Applied Multivariate Analysis, Notes originally for the course of Lent 2004, MPhil in Statistical Science, gradually updated

P.M.E.Altham, Statistical Laboratory, University of Cambridge.

September 23, 2013

Contents

Preface

Note added 2013: these are essentially my original notes, but I have just done a little tidying up, and have included a couple of extra graphs for clarity.
All of the statistical techniques described may be implemented in R: see `http://www.statslab.cam.ac.uk/~pat/misc.pdf` for examples.
I have also appended the Exercises sheet at the end of this set of notes, for convenience.
There are 6 Chapters in all, intended for a 16-hour course, of which about 8 hours should be practical classes: I used R or S-plus for these.

Chapter 1

Properties of the multivariate normal distribution

The multivariate normal distribution is the basis for many of the classical techniques in multivariate analysis. It has many beautiful properties. Here we mention only a few of these properties, with an eye to the statistical inference that will come in subsequent Chapters.

Definition and Notation.
We write

$$X \sim N_p(\mu, V)$$

if the $p-$dimensional random vector X has the pdf

$$f(x|\mu, V) \propto \exp[-(x-\mu)^T V^{-1}(x-\mu)]/2$$

for $x \epsilon R^p$.
The constant of proportionality is $1/\sqrt{(2\pi)^p |V|}$, and we use the notation $|V|$ as the determinant of the matrix V.
Then this pdf has ellipsoidal contours, ie

$$f(x|\mu, V) = \text{constant}$$

is the equation

$$(x-\mu)^T V^{-1}(x-\mu) = \text{constant}$$

which is an ellipse (for $p = 2$) or an ellipsoid (for $p > 2$) centred on the point μ, with shape determined by the matrix V.
The characteristic function of X is say $\phi(t) = E(\exp it^T X)$, and you can check that

$$\phi(t) = \exp(it^T \mu - t^T V t/2)$$

(using the fact that $\int_x f(x|\mu, V)dx = 1$).
Furthermore, by differentiating $\phi(t)$ with respect to t and setting $t = 0$, you can see that

$$E(X) = \mu,$$

similarly, differentiating again and setting $t = 0$ shows you that

$$E(X - \mu)(X - \mu)^T = V.$$

V is the covariance matrix of X.
By definition, $u^T V u \geq 0$ for any vector u, ie the matrix V is positive semi-definite.
Here is one possible *characterisation* of the multivariate normal distribution:
X is multivariate normal if and only if
for any fixed vector a, $a^T X$ is univariate normal.

Partitioning the normal vector X
Take X_1 as the first p_1 elements of X, and X_2 as the last p_2 elements, where $p = p_1 + p_2$.
Assume as before that $X \sim N(\mu, V)$, and now suppose that $\mu^T = (\mu_1^T, \mu_2^T)$, with V partitioned in a corresponding fashion,

$$V = \begin{pmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{pmatrix}$$

then, for $i = 1, 2$, $X_i \sim N(\mu_i, V_{ii})$,
and

$$cov(X_1, X_2) = E(X_1 - \mu_1)(X_2 - \mu_2)^T = V_{12} = V_{21}^T$$

so that X_1, X_2 are independent iff V_{12} is a matrix with every element 0.

Linear transformation of a normal X
If $X \sim N(\mu, V)$ and C is an $m \times p$ constant matrix, then $CX \sim N(C\mu, CVC^T)$.

Diagonalisation
Suppose $X \sim N(\mu, V)$ and V is a positive-definite matrix, with eigen-values $\lambda_1, \ldots, \lambda_p$ say (which are then > 0, since V is positive-definite). Let $u_1, \ldots, u_p$ be the corresponding eigen-vectors of V, thus

$$V u_i = \lambda_i u_i, \text{ for } 1 \leq i \leq p,$$

and

$$u_i^T u_j = 0 \text{ for } i \neq j, 1 \text{ for } i = j,$$

ie the eigen-vectors are mutually orthogonal, and each is of length 1.
Define U as the $p \times p$ matrix whose columns are $(u_1, \ldots, u_p)$. Then

$$U^T X \sim N_p(U^T \mu, U^T V U).$$

But

$$u_j^T V u_i = \lambda_i u_j^T u_i$$

and this is λ_i for $i = j$, 0 otherwise. Hence

$$U^T V U = diag(\lambda_1, \ldots, \lambda_p).$$

Thus, given V, we can always construct an orthogonal matrix U such that if $Y = U^T X$, then $Y_1, \ldots, Y_p$ are independent normal variables (with variances, $\lambda_1, \ldots, \lambda_p$ in fact).
Exercises.
i) Given $X \sim N(\mu, V)$, modify the above proof to construct a matrix D such that

$$DX \sim N(D\mu, I_p)$$

where I_p is the $p \times p$ identity matrix.
Hence
ii) show that $(X - \mu)^T V^{-1}(X - \mu)$ is distributed as χ^2, with p df.

