Institute of Phonetic Sciences,
University of Amsterdam,
Proceedings 23 (1999), 77–89.

# ACCURATE ALGORITHMS FOR PERFORMING PRINCIPAL COMPONENT ANALYSIS AND DISCRIMINANT ANALYSIS

*David Weenink*

## Abstract

We discuss two algorithms for performing principal component analysis and discriminant analysis. Both algorithms are based on singular value decomposition (svd). We calculate principal components and discriminants directly from the data matrix without forming the intermediate covariance matrices. In this way we do not loose accuracy. The methods described here have been implemented in the speech analysis program Praat.

## 1 Introduction

Principal component analysis (PCA) and discriminant analysis belong to the basic repertoire of multivariate data analysis. From a mathematical standpoint both methods try to construct an optimal orthogonal basis for the multidimensional data. They only differ in the choice of the optimality criterion. For principal component analysis the data are not labeled and we try to find orthogonal directions in which the variance is a maximum. These directions are called the principal directions and are unique up to a reflection. However, when the data are isotropically distributed in a subspace then there are no preferred directions in that subspace. In popular language one says that the first principal direction *explains* most of the variance. This means that when one projects the multidimensional data onto this single dimension the variance along this direction is a maximum, i.e. from all the possible directions in the multidimensional space no other direction shows this much variance for projected data. Translated into mathematical terms one tries to solve the following eigensystem

$$\boldsymbol{\Sigma}\mathbf{x} - \lambda\mathbf{x} = 0 \tag{1}$$

for the eigenvectors $\mathbf{x}$ and eigenvalues $\lambda$. The matrix $\boldsymbol{\Sigma}$ is the data covariance matrix which is a symmetric matrix.

For discriminant analysis we have labeled data, i.e., each row belongs to a certain group or category, and we try to find orthogonal directions among which discrimination between the different groups is optimal. In other words, one looks for directions in space where the ratio of the between-variance $\mathbf{B}$ and the within-variance $\mathbf{W}$ forms a maximum. This translates to the following eigensystem:

$$\mathbf{B}\mathbf{x} - \lambda\mathbf{W}\mathbf{x} = 0 \tag{2}$$

In this paper we will try to explain the origin of these equations and how we can solve them in an efficient and numerically stable way, without getting into mathematical detail.

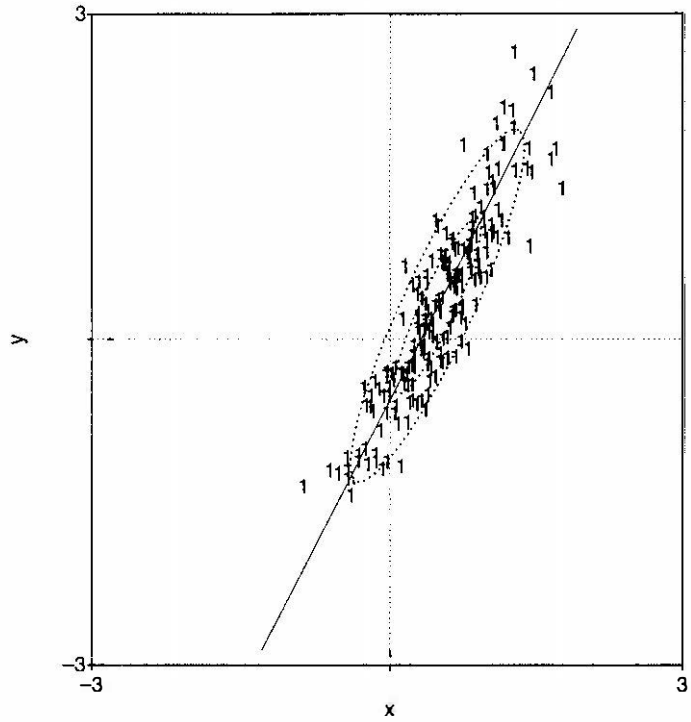


Fig. 1. Bivariate normally distributed random data centered at (0.5, 0.3) in the $xy$-plane with principal direction at an angle of 60 degrees with respect to the horizontal $x$-axis. The ellipses are the $1\sigma$ and $2\sigma$ ellipses that include approximately 39.3% and 86.5% of the data, respectively.

## 2 Principal component analysis

In figure 1 we have drawn 200 points in the $xy$-plane following a binormal distribution. These points were generated as follows.

- Both columns, $x$ and $y$, of a data matrix $\mathbf{A}$ with 200 rows are filled with Gaussian random deviates, centered at the origin, with $\sigma_x = 1$ and $\sigma_y = 0.2$.
- The points are rotated counterclockwise with an angle $\alpha$ of 60 degrees.
- The points are translated along the vector (0.5, 0.3).

When we calculate the variances in the new $x$ and $y$ column from $\mathbf{A}$ they will approximately be 0.53 ($= \sigma_x^2 \cos^2 \alpha + \sigma_y^2 \sin^2 \alpha$) and 0.87 ($= \sigma_x^2 \sin^2 \alpha + \sigma_y^2 \cos^2 \alpha$).

As can be seen from the figure, now the direction with maximum variance is not along the $x$ axis anymore, but along a vector that makes an angle of approximately 60 degrees with the horizontal $x$-axis. If we created a new variable by projecting the two-dimensional points onto this direction then the new one-dimensional variable would explain approximately 96% of the total variance ($96\% = \sigma_x^2/(\sigma_y^2 + \sigma_x^2)100\%$). This offers a possibility for data reduction. When we are satisfied with a description that explains 96% of the variance then we only need one variable, the first principal component, instead of the original two, the $x$ and $y$. A PCA-algorithm finds the orthogonal directions that have maximum variance.

