Version: March 18, 2014

Notes for Applied Multivariate Analysis: Linear Algebra Component

0.1 Multiple Regression

One of the most common topics in any beginning statistics class is *multiple regression* that we now formulate (in matrix terms) as the relation between a dependent random variable Y and a collection of K independent variables, $X_1, X_2, \ldots, X_K$. Suppose we have N subjects on which we observe Y, and arrange these values into an $N \times 1$ vector:

$$\mathbf{Y} = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_N \end{pmatrix}$$

The observations on the K independent variables are also placed in vectors:

$$\mathbf{X}_1 = \begin{pmatrix} X_{11} \\ X_{21} \\ \vdots \\ X_{N1} \end{pmatrix};\ \mathbf{X}_2 = \begin{pmatrix} X_{12} \\ X_{22} \\ \vdots \\ X_{N2} \end{pmatrix};\ \ldots;\ \mathbf{X}_K = \begin{pmatrix} X_{1K} \\ X_{2K} \\ \vdots \\ X_{NK} \end{pmatrix}$$

It would be simple if the vector $\mathbf{Y}$ were linearly dependent on $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_K$ because then

$$\mathbf{Y} = b_1\mathbf{X}_1 + b_2\mathbf{X}_2 + \cdots + b_K\mathbf{X}_K$$

for some values $b_1, \ldots, b_K$. We could always write for *any* values of $b_1, \ldots, b_K$:

$$\mathbf{Y} = b_1\mathbf{X}_1 + b_2\mathbf{X}_2 + \cdots + b_K\mathbf{X}_K + \mathbf{e}$$

where

$$\mathbf{e} = \begin{pmatrix} e_1 \\ \vdots \\ e_N \end{pmatrix}$$

is an error vector. To formulate our task as an optimization problem (least-squares), we wish to find a good set of weights, $b_1, \ldots, b_K$, so the length of $\mathbf{e}$ is minimized, i.e., $\mathbf{e}'\mathbf{e}$ is made as small as possible.

As notation, let

$$\mathbf{Y}_{N\times 1} = \mathbf{X}_{N\times K}\mathbf{b}_{K\times 1} + \mathbf{e}_{N\times 1}$$

where

$$\mathbf{X} = \begin{pmatrix} \mathbf{X}_1 & \ldots & \mathbf{X}_K \end{pmatrix};\ \mathbf{b} = \begin{pmatrix} b_1 \\ \vdots \\ b_K \end{pmatrix}$$

To minimize $\mathbf{e}'\mathbf{e} = (\mathbf{Y} - \mathbf{X}\mathbf{b})'(\mathbf{Y} - \mathbf{X}\mathbf{b})$, we use the vector $\mathbf{b}$ that satisfies what are called the normal equations:

$$\mathbf{X}'\mathbf{X}\mathbf{b} = \mathbf{X}'\mathbf{Y}$$

If $\mathbf{X}'\mathbf{X}$ is nonsingular (i.e., $\det(\mathbf{X}'\mathbf{X}) \neq 0$; or equivalently, $\mathbf{X}_1, \ldots, \mathbf{X}_K$ are linearly independent), then

$$\mathbf{b} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$$

The vector that is "closest" to $\mathbf{Y}$ in our least-squares sense, is $\mathbf{X}\mathbf{b}$; this is a linear combination of the columns of $\mathbf{X}$ (or in other jargon,

$\mathbf{Xb}$ defines the *projection* of $\mathbf{Y}$ into the space defined by (all linear combinations of) the columns of $\mathbf{X}$.

In statistical uses of multiple regression, the estimated variance-covariance matrix of the regression coefficients, $b_1, \ldots, b_K$, is given as $(\frac{1}{N-K})\mathbf{e}'\mathbf{e}(\mathbf{X}'\mathbf{X})^{-1}$, where $(\frac{1}{N-K})\mathbf{e}'\mathbf{e}$ is an (unbiased) estimate of the error variance for the distribution from which the errors are assumed drawn. Also, in multiple regression instances that usually involve an additive constant, the latter is obtained from a weight attached to an independent variable defined to be identically one.

In multivariate multiple regression where there are, say, T dependent variables (each represented by an $N \times 1$ vector), the dependent vectors are merely concatenated together into an $N \times T$ matrix, $\mathbf{Y}_{N\times T}$; the solution to the normal equations now produces a matrix $\mathbf{B}_{K\times T} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$ of regression coefficients. In effect, this general expression just uses each of the dependent variables separately and adjoins all the results.

0.2 Eigenvectors and Eigenvalues

Suppose we are given a square matrix, $\mathbf{A}_{U\times U}$, and consider the polynomial $\det(\mathbf{A} - \lambda\mathbf{I})$ in the unknown value λ, referred to as Laplace's expansion:

$$\det(\mathbf{A} - \lambda\mathbf{I}) = (-\lambda)^U + S_1(-\lambda)^{U-1} + \cdots + S_{U-1}(-\lambda)^{-1} + S_U(-\lambda)^0$$

where S_u is the sum of all $u \times u$ principal minor determinants. A *principal* minor determinant is obtained from a submatrix formed from $\mathbf{A}$ that has u diagonal elements left in it. Thus, S_1 is the trace of $\mathbf{A}$ and S_U is the determinant.

There are U roots, $\lambda_1, \ldots, \lambda_U$, of the equation $\det(\mathbf{A} - \lambda\mathbf{I}) = 0$, given that the left-hand-side is a U^{th} degree polynomial. The roots are called the *eigenvalues* of $\mathbf{A}$. There are a number of properties of eigenvalues that prove generally useful:

(A) $\det \mathbf{A} = \Pi_{u=1}^{U} \lambda_u$; $\text{trace}(\mathbf{A}) = \Sigma_{u=1}^{U} \lambda_u$;

(B) if $\mathbf{A}$ is symmetric with real elements, then all λ_u are real;

(C) if $\mathbf{A}$ is positive definite, then all λ_u are positive (strictly greater than zero); if $\mathbf{A}$ is positive semi-definite, then all λ_u are nonnegative (greater than or equal to zero);

(D) if $\mathbf{A}$ is symmetric and positive semi-definite with rank R, then there are R positive roots and $U - R$ zero roots;

(E) the nonzero roots of $\mathbf{AB}$ are equal to those of $\mathbf{BA}$; thus, the trace of $\mathbf{AB}$ is equal to the trace of $\mathbf{BA}$;

(F) eigenvalues of a diagonal matrix are the diagonal elements themselves;

(G) for any $U \times V$ matrix $\mathbf{B}$, the ranks of $\mathbf{B}$, $\mathbf{B}'\mathbf{B}$, and $\mathbf{B}\mathbf{B}'$ are all the same. Thus, because $\mathbf{B}'\mathbf{B}$ (and $\mathbf{B}\mathbf{B}'$) are symmetric and positive semi-definite (i.e., $\mathbf{x}'(\mathbf{B}'\mathbf{B})\mathbf{x} \geq 0$ because $(\mathbf{B}\mathbf{x})'(\mathbf{B}\mathbf{x})$ is a sum-of-squares which is always nonnegative), we can use (D) to find the rank of $\mathbf{B}$ by counting the positive roots of $\mathbf{B}'\mathbf{B}$.

We carry through a small example below:

$$\mathbf{A} = \begin{pmatrix} 7 & 0 & 1 \\ 0 & 7 & 2 \\ 1 & 2 & 3 \end{pmatrix}$$

$$S_1 = \text{trace}(\mathbf{A}) = 17$$

$$S_2 = \det(\begin{pmatrix} 7 & 0 \\ 0 & 7 \end{pmatrix}) + \det(\begin{pmatrix} 7 & 1 \\ 1 & 3 \end{pmatrix}) + \det(\begin{pmatrix} 7 & 2 \\ 2 & 3 \end{pmatrix}) = 49 + 20 + 17 = 86$$

$$S_3 = \det(\mathbf{A}) = 147 + 0 + 0 - 7 - 28 - 0 = 112$$

Thus,

$$\det(\mathbf{A} - \lambda\mathbf{I}) = (-\lambda)^3 + 17(-\lambda)^2 + 86(-\lambda)^1 + 112 =$$

$$-\lambda^3 + 17\lambda^2 - 86\lambda + 112 = -(\lambda - 2)(\lambda - 8)(\lambda - 7) = 0$$

which gives roots of 2, 8, and 7.

If λ_u is an eigenvalue of $\mathbf{A}$, then the equations $[\mathbf{A} - \lambda_u\mathbf{I}]\mathbf{x}_u = 0$ have a nontrivial solution (i.e., the determinant of $\mathbf{A} - \lambda_u\mathbf{I}$ vanishes, and so the inverse of $\mathbf{A} - \lambda_u\mathbf{I}$ does not exist). The solution is called an *eigenvector* (associated with the corresponding eigenvalue), and can be characterized by the following condition:

$$\mathbf{A}\mathbf{x}_u = \lambda_u\mathbf{x}_u$$

An eigenvector is determined up to a scale factor only, so typically we normalize to unit length (which then gives a $\pm$ option for the two possible unit length solutions).

We continue our simple example and find the corresponding eigenvalues: when $\lambda = 2$, we have the equations (for $[\mathbf{A} - \lambda\mathbf{I}]\mathbf{x} = \mathbf{0}$)

$$\begin{pmatrix} 5 & 0 & 1 \\ 0 & 5 & 2 \\ 1 & 2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

with an arbitrary solution of

$$\begin{pmatrix} -\frac{1}{5}a \\ -\frac{2}{5}a \\ a \end{pmatrix}$$

Choosing a to be $+\frac{5}{\sqrt{30}}$ to obtain one of the two possible normalized solutions, we have as our final eigenvector for $\lambda = 2$:

$$\begin{pmatrix} -\frac{1}{\sqrt{30}} \\ -\frac{2}{\sqrt{30}} \\ \frac{5}{\sqrt{30}} \end{pmatrix}$$

For $\lambda = 7$ we will use the normalized eigenvector of

$$\begin{pmatrix} -\frac{2}{\sqrt{5}} \\ \frac{1}{\sqrt{5}} \\ 0 \end{pmatrix}$$

and for $\lambda = 8$,

$$\begin{pmatrix} \frac{1}{\sqrt{6}} \\ \frac{2}{\sqrt{6}} \\ \frac{1}{\sqrt{6}} \end{pmatrix}$$

One of the interesting properties of eigenvalues/eigenvectors for a symmetric matrix $\mathbf{A}$ is that if λ_u and λ_v are distinct eigenvalues, then the corresponding eigenvectors, $\mathbf{x}_u$ and $\mathbf{x}_v$, are orthogonal (i.e., $\mathbf{x}_u'\mathbf{x}_v = 0$). We can show this in the following way: the defining conditions of

$$\mathbf{A}\mathbf{x}_u = \lambda_u \mathbf{x}_u$$

$$\mathbf{A}\mathbf{x}_v = \lambda_v \mathbf{x}_v$$

lead to

$$\mathbf{x}_v'\mathbf{A}\mathbf{x}_u = \mathbf{x}_v'\lambda_u\mathbf{x}_u$$

$$\mathbf{x}_u'\mathbf{A}\mathbf{x}_v = \mathbf{x}_u'\lambda_v\mathbf{x}_v$$

Because $\mathbf{A}$ is symmetric and the left-hand-sides of these two expressions are equal (they are one-by-one matrices and equal to their own transposes), the right-hand-sides must also be equal. Thus,

$$\mathbf{x}_v'\lambda_u\mathbf{x}_u = \mathbf{x}_u'\lambda_v\mathbf{x}_v \Rightarrow$$

$$\mathbf{x}_v'\mathbf{x}_u\lambda_u = \mathbf{x}_u'\mathbf{x}_v\lambda_v$$

Due to the equality of $\mathbf{x}_v'\mathbf{x}_u$ and $\mathbf{x}_u'\mathbf{x}_v$, and by assumption, $\lambda_u \neq \lambda_v$, the inner product $\mathbf{x}_v'\mathbf{x}_u$ must be zero for the last displayed equality to hold.

