method (MP), the LDA/GSVD method (GSVD) and our algorithm with growing number of transformation vectors (dimension)

Dimension	MP	GSVD	Alg1
1	0.58	0.53	0.74
2	0.66	0.91	0.91
3	0.70	1.08	1.08
4	0.78	1.12	1.12

Table 7
Medline data: traces of within-group covariance matrices ($c^{\mathrm{T}}Wc$) achieved by the Moore–Penrose method (MP), the LDA/GSVD method (GSVD) and our algorithm with growing number of transformation vectors (dimension)

Dimension	MP	GSVD	Alg1
1	4.72e − 06	0	0
2	1.07e − 05	0	0
3	4.13e − 04	4.72e − 06	4.72e − 06
4	7.61e − 04	1.06e − 05	1.06e − 05

Table 8
Medline data: successful classification rates with L_2-norm similarity

Dimension	MP (%)	GSVD (%)	Alg1 (%)
1	51.0	31.9	48.5
2	50.2	54.6	55.0
3	63.4	74.6	74.6
4	86.7	87.5	87.5

5. Conclusions

We studied implementation of an FLDA-based classification method for the high dimension/small sample size case. We showed and confirmed with experiments that this method is closer to original FLDA than other popular FLDA-based methods. We optimized its implementation with regard to computational and storage costs using many tools, among others elimination of the common null space and sparse numerical algorithms. The resulting algorithm is prepared to be applied to very high dimensional data. It is especially fast with a sparse sample matrix. We demonstrated on examples the accelerating effect of the tools we used and we showed our implementation is faster than that of other reference methods. If the sample matrix is dense, feasibility of our algorithm depends on whether it can cope with multiplication of full $p \times n$ matrices with each other. For sample matrices that are sparse enough, the only bottleneck is the solution of a symmetric $n \times n$ eigenproblem.

Acknowledgments

We thank Prof. Eldén for turning our attention to relevant literature and we thank Alois Schloegl for his contribution in tracing an important bug in our programs. This work was supported by the Program Information Society under project 1ET400300415 (first author) and the MSMT CR Project LC536 (second author).

References

Anderson, E., Bai, Z., Bischof, C., Blackford, L.S., Demmel, J., Dongarra, J., Croz, J., Du, J., Greeenhaum, A., Hammarling, S., McKenney, A., 2000. Lapack Users' Guide. SIAM, Philadelphia.

Bai, Z., Demmel, J., Dongarra, J., Ruhe, A., van der Vorst, H. (Eds.), 2000. Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide, 2000, SIAM, Philadelphia.

Bensmail, H., Celeux, G., 1996. Regularized Gaussian discriminant analysis through eigenvalue decomposition. J. Amer. Statist. Assoc. 91, 1743–1748.

Chen, L.-F., Liao, H.-Y.M., Ko, M.-T., Lin, J.-C., Yu, G.-J., 2000. A new LDA-based face recognition system which can solve the small sample size problem. Pattern Recognition 33 (10), 1713–1726.

Cheng, Y.-Q., Zhuang, Y.-M., Yang, J.-Y., 1992. Optimal Fisher discriminant analysis using the rank decomposition. Pattern Recognition 25 (1), 101–111.

Duda, R.O., Hart, P.E., Stork, D.G., 2000. Pattern Recognition. second ed. Wiley, New York.

Fisher, R.A., 1936. The use of multiple measurements in taxonomic problems. Ann. Eugenics 7, 179–188.

Friedman, J.H., 1989. Regularized discriminant analysis. J. Amer. Statist. Assoc. 84, 165–175.

Golub, G.H., van Loan, C.F., 1996. Matrix Computations. third ed. John Hopkins University Press, Baltimore.

Guo, Y.-F., Li, S.-J., Yang, J.-Y., Shu, T.-T., Wu, L.-D., 2003. A generalized Foley–Sammon transform based on generalized Fisher discriminant criterion and its application to face recognition. Pattern Recognition Lett. 24, 147–158.

Hastie, T., Tibshirani, R., 2003. Expression arrays and the $p \gg n$ problem. See ⟨http://www-stat.stanford.edu/∼hastie/Papers/pgtn.pdf⟩.

Hoffbeck, J.P., Landgrebe, D.A., 1996. Covariance matrix estimation and classification with limited training data. IEEE Trans. Pattern Anal. Mach. Intell. 18 (7), 763–767.

Hong, Z.-Q., Yang, J.-Y., 1991. Optimal discriminant plane for a small number of samples and design method of classifier on the plane. Pattern Recognition 24 (4), 317–324.

Howland, P., Park, H., 2004. Generalizing discriminant analysis using the generalized singular value decomposition. IEEE Trans. Pattern Anal. Mach. Intell. 26 (8), 995–1006.

Howland, P., Jeon, M., Park, H., 2003. Structure preserving dimension reduction for clustered text data based on the generalized singular value decomposition. SIAM J. Matrix Anal. Appl. 25 (1), 165–179.

Howland, P., Wang, J., Park, H., 2006. Solving the small sample size problem in face recognition using generalized discriminant analysis. Pattern Recognition 39, 277–287.

Johnson, R.A., Wichern, D.W., 1998. Applied Multivariate Statistical Analysis. Prentice-Hall, Engelwood Cliffs, NJ.

Kim, H., Howland, P., Park, P., 2005. Dimension reduction in text classification with support vector machines. J. Mach. Learn. Res. 6, 37–53.

Krzanowski, W.J., Jonathan, P., McCarthy, W.V., Thomas, M.R., 1995. Discriminant analysis with singular covariance matrices: methods and applications to spectroscopic data. Appl. Statist. 44 (1), 101–115.

Li, Y., Kittler, J., Matas, J., 1999. Effective implementation of linear discriminant analysis for face recognition and verification. In: Leonardis, A., Solina, F. (Eds.), Lecture Notes in Computer Science 1689. Springer, Berlin, pp. 234–242.

MathWorks, Inc., 1984–2005. MATLAB 7.0, ⟨http://www.mathworks.com/products/matlab/⟩.

Moler, C.B., Stewart, G.W., 1973. An algorithm for generalized matrix eigenvalue problems. SIAM J. Numer. Anal. 10 (2), 241–256.

Paige, C.C., Saunders, M.A., 1981. Towards a generalized singular value decomposition. SIAM J. Numer. Anal. 18 (3), 398–405.

Parlett, B.N., 1998. The Symmetric Eigenvalue Problem. SIAM, Philadelphia.

R Development Core Team, 2005. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna.

Ripley, B.D., 1996. Pattern Recognition and Neural Networks. Cambridge University Press, Cambridge.

Tibshirani, R., Hastie, T., Narasimhan, B., Chu, G., 2002. Diagnosis of multiple cancer types by shrunken centroids of gene expression. Proc. Nat. Acad. Sci. USA 99 (10), 6567–6572.

Venables, W.N., Ripley, B.D., 2002. Modern Applied Statistics with S. fourth ed. Springer, New York.

Yang, J., Yang, J.-Y., 2003. Why can LDA be performed in PCA transformed space? Pattern Recognition 36 (2), 563–566.

Yang, J., Yu, H., Kunz, W., 2000. An efficient LDA algorithm for face recognition. Sixth International Conference on Control, Automation, Robotics and Vision (ICARCV2000).