Equation (1) can be derived as follows: we start with a general $m \times n$ data matrix $\mathbf{A}$. Let the first principal component $\mathbf{x}$ be the direction that maximizes the variance. The dimension of $\mathbf{x}$ equals $n$, the number of columns in $\mathbf{A}$. We project our data onto this direction and get a new vector $\mathbf{y}$ of projected data points $\mathbf{y} = \mathbf{Ax}$. The number of elements in $\mathbf{y}$ now equals $m$, the number of rows in $\mathbf{A}$. The variance (length) of this vector $\mathbf{y}$ should be the maximum of all lengths possible, i.e.,

$$\mathbf{y}'\mathbf{y} = (\mathbf{Ax})'(\mathbf{Ax}) = \mathbf{x}'\mathbf{A}'\mathbf{Ax} = \mathbf{x}'\mathbf{\Sigma x} \tag{3}$$

should be a maximum, where $\mathbf{A}'$ means the transpose of $\mathbf{A}$. In order to obtain meaningful solutions we have to constrain the length of the vector $\mathbf{x}$ we are looking for. If no constraints were imposed on $\mathbf{x}$, any $\mathbf{x}$ of infinite length would satisfy. One normally adds the constraint that $\mathbf{x}$ be a unit vector, $\mathbf{x}'\mathbf{x} = 1$. With the help of the Lagrange multiplier $\lambda$ this constraint can be included and the equation to maximize can be written as follows

$$\mathbf{x}'\mathbf{\Sigma x} - \lambda(\mathbf{x}'\mathbf{x} - 1).$$

Taking the derivative with respect to $\mathbf{x}$ and setting this derivative equal to zero we end up with:

$$\mathbf{\Sigma x} - \lambda\mathbf{x} = 0$$

which is the desired equation (1). There are many ways to solve eigensystems of this type where the matrix $\mathbf{\Sigma}$ is symmetrical. We can use a method due to Jacobi or we might first reduce $\mathbf{\Sigma}$ to tridiagonal form with a Householder reduction and then solve the resulting system with $\mathbf{QR}$ transformations (Press et al., 1996; Golub & van Loan, 1996). However, when we have the original data matrix $\mathbf{A}$ at our disposal, explicitly forming the product matrix, $\mathbf{A}'\mathbf{A} = \mathbf{\Sigma}$ is inadvisable because of a potential loss of information in finite precision arithmetic. When we define the condition number $c$ of a matrix $\mathbf{A}$ as the ratio of the largest eigenvalue to the smallest eigenvalue, then the condition number of the matrix product $\mathbf{A}'\mathbf{A}$ will be $c^2$. When $1/c^2$ is smaller than the machine precision $\epsilon$ the corresponding eigenvalue will be lost. This is where the singular value decomposition (SVD) enters. The SVD of an $m \times n$ matrix $\mathbf{A}$ has the form

$$\mathbf{A} = \mathbf{UDV}', \tag{4}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices of order $m$ and $n$, respectively, and $\mathbf{D}$ is an $m \times n$ nonnegative diagonal matrix. (For convenience we consider $m \geq n$. We also assume that our matrices are real. See Golub & van Loan (1996) for more details.) The diagonal elements $d_i$ of $\mathbf{D}$ are called the singular values of $\mathbf{A}$ and by convention are ordered so that $d_1 \geq d_2 \geq \cdots \geq d_n \geq 0$. The columns of $\mathbf{U}$ and $\mathbf{V}$ are orthonormal eigenvectors of $\mathbf{AA}'$ and $\mathbf{A}'\mathbf{A}$, respectively. We now use the SVD of $\mathbf{A}$ to calculate $\mathbf{\Sigma}$

$$\mathbf{\Sigma} = \mathbf{A}'\mathbf{A} = (\mathbf{UDV}')'(\mathbf{UDV}') = \mathbf{VDU}'\mathbf{UDV}' = \mathbf{VD}^2\mathbf{V}'. \tag{5}$$

This is a familiar result, it shows that any real symmetric matrix $\mathbf{\Sigma}$ can brought into diagonal form by a rotation. Now $\mathbf{D}^2$ is a matrix whose diagonal values are $d_i^2$, the squares of the singular values of $\mathbf{A}$. When we substitute the result from equation (5) into equation 1 we obtain

$$\mathbf{VD}^2\mathbf{V}'\mathbf{x} - \lambda\mathbf{x} = 0,$$

now multiplying with $\mathbf{V}'$ and using $\mathbf{V}$'s orthogonality results in

$$\mathbf{D}^2\mathbf{V}'\mathbf{x} - \lambda\mathbf{V}'\mathbf{x} = 0. \tag{6}$$

The solution for the equation (6) is now obvious. The vectors $\mathbf{x}$ that satisfy equation (1) are the column vectors from $\mathbf{V}$. The corresponding eigenvalues are the squares of

the singular values in the diagonal matrix $\mathbf{D}$. This result shows that by taking the SVD of $\mathbf{A}$ we can easily obtain the eigenvalues and eigenvectors of the matrix $\mathbf{\Sigma} = \mathbf{A}'\mathbf{A}$. Algorithms for determining the SVD of a rectangular matrix can be found in Press et al. (1996) and Golub & van Loan (1996). We do not loose any numerical precision by taking the SVD of $\mathbf{A}$. Every rectangular matrix can be decomposed according to equation (4). The advantage of the SVD-algorithm is its numerical stability. An additional advantage of the SVD algorithms is that they are robust against $\mathbf{A}$ not being of maximum rank, or, which is the same, $\mathbf{\Sigma}$ being singular. The singular values $d_i$ show how well behaved the matrix is. When the matrix is not of full rank the quotient of the largest and the smallest singular values will be very large.