In summary of the above discussion, for every real symmetric matrix $\mathbf{A}_{U\times U}$, there exists an orthogonal matrix $\mathbf{P}$ (i.e., $\mathbf{P}'\mathbf{P} = \mathbf{P}\mathbf{P}' = \mathbf{I}$) such that $\mathbf{P}'\mathbf{A}\mathbf{P} = \mathbf{D}$, where $\mathbf{D}$ is a diagonal matrix containing the eigenvalues of $\mathbf{A}$, and

$$\mathbf{P} = \begin{pmatrix} \mathbf{p}_1 & \ldots & \mathbf{p}_U \end{pmatrix}$$

where $\mathbf{p}_u$ is a normalized eigenvector associated with λ_u for $1 \leq u \leq U$. If the eigenvalues are not distinct, it is still possible to choose the eigenvectors to be orthogonal. Finally, because $\mathbf{P}$ is an orthogonal matrix (and $\mathbf{P}'\mathbf{A}\mathbf{P} = \mathbf{D} \Rightarrow \mathbf{P}\mathbf{P}'\mathbf{A}\mathbf{P}\mathbf{P}' = \mathbf{P}\mathbf{D}\mathbf{P}'$), we can finally represent $\mathbf{A}$ as

$$\mathbf{A} = \mathbf{P}\mathbf{D}\mathbf{P}'$$

In terms of the small numerical example being used, we have for $\mathbf{P}'\mathbf{A}\mathbf{P} = \mathbf{D}$:

$$\begin{pmatrix} -\frac{1}{\sqrt{30}} & -\frac{2}{\sqrt{30}} & \frac{5}{\sqrt{30}} \\ -\frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} & 0 \\ \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} \end{pmatrix} \begin{pmatrix} 7 & 0 & 1 \\ 0 & 7 & 2 \\ 1 & 2 & 3 \end{pmatrix} \begin{pmatrix} -\frac{1}{\sqrt{30}} & -\frac{2}{\sqrt{5}} & \frac{1}{\sqrt{6}} \\ -\frac{2}{\sqrt{30}} & \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{6}} \\ \frac{5}{\sqrt{30}} & 0 & \frac{1}{\sqrt{6}} \end{pmatrix} =$$

$$\begin{pmatrix} 2 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 8 \end{pmatrix}$$

and for $\mathbf{PDP}' = \mathbf{A}$:

$$\begin{pmatrix} -\frac{1}{\sqrt{30}} & -\frac{2}{\sqrt{5}} & \frac{1}{\sqrt{6}} \\ -\frac{2}{\sqrt{30}} & \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{6}} \\ \frac{5}{\sqrt{30}} & 0 & \frac{1}{\sqrt{6}} \end{pmatrix} \begin{pmatrix} 2 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 8 \end{pmatrix} \begin{pmatrix} -\frac{1}{\sqrt{30}} & -\frac{2}{\sqrt{30}} & \frac{5}{\sqrt{30}} \\ -\frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} & 0 \\ \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} \end{pmatrix} =$$

$$\begin{pmatrix} 7 & 0 & 1 \\ 0 & 7 & 2 \\ 1 & 2 & 3 \end{pmatrix}$$

The representation of $\mathbf{A}$ as $\mathbf{PDP}'$ leads to several rather nice computational "tricks." First, if $\mathbf{A}$ is p.s.d., we can define

$$\mathbf{D}^{1/2} \equiv \begin{pmatrix} \sqrt{\lambda_1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sqrt{\lambda_U} \end{pmatrix}$$

and represent $\mathbf{A}$ as

$$\mathbf{A} = \mathbf{PD}^{1/2}\mathbf{D}^{1/2}\mathbf{P}' = \mathbf{PD}^{1/2}(\mathbf{PD}^{1/2})' = \mathbf{LL}', \text{say.}$$

In other words, we have "factored" $\mathbf{A}$ into $\mathbf{LL}'$, for

$$\mathbf{L} = \mathbf{PD}^{1/2} = \begin{pmatrix} \sqrt{\lambda_1}\mathbf{p}_1 & \sqrt{\lambda_2}\mathbf{p}_2 & \dots & \sqrt{\lambda_U}\mathbf{p}_U \end{pmatrix}$$

Secondly, if $\mathbf{A}$ is p.d., we can define

$$\mathbf{D}^{-1} \equiv \begin{pmatrix} \frac{1}{\lambda_1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \frac{1}{\lambda_U} \end{pmatrix}$$

and represent $\mathbf{A}^{-1}$ as

$$\mathbf{A}^{-1} = \mathbf{P}\mathbf{D}^{-1}\mathbf{P}'$$

To verify,

$$\mathbf{A}\mathbf{A}^{-1} = (\mathbf{P}\mathbf{D}\mathbf{P}')(\mathbf{P}\mathbf{D}^{-1}\mathbf{P}') = \mathbf{I}$$

Thirdly, to define a "square root" matrix, let $\mathbf{A}^{1/2} \equiv \mathbf{P}\mathbf{D}^{1/2}\mathbf{P}'$. To verify, $\mathbf{A}^{1/2}\mathbf{A}^{1/2} = \mathbf{P}\mathbf{D}\mathbf{P}' = \mathbf{A}$.

There is a generally interesting way to represent the multiplication of two matrices considered as collections of column and row vectors, respectively, where the final answer is a sum of outer products of vectors. This view will prove particularly useful in our discussion of principal component analysis. Suppose we have two matrices $\mathbf{B}_{U\times V}$, represented as a collection of its V columns:

$$\mathbf{B} = \begin{pmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \dots & \mathbf{b}_V \end{pmatrix}$$

and $\mathbf{C}_{V\times W}$, represented as a collection of its V rows:

$$\mathbf{C} = \begin{pmatrix} \mathbf{c}_1' \\ \mathbf{c}_2' \\ \vdots \\ \mathbf{c}_V' \end{pmatrix}$$

The product $\mathbf{B}\mathbf{C} = \mathbf{D}$ can be written as

$$\mathbf{BC} = \begin{pmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \dots & \mathbf{b}_V \end{pmatrix} \begin{pmatrix} \mathbf{c}_1' \\ \mathbf{c}_2' \\ \vdots \\ \mathbf{c}_V' \end{pmatrix} =$$

$$\mathbf{b}_1\mathbf{c}_1' + \mathbf{b}_2\mathbf{c}_2' + \cdots + \mathbf{b}_V\mathbf{c}_V' = \mathbf{D}$$

As an example, consider the *spectral decomposition* of $\mathbf{A}$ considered above as $\mathbf{PDP}'$, and where from now on, without loss of any generality, the diagonal entries in $\mathbf{D}$ are ordered as $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_U$. We can represent $\mathbf{A}$ as

$$\mathbf{A}_{U\times U} = \begin{pmatrix} \sqrt{\lambda_1}\mathbf{p}_1 & \dots & \sqrt{\lambda_U}\mathbf{p}_U \end{pmatrix} \begin{pmatrix} \sqrt{\lambda_1}\mathbf{p}_1' \\ \vdots \\ \sqrt{\lambda_U}\mathbf{p}_U' \end{pmatrix} =$$

$$\lambda_1\mathbf{p}_1\mathbf{p}_1' + \cdots + \lambda_U\mathbf{p}_U\mathbf{p}_U'$$

If $\mathbf{A}$ is p.s.d. and of rank R, then the above sum obviously stops at R components. In general, the matrix $\mathbf{B}_{U\times U}$ that is a rank K ($\leq R$) least-squares approximation to $\mathbf{A}$ can be given by

$$\mathbf{B} = \lambda_1\mathbf{p}_1\mathbf{p}_1' + \cdots + \lambda_k\mathbf{p}_K\mathbf{p}_K'$$

and the value of the loss function:

$$\sum_{v=1}^{U}\sum_{u=1}^{U}(a_{uv} - b_{uv})^2 = \lambda_{K+1}^2 + \cdots + \lambda_U^2$$

0.3 The Singular Value Decomposition of a Matrix

The *singular value decomposition* (SVD) or the *basic structure* of a matrix refers to the representation of *any* rectangular $U \times V$ matrix, say, $\mathbf{A}$, as a triple product:

$$\mathbf{A}_{U\times V} = \mathbf{P}_{U\times R}\mathbf{\Delta}_{R\times R}\mathbf{Q}'_{R\times V}$$

where the R columns of $\mathbf{P}$ are orthonormal; the R rows of $\mathbf{Q}'$ are orthonormal; $\mathbf{\Delta}$ is diagonal with ordered positive entries, $\delta_1 \geq \delta_2 \geq \cdots \geq \delta_R > 0$; and R is the rank of $\mathbf{A}$. Or, alternatively, we can "fill up" this decomposition as

$$\mathbf{A}_{U\times V} = \mathbf{P}^*_{U\times U}\mathbf{\Delta}^*_{U\times V}\mathbf{Q}^{*'}_{V\times V}$$

where the columns of $\mathbf{P}^*$ and rows of $\mathbf{Q}^{*'}$ are still orthonormal, and the diagonal matrix $\mathbf{\Delta}$ forms the upper-left-corner of $\mathbf{\Delta}^*$:

$$\mathbf{\Delta}^* = \begin{pmatrix} \mathbf{\Delta} & \emptyset \\ \emptyset & \emptyset \end{pmatrix}$$

here, $\emptyset$ represents an appropriately dimensioned matrix of all zeros. In analogy to the least-squares result of the last section, if a rank K ($\leq R$) matrix approximation to $\mathbf{A}$ is desired, say $\mathbf{B}_{U\times V}$, the first K ordered entries in $\mathbf{\Delta}$ are taken:

$$\mathbf{B} = \delta_1\mathbf{p}_1\mathbf{q}'_1 + \cdots + \delta_K\mathbf{p}_K\mathbf{q}'_K$$

and the value of the loss function:

$$\sum_{v=1}^{V}\sum_{u=1}^{U}(a_{uv} - b_{uv})^2 = \delta^2_{K+1} + \cdots + \delta^2_R$$

This latter result of approximating one matrix (least-squares) by another of lower rank, is referred to as the Eckart-Young theorem in the psychometric literature.

Once one has the SVD of a matrix, a lot of representation needs can be expressed in terms of it. For example, suppose $\mathbf{A} = \mathbf{P}\mathbf{\Delta}\mathbf{Q}'$;

the spectral decomposition of $\mathbf{AA'}$ can then be given as

$$(\mathbf{P\Delta Q'})(\mathbf{P\Delta Q'})' = \mathbf{P\Delta Q'Q\Delta P'} = \mathbf{P\Delta\Delta P'} = \mathbf{P\Delta}^2\mathbf{P'}$$

Similarly, the spectral decomposition of $\mathbf{A'A}$ is expressible as $\mathbf{Q\Delta}^2\mathbf{Q'}$.

0.4 Common Multivariate Methods in Matrix Terms

In this section we give brief overviews of some common methods of multivariate analysis in terms of the matrix ideas we have introduced thus far in this chapter. We come back to a few of these topics later and develop them in more detail.

0.4.1 Principal Components

Suppose we have a data matrix $\mathbf{X}_{N\times P} = \{x_{ij}\}$, with x_{ij} referring as usual to the observation for subject i on variable or column j:

$$\mathbf{X}_{N\times P} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1P} \\ x_{21} & x_{22} & \cdots & x_{2P} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{NP} \end{pmatrix}$$

The columns can be viewed as containing N observations on each of P random variables that we denote generically by $X_1, X_2, \ldots, X_P$. We let $\mathbf{A}$ denote the $P\times P$ sample covariance matrix obtained among the variables from $\mathbf{X}$, and let $\lambda_1 \geq \cdots \geq \lambda_P \geq 0$ be its P eigenvalues and $\mathbf{p}_1, \ldots, \mathbf{p}_P$ the corresponding normalized eigenvectors. Then, the linear combination

$$\mathbf{p}_k' \begin{pmatrix} X_1 \\ \vdots \\ X_P \end{pmatrix}$$

is called the k^{th} (sample) *principal component*.

There are (at least) two interesting properties of principal components to bring up at this time:

A) The k^{th} principal component has maximum variance among all linear combinations defined by unit length vectors orthogonal to $\mathbf{p}_1, \ldots, \mathbf{p}_{k-1}$; also, it is uncorrelated with the components up to $k-1$;

B) $\mathbf{A} \approx \lambda_1 \mathbf{p}_1 \mathbf{p}_1' + \cdots + \lambda_K \mathbf{p}_K \mathbf{p}_K'$ gives a least-squares rank K approximation to $\mathbf{A}$ (a special case of the Eckart-Young theorem for an arbitrary symmetric matrix).

0.4.2 Discriminant Analysis

Suppose we have a one-way analysis-of-variance (ANOVA) layout with J groups (n_j subjects in group j, $1 \leq j \leq J$), and P measurements on each subject. If x_{ijk} denotes person i, in group j, and the observation of variable k ($1 \leq i \leq n_j$; $1 \leq j \leq J$; $1 \leq k \leq P$), then define the Between-Sum-of-Squares matrix

$$\mathbf{B}_{P \times P} = \{ \sum_{j=1}^{J} n_j (\bar{x}_{\cdot jk} - \bar{x}_{\cdot\cdot k})(\bar{x}_{\cdot jk'} - \bar{x}_{\cdot\cdot k'}) \}_{P \times P}$$

and the Within-Sum-of-Squares matrix

$$\mathbf{W}_{P \times P} = \{ \sum_{j=1}^{J} \sum_{i=1}^{n_j} (x_{ijk} - \bar{x}_{\cdot jk})(x_{ijk'} - \bar{x}_{\cdot jk'}) \}_{P \times P}$$

For the matrix product $\mathbf{W}^{-1}\mathbf{B}$, let $\lambda_1, \ldots, \lambda_T \geq 0$ be the eigenvectors ($T = \min(P, J-1)$, and $\mathbf{p}_1, \ldots, \mathbf{p}_T$ the corresponding normalized eigenvectors. Then, the linear combination

$$\mathbf{p}_k' \begin{pmatrix} X_1 \\ \vdots \\ X_P \end{pmatrix}$$

is called the k^{th} *discriminant function*. It has the valuable property of maximizing the univariate F-ratio subject to being uncorrelated with the earlier linear combinations. A variety of applications of discriminant functions exists in classification that we will come back to later. Also, standard multivariate ANOVA significance testing is based on various functions of the eigenvalues $\lambda_1, \ldots, \lambda_T$ and their derived sampling distributions.