Less familiar facts about the normal distribution
Conditioning.
Take $X \sim N(\mu, V)$ and partition the vector X as before, so that $X^T = (X_1^T, X_2^T)$. We will prove that

$$X_1 | (X_2 = x_2) \sim N(\nu_1, V_{11.2}) \text{ say}$$

where
$\nu_1 = \mu_1 + V_{12}V_{22}^{-1}(x_2 - \mu_2)$, the conditional mean vector,
and $V_{11.2} = V_{11} - V_{12}V_{22}^{-1}V_{21}$, the conditional covariance matrix.
(Observe that $V_{11.2}$ is free of x_2.)
Proof
Note that we can always derive a conditional pdf as

$$f(x_1|x_2) = f(x_1, x_2)/f(x_2)$$

(ie joint pdf divided by marginal pdf), but in the current proof we employ a MORE CUNNING argument. (Fine if you know how to get started.)
Suppose the vector Y has components Y_1, Y_2 say, where

$$Y_1 = X_1 - V_{12}V_{22}^{-1}X_2, \text{ and } Y_2 = X_2.$$

Thus we have written $Y = CX$ say, where

$$C = \begin{pmatrix} I_{p_1} & -V_{12}V_{22}^{-1} \\ 0 & I_{p_2} \end{pmatrix}$$

and so $Y \sim N(C\mu, CVC^T)$, and you can check that

$$C\mu = \begin{pmatrix} \mu_1 - V_{12}V_{22}^{-1}\mu_2 \\ \mu_2 \end{pmatrix}$$

and

$$CVC^T = \begin{pmatrix} V_{11.2} & 0 \\ 0 & V_{22} \end{pmatrix}$$

(Multiply out as if we had 2×2 matrices).
Hence Y_1 and Y_2 are independent vectors, ie $X_1 - V_{12}V_{22}^{-1}X_2$ is independent of X_2.
Thus the distribution of $(X_1 - V_{12}V_{22}^{-1}X_2)|(X_2 = x_2)$ is the same as the distribution of $(X_1 - V_{12}V_{22}^{-1}X_2)$, which is Normal with covariance matrix $V_{11.2}$.
Hence $(X_1 - V_{12}V_{22}^{-1}X_2)|(X_2 = x_2)$ is Normal with covariance matrix $V_{11.2}$.
Now $E(Y_1) = \mu_1 - V_{12}V_{22}^{-1}\mu_2$.
Hence $X_1|(X_2 = x_2)$ has distribution $N(\mu_1 + V_{12}V_{22}^{-1}(x_2 - \mu_2), V_{11.2})$, which is the required result.
Note
i)$E(X_1|X_2 = x_2) = \mu_1 + V_{12}V_{22}^{-1}(x_2 - \mu_2)$, a linear function of x_2, as we should expect,
ii)

$$var(X_1|X_2 = x_2) = V_{11} - V_{12}V_{22}^{-1}V_{21} \leq V_{11} = var(X_1)$$

(ie conditional variance is $\leq$ marginal variance)
in the sense that we take $A \leq B$ for matrices A, B if $B - A$ is a positive definite matrix.
Here

$$var(X_1|X_2 = x_2) = var(X_1)$$

iff $V_{12} = 0$, in which case X_1, X_2 are independent.
The correlation coefficient.
We take

$$X = \begin{pmatrix} X_1 \\ \vdots \\ X_p \end{pmatrix}$$

as before, and take V as the covariance matrix of X.
Definition The Pearson correlation coefficient between X_i, X_j is

$$\rho_{ij} = corr(X_i, X_j) = v_{ij}/\sqrt{(v_{ii}v_{jj})}.$$

(ρ_{ij}) is the correlation matrix of X.
Check: by definition, $\rho_{ij}^2 \leq 1$, with $=$ iff X_i is a linear function of X_j.
With

$$X = \begin{pmatrix} X_1 \\ X_2 \end{pmatrix}$$

with X_1, X_2 now of dimensions p_1, p_2 respectively, we know that

$$var(X_1|X_2 = x_2) = V_{11} - V_{12}V_{22}^{-1}V_{21}.$$

We could use this latter matrix to find the conditional correlation of, say, X_{1i}, X_{1j}, conditional on $X_2 = x_2$.
Exercise.
Suppose $X_1 = \epsilon_1 + aY$ and $X_2 = Y$, where $Y \sim N(0, V_{22})$ and $\epsilon_1 \sim N(0, I_{p_1})$, independently of Y, and a is a $p_1 \times p_2$ constant matrix. Show that, conditional

on $X_2 = x_2$, the components of X_1 are independent.
Clearly, for this example

$$var(X) = \begin{pmatrix} I + aV_{22}a^T & aV_{22} \\ V_{22}a^T & V_{22} \end{pmatrix}.$$