The complete procedure for performing a PCA analysis, starting from the $m \times n$ data matrix $\mathbf{A}$ is now the following:

- Centralize the data in $\mathbf{A}$. For each element $a_{ij}$ in $\mathbf{A}$ subtract the column mean $a_{\cdot j} = \sum_1^m a_{ij}/m$. This gives a new *centralized* data matrix $\mathbf{C}$ with elements $c_{ij} = a_{ij} - a_{\cdot j}$.
- Calculate the singular value decomposition $\mathbf{C} = \mathbf{UDV}'$. Sort the singular values $d_i$ and corresponding columns of $\mathbf{V}$. Now $d_1$ will be the largest singular value and $d_n$ the smallest. Save only those singular values $d_i$ that satisfy $d_i/d_1 > \max(m, n)\epsilon$, where $\epsilon \approx 2.2 \times 10^{-16}$ is the machine precision for double precision floating point .
- Store the eigenvalues, $d_j^2$, with their corresponding eigenvector, $\mathbf{e}_j$. The most important direction will be the first eigenvector $\mathbf{e_1}$. The projection of the data matrix on $\mathbf{e_1}$ equals $\mathbf{y} = \mathbf{Ae_1}$, and is called the first principal component.

## 3 Discriminant analysis

As in the previous section, we will start with a $m \times n$ data matrix $\mathbf{A}$. Now, however, each row in the matrix is labeled as belonging to a certain group. There are $g$ different groups, each group has $n_i$ elements. Clearly $\sum_i^g n_i = m$. When we plot each row from $\mathbf{A}$ as a point in a space of dimension $n$ and label the point with its corresponding label, we will see points spread around the origin. Normally points belonging to the same group lie in each others neighborhood. The purpose of discriminant analysis is to find orthogonal directions in space such that along these directions the separation of the groups is optimal. In general, directions of maximum variance and directions of maximal separation need not have anything in common. To illustrate this we have drawn 200 points in figure 2 that were generated according to two different distributions. These points are labeled "1" or "2", depending on the distribution they belong to. Both distributions consist of 100 points that were generated as binormally distributed around the origin with $\sigma_x = 1$ and $\sigma_y = 0.2$ and subsequently rotated 60 degrees counterclockwise, in analogy with the data from figure 1. Next the points labeled "1" were translated along the vector (−0.5, 0.5) and the points labeled "2" were translated along the vector (0.2, −0.5). The $1\sigma$ ellipses for the points labeled "1" and "2" have also been drawn as well as the $1\sigma$ ellipse for the combined data set. The direction of maximum variance is the plain line that goes from the lower left to the upper right, along the long axis of the ellipse of the combined data. It is obvious that in this figure the direction in which best separation of the point labeled "1" and "2" is achieved, is along the dashed line that crosses the previous line.

The quantification of *best separation* is the following. Suppose that the $n$ dimensional vector $\mathbf{x}$ is the direction that gives maximum separation. When we project the

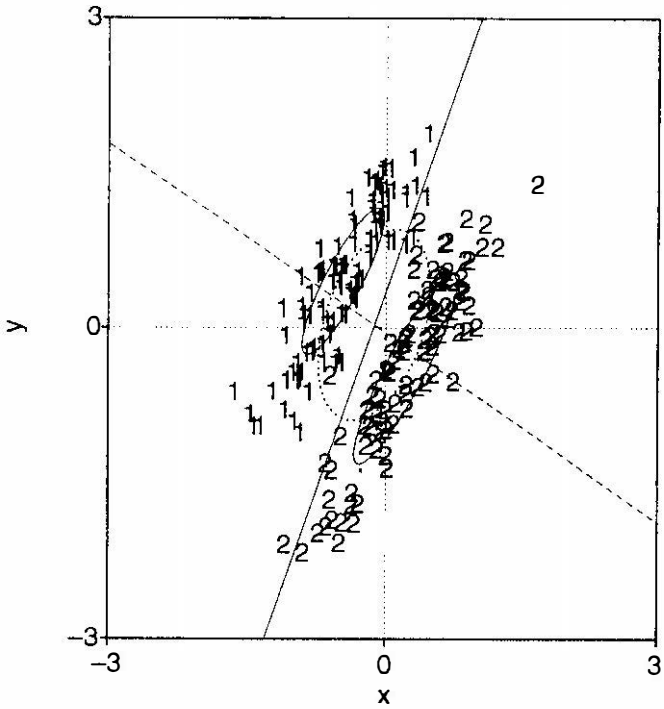


Fig. 2. Two bivariate normally distributed random data sets in the $xy$-plane. Both sets have their principal direction at an angle of approximately 60 degrees with respect to the horizontal $x$-axis. The data labeled "1" are centered at $(-0.5, 0.5)$ while the data labeled "2" are centered at $(0.2, -0.5)$. Both sets have $\sigma_1 = 1$ along the first principal axis and $\sigma_2 = 0.2$ along the second principal axis. The two small ellipses are the individual $1\sigma$ ellipses of the two data sets that cover approximately 39.3% of the data. The larger ellipse is the $1\sigma$ ellipse computed from the two data sets combined. The plain line bottom-left-to-top-right that approximately parallels the long axes of the ellipses corresponds to the direction of maximum variance. The dashed line is the direction along which discrimination is a maximum.

data onto this vector $\mathbf{x}$ all point now lie on a line. The points that belong to the same group hopefully cluster and lie close to each other. Good separation between the groups has been achieved when the clusters lie as far apart from each other as possible and the spread within each cluster is minimal. This amounts to saying that we want the variance of the group means along this direction to be as large as possible and at the same time the variances within the groups be as small as possible. It is mathematically more tractable if we express this as: find the maximum value for the F-ratio, i.e., the ratio of the variance of the group means and the variance within the groups.