0.4.3 Canonical Correlation

Suppose the collection of P random variables that we have observed over the N subjects is actually in the form of two "batteries," $X_1, \ldots, X_Q$ and $X_{Q+1}, \ldots, X_P$, and the observed covariance matrix $\mathbf{A}_{P\times P}$ is partitioned into four parts:

$$\mathbf{A}_{P\times P} = \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}'_{12} & \mathbf{A}_{22} \end{pmatrix}$$

where $\mathbf{A}_{11}$ is $Q \times Q$ and represents the observed covariances among the variables in the first battery; $\mathbf{A}_{22}$ is $(P-Q)\times(P-Q)$ and represents the observed covariances among the variables in the second battery; $\mathbf{A}_{12}$ is $Q \times (P-Q)$ and represents the observed covariances between the variables in the first and second batteries. Consider the following two equations in unknown vectors $\mathbf{a}$ and $\mathbf{b}$, and unknown scalar λ:

$$\mathbf{A}_{11}^{-1}\mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}'_{12}\mathbf{a} = \lambda\mathbf{a}$$

$$\mathbf{A}_{22}^{-1}\mathbf{A}'_{12}\mathbf{A}_{11}^{-1}\mathbf{A}_{12}\mathbf{b} = \lambda\mathbf{b}$$

There are T solutions to these expressions (for $T = \min(Q, (P - Q))$), given by normalized unit-length vectors, $\mathbf{a}_1, \ldots, \mathbf{a}_T$ and $\mathbf{b}_1, \ldots, \mathbf{b}_T$; and a set of common $\lambda_1 \geq \cdots \geq \lambda_T \geq 0$.

The linear combinations of the first and second batteries defined by $\mathbf{a}_k$ and $\mathbf{b}_k$ are the k^{th} *canonical variates* and have squared correlation of λ_k; they are uncorrelated with all other canonical variates (defined either in the first or second batteries). Thus, $\mathbf{a}_1$ and $\mathbf{b}_1$ are the first canonical variates with squared correlation of λ_1; among all linear combinations defined by unit-length vectors for the variables in the two batteries, this squared correlation is the highest it can be. (We note that the coefficient matrices $\mathbf{A}_{11}^{-1}\mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{12}'$ and $\mathbf{A}_{22}^{-1}\mathbf{A}_{12}'\mathbf{A}_{11}^{-1}\mathbf{A}_{12}$ are not symmetric; thus, special symmetrizing and equivalent equation systems are typically used to obtain the solutions to the original set of expressions.)

0.4.4 Algebraic Restrictions on Correlations

A matrix $\mathbf{A}_{P\times P}$ that represents a covariance matrix among a collection of random variables, $X_1, \ldots, X_P$ is p.s.d.; and conversely, any p.s.d. matrix represents the covariance matrix for some collection of random variables. We partition $\mathbf{A}$ to isolate its last row and column as

$$\mathbf{A} = \begin{pmatrix} \mathbf{B}_{(P-1)\times(P-1)} & \mathbf{g}_{(P-1)\times 1} \\ \mathbf{g}' & a_{PP} \end{pmatrix}$$

$\mathbf{B}$ is the $(P-1)\times(P-1)$ covariance matrix among the variables $X_1, \ldots, X_{P-1}$; $\mathbf{g}$ is $(P-1)\times 1$ and contains the cross-covariance between the the first $P-1$ variables and the P^{th}; a_{PP} is the variance for the P^{th} variable.

Based on the observation that determinants of p.s.d. matrices are nonnegative, and a result on expressing determinants for partitioned matrices (that we do not give here), it must be true that

$$\mathbf{g}'\mathbf{B}^{-1}\mathbf{g} \leq a_{PP}$$

or if we think correlations rather than merely covariances (so the main diagonal of $\mathbf{A}$ consists of all ones):

$$\mathbf{g}'\mathbf{B}^{-1}\mathbf{g} \leq 1$$

Given the correlation matrix $\mathbf{B}$, the possible values the correlations in $\mathbf{g}$ could have are in or on the ellipsoid defined in $P-1$ dimensions by $\mathbf{g}'\mathbf{B}^{-1}\mathbf{g} \leq 1$. The important point is that we do not have a "box" in $P-1$ dimensions containing the correlations with sides extending the whole range of ± 1; instead, some restrictions are placed on the observable correlations that gets defined by the size of the correlation in $\mathbf{B}$. For example, when $P = 3$, a correlation between variables X_1 and X_2 of $r_{12} = 0$ gives the "degenerate" ellipse of a circle for constraining the correlation values between X_1 and X_2 and the third variable X_3 (in a two-dimensional r_{13} versus r_{23} coordinate system); for $r_{12} = 1$, the ellipse flattens to a line in this same two-dimensional space.

Another algebraic restriction that can be seen immediately is based on the formula for the partial correlation between two variables, "holding the third constant":

$$\frac{r_{12} - r_{13}r_{23}}{\sqrt{(1 - r_{13}^2)(1 - r_{23}^2)}}$$

Bounding the above by ± 1 (because it is a correlation) and "solving" for r_{12}, gives the algebraic upper and lower bounds of

$$r_{12} \leq r_{13}r_{23} + \sqrt{(1 - r_{13}^2)(1 - r_{23}^2)}$$

$$r_{13}r_{23} - \sqrt{(1 - r_{13}^2)(1 - r_{23}^2)} \leq r_{12}$$

0.4.5 The Biplot

Let $\mathbf{A} = \{a_{ij}\}$ be an $n \times m$ matrix of rank r. We wish to find a second matrix $\mathbf{B} = \{b_{ij}\}$ of the same size, $n \times m$, but of rank t, where $t \leq r$, such that the least squares criterion, $\Sigma_{i,j}(a_{ij} - b_{ij})^2$, is as small as possible overall all matrices of rank t.

The solution is to first find the singular value decomposition of $\mathbf{A}$ as $\mathbf{UDV}'$, where $\mathbf{U}$ is $n \times r$ and has orthonormal columns, $\mathbf{V}$ is $m \times r$ and has orthonormal columns, and $\mathbf{D}$ is $r \times r$, diagonal, with positive values $d_1 \geq d_2 \geq \cdots \geq d_r > 0$ along the main diagonal. Then, $\mathbf{B}$ is defined as $\mathbf{U}^*\mathbf{D}^*\mathbf{V}^{*\prime}$, where we take the first t columns of $\mathbf{U}$ and $\mathbf{V}$ to obtain $\mathbf{U}^*$ and $\mathbf{V}^*$, respectively, and the first t values, $d_1 \geq \cdots \geq d_t$, to form a diagonal matrix $\mathbf{D}^*$.

The approximation of $\mathbf{A}$ by a rank t matrix $\mathbf{B}$, has been one mechanism for representing the row and column objects defining $\mathbf{A}$ in a low-dimensional space of dimension t through what can be generically labeled as a biplot (the prefix "bi" refers to the representation of both the row and column objects together in the same space). Explicitly, the approximation of $\mathbf{A}$ and $\mathbf{B}$ can be written as

$$\mathbf{B} = \mathbf{U}^*\mathbf{D}^*\mathbf{V}^{*\prime} = \mathbf{U}^*\mathbf{D}^{*\alpha}\mathbf{D}^{(1-\alpha)}\mathbf{V}^{*\prime} = \mathbf{PQ}' \ ,$$

where α is some chosen number between 0 and 1, $\mathbf{P} = \mathbf{U}^*\mathbf{D}^{*\alpha}$ and is $n \times t$, $\mathbf{Q} = (\mathbf{D}^{(1-\alpha)}\mathbf{V}^{*\prime})'$ and is $m \times t$.

The entries in $\mathbf{P}$ and $\mathbf{Q}$ define coordinates for the row and column objects in a t-dimensional space that, irrespective of the value of α chosen, have the following characteristic:

If a vector is drawn from the origin through the i^{th} row point and the m column points are projected onto this vector, the collection of such projections is proportional to the i^{th} row of the approximating matrix $\mathbf{B}$. The same is true for projections of row points onto vectors from the origin through each of the column points.

0.4.6 The Procrustes Problem

Procrustes (the subduer), son of Poseidon, kept an inn benefiting from what he claimed to be a wonderful all-fitting bed. He lopped off excessive limbage from tall guests and either flattened short guests by hammering or stretched them by racking. The victim fitted the bed perfectly but, regrettably, died. To exclude the embarrassment of an initially exact-fitting guest, variants of the legend allow Procrustes two, different-sized beds. Ultimately, in a crackdown on robbers and monsters, the young Theseus fitted Procrustes to his own bed. (Gower and Dijksterhuis, 2004)

Suppose we have two matrices, $\mathbf{X}_1$ and $\mathbf{X}_2$, each considered (for convenience) to be of the same size, $n \times p$. If you wish, $\mathbf{X}_1$ and $\mathbf{X}_2$ can be interpreted as two separate p-dimensional coordinate sets for the same set of n objects. Our task is to match these two configurations optimally, with the criterion being least-squares: find a transformation matrix, $\mathbf{T}_{p\times p}$, such that $\parallel \mathbf{X}_1\mathbf{T} - \mathbf{X}_2 \parallel$ is minimized, where $\parallel \cdot \parallel$ denotes the sum-of-squares of the incorporated matrix, i.e., if $\mathbf{A} = \{a_{uv}\}$, then $\parallel \mathbf{A} \parallel = \text{trace}(\mathbf{A}'\mathbf{A}) = \Sigma_{u,v}\, a_{uv}^2$. For conve-

nience, assume both $\mathbf{X}_1$ and $\mathbf{X}_2$ have been normalized so $\| \mathbf{X}_1 \| = \| \mathbf{X}_2 \| = 1$, and the columns of $\mathbf{X}_1$ and $\mathbf{X}_2$ have sums of zero.

Two results are central:

(a) When $\mathbf{T}$ is unrestricted, we have the multivariate multiple regression solution

$$\mathbf{T}^* = (\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{X}_1'\mathbf{X}_2 \ ;$$

(b) When $\mathbf{T}$ is orthogonal, we have the Schönemann solution done for his thesis in the Quantitative Division at Illinois in 1965 (published in *Psychometrika* in 1966):

for the SVD of $\mathbf{X}_2'\mathbf{X}_1 = \mathbf{USV}'$, we let $\mathbf{T}^* = \mathbf{VU}'$.

0.4.7 Matrix Rank Reduction

Lagrange's Theorem (as inappropriately named by C. R. Rao, because it should really be attributed to Guttman) can be stated as follows:

Let $\mathbf{G}$ be a nonnegative-definite (i.e., a symmetric positive semi-definite) matrix of order $n \times n$ and of rank $r > 0$. Let $\mathbf{B}$ be of order $n \times s$ and such that $\mathbf{B}'\mathbf{GB}$ is non-singular. Then the residual matrix

$$\mathbf{G_1} = \mathbf{G} - \mathbf{GB}(\mathbf{B}'\mathbf{GB})^{-1}\mathbf{B}'\mathbf{G} \tag{1}$$

is of rank $r - s$ and is nonnegative definite.

Intuitively, this theorem allows you to "take out" "factors" from a covariance (or correlation) matrix.

There are two somewhat more general results (from Guttman) on matrix rank reduction that prove useful:

Let $\mathbf{S}$ be any matrix of order $n \times N$ and of rank $r > 0$. Let $\mathbf{X}$ and $\mathbf{Y}$ be of orders $s \times n$ and $s \times N$, respectively (where $s \leq r$), and such that $\mathbf{XSY'}$ is nonsingular. Then the residual matrix

$$\mathbf{S}_1 = \mathbf{S} - \mathbf{SY'}(\mathbf{XSY'})^{-1}\mathbf{XS}$$

is exactly of rank $r - s$.

If $\mathbf{S}$ is of order $n \times N$ and of rank r, $\mathbf{F}$ of order $n \times r$ (and of rank r), and $\mathbf{SS'} = \mathbf{FF'}$, then there is a unique matrix $\mathbf{P}$ of order $r \times N$ such that

$$\mathbf{S} = \mathbf{FP} \ .$$

The matrix $\mathbf{P} = (\mathbf{F'F})^{-1}\mathbf{F'S}$ satisfies $\mathbf{PP'} = \mathbf{I}$ (i.e., $\mathbf{P}$ has orthonormal rows).