Two useful expressions from V^{-1}
First we find an expression for the **conditional correlation**, say of X_1, X_2 conditional on the values of the remaining variables.
Suppose $X \sim N(0, V)$, and write X_i as the ith component of X. Then

$$f(x) \propto \exp -x^T ax/2,$$

where we have defined $a = V^{-1}$. Thus, expanding out the quadratic expression, we see that

$$f(x_1, x_2, z) \propto \exp -(a_{11}x_1^2 + 2a_{12}x_1x_2 + a_{22}x_2^2 + \text{terms linear in } x_1, x_2)/2,$$

where $z^T = (x_3, \ldots, x_p)$. Thus

$$f(x_1, x_2|z) = \frac{f(x_1, x_2, z)}{f(z)} \propto exp-(a_{11}(x_1-\mu_1)^2+2a_{12}(x_1-\mu_1)(x_2-\mu_2)+a_{22}(x_2-\mu_2)^2)/2$$

where we have defined $\mu_1 = E(X_1|z)$ and $\mu_2 = E(X_2|z)$, thus μ_1, μ_2 are linear functions of z, but not of interest to us at present. We compare the above expression for $f(x_1, x_2|z)$ with the bivariate normal density to find an expression for $corr(X_1, X_2|z)$ in terms of elements of a.
Suppose Y is bivariate normal, with $E(Y_i) = m_i$, and $var(Y_i) = \sigma_i^2$, and $cor(Y_1, Y_2) = \rho$, thus

$$\begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} \text{ has pdf } g(y_1, y_2) \propto \exp -(b_1^2 - 2\rho b_1 b_2 + b_2^2)/2(1 - \rho^2)$$

where we have written $b_i = (y_i - m_i)/\sigma_i$ for $i = 1, 2$.
Look at these two expressions for a density function and compare coefficients in the quadratic. You will then see that

$$-\rho = a_{12}/\sqrt{a_{11}a_{22}},$$

ie $corr(X_1, X_2|Z = z) = -a_{12}/\sqrt{a_{11}a_{22}}$, where a is the inverse of V, the covariance matrix.
Similarly, if we now define

$$z = \begin{pmatrix} x_2 \\ \vdots \\ x_p \end{pmatrix},$$

you will now see by a similar argument, that

$$f(x_1, z) \propto \exp -(a_{11}x_1^2 + \ldots)/2$$

and hence $var(X_1|Z = z) = 1/a_{11}$, where $a = V^{-1}$ as before.
If $1/a_{11}$ is small (compared with $var(X_1)$), then X_1 will be (almost) a linear function of $X_2, \ldots, X_p$.
In R, the matrix V has inverse

```
solve(V)
```

(recall that the original use of matrix inverse is to SOLVE a system of linear equations.)
Check If we write

$$V = \begin{pmatrix} v_{11} & b^T \\ b & V_{22} \end{pmatrix},$$

then

$$var(X_1|X_2 = x_2, \ldots, X_p = x_p) = v_{11} - b^T V_{22}^{-1} b.$$

Exercises
i) Suppose

$$\begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix} \sim N(.,.),$$

with $\rho_{ij} = corr(X_i, X_j)$.
Show that

$$Corr(X_1, X_2|X_3 = x_3) = (\rho_{12} - \rho_{13}\rho_{23})/\sqrt{((1-\rho_{13}^2)(1-\rho_{23}^2)}.$$

Hence show that if $\rho_{12} - \rho_{13}\rho_{23} = 0$, then we may represent the dependence graph of (X_1, X_2, X_3) as

```
X1----X3-----X2
```

ie X_1 and X_2 are 'linked' only through X_3.
(This is rather a 'poor-man's graphic': doubtless you can do it in a better way.)