In the language of matrices: we first calculate from $\mathbf{A}$ the $g$ group means and subsequently centralize $\mathbf{A}$. Call $\mathbf{C}$ the $m \times n$ matrix that results when we subtract from each row vector in $\mathbf{A}$ it's corresponding group mean. When we plot these points all clusters will be centered at the origin and no separate cluster would be noticeable. (For the data in figure 2 this would mean that all data now would lie in one elliptic region.) Now project these data onto $\mathbf{x}$ and form the $m$ dimensional vector $\mathbf{y}$ as $\mathbf{y} = \mathbf{Cx}$. The variance of $\mathbf{y}$ is a measure for the spread:

$$\mathbf{y}'\mathbf{y} = (\mathbf{Cx})'\mathbf{Cx} = \mathbf{x}'\mathbf{C}'\mathbf{Cx} = \mathbf{x}'\mathbf{Wx}.$$

$\mathbf{W}$ is the matrix with the so called within-group sums of squares and cross products (sscp). We now form the $g \times n$ matrix $\mathbf{M}$ with group means. The projection of the

group means on $\mathbf{x}$ leads to a new vector $\mathbf{z}$ defined as $\mathbf{z} = \mathbf{M}\mathbf{x}$. The variance of $\mathbf{z}$ is a measure for how far the groups lie from each other:

$$\mathbf{z}'\mathbf{z} = (\mathbf{M}\mathbf{x})'\mathbf{M}\mathbf{x} = \mathbf{x}'\mathbf{M}'\mathbf{M}\mathbf{x} = \mathbf{x}'\mathbf{B}\mathbf{x}.$$

$\mathbf{B}$ is the matrix with the between-group sums of squares and cross products. Now we have to find the vector $\mathbf{x}$ that maximizes the ratio $\phi(\mathbf{x})$ defined as[1]

$$\phi(\mathbf{x}) = \frac{\mathbf{x}'\mathbf{B}\mathbf{x}}{\mathbf{x}'\mathbf{W}\mathbf{x}} \tag{7}$$

However, to obtain meaningful solutions, we first put in the constraint $\mathbf{x}'\mathbf{x} = 1$ to obtain:

$$\phi(\mathbf{x}) = \frac{\mathbf{x}'\mathbf{B}\mathbf{x}}{\mathbf{x}'\mathbf{W}\mathbf{x}} - \lambda(\mathbf{x}'\mathbf{x} - 1)$$

The $\mathbf{x}$ that maximizes the equation above can be found by differentiation of $\phi(\mathbf{x})$ with respect to $\mathbf{x}$ and putting the result equal to zero. We then find the following eigenvalue equation:

$$\mathbf{B}\mathbf{x} - \lambda\mathbf{W}\mathbf{x} = 0 \tag{8}$$

To solve for the eigenvalues and eigenvectors of this equation in a numerically stable way is not a trivial matter because either the $\mathbf{B}$ or the $\mathbf{W}$ matrix might be singular. Golub & van Loan (1996) discuss methods for solving this type of equation. In general the eigenvectors of this equation are not orthogonal and we have to perform a subsequent orthogonalization step.

When $\mathbf{W}$ is not singular we can multiply by its inverse and obtain

$$\mathbf{W}^{-1}\mathbf{B}\mathbf{x} - \lambda\mathbf{x} = 0.$$

Although this equation has the same form as equation (1) there is also an important difference. The product $\mathbf{W}^{-1}\mathbf{B}$ of two symmetric matrices, does not result in a symmetric matrix and, therefore, the methods used before to solve equation (1) for the eigenvalues and eigenvectors are not valid here (apart from the fact that it would be numerically very unwise to explicitly form $\mathbf{W}^{-1}\mathbf{B}$).

Now for the special case that in equation (8) both matrices $\mathbf{B}$ and $\mathbf{W}$ are symmetric and 8 can be written as

$$\mathbf{M}'\mathbf{M}\mathbf{x} - \lambda\mathbf{C}'\mathbf{C}\mathbf{x} = 0 \tag{9}$$

there is an elegant solution possible without forming explicitly the matrix products $\mathbf{M}'\mathbf{M}$ and $\mathbf{C}'\mathbf{C}$ that could ruin our numerical precision. This solution is the generalization of the method we used with PCA-analysis and is called the *generalized singular value decomposition* (GSVD).

## 4 The generalized singular value decomposition

The generalized singular value decomposition (GSVD) decomposes two matrices at the same time into a common row basis. In the following we borrow the notation from Bai & Demmel (1993). The GSVD of an $m \times n$ matrix $\mathbf{A}$ and a $p \times n$ matrix $\mathbf{B}$ is given by

$$\mathbf{A} = \mathbf{U}\Sigma_1\mathbf{R}\mathbf{Q}' \quad \text{and} \quad \mathbf{B} = \mathbf{V}\Sigma_2\mathbf{R}\mathbf{Q}', \tag{10}$$

[1] Many more problems in multivariate data analysis can be formulated in the way of equation (7): find the optimum of the ratio $\phi(\mathbf{x}) = \mathbf{x}'\mathbf{E}\mathbf{x}/\mathbf{x}'\mathbf{F}\mathbf{x}$, where the matrices $\mathbf{E}$ and $\mathbf{F}$ are specific summaries of the data.