0.4.8 Torgerson Metric Multidimensional Scaling

Let $\mathbf{A}$ be a symmetric matrix of order $n \times n$. Suppose we want to find a matrix $\mathbf{B}$ of rank 1 (of order $n \times n$) in such a way that the sum of the squared discrepancies between the elements of $\mathbf{A}$ and the corresponding elements of $\mathbf{B}$ (i.e., $\Sigma_{j=1}^{n}\Sigma_{i=1}^{n}(a_{ij} - b_{ij})^2$) is at a minimum. It can be shown that the solution is $\mathbf{B} = \lambda\mathbf{kk'}$ (so all columns in $\mathbf{B}$ are multiples of $\mathbf{k}$), where λ is the largest eigenvalue of $\mathbf{A}$ and $\mathbf{k}$ is the corresponding normalized eigenvector. This theorem can be generalized. Suppose we take the first r largest eigenvalues and the corresponding normalized eigenvectors. The eigenvectors are collected in an $n \times r$ matrix $\mathbf{K} = \{\mathbf{k}_1, \ldots, \mathbf{k}_r\}$ and the eigenvalues in a diagonal matrix $\mathbf{\Lambda}$. Then $\mathbf{K\Lambda K'}$ is an $n \times n$ matrix of rank r and is a least-squares solution for the approximation of $\mathbf{A}$ by a matrix of rank r. It is assumed, here, that the eigenvalues are all positive. If

$\mathbf{A}$ is of rank r by itself and we take the r eigenvectors for which the eigenvalues are different from zero collected in a matrix $\mathbf{K}$ of order $n \times r$, then $\mathbf{A} = \mathbf{K\Lambda K'}$. Note that $\mathbf{A}$ could also be represented by $\mathbf{A} = \mathbf{LL'}$, where $\mathbf{L} = \mathbf{K\Lambda^{1/2}}$ (we factor the matrix), or as a sum of r $n \times n$ matrices — $\mathbf{A} = \lambda_1 \mathbf{k_1 k_1'} + \cdots + \lambda_r \mathbf{k_r k_r'}$.

Metric Multidimensional Scaling – Torgerson's Model (Gower's Principal Coordinate Analysis)

Suppose I have a set of n points that can be perfectly represented spatially in r dimensional space. The i^{th} point has coordinates $(x_{i1}, x_{i2}, \ldots, x_{ir})$. If $d_{ij} = \sqrt{\Sigma_{k=1}^{r}(x_{ik} - x_{jk})^2}$ represents the Euclidean distance between points i and j, then

$$d_{ij}^* = \sum_{k=1}^{r} x_{ik} x_{jk}, \text{where}$$

$$d_{ij}^* = -\frac{1}{2}(d_{ij}^2 - A_i - B_j + C); \tag{2}$$

$$A_i = (1/n) \sum_{j=1}^{n} d_{ij}^2;$$

$$B_j = (1/n) \sum_{i=1}^{n} d_{ij}^2;$$

$$C = (1/n^2) \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij}^2.$$

Note that $\{d_{ij}^*\}_{n \times n} = \mathbf{XX'}$, where $\mathbf{X}$ is of order $n \times r$ and the entry in the i^{th} row and k^{th} column is x_{ik}.

So, the Question: If I give you $\mathbf{D} = \{d_{ij}\}_{n \times n}$, find me *a* set of coordinates to do it. The Solution: Find $\mathbf{D}^* = \{d_{ij}^*\}$, and take its Spectral Decomposition. This is *exact* here.

To use this result to obtain a spatial representation for a set of n objects given *any* "distance-like" measure, p_{ij}, between objects i and j, we proceed as follows:

(a) Assume (i.e., pretend) the Euclidean model holds for p_{ij}.

(b) Define p^*_{ij} from p_{ij} using (2).

(c) Obtain a spatial representation for p^*_{ij} using a suitable value for r, the number of dimensions (at most, r can be no larger than the number of positive eigenvalues for $\{p^*_{ij}\}_{n\times n}$):

$$\{p^*_{ij}\} \approx \mathbf{X}\mathbf{X}'$$

(d) Plot the n points in r dimensional space.

0.4.9 A Guttman Multidimensional Scaling Result

If $\mathbf{B}$ is a symmetric matrix of order n, having all its elements non-negative, the following quadratic form defined by the matrix $\mathbf{A}$ must be positive semi-definite:

$$\sum_{i,j} b_{ij}(x_i - x_j)^2 = \sum_{i,j} x_i a_{ij} x_j,$$

where

$$a_{ij} = \begin{cases} \Sigma^n_{k=1;k\neq i}\, b_{ik} & (i = j) \\ -b_{ij} & (i \neq j) \end{cases}$$

If all elements of $\mathbf{B}$ are positive, then $\mathbf{A}$ is of rank $n-1$, and has one smallest eigenvalue equal to zero with an associated eigenvector having all constant elements. Because all (other) eigenvectors must be orthogonal to the constant eigenvector, the entries in these other eigenvectors must sum to zero.

This Guttman result can be used for a method of multidimensional scaling (mds), and is one that seems to get reinvented periodically in the literature. Generally, this method has been used to provide rational starting points in iteratively-defined nonmetric mds. More recently, the Guttman strategy (although not attributed to him as such) has been applied to graphs and the corresponding 0/1 adjacency matrix (treated as a similarity measure). In this case, we have what are called Laplacian eigenmaps, where the graphs are imbedded into a space by using the coordinates from the *smallest* nonzero eigenvectors.

0.4.10 A Few General MATLAB Routines to Know About

For Eigenvector/Eigenvalue Decompositions:

$[\mathbf{V}, \mathbf{D}] = \text{eig}(\mathbf{A})$, where $\mathbf{A} = \mathbf{VDV}'$, for $\mathbf{A}$ square; $\mathbf{V}$ is orthogonal and contains eigenvectors (as columns); $\mathbf{D}$ is diagonal and contains the eigenvalues.

For Singular Value Decompositions:

$[\mathbf{U}, \mathbf{S}, \mathbf{V}] = \text{svd}(\mathbf{B})$, where $\mathbf{B} = \mathbf{USV}'$; the columns of $\mathbf{U}$ and the rows of $\mathbf{V}'$ are orthonormal; $\mathbf{S}$ is diagonal and contains the nonnegative singular values (ordered from *largest to smallest*).

The help comments for the Procrustes routine in the Statistics Toolbox are given verbatim below. Note the very general transformation provided in the form of a MATLAB Structure that involves optimal rotation, translation, and scaling.

```
help procrustes
procrustes Procrustes Analysis
   D = procrustes(X, Y) determines a linear transformation (translation,
```

```
reflection, orthogonal rotation, and scaling) of the points in the
matrix Y to best conform them to the points in the matrix X.  The
"goodness-of-fit" criterion is the sum of squared errors.  procrustes
returns the minimized value of this dissimilarity measure in D.  D is
standardized by a measure of the scale of X, given by

   sum(sum((X - repmat(mean(X,1), size(X,1), 1)).^2, 1))

i.e., the sum of squared elements of a centered version of X.  However,
if X comprises repetitions of the same point, the sum of squared errors
is not standardized.

X and Y are assumed to have the same number of points (rows), and
procrustes matches the i'th point in Y to the i'th point in X.  Points
in Y can have smaller dimension (number of columns) than those in X.
In this case, procrustes adds columns of zeros to Y as necessary.

[D, Z] = procrustes(X, Y) also returns the transformed Y values.

[D, Z, TRANSFORM] = procrustes(X, Y) also returns the transformation
that maps Y to Z.  TRANSFORM is a structure with fields:
   c:  the translation component
   T:  the orthogonal rotation and reflection component
   b:  the scale component
That is, Z = TRANSFORM.b * Y * TRANSFORM.T + TRANSFORM.c.

[...] = procrustes(..., 'Scaling',false) computes a procrustes solution
that does not include a scale component, that is, TRANSFORM.b == 1.
procrustes(..., 'Scaling',true) computes a procrustes solution that
does include a scale component, which is the default.

[...] = procrustes(..., 'Reflection',false) computes a procrustes solution
that does not include a reflection component, that is, DET(TRANSFORM.T) is
1.  procrustes(..., 'Reflection','best') computes the best fit procrustes
solution, which may or may not include a reflection component, 'best' is
the default.  procrustes(..., 'Reflection',true) forces the solution to
include a reflection component, that is, DET(TRANSFORM.T) is -1.

Examples:

   % Create some random points in two dimensions
   n = 10;
   X = normrnd(0, 1, [n 2]);

   % Those same points, rotated, scaled, translated, plus some noise
```

```
S = [0.5 -sqrt(3)/2; sqrt(3)/2 0.5]; % rotate 60 degrees
Y = normrnd(0.5*X*S + 2, 0.05, n, 2);

% Conform Y to X, plot original X and Y, and transformed Y
[d, Z, tr] = procrustes(X,Y);
plot(X(:,1),X(:,2),'rx', Y(:,1),Y(:,2),'b.', Z(:,1),Z(:,2),'bx');
```

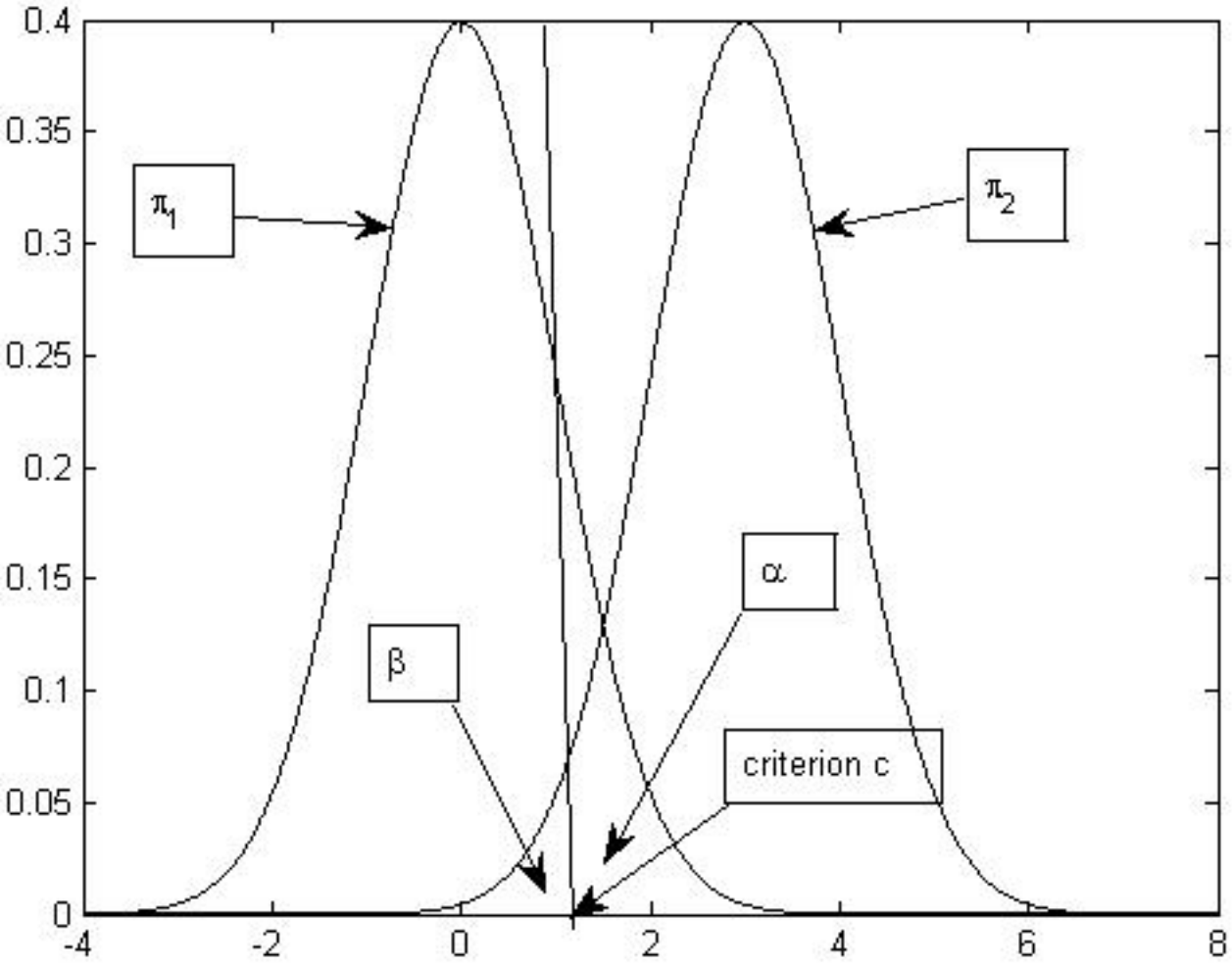
Notes on Discrimination and Classification

The term "discrimination" (in a nonpejorative statistical sense) refers to the task of discrimination among groups through linear combinations of variables that maximize some criterion, usually F-ratios. The term "classification" refers to the task of allocating observations to existing groups, typically to minimize the cost and/or probability of misclassification. These two topics are intertwined, but it is most convenient to start with the topic of classification.