where $\mathbf{R}$ is $n \times n$ upper triangular and nonsingular, and

$$\Sigma_1 = \begin{array}{c} \\ l \\ k \\ m-l-k \end{array} \begin{array}{c} \begin{array}{ccc} l & k & n-l-k \end{array} \\ \begin{pmatrix} \mathbf{I}_1 & & \\ & \mathbf{D}_1 & \\ & & \mathbf{O}_1 \end{pmatrix} \end{array},$$

$$\Sigma_2 = \begin{array}{c} \\ p-n+l \\ k \\ n-l-k \end{array} \begin{array}{c} \begin{array}{ccc} l & k & n-l-k \end{array} \\ \begin{pmatrix} \mathbf{I}_2 & & \\ & \mathbf{D}_2 & \\ & & \mathbf{O}_2 \end{pmatrix} \end{array},$$

The $l \times l$ matrix $\mathbf{I}_1$ and the $(n-l-k) \times (n-l-k)$ matrix $\mathbf{I}_2$ are identity matrices. The $(m-l-k) \times (n-l-k)$ matrix $\mathbf{O}_1$ and the $(p-n+l) \times l$ matrix $\mathbf{O}_2$ are zero matrices. $\mathbf{D}_1 = \mathrm{diag}(\alpha_{l+1}, \cdots, \alpha_{l+k})$ and $\mathbf{D}_2 = \mathrm{diag}(\beta_{l+1}, \cdots, \beta_{l+k})$ are diagonal matrices, where

$$1 > \alpha_{l+k} \geq \cdots \geq \alpha_{l+k} > 0, \quad 0 < \beta_{l+1} \leq \cdots \leq \beta_{l+k} < 1, \quad \alpha_i^2 + \beta_i^2 = 1. \qquad (11)$$

The GSVD is a generalization of the SVD in the sense that: if $\mathbf{B}$ is the identity matrix then the GSVD reduces to the SVD of $\mathbf{A}$. The pairs $(\alpha_i, \beta_i)$ are called the singular value pairs. The quotient $\lambda_i = \alpha_i/\beta_i$ is called a generalized singular value. If $\beta_i = 0$ then the generalized singular value $\alpha_i/\beta_i$ is infinite. If $\mathbf{B}$ is square and nonsingular, then the GSVD of $\mathbf{A}$ and $\mathbf{B}$ reduces to the SVD of $\mathbf{AB}^{-1}$ :

$$\mathbf{AB}^{-1} = (\mathbf{U}\Sigma_1\mathbf{RQ}')(\mathbf{V}\Sigma_2\mathbf{RQ}')^{-1} = \mathbf{U}(\Sigma_1\Sigma_2^{-1})\mathbf{V}'.$$

Now we will show that the generalized eigenvalues and eigenvectors of $\mathbf{A}'\mathbf{A} - \lambda\mathbf{B}'\mathbf{B}$ can be expressed in terms of the GSVD. We define $\mathbf{X}$ as

$$\mathbf{X} = \mathbf{QR}^{-1}.$$

Then using equation (10) we find that

$$\mathbf{X}'\mathbf{A}'\mathbf{AX} = \Sigma_1'\Sigma_1 \quad \text{and} \quad \mathbf{X}'\mathbf{B}'\mathbf{BX} = \Sigma_2'\Sigma_2.$$

Therefore the columns of $\mathbf{X}$ are the eigenvectors of $\mathbf{A}'\mathbf{A} - \lambda\mathbf{B}'\mathbf{B}$, and the nontrivial eigenvalues are the squares of the generalized singular values. The matrix $\mathbf{X}$ is not an orthogonal matrix anymore. When we want an orthogonal set of eigenvectors, we have to use the columns of $\mathbf{Q}$. The GSVD of two matrices $\mathbf{A}$ and $\mathbf{B}$ can be calculated with the help of (modified versions of) the LAPACK routines `dtgsja` and `dggsvp` (Anderson et al., 1995). Routine `dtgsja` calculates the GSVD from two upper-triangle matrices $\mathbf{A}$ and $\mathbf{B}$ into the form of equation (10). The program `dggsvp` preprocesses two matrices $\mathbf{A}$ and $\mathbf{B}$ and brings them in upper-triangle form.

The complete procedure for performing a linear discriminant analysis, starting from the $m \times n$ data matrix $\mathbf{A}$ with $g$ different groups is now the following:

- Centralize the data in $\mathbf{A}$ per group. This results in a new centralized matrix $\mathbf{C}$ with elements: $c_{ij} = a_{ij} - a_{\cdot j}^k$, where $a_{\cdot j}^k$ is the average value of the elements in column $j$ that belong to group $k$.

- Collect the averages $a_{.j}^{k}$ in a $k \times n$ matrix $\mathbf{M}$.
- Calculate the generalized singular value decomposition of matrices $\mathbf{C}$ and $\mathbf{M}$.
  - Use a modified version of `dggsvp` to bring $\mathbf{C}$ and $\mathbf{M}$ into upper-triangular form.[2]
  - Use the output of `dggsvp` as input for a modified version of `dtgsja` to calculate $\mathbf{C} = \mathbf{U}\Sigma_1\mathbf{R}\mathbf{Q}'$ and $\mathbf{M} = \mathbf{V}\Sigma_2\mathbf{R}\mathbf{Q}'$.
- Calculate the generalized eigenvalues $\lambda_i^2$ from the $\alpha_i$'s and $\beta_i$'s of equation (11).

# 5 Discriminant analysis in the Praat program

## 5.1 Introduction

In this section we will demonstrate how we perform a discriminant analysis in the Praat program by Boersma & Weenink (1996). This section is also available as a tutorial on discriminant analysis within the Praat program itself (under "Help/Tutorials").

We will use the multivariate data set from Pols et al. (1973) with the first three formant frequency values and the levels in dB of the 12 Dutch monophthong vowels as spoken in /h_t/ context by 50 male speakers. This data set has been incorporated into the Praat program and can be called into play with the `Create TableOfReal from Pols data (50 males)...` command that can be found in the "New / TableOfReal" menu. In the list of objects a new TableOfReal object will appear, named `pols_50males`. After pressing the "Info" button, the "Info window" will show you that this table has 6 columns and 600 rows (50 speakers $\times$ 12 vowels). The first three columns contain the formant frequencies in Hz, the last three columns contain the levels of the first three formants given in decibels below the overall sound pressure level of the measured vowel segment. Each row is labelled with a vowel label. Pols et al. use logarithms of formant frequency values, we will do the same.[3] The following script summarizes our achievements up till now:

```
Create TableOfReal from Pols data (50 males)...  yes
Formula...  if col < 4 then log10 (self) else self endif
```

To get an indication how these data look, we make a scatter plot of the first log-formant-frequency against the second log-formant-frequency. With the next script fragment you can reproduce figure 3.

```
Viewport...  0 5 0 5
select TableOfReal pols_50males
Draw scatter plot...  1 2 0 0 2.3 3.0 2.8 3.5 no yes
```

This plot equals figure 3 in the Pols et al. study.

[2] We didn't like the automatically (from the FORTRAN sources) translated C-version of LAPACK. We have translated the routines that we needed from the FORTRAN-based LAPACK implementation to the C-language ourselves. This involved several changes in the routines because FORTRAN uses column-wise storage of matrices while C uses row-wise storage. FORTRAN always uses call-by-reference while C uses call-by-value. We did, however, maintain the FORTRAN way of letting array indices default start at 1.

[3] The measurement units in the first three columns and in the last three columns differ. For a discriminant analysis it is not necessary to standardize the columns, for a PCA one would normally standardize all columns first.

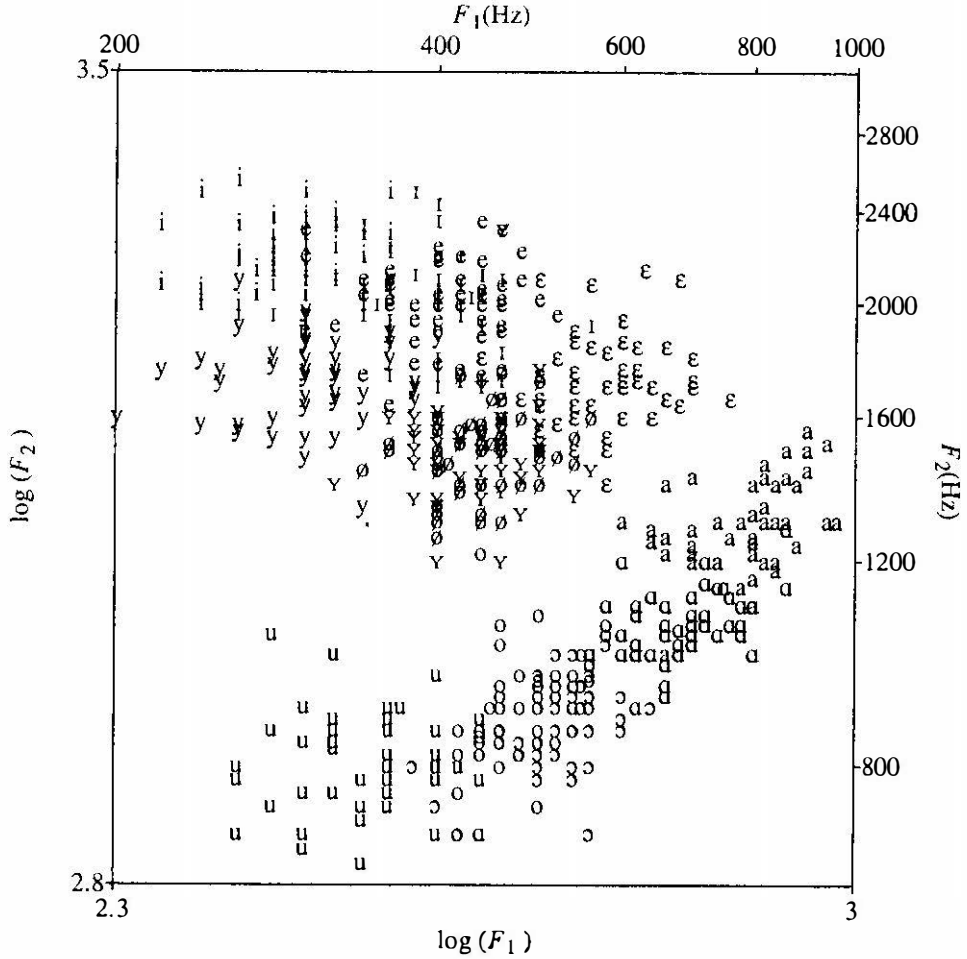


Fig. 3. First formant frequency versus second formant frequency of the Pols et al. data set on a logarithmic scale.