In the picture to follow, we have two populations, called π_1 and π_2; π_1 is characterized by a normal distribution with mean μ_1, and variance σ_X^2 (the density is denoted by $f_1(x)$); π_2 is characterized by a normal distribution with mean μ_2, and (common) variance σ_X^2 (the density is denoted by $f_2(x)$). I have an observation, say x_0, and wish to decide where it should go, either to π_1 or π_2. Assuming implicitly that $\mu_1 \leq \mu_2$, we choose a criterion point, c, and allocate to π_1 if $x_0 \leq c$, and to π_2 if $> c$. The probabilities of misclassification can be given in the following chart (and in the figure):

		True State	
		π_1	π_2
Decision	π_1	$1-\alpha$	β
	π_2	α	$1-\beta$

If I want to choose c so that $\alpha + \beta$ is smallest, I would select the point at which the densities are equal. A more complicated way of saying this decision rule is to allocate to π_1 if $f_1(x_0)/f_2(x_0) \geq 1$; if < 1, then allocate to π_2. Suppose now that the prior probabilities


0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-4
-2
0
2
4
6
8
π1
π2
α
β
criterion c

of being drawn from π_1 and π_2 are p_1 and p_2, where $p_1 + p_2 = 1$. I wish to choose c so the Total Probability of Misclassification (TPM) is minimized, i.e., $p_1\alpha + p_2\beta$. The rule would be to allocate to π_1 if $f_1(x_0)/f_2(x_0) \geq p_2/p_1$; if $< p_2/p_1$, then allocate to π_2. Finally, if we include costs of misclassification, $c(1|2)$ (for assigning to π_1 when actually coming from π_2), and $c(2|1)$ (for assigning to π_2 when actually coming from π_1), we can choose c to minimize the Expected Cost of Misclassification (ECM), $c(2|1)p_1\alpha + c(1|2)p_1\beta$, with the associated rule of allocating to π_1 if $f_1(x_0)/f_2(x_0) \geq (c(1|2)/c(2|1))(p_2/p_1)$; if $< (c(1|2)/c(2|1))(p_2/p_1)$, then allocate to π_2.

Using logs, the last rule can be restated: allocate to π_1 if $\log(f_1(x_0)/f_2(x_0)) \geq \log((c(1|2)/c(2|1))(p_2/p_1))$. The left-hand-side is equal to $(\mu_1 - \mu_2)(\sigma_X^2)^{-1}x_0 - (1/2)(\mu_1 - \mu_2)(\sigma_X^2)^{-1}(\mu_1 + \mu_2)$, so the rule can be restated further: allocate to π_1 if

$$x_0 \leq \{(1/2)(\mu_1 - \mu_2)(\sigma_X^2)^{-1}(\mu_1 + \mu_2)$$

$$- \log((c(1|2)/c(2|1))(p_2/p_1))\}\{\frac{\sigma_X^2}{-(\mu_1 - \mu_2)}\}$$

or

$$x_0 \leq \{(1/2)(\mu_1+\mu_2) - \log((c(1|2)/c(2|1))(p_2/p_1))\}\{\frac{\sigma_X^2}{(\mu_2 - \mu_1)}\} = c\,.$$

If the costs of misclassification are equal (i.e., $c(1|2) = c(2|1)$), then the allocation rule is based on classification functions: allocate to π_1 if

$$[\frac{\mu_1}{\sigma_X^2}x_0 - (1/2)\frac{\mu_1^2}{\sigma_X^2} + \log(p_1)] - [\frac{\mu_2}{\sigma_X^2}x_0 - (1/2)\frac{\mu_2^2}{\sigma_X^2} + \log(p_2)] \geq 0\ .$$

Moving toward the multivariate framework, suppose population π_1 is characterized by a $p \times 1$ vector of random variables, $\mathbf{X} \sim \text{MVN}(\boldsymbol{\mu_1}, \boldsymbol{\Sigma})$; population π_2 is characterized by a $p \times 1$ vector of random variables, $\mathbf{X} \sim \text{MVN}(\boldsymbol{\mu_2}, \boldsymbol{\Sigma})$. We have a similar allocation rule as in the univariate case: allocate to π_1 if $\mathbf{a}\mathbf{x}_0 - \mathbf{a}[(\boldsymbol{\mu_1} + \boldsymbol{\mu_2})/2] \geq (c(1|2)/c(2|1))(p_2/p_1)$, where

$$\mathbf{a} = (\boldsymbol{\mu_1} - \boldsymbol{\mu_2})'\boldsymbol{\Sigma}^{-1} \ .$$

Or, if the misclassification costs are equal, allocate to π_1 if $\mathbf{a}\mathbf{x}_0 - \mathbf{a}[(\boldsymbol{\mu_1} + \boldsymbol{\mu_2})/2] \geq [\log(p_2) - \log(p_1)]$. In effect, we define regions of classification, say R_1 and R_2; if an observation falls into region R_i, it is allocated to group i, for $i = 1, 2$ There are a number of ways of restating this last rule (assuming equal misclassification costs, this is choosing to minimize the Total Probability of Misclassification (TPM)):

A) Evaluate the classification functions for both groups and assign according to which is higher: allocate to π_1 if

$$[\boldsymbol{\mu}_1'\boldsymbol{\Sigma}^{-1}\mathbf{x}_0 - (1/2)\boldsymbol{\mu}_1\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1) + \log(p_1)] -$$
$$[\boldsymbol{\mu}_2'\boldsymbol{\Sigma}^{-1}\mathbf{x}_0 - (1/2)\boldsymbol{\mu}_2\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2) + \log(p_2)] \geq 0 \ .$$

B) Define the posterior probability of being in group i, for $i = 1, 2$, $P(\pi_i|\mathbf{x}_0)$ as $(f_i p_i)/(f_1 p_1 + f_2 p_2)$. We allocate to the group with the largest posterior probability.

C) We can restate our allocation rule according to Mahalanobis distances: define the squared Mahalanobis distance of $\mathbf{x}_0$ to $\mu_i, i = 1, 2$, as

$$(\mathbf{x}_0 - \boldsymbol{\mu}_i)'\boldsymbol{\Sigma}^{-1}(\mathbf{x}_0 - \boldsymbol{\mu}_i) \ .$$

Allocate to π_i for the largest quantity of the form:

$$-(1/2)[(\mathbf{x}_0 - \boldsymbol{\mu}_i)'\boldsymbol{\Sigma}^{-1}(\mathbf{x}_0 - \boldsymbol{\mu}_i)] + \log(p_i) \ .$$

When the covariance matrices are not equal in the two populations (i.e., $\boldsymbol{\Sigma}_1 \neq \boldsymbol{\Sigma}_2$), the allocation rules get a little more complicated. The classification rules are now called "quadratic", and may produce regions of allocation that may not be contiguous. This is a little strange, but it can be done, and we can still split the allocation rule into two classification functions (assuming, as usual, equal costs of misclassification):

Assign to π_1 if

$$-(1/2)\mathbf{x}_0'(\boldsymbol{\Sigma}_1^{-1} - \boldsymbol{\Sigma}_2^{-1})\mathbf{x}_0 + (\boldsymbol{\mu_1}'\boldsymbol{\Sigma}_1^{-1} - \boldsymbol{\mu_2}'\boldsymbol{\Sigma}_1^{-1})\mathbf{x}_0 \ - \ k \geq$$
$$\log((c(1|2)/c(2|1))(p_2/p_1)) \ ,$$

where

$$k = (1/2)\log(\frac{|\boldsymbol{\Sigma}_1|}{|\boldsymbol{\Sigma}_2|}) + (1/2)(\boldsymbol{\mu_1}'\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu_1} - \boldsymbol{\mu_2}'\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu_2}) \ .$$

Moving to the sample, we could just use estimated quantities and hope our rule does well — we use $\mathbf{S}_{pooled}$, assuming equal covariance matrices in the two populations, and sample means, $\hat{\boldsymbol{\mu}}_\mathbf{1}$ and $\hat{\boldsymbol{\mu}}_\mathbf{2}$. In fact, we can come up with the misclassification table based on the given sample and how they allocate the given n observations to the two groups:

		Group	
		π_1	π_2
Decision	π_1	a	b
	π_2	c	d
		n_1	n_2

The apparent error rate (APR) is $(b + c)/n$, which is overly optimistic because it is optimized with respect to *this* sample. To cross-validate, we could use a "hold out one-at-a-time" strategy (i.e., a sample reuse procedure commonly referred to as the "jackknife"):

		Group	
		π_1	π_2
Decision	π_1	a^*	b^*
	π_2	c^*	d^*
		n_1	n_2

To estimate the actual error rate (AER), we would use $(b^* + c^*)/n$.

Suppose we have g groups; p_i is the a priori probability of group i, $1 \leq i \leq g$; $c(k|i)$ is the cost of classifying an i as a k. The decision rule that minimizes the expected cost of misclassification (ECM) is: allocate $\mathbf{x}_0$ to population π_k, $1 \leq k \leq g$, if

$$\sum_{i=1;i\neq k}^{g} p_i f_i(\mathbf{x}_0) c(k|i)$$

is smallest.

There are, again, alternative ways of stating this allocation rule;

we will assume for convenience that the costs of misclassification are equal:

Allocate to group k if the posterior probability,

$$P(\pi_k|\mathbf{x}_0) = \frac{p_k f_k(\mathbf{x}_0)}{\Sigma_{i=1}^{g} p_i f_i(\mathbf{x}_0)} \ ,$$

is largest.

If in population k, $\mathbf{X} \sim \text{MVN}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, we allocate to group k if $\log(p_k f_k(\mathbf{x}_0)) =$

$$-(1/2)\log(|\boldsymbol{\Sigma}_k|) - (1/2)(\mathbf{x}_0 - \boldsymbol{\mu}_k)'\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_0 - \boldsymbol{\mu}_k) + \log(p_i) + \text{constant} \ ,$$

is largest.

If all the $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}$ for all k, then we allocate to π_k if

$$\boldsymbol{\mu}_k'\boldsymbol{\Sigma}_k^{-1}\mathbf{x}_0 - (1/2)\boldsymbol{\mu}_k'\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k + \log(p_k) \ ,$$

is largest.

It is interesting that we can do this in a pairwise way as well: allocate to π_k if

$$(\boldsymbol{\mu}_k - \boldsymbol{\mu}_i)'\boldsymbol{\Sigma}_k^{-1}\mathbf{x}_0 - (1/2)(\boldsymbol{\mu}_k - \boldsymbol{\mu}_i)'\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{\mu}_k + \boldsymbol{\mu}_i) \geq \log(p_i/p_k) \ ,$$

for all $i = 1, \ldots, g$.

0.4.11 Discriminant Analysis

Suppose we have a one-way analysis-of-variance (ANOVA) layout with J groups (n_j subjects in group j, $1 \leq j \leq J$), and p measurements on each subject. If x_{ijk} denotes person i, in group j, and the

observation of variable k ($1 \leq i \leq n_j$; $1 \leq j \leq J$; $1 \leq k \leq p$), then define the Between-Sum-of-Squares matrix

$$\mathbf{B}_{p\times p} = \{\sum_{j=1}^{J} n_j(\bar{x}_{\cdot jk} - \bar{x}_{\cdot\cdot k})(\bar{x}_{\cdot jk'} - \bar{x}_{\cdot\cdot k'})\}_{p\times p}$$

and the Within-Sum-of-Squares matrix

$$\mathbf{W}_{p\times p} = \{\sum_{j=1}^{J}\sum_{i=1}^{n_j}(x_{ijk} - \bar{x}_{\cdot jk})(x_{ijk'} - \bar{x}_{\cdot jk'})\}_{p\times p}$$

For the matrix product $\mathbf{W}^{-1}\mathbf{B}$, let $\lambda_1, \ldots, \lambda_T \geq 0$ be the eigenvectors ($T = \min(p, J-1)$, and $\mathbf{p}_1, \ldots, \mathbf{p}_T$ the corresponding normalized eigenvectors. Then, the linear combination

$$\mathbf{p}_k' \begin{pmatrix} X_1 \\ \vdots \\ X_p \end{pmatrix}$$

is called the k^{th} *discriminant function*. It has the valuable property of maximizing the univariate F-ratio subject to being uncorrelated with the earlier linear combinations.

There are a number of points to make about (Fisher's) Linear Discriminant Functions:

A) Typically, we define a sample pooled variance-covariance matrix, $\mathbf{S}_{pooled}$, as $(\frac{1}{n-J})\mathbf{W}$. And generally, the eigenvalues are scaled so that $\mathbf{p}_k'\mathbf{S}_{pooled}\mathbf{p}_k = 1$.