### 5.2 How to perform a discriminant analysis

To perform a discriminant analysis we select the TableOfReal from the list of objects and choose from the dynamic menu the option `To Discriminant`. This command is available in the "Multivariate statistics –" action button in the dynamic menu. The resulting Discriminant object will bear the same name as the TableOfReal object. The following script summarizes:

```
select TableOfReal pols_50males
To Discriminant
```

### 5.3 How to measure the correlation between the variables

To measure the correlation between the variables, we select the TableOfReal object and choose `To SSCP...`. The resulting SSCP object contains the sums of squares and cross-products of the column variables of the TableOfReal. Up to a simple scaling, we now almost have the correlation matrix. Subsequently choosing `To Correlation` results in the desired Correlation object. When we now choose `Draw as numbers...`, the numbers in the Picture window reproduce the numbers shown in the lower-left part of table III in the Pols et al. study. To calculate the numbers in the upper-right part of their table III, we first have to get the group centroids. We select the Discriminant object and choose the action `Extract group centroids`. This results in a Table-

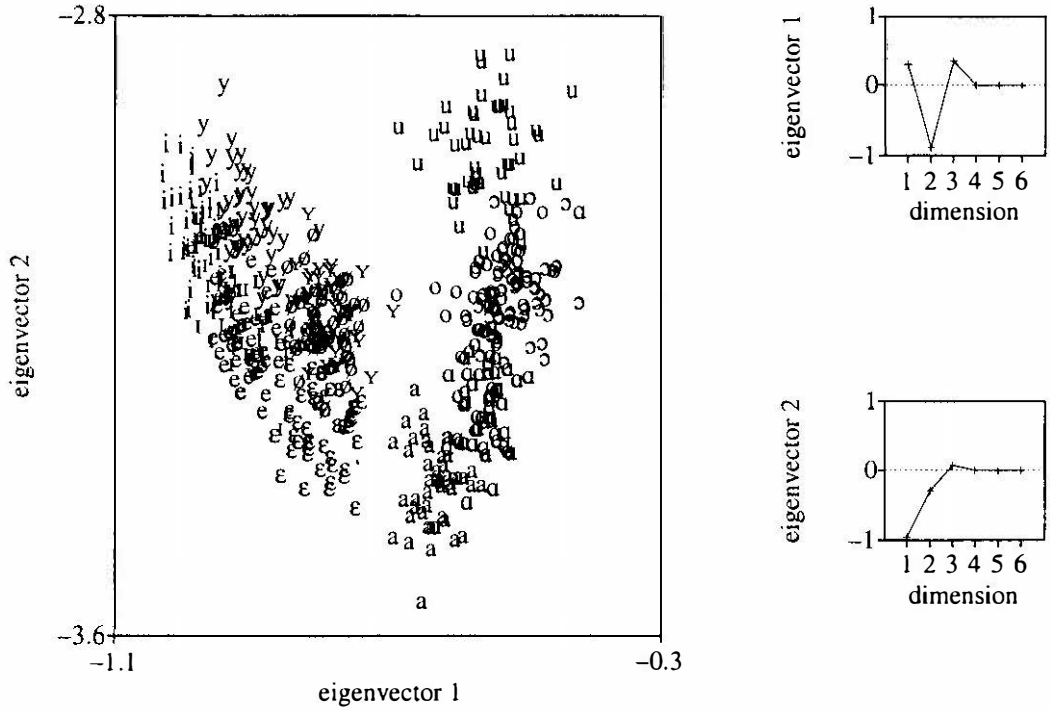


Fig. 4. Projection of the 6-dimensional Pols et al. data set on the plane spanned by the first two eigenvectors of the discriminant analysis.

OfReal object in which each row represents a group centroid. We then generate the Correlation matrix based on these centroids in the same way as was depicted above. The following scripts summarizes the procedure:

```
select TableOfReal pols_50males
To SSCP...  0 0 0 0
To Correlation
# draw lower-left part of Correlation matrix of data
Draw as numbers if...  1 0 free 3 col <= row
select Discriminant pols_50males
Extract group centroids
To SSCP...  0 0 0 0
To Correlation
# draw upper-right part of Correlation matrix of the means
Draw as numbers if...  1 0 free 3 row < col
```

### 5.4 How to project data on the discriminant space

To project the data on the discriminant space we select from the list of objects the TableOfReal and the Discriminant object together and choose: `To Configuration...`. The axes in the Configuration are the eigenvectors from the Discriminant. Figure 4 shows the data when projected onto the plane spanned by the first two dimensions of the Configuration. The plot on the left looks very similar to the $F_1$ vs. $F_2$ plot of figure 3. The eigenvectors show that indeed the $F_1$ and the $F_2$ variables have the largest weight (eigenvector 1 is dominated by $F_2$ and eigenvector 2 is dominated by $F_1$). The following script summarizes the procedure:

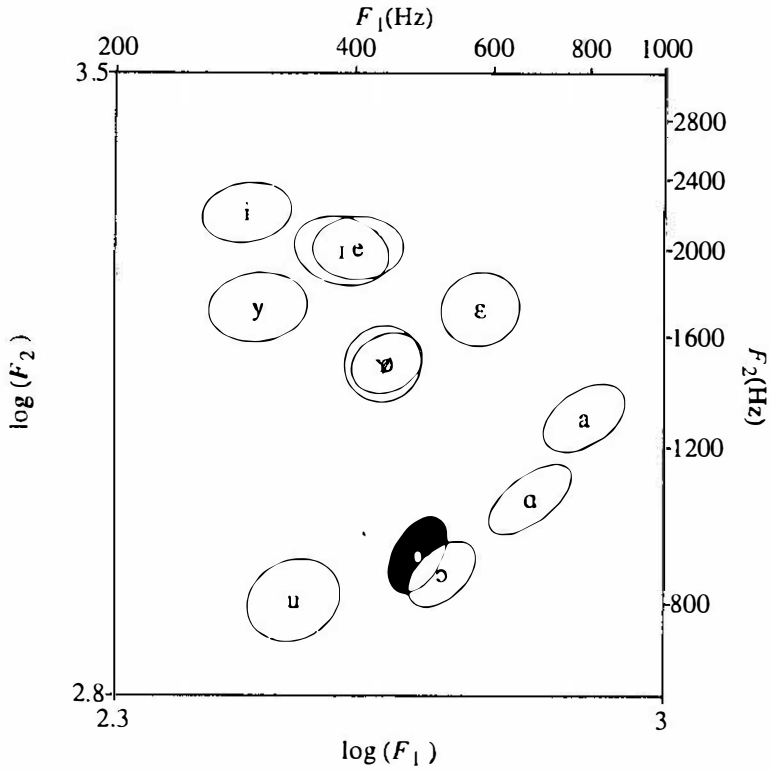


Fig. 5. Concentration ellipses of the Pols et al. data set in the plane spanned by the first and second formant frequency on a logarithmic scale.

```
select TableOfReal pols_50males
plus Discriminant pols_50males
To Configuration...  0
Viewport...  0 5 0 5
Draw...  1 2 -1.1 -0.3 -3.6 -2.8 yes
```