B) When $J = 2$, the eigenvector, $\mathbf{p}_1'$, is equal to $(\hat{\boldsymbol{\mu}}_{\mathbf{1}} - \hat{\boldsymbol{\mu}}_{\mathbf{2}})'\mathbf{S}_{pooled}$. This set of weights maximized the square of the t ratio in a two-group separation problem (i.e., discriminating between the two groups). We

also maximize the square of the effect size for this linear combination; the maximum for such an effect size is

$$(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)' \mathbf{S}_{pooled}^{-1} (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)' \ ,$$

where $\bar{\mathbf{x}}_1$ and $\bar{\mathbf{x}}_2$ are the sample centroids in groups 1 and 2 for the p variables. Finally, if we define $Y = 1$ if an observation falls into group 1, and $= 0$ if in group 2, the set of weights in $\mathbf{p}_1'$ is proportional to the regression coefficients in predicting Y from $X_1, \ldots, X_p$.

C) The classification rule based on Mahalanobis distance (and assuming equal prior probabilities and equal misclassification values), could be restated equivalently using plain Euclidean distances in discriminant function space.

Notes on Principal Component Analysis

A preliminary introduction to principal components was given in our brief discussion of the spectral decomposition (i.e., the eigenvector/eigenvalue decomposition) of a matrix and what it might be used for. We will now be a bit more systematic, and begin by making three introductory comments:

(a) Principal component analysis (PCA) deals with only one set of variables without the need for categorizing the variables as being independent or dependent. There is asymmetry in the discussion of the general linear model; in PCA, however, we analyze the relationships among the variables in one set and *not* between two.

(b) As always, everything can be done computationally without the Multivariate Normal (MVN) assumption; we are just getting descriptive statistics. When significance tests and the like are desired, the MVN assumption becomes indispensable. Also, MVN gives some very nice interpretations for what the principal components are in terms of our constant density ellipsoids.

(c) Finally, it is probably best if you are doing a PCA, not to refer to these as "factors". A lot of blood and ill-will has been spilt and spread over the distinction between component analysis (which involves linear combinations of *observable* variables), and the estimation of a factor model (which involves the use of underlying latent variables or factors, and the estimation of the factor structure). We will get sloppy ourselves later, but some people really get exercised about these things.

We will begin working with the population (but everything translates more-or-less directly for a sample):

Suppose $[X_1, X_2, \ldots, X_p] = \mathbf{X}'$ is a set of p random variables, with mean vector $\boldsymbol{\mu}$ and variance-covariance matrix $\boldsymbol{\Sigma}$. I want to define p linear combinations of $\mathbf{X}'$ that represent the information in $\mathbf{X}'$ more parsimoniously. Specifically, find $\mathbf{a}_1, \ldots, \mathbf{a}_p$ such that $\mathbf{a}_1'\mathbf{X}, \ldots, \mathbf{a}_p'\mathbf{X}$ gives the same information as $\mathbf{X}'$, but the new random variables, $\mathbf{a}_1'\mathbf{X}, \ldots, \mathbf{a}_p'\mathbf{X}$, are "nicer".

Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$ be the p roots (eigenvalues) of the matrix $\boldsymbol{\Sigma}$, and let $\mathbf{a}_1, \ldots, \mathbf{a}_p$ be the corresponding eigenvectors. If some roots are not distinct, I can still pick corresponding eigenvectors to be orthogonal. Choose an eigenvector $\mathbf{a}_i$ so $\mathbf{a}_i'\mathbf{a}_i = 1$, i.e., a normalized eigenvector. Then, $\mathbf{a}_i'\mathbf{X}$ is the i^{th} principal component of the random variables in $\mathbf{X}'$.

Properties:

1) $\text{Var}(\mathbf{a}_i'\mathbf{X}) = \mathbf{a}_i'\boldsymbol{\Sigma}\mathbf{a}_i = \lambda_i$

We know $\boldsymbol{\Sigma}\mathbf{a}_i = \lambda_i\mathbf{a}_i$, because $\mathbf{a}_i$ is the eigenvector for λ_i; thus, $\mathbf{a}_i'\boldsymbol{\Sigma}\mathbf{a}_i = \mathbf{a}_i'\lambda_i\mathbf{a}_i = \lambda_i$. In words, the variance of the i^{th} principal component is λ_i, the root.

Also, for all vectors $\mathbf{b}_i$ such that $\mathbf{b}_i$ is orthogonal to $\mathbf{a}_1, \ldots, \mathbf{a}_{i-1}$, and $\mathbf{b}_i'\mathbf{b}_i = 1$, $\text{Var}(\mathbf{b}_i'\mathbf{X})$ is the greatest it can be (i.e., λ_i) when $\mathbf{b}_i = \mathbf{a}_i$.

2) $\mathbf{a}_i$ and $\mathbf{a}_j$ are orthogonal, i.e., $\mathbf{a}_i'\mathbf{a}_j = 0$

3) $\text{Cov}(\mathbf{a}_i'\mathbf{X}, \mathbf{a}_j'\mathbf{X}) = \mathbf{a}_i'\boldsymbol{\Sigma}\mathbf{a}_j = \mathbf{a}_i'\lambda_j\mathbf{a}_j = \lambda_j\mathbf{a}_i'\mathbf{a}_j = 0$

4) $\text{Tr}(\mathbf{\Sigma}) = \lambda_1 + \cdots + \lambda_p$ = sum of variances for all p principal components, and for $X_1, \ldots, X_p$. The importance of the i^{th} principal component is

$$\lambda_i / \text{Tr}(\mathbf{\Sigma}) \ ,$$

which is equal to the variance of the i^{th} principal component divided by the total variance in the system of p random variables, $\mathbf{X}_1, \ldots, \mathbf{X}_p$; it is the proportion of the total variance explained by the i^{th} component.

If the first few principal components account for most of the variation, then we might interpret these components as "factors" underlying the whole set $\mathbf{X}_1, \ldots, \mathbf{X}_p$. This is the basis of *principal factor analysis*.

The question of how many components (or factors, or clusters, or dimensions) usually has no definitive answer. Some attempt has been made to do what are called Scree plots, and graphically see how many components to retain. A plot is constructed of the value of the eigenvalue on the y-axis and the number of the eigenvalue (e.g., 1, 2, 3, and so on) on the x-axis, and you look for an "elbow" to see where to stop. Scree or talus is the pile of rock debris (detritus) at the foot of a cliff, i.e., the sloping mass of debris at the bottom of the cliff. I, unfortunately, can never see an "elbow"!

If we let a population correlation matrix corresponding to $\mathbf{\Sigma}$ be denoted as $\mathbf{P}$, then $\text{Tr}(\mathbf{P}) = p$, and we might use only those principal components that have variance of $\lambda_i \geq 1$ — otherwise, the component would "explain" less variance than would a single variable.

A major rub — if I do principal components on the correlation

matrix, $\mathbf{P}$, *and* on the original variance-covariance matrix, $\mathbf{\Sigma}$, the structures obtained are generally different. This is one reason the "true believers" might prefer a factor analysis model over a PCA because the former holds out some hope for an invariance (to scaling). Generally, it seems more reasonable to always use the population correlation matrix, $\mathbf{P}$; the units of the original variables become irrelevant, and it is much easier to interpret the principal components through their coefficients.

The j^{th} principal component is $\mathbf{a}_j'\mathbf{X}$:

$\text{Cov}(\mathbf{a}_j'\mathbf{X}, X_i) = \text{Cov}(\mathbf{a}_j'\mathbf{X}, \mathbf{b}'\mathbf{X})$, where $\mathbf{b}' = [0 \cdots 0\ 1\ 0 \cdots 0]$, with the 1 in the i^{th} position, $= \mathbf{a}_j'\mathbf{\Sigma}\mathbf{b} = \mathbf{b}'\mathbf{\Sigma}\mathbf{a}_j = \mathbf{b}'\lambda_j\mathbf{a}_j = \lambda_j$ times the i^{th} component of $\mathbf{a}_j = \lambda_j a_{ij}$. Thus, $\text{Cor}(\mathbf{a}_j'\mathbf{X}, X_i) =$

$$\frac{\lambda_j a_{ij}}{\sqrt{\lambda_j}\sigma_i} = \frac{\sqrt{\lambda_j}a_{ij}}{\sigma_i} \ ,$$

where σ_i is the standard deviation of X_i. This correlation is called the *loading* of X_i on the j^{th} component. Generally, these correlations can be used to see the contribution of each variable to each of the principal components.

If the population covariance matrix, $\mathbf{\Sigma}$, is replaced by the sample covariance matrix, $\mathbf{S}$, we obtain sample principal components; if the population correlation matrix, $\mathbf{P}$, is replaced by the sample correlation matrix, $\mathbf{R}$, we again obtain sample principal components. These structures are generally different.

The covariance matrix $\mathbf{S}$ (or $\mathbf{\Sigma}$) can be represented by

$$\mathbf{S} = [\mathbf{a}_1, \ldots, \mathbf{a}_p] \begin{bmatrix} \sqrt{\lambda_1} & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & \sqrt{\lambda_p} \end{bmatrix} \begin{bmatrix} \sqrt{\lambda_1} & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & \sqrt{\lambda_p} \end{bmatrix} \begin{bmatrix} \mathbf{a}_1' \\ \vdots \\ \mathbf{a}_p' \end{bmatrix} \equiv \mathbf{L}\mathbf{L}'$$

or as the sum of p, $p \times p$ matrices,

$$\mathbf{S} = \lambda_1 \mathbf{a}_1 \mathbf{a}_1' + \cdots + \lambda_p \mathbf{a}_p \mathbf{a}_p' \; .$$

Given the ordering of the eigenvalues as $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$, the least-squares approximation to $\mathbf{S}$ of rank r is $\lambda_1 \mathbf{a}_1 \mathbf{a}_1' + \cdots + \lambda_r \mathbf{a}_r \mathbf{a}_r'$, and the residual matrix, $\mathbf{S} - \lambda_1 \mathbf{a}_1 \mathbf{a}_1' - \cdots - \lambda_r \mathbf{a}_r \mathbf{a}_r'$, is $\lambda_{r+1} \mathbf{a}_{r+1} \mathbf{a}_{r+1}' + \cdots + \lambda_p \mathbf{a}_p \mathbf{a}_p'$.

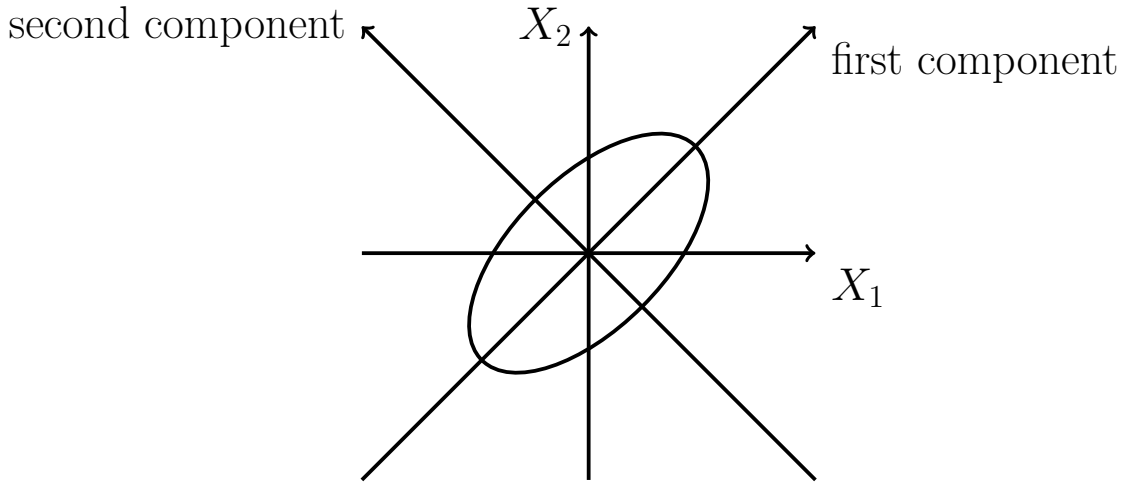
Note that for an arbitrary matrix, $\mathbf{B}_{p \times q}$, the $\mathrm{Tr}(\mathbf{B}\mathbf{B}') =$ sum of squares of the entries in $\mathbf{B}$. Also, for two matrices, $\mathbf{B}$ and $\mathbf{C}$, if both of the products $\mathbf{B}\mathbf{C}$ and $\mathbf{C}\mathbf{B}$ can be taken, then $\mathrm{Tr}(\mathbf{B}\mathbf{C})$ is equal to $\mathrm{Tr}(\mathbf{C}\mathbf{B})$. Using these two results, the least-squares criterion value can be given as

$$\mathrm{Tr}([\lambda_{r+1} \mathbf{a}_{r+1} \mathbf{a}_{r+1}' + \cdots + \lambda_p \mathbf{a}_p \mathbf{a}_p'][\lambda_{r+1} \mathbf{a}_{r+1} \mathbf{a}_{r+1}' + \cdots + \lambda_p \mathbf{a}_p \mathbf{a}_p']') = \sum_{k \geq r+1} \lambda_k^2 \; .$$

This measure is one of how bad the rank r approximation might be (i.e., the proportion of unexplained sum-of-squares when put over $\sum_{k=1}^{p} \lambda_k^2$).