When you are only interested in the projection, there also is a short cut under the "Multivariate statistics –" button that deletes the intermediate Discriminant object:

```
select TableOfReal pols_50males
To Configuration (lda)...  2
```

### 5.5 How to draw concentration ellipses

To draw concentration ellipses for the different groups, we select from the list of objects the Discriminant object and choose `Draw sigma ellipses...`. In the form you can fill out the coverage of the ellipse by way of the parameter *numberOfSigmas*. You can also select the projection plane. Figure 5 shows the $1\sigma$ concentration ellipses in the standardized $\log(F_1)$ versus $\log(F_2)$ plane. When the data are multinormally distributed and projected onto a plane, an ellipse whose axes have length $2 \cdot numberOfSigmas$ covers approximately $(1 - e^{-numberOfSigmas^2/2}) \cdot 100\%$ of the data. The following code summarizes:

```
select Discriminant pols_50males
Draw sigma ellipses...  1.0 no 1 2 2.3 3.0 2.8 3.5 yes
```

### 5.6 How to classify

The following script summarizes how to classify:

```
select Discriminant pols_50males
plus TableOfReal pols_50males
To ClassificationTable... no yes
To Confusion
Get percentage correct
```

First we select together the classifier object and the data to be classified as is shown by the first two lines in the script.[4] In the example above, the test data set equals the train data set. Next we make a ClassificationTable that will contain the posterior probabilities of group membership $p_j$. The $p_j$ are defined as:

$$p_j = p(j|\mathbf{x}) = \frac{\exp(-d_j^2(\mathbf{x})/2)}{\sum_{k=1}^{g} \exp(-d_k^2(\mathbf{x})/2)},$$

where $d_i^2(\mathbf{x})$ is the generalized squared "distance" function

$$d_i^2(\mathbf{x}) = (\mathbf{x} - \boldsymbol{\mu}_i)' \boldsymbol{\Sigma}_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i) + \ln |\boldsymbol{\Sigma}_i| - 2 \ln apriori_i. \tag{12}$$

The first term in this "distance" function is the Mahalanobis distance, based on the individual covariance matrix $\boldsymbol{\Sigma}_i$. The following two terms are the logarithm of the determinant of $\boldsymbol{\Sigma}_i$, which might be negative, and the a priori probability $apriori_i$ of $\mathbf{x}$ belonging to group $i$, respectively. For each input vector $\mathbf{x}$ (each row in $\mathbf{A}$) we can calculate the $p_j$. As equation (12) indicates, we use the individual covariance matrices in the distance calculation. This has the consequence that this "distance" is not symmetric between different groups, it might even turn out to be negative.

Performing classification results in a percentage correct of 79.2%. This is 0.3% less than the 79.5% that can be found in table IV of the Pols et al. study.

## 6 Conclusion

We have described recent algorithms for performing principal components analysis and linear discriminant analysis. Both algorithms are based on singular value decomposition of the data matrix. There is no need for covariance matrices to be formed. It was shown that leaving out the latter operation has numerical advantages. Both algorithms were implemented in the computer program Praat and an example was shown how to use the latter algorithm effectively. The algorithms are designed for data sets that fit into the memory of the computer and are relatively fast on modern computers. Say, on a modern computer with 64 MB of memory approximately 24 MB can be used for the data matrix (we first make a copy of the data matrix and transform the copy). We can then handle a data matrix with three million cells with double precision numbers. This could for example be a matrix with 300 000 rows and 10 columns. By temporarily saving the original matrix we could even double the available storage space. Nevertheless there will always be data sets too large to handle in this way and then we have to fall back to other techniques.

[4] The data set to test the classifier may be any data set whose data format conforms to the data set that was used to train the discriminant classifier, i.e., the same variables in the same columns.

## Acknowledgement

The author wants to thank Louis Pols and Jan van Dijk for their critical reviews and constructive comments during this study.

## References

Anderson, E., Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. du Croz, A. Greenbaum, S. Hammarling, A. McKennedy, S. ●strouchov & D. Sorensen (1995): *LAPACK Users' Guide - Release 2.0*, Society for Industrial and Applied Mathematics.

Bai, Z. & J. W. Demmel (1993): "Computing the generalized singular value decomposition", *SIAM J. Sci. Comput.* **14**: 1464–1486.

Boersma, P. P. G. & D. J. M. Weenink (1996): *Praat, a system for doing phonetics by computer, version 3.4*, report 132, Institute Of Phonetic Sciences University of Amsterdam (up-to-date version of the manual at `http://www.fon.hum.uva.nl/praat/`).

Golub, G. H. & C. F. van Loan (1996): *Matrix Computations*, The John Hopkins University Press, 3rd edn.

Pols, L. C. W., H. Tromp & R. Plomp (1973): "Frequency analysis of Dutch vowels from 50 male speakers", *J. Acoust. Soc. Am.* **53**: 1093–1101.

Press, W. H., S. A. Teukolsky, W. T. Vetterling & B. P. Flannery (1996): *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, 2nd edn.