For a geometric interpretation of principal components, suppose we have two variables, X_1 and X_2, that are centered at their respective means (i.e., the means of the scores on X_1 and X_2 are zero). In

the diagram below, the ellipse represents the scatter diagram of the sample points. The first principal component is a line through the widest part; the second component is the line at right angles to the first principal component. In other words, the first principal component goes through the fattest part of the "football," and the second principal component through the next fattest part of the "football" and orthogonal to the first; and so on. Or, we take our original frame of reference and do a rigid transformation around the origin to get a new set of axes; the origin is given by the sample means (of zero) on the two X_1 and X_2 variables. (To make these same geometric points, we could have used a constant density contour for a bivariate normal pair of random variables, X_1 and X_2, with zero mean vector.)



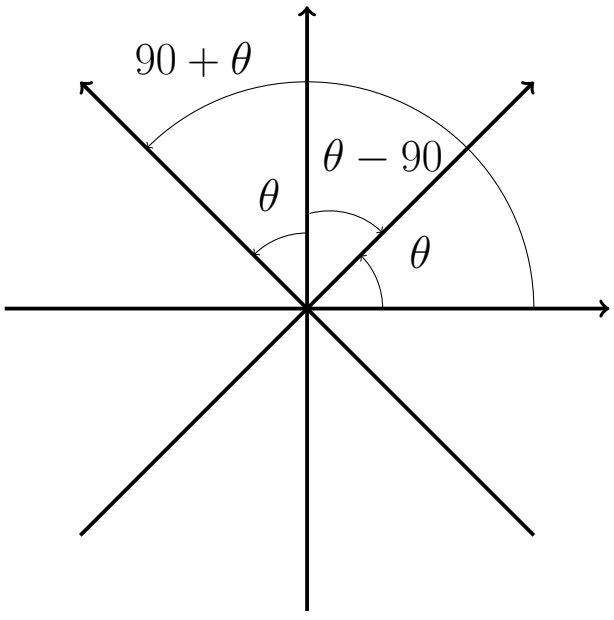
As an example of how to find the placement of the components in the picture given above, suppose we have the two variables, X_1 and X_2, with variance-covariance matrix

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{pmatrix} .$$

Let a_{11} and a_{21} denote the weights from the first eigenvector of $\mathbf{\Sigma}$; a_{12} and a_{22} are the weights from the second eigenvector. If these are placed in a 2×2 orthogonal (or rotation) matrix $\mathbf{T}$, with the first column containing the first eigenvector weights and the second column the second eigenvector weights, we can obtain the direction cosines of the new axes system from the following:

$$\mathbf{T} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = \begin{pmatrix} \cos(\theta) & \cos(90+\theta) \\ \cos(\theta - 90) & \cos(\theta) \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}.$$

These are the cosines of the angles with the positive (horizontal and vertical) axes. If we wish to change the orientation of a transformed axis (i.e., to make the arrow go in the other direction), we merely use a multiplication of the relevant eigenvector values by -1 (i.e., we choose the other normalized eigenvector for that same eigenvalue, which still has unit length).



If we denote the data matrix in this simple two variable problem as $\mathbf{X}_{n \times 2}$, where n is the number of subjects and the two columns

represent the values on variables X_1 and X_2 (i.e., the coordinates of each subject on the original axes), the $n \times 2$ matrix of coordinates of the subjects on the transformed axes, say $\mathbf{X}_{trans}$ can be given as $\mathbf{XT}$.

For another interpretation of principal components, the first component could be obtained by minimizing the sum of squared perpendicular residuals from a line (and in analogy to simple regression where the sum of squared vertical residuals from a line is minimized). This notion generalizes to more than than one principal component and in analogy to the way that multiple regression generalizes simple regression — vertical residuals to hyperplanes are used in multiple regression, and perpendicular residuals to hyperplanes are used in PCA.

There are a number of specially patterned matrices that have interesting eigenvector/eigenvalue decompositions. For example, for the $p \times p$ diagonal variance-covariance matrix

$$\Sigma_{p\times p} = \begin{pmatrix} \sigma_1^2 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & \sigma_p^2 \end{pmatrix},$$

the roots are $\sigma_1^2, \ldots, \sigma_p^2$, and the eigenvector corresponding to σ_i^2 is $[0\ 0 \ldots 1 \ldots 0]'$ where the single 1 is in the i^{th} position. If we have a correlation matrix, the root of 1 has multiplicity p, and the eigenvectors could also be chosen as these same vectors having all zeros except for a single 1 in the i^{th} position, $1 \leq i \leq p$.

If the $p \times p$ variance-covariance matrix demonstrates compound

symmetry,

$$\Sigma_{p\times p} = \sigma^2 \begin{pmatrix} 1 & \cdots & \rho \\ \vdots & & \vdots \\ \rho & \cdots & 1 \end{pmatrix},$$

or is an equicorrelation matrix,

$$\mathbf{P} = \begin{pmatrix} 1 & \cdots & \rho \\ \vdots & & \vdots \\ \rho & \cdots & 1 \end{pmatrix},$$

then the $p-1$ smallest roots are all equal. For example, for the equicorrelation matrix, $\lambda_1 = 1 + (p-1)\rho$, and the remaining $p-1$ roots are all equal to $1-\rho$. The $p \times 1$ eigenvector for λ_1 is $[\frac{1}{\sqrt{p}}, \ldots, \frac{1}{\sqrt{p}}]'$, and defines an average. Generally, for any variance-covariance matrix with all entries greater than zero (or just non-negative), the entries in the first eigenvector must all be greater than zero (or non-negative). This is known as the Perron-Frobenius theorem.

Although we will not give these tests explicitly here (they can be found in Johnson and Wichern's (2007) multivariate text), they are inference methods to test the null hypothesis of an equicorrelation matrix (i.e., the last $p-1$ eigenvalues are equal); that the variance-covariance matrix is diagonal or the correlation matrix is the identity (i.e., all eigenvalues are equal); or a sphericity test of independence that all eigenvalues are equal and $\mathbf{\Sigma}$ is σ^2 times the identity matrix.

0.5 Analytic Rotation Methods

Suppose we have a $p \times m$ matrix, $\mathbf{A}$, containing the correlations (loadings) between our p variables and the first m principal components. We are seeking an orthogonal $m \times m$ matrix $\mathbf{T}$ that defines a rotation of the m components into a new $p \times m$ matrix, $\mathbf{B}$, that contains the correlations (loadings) between the p variables and the newly rotated axes: $\mathbf{AT} = \mathbf{B}$. A rotation matrix $\mathbf{T}$ is sought that produces "nice" properties in $\mathbf{B}$, e.g., a "simple structure", where generally the loadings are positive and either close to 1.0 or to 0.0.

The most common strategy is due to Kaiser, and calls for maximizing the normal varimax criterion:

$$\frac{1}{p} \sum_{j=1}^{m} [\sum_{i=1}^{p} (b_{ij}/h_i)^4 - \frac{\gamma}{p} \{\sum_{i=1}^{p} (b_{ij}/h_i)^2\}^2] \ ,$$

where the parameter $\gamma = 1$ for varimax, and $h_i = \sqrt{\sum_{j=1}^{m} b_{ij}^2}$ (this is called the square root of the communality of the i^{th} variable in a factor analytic context). Other criteria have been suggested for this so-called orthomax criterion that use different values of γ — 0 for quartimax, $m/2$ for equamax, and $p(m-1)/(p+m-2)$ for parsimax. Also, various methods are available for attempting oblique rotations where the transformed axes do not need to maintain orthogonality, e.g., oblimin in SYSTAT; Procrustes in MATLAB.

Generally, varimax seems to be a good default choice. It tends to "smear" the variance explained across the transformed axes rather evenly. We will stick with varimax in the various examples we do later.

0.6 Little Jiffy

Chester Harris named a procedure posed by Henry Kaiser for "factor analysis," Little Jiffy. It is defined very simply as "principal components (of a correlation matrix) with associated eigenvalues greater than 1.0 followed by normal varimax rotation". To this date, it is the most used approach to do a factor analysis, and could be called "the principal component solution to the factor analytic model".

More explicitly, we start with the $p \times p$ sample correlation matrix $\mathbf{R}$ and assume it has r eigenvalues greater than 1.0. $\mathbf{R}$ is then approximated by a rank r matrix of the form:

$$\mathbf{R} \approx \lambda_1 \mathbf{a}_1 \mathbf{a}_1' + \cdots + \lambda_r \mathbf{a}_r \mathbf{a}_r' =$$
$$(\sqrt{\lambda_1}\mathbf{a}_1)(\sqrt{\lambda_1}\mathbf{a}_1') + \cdots + (\sqrt{\lambda_r}\mathbf{a}_r)(\sqrt{\lambda_r}\mathbf{a}_r') =$$
$$\mathbf{b}_1\mathbf{b}_1' + \cdots + \mathbf{b}_r\mathbf{b}_r' =$$
$$(\mathbf{b}_1, \ldots, \mathbf{b}_r) \begin{pmatrix} \mathbf{b}_1' \\ \vdots \\ \mathbf{b}_r' \end{pmatrix} = \mathbf{B}\mathbf{B}' \ ,$$

where

$$\mathbf{B}_{p\times r} = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1r} \\ b_{21} & b_{22} & \cdots & b_{2r} \\ \vdots & \vdots & & \vdots \\ b_{p1} & b_{p2} & \cdots & b_{pr} \end{pmatrix} \ .$$

The entries in $\mathbf{B}$ are the loadings of the row variables on the column components.

For any $r \times r$ orthogonal matrix $\mathbf{T}$, we know $\mathbf{T}\mathbf{T}' = \mathbf{I}$, and

$$\mathbf{R} \approx \mathbf{B}\mathbf{I}\mathbf{B}' = \mathbf{B}\mathbf{T}\mathbf{T}'\mathbf{B}' = (\mathbf{B}\mathbf{T})(\mathbf{B}\mathbf{T})' = \mathbf{B}^*_{p\times r}\mathbf{B}^{*'}_{r\times p} \ .$$

For example, varimax is one method for constructing $\mathbf{B}^*$. The columns of $\mathbf{B}^*$ when normalized to unit length, define r linear composites of the observable variables, where the sum of squares within columns of $\mathbf{B}^*$ defines the variance for that composite. The composites are still orthogonal.

0.7 Principal Components in Terms of the Data Matrix

For convenience, suppose we transform our $n \times p$ data matrix $\mathbf{X}$ into the z-score data matrix $\mathbf{Z}$, and assuming $n > p$, let the SVD of $\mathbf{Z}_{n\times p} = \mathbf{U}_{n\times p}\mathbf{D}_{p\times p}\mathbf{V}'_{p\times p}$. Note that the $p \times p$ correlation matrix

$$\mathbf{R} = \frac{1}{n}\mathbf{Z}'\mathbf{Z} = \frac{1}{n}(\mathbf{V}\mathbf{D}\mathbf{U}')(\mathbf{U}\mathbf{D}\mathbf{V}') = \mathbf{V}(\frac{1}{n}\mathbf{D}^2)\mathbf{V}' \ .$$

So, the rows of $\mathbf{V}'$ are the principal component weights. Also,

$$\mathbf{Z}\mathbf{V} = \mathbf{U}\mathbf{D}\mathbf{V}'\mathbf{V} = \mathbf{U}\mathbf{D} \ .$$

In other words, $(\mathbf{U}\mathbf{D})_{n\times p}$ are the scores for the n subjects on the p principal components.

What's going on in "variable" space: Suppose we look at a rank 2 approximation of $\mathbf{Z}_{n\times p} \approx \mathbf{U}_{n\times 2}\mathbf{D}_{2\times 2}\mathbf{V}'_{2\times p}$. The i^{th} subject's row data vector sits somewhere in p-dimensional "variable" space; it is approximated by a linear combination of the two eigenvectors (which gives another point in p dimensions), where the weights used in the linear combination come from the i^{th} row of $(\mathbf{U}\mathbf{D})_{n\times 2}$. Because we do least-squares, we are minimizing the squared Euclidean distances between the subject's row vector and the vector defined by the par-

ticular linear combination of the two eigenvectors. These approximating vectors in p dimensions are all in a plane defined by all linear combinations of the two eigenvectors. For a rank 1 approximation, we merely have a multiple of the first eigenvector (in p dimensions) as the approximating vector for a subject's row vector.

What's going on in "subject space": Suppose we begin by looking at a rank 1 approximation of $\mathbf{Z}_{n\times p} \approx \mathbf{U}_{n\times 1}\mathbf{D}_{1\times 1}\mathbf{V}'_{1\times p}$. The j^{th} column (i.e., variable) of $\mathbf{Z}$ is a point in n-dimensional "subject space," and is approximated by a multiple of the scores on the first component, $(\mathbf{UD})_{n\times 1}$. The multiple used is the j^{th} element of the $1 \times p$ vector of first component weights, $\mathbf{V}'_{1\times p}$. Thus, each column of the $n \times p$ approximating matrix, $\mathbf{U}_{n\times 1}\mathbf{D}_{1\times 1}\mathbf{V}'_{1\times p}$, is a multiple of the same vector giving the scores on the first component. In other words, we represent each column (variable) by a multiple of one specific vector, where the multiple represents where the projection lies on this one single vector (the term "projection" is used because of the least-squares property of the approximation). For a rank 2 approximation, each column variable in $\mathbf{Z}$ is represented by a point in the plane defined by all linear combinations of the two component score columns in $\mathbf{U}_{n\times 2}\mathbf{D}_{2\times 2}$; the point in that plane is determined by the weights in the j^{th} column of $\mathbf{V}'_{2\times p}$. Alternatively, $\mathbf{Z}$ is approximated by the sum of two $n \times p$ matrices defined by columns being multiples of the first or second component scores.

As a way of illustrating a graphical way of representing principal components of a data matrix (through a biplot), suppose we have the rank 2 approximation, $\mathbf{Z}_{n\times p} \approx \mathbf{U}_{n\times 2}\mathbf{D}_{2\times 2}\mathbf{V}'_{2\times p}$, and consider a two-dimensional Cartesian system where the horizontal axis cor-

responds to the first component and the vertical axis corresponds to the second component. Use the n two-dimensional coordinates in $(\mathbf{U}_{n\times 2}\mathbf{D}_{2\times 2})_{n\times 2}$ to plot the rows (subjects), let $\mathbf{V}_{p\times 2}$ define the two-dimensional coordinates for the p variables in this same space. As in any biplot, if a vector is drawn from the origin through the i^{th} row (subject) point, and the p column points are projected onto this vector, the collection of such projections is proportional to the i^{th} row of the $n \times p$ approximation matrix $(\mathbf{U}_{n\times 2}\mathbf{D}_{2\times 2}\mathbf{V}'_{2\times p})_{n\times p}$.

The emphasis in this notes has been on the descriptive aspects of principal components. For a discussion of the statistical properties of these entities, consult Johnson and Wichern (2007) — confidence intervals on the population eigenvalues; testing equality of eigenvalues; assessing the patterning present in an eigenvector; and so on.

Notes on Factor Analysis

The first question we need to address is why go to the trouble of developing a specific factor analysis model when principal components and "Little Jiffy" seem to get at this same problem of defining factors:

(1) In a principal component approach, the emphasis is completely on linear combinations of the observable random variables. There is no underlying (latent) structure of the variables that I try to estimate. Statisticians generally love models and find principal components to be somewhat inelegant and nonstatistical.

(2) The issue of how many components should be extracted is always an open question. With explicit models having differing numbers of "factors," we might be able to see which of the models fits "best" through some formal statistical mechanism.

(3) Depending upon the scale of the variables used (i.e., the variances), principal components may vary and there is no direct way of relating the components obtained on the correlation matrix and the original variance-covariance matrix. With some forms of factor analysis, such as maximum likelihood (ML), it is possible to go between the results obtained from the covariance matrix and the correlations by dividing or multiplying by the standard deviations of the variables. In other words, we can have a certain type of "scale invariance" if we choose, for example, the maximum likelihood approach.

(4) If one wishes to work with a correlation matrix and have a means of testing whether a particular model is adequate or to develop

confidence intervals and the like, it is probably preferable to use the ML approach. In PCA on a correlation matrix, the results that are usable for statistical inference are limited and very strained generally (and somewhat suspect).

To develop the factor analysis model, assume the p *observable* random variables, $\mathbf{X}' = [X_1, \ldots, X_p]$, are MVN($\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$). Without loss of generality, we can assume that $\boldsymbol{\mu}$ is the zero vector. Also, suppose that each X_i can be represented by a linear combination of some m *unobservable* or latent random variables, $\mathbf{Y}' = [Y_1, \ldots, Y_m]$, plus an error term, e_i:

$$X_i = \lambda_{i1}Y_1 + \cdots + \lambda_{im}Y_m + e_i, \text{ for } 1 \leq i \leq p \ .$$

Here, $Y_1, \ldots, Y_m$ are the common factor variables; $e_1, \ldots, e_p$ are the specific factor variables; λ_{ij} is the *loading* (i.e., the covariance) of the i^{th} response variable, X_i, on the j^{th} common factor variable.

If $\mathbf{e}' = [e_1, \ldots, e_p]$, then $\mathbf{X} = \boldsymbol{\Lambda}\mathbf{Y} + \mathbf{e}$, where

$$\boldsymbol{\Lambda} = \begin{pmatrix} \lambda_{11} & \cdots & \lambda_{1m} \\ \vdots & & \vdots \\ \lambda_{p1} & \cdots & \lambda_{pm} \end{pmatrix} \ .$$

For notation, we let the variance of e_i be ψ_i, $1 \leq i \leq p$, and refer to ψ_i as the *specific variance* of the i^{th} response variable; $e_i \sim$ N(0,ψ_i) and all the e_is are independent of each other; $Y_i \sim$ N(0,1) and all the Y_is are independent of each other and of the e_is. Also, we define the diagonal matrix containing the specific variances to be

$$\boldsymbol{\Psi} = \begin{pmatrix} \psi_1 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & \psi_p \end{pmatrix} \ .$$

$$\text{Var}(X_i) = \text{Var}(\lambda_{i1}Y_1 + \cdots \lambda_{im}Y_m + e_i) =$$

$$\text{Var}(\lambda_{i1}Y_1) + \cdots + \text{Var}(\lambda_{im}Y_m) + \text{Var}(e_i) =$$

$$\lambda_{i1}^2 + \cdots + \lambda_{im}^2 + \psi_i \ .$$

The expression, $\Sigma_{j=1}^m \lambda_{ij}^2$, is called the communality of the i^{th} variable, X_i.

Because terms involving different unobservable and specific variables are zero because of independence, we have

$$\text{Cov}(X_i, X_j) = \text{Cov}(\lambda_{i1}Y_1 + \cdots \lambda_{im}Y_m + e_i, \lambda_{j1}Y_1 + \cdots \lambda_{jm}Y_m + e_j) =$$

$$\lambda_{i1}\lambda_{j1} + \cdots + \lambda_{im}\lambda_{jm} \ .$$

As a way of summarizing the results just given for the variances and covariances of the observable variables in terms of the loadings and specific variances, the factor analytic model is typically written as

$$\mathbf{\Sigma}_{p\times p} = \mathbf{\Lambda}_{p\times m}\mathbf{\Lambda}'_{m\times p} + \mathbf{\Psi}_{p\times p} \ .$$

There is a degree of indeterminacy in how this model is phrased, because for any $m \times m$ orthogonal matrix $\mathbf{T}$, we have the same type of decomposition of $\mathbf{\Sigma}$ as

$$\mathbf{\Sigma}_{p\times p} = (\mathbf{\Lambda T})_{p\times m}(\mathbf{\Lambda T})'_{m\times p} + \mathbf{\Psi}_{p\times p} \ .$$

Thus, we have a rotation done by $\mathbf{T}$ to generate a new loading matrix, $\mathbf{\Lambda T}$.

0.8 Iterated Principal (Axis) Factor Analysis

Suppose I assume the factor analytic model to hold for the population correlation matrix, $\mathbf{P} = \mathbf{\Lambda}\mathbf{\Lambda}' + \mathbf{\Psi}$, and am given the sample correlation matrix, $\mathbf{R}$. The Guttman lower bound to the communality of a variable is the squared multiple correlation of that variable with the others, and can be used to give an initial estimate, $\hat{\mathbf{\Psi}}$, of the matrix of specific variances by subtracting these lower bounds from 1.0 (the main diagonal entries in $\mathbf{R}$). A component analysis (with m components) is carried out on $\mathbf{R} - \hat{\mathbf{\Psi}}$ and then normalized to produce a factoring, say, $\mathbf{BB}'$. We estimate $\mathbf{\Psi}$ by using the diagonal of $\mathbf{R} - \mathbf{BB}'$, and iterate the process until convergence. (Little Jiffy (the principal component solution to the factor analysis model) could be viewed as a "one shot" process, with specific variances set at 0.0.)

0.9 Maximum Likelihood Factor Analysis (MLFA)

The method of MLFA holds out the hope of being a scale-invariant method, implying that the results from a correlation or the covariance matrix can be transformed into each other though simple multiplications by the variable standard deviations. So if λ_{ij} is a loading from a (population) correlation matrix, then $\lambda_{ij}\sigma_i$ is the corresponding loading from the (population) covariance matrix.

MLFA begins with the assumption that $\mathbf{X}_{p\times 1} \sim \text{MVN}(\mathbf{0}, \mathbf{\Sigma}_{p\times p} = \mathbf{\Lambda}_{p\times m}\mathbf{\Lambda}'_{m\times p} + \mathbf{\Psi}_{p\times p})$. If there is a unique diagonal matrix, $\mathbf{\Psi}$, with positive elements such that the m largest roots (eigenvalues) of $\mathbf{\Sigma}^* = \mathbf{\Psi}^{-1/2}\mathbf{\Sigma}\mathbf{\Psi}^{-1/2}$ are distinct and greater than unity, and the $p - m$

remaining roots are each unity (this is true if the model holds), then $\mathbf{\Lambda} = \mathbf{\Psi}^{1/2}\mathbf{\Omega}\mathbf{\Delta}^{1/2}$, where $\mathbf{\Sigma}^* - \mathbf{I} = \mathbf{\Omega}_{p\times m}\mathbf{\Delta}_{m\times m}\mathbf{\Omega}'_{m\times p}$. In other words, once you get $\mathbf{\Psi}$, you are "home free" because $\mathbf{\Lambda}$ comes along by a formula.

So, we start with some $\mathbf{\Psi}$ (and generating $\mathbf{\Lambda}$ automatically), and improve upon this initial value by maximizing the log-likelihood

$$\ell(\mathbf{\Lambda}, \mathbf{\Psi}) = -\frac{n}{2}(\ln|\mathbf{\Sigma}| + \mathrm{Tr}(\mathbf{S}\mathbf{\Sigma}^{-1})) + \text{constant} .$$

Equivalently, we can minimize

$$\mathrm{F}(\mathbf{\Lambda}, \mathbf{\Psi}) = \ln|\mathbf{\Sigma}| + \mathrm{Tr}(\mathbf{S}\mathbf{\Sigma}^{-1})) - \ln|\mathbf{S}| - p .$$

The particular iterative optimization procedure used to obtain better and better values for $\mathbf{\Psi}$ is typically the Davidon-Fletcher-Powell method.

In practice, one has a large sample likelihood ratio test available of

$$H_0 : \mathbf{\Sigma} = \mathbf{\Lambda}\mathbf{\Lambda}' + \mathbf{\Psi} ,$$

using a test statistic of $(n - (2p+5)/6 - 2m/3)\mathrm{F}(\hat{\mathbf{\Lambda}}, \hat{\mathbf{\Psi}})$, compared to a chi-squared random variable with $\frac{1}{2}[(p-m)^2 - (p+m)]$ degrees of freedom. Generally, the residuals one gets from an MLFA tend to be smaller than from a PCA, even though the cumulative variance explained in a PCA is usually larger; these are somewhat different criteria of fit.

In MLFA, one typically needs a rotation (oblique or orthogonal) to make the originally generated factors intelligible. Also, we now have various forms of confirmatory factor analysis (CFA) where some of

the loadings might be fixed and others free to vary. CFA seems to be all the rage in scale development, but I would still like to see what a PCA tells you in an exploratory and optimized context. Finally, and although we talked about using and plotting component scores on our subjects in PCA, the comparable factor scores here should *not* be used. There has been an enormous controversy about their indeterminacy; among people who are thinking straight (e.g., SYSTAT and Leland Wilkinson), factor scores are just not given.

When one allows correlated factors (e.g., using an oblique rotation), the factor analytic model is generalized to

$$\mathbf{\Sigma} = \mathbf{\Lambda}\mathbf{\Phi}\mathbf{\Lambda}' + \mathbf{\Psi}$$

where $\mathbf{\Phi}$ is the $m \times m$ covariance matrix among the m factors. In terms of terminology, the matrix, $\mathbf{\Lambda}$, is called the factor *pattern* matrix; $\mathbf{\Lambda\Phi}$ is called the factor *structure* matrix and contains the covariances between the observed variables and the m common factors.

There is one property of MLFA that sometimes (in fact, often) rears its ugly head, involving what are called Heywood cases (or improper solutions) in which the optimization procedure wants to make some of the ψ_is go negative. When this appears to be happening, the standard strategy is to remove the set of variables for which the ψ_is want to go negative, set them equal to zero exactly; the removed set is then subjected to a principal component analysis, and a "kluge" made of the principal components and the results from an MLFA on a covariance matrix residualized from the removed set. Obviously, the nice scale invariance of a true MLFA approach disappears when

these improper solutions are encountered. You can tell immediately that you have this kind of hybrid solution when some of the specific variances are exactly zero.