This would be the case if, for example,
$X_1 = \alpha_1 X_3 + \epsilon_1$,
$X_2 = \alpha_2 X_3 + \epsilon_2$
where $\epsilon_1, \epsilon_2, X_3$ are independent random variables.
ii) Suppose $X \sim N(0, V)$ and $X^T = (X_1, X_2^T)$, where X_1, X_2 are of dimensions $1, p-1$ respectively.
The **multiple correlation coefficient** between X_1, X_2 is defined as the maximum value of $corr(X_1, \alpha^T X_2)$, maximising wrt the vector α.
Show that this maximising α is given by

$$\alpha^T \propto V_{12} V_{22}^{-1}$$

where we have partitioned V in the obvious way, and find the resulting multiple correlation coefficient.
Hint: $cov(X_1, \alpha^T X_2) = \alpha^T V_{21}$, and $var(\alpha^T X_2 \alpha) = \alpha^T V_{22} \alpha$. So the problem is equivalent to:
maximise $\alpha^T V_{21}$ subject to $\alpha^T V_{22} \alpha = 1$. We write down the corresponding Lagrangian.

Chapter 2

Estimation and Testing for the multivariate normal distribution

2.1 Maximum Likelihood Estimation (mle)

Let $x_1, \ldots, x_n$ be a random sample (rs) from $N_p(\mu, V)$. Then

$$f(x_1, \ldots, x_n|\mu, V) \propto 1/|V|^{n/2} \exp -\Sigma_1^n (x_i - \mu)^T V^{-1}(x_i - \mu)/2.$$

Now

$$\Sigma(x_i - \mu)^T V^{-1}(x_i - \mu) = \Sigma(x_i - \bar{x} + \bar{x} - \mu)^T V^{-1}(x_i - \bar{x} + \bar{x} - \mu)$$

where $\bar{x} = \Sigma x_i/n$. Thus

$$\Sigma(x_i - \mu)^T V^{-1}(x_i - \mu) = \Sigma(x_i - \bar{x})^T V^{-1}(x_i - \bar{x}) + n(\bar{x} - \mu)^T V^{-1}(\bar{x} - \mu).$$

(Check this.) Hence $-2 \log f(x_1, \ldots, x_n|\mu, V)$

$$= n \log |V| + \Sigma(x_i - \bar{x})^T V^{-1}(x_i - \bar{x}) + n(\bar{x} - \mu)^T V^{-1}(\bar{x} - \mu).$$

Recall, the trace of a square matrix is the sum of its diagonal elements. Now, for any vector u, $u^T V^{-1} u$ is a scalar quantity, and hence is equal to its **trace**, ie

$$u^T V^{-1} u = tr(u^T V^{-1} u)$$

and this in turn is equal to

$$tr(V^{-1} u u^T).$$

Further, for any matrices, $tr(A + B) = tr(A) + tr(B)$.
Hence we may write -2 log-likelihood as

$$= n \log(|V|) + tr(V^{-1}\Sigma(x_i - \bar{x})(x_i - \bar{x})^T + n(\bar{x} - \mu)^T V^{-1}(\bar{x} - \mu)$$

$$= n \log(|V|) + tr(V^{-1} n S) + n(\bar{x} - \mu)^T V^{-1}(\bar{x} - \mu),$$

where

$$S = (1/n)\Sigma(x_i - \bar{x})(x_i - \bar{x})^T,$$

the **sample covariance matrix**.
Thus we see that
(i) $(\bar{x}, S)$ is sufficient for (μ, V), by the factorisation theorem, and
(ii) it is easy to minimise -2 loglikelihood, with respect to μ, for V a fixed positive definite matrix.
Clearly,

$$(\bar{x} - \mu)^T V^{-1} (\bar{x} - \mu) \geq 0,$$

with $=$ if and only if $\mu = \bar{x}$.
Hence $\hat{\mu} = \bar{x}$, (whether or not V is known).
Finally, we wish to minimise the expression

$$l(V) = nlog|V| + ntr(V^{-1}S),$$

with respect to V, for V a positive-definite matrix. (Of course, experience with the univariate normal tells us to expect the answer $\hat{V} = S$.)
Observe that, at $V = S$,

$$l(V) = nlog|S| + ntr(I_p) = nlog|S| + np.$$

a) Here is the slick way to achieve the minimisation.

Note that we may write

$$l(V) = -n\ log|V^{-1}| + n\ tr(V^{-1}S),$$

and hence

$$l(V) = -n\ log|V^{-1}S| + n\ tr(V^{-1}S) + \text{constant}.$$

But each of V, S is symmetric and positive-definite. Suppose λ is an eigen-value of $V^{-1}S$, with $V^{-1}Su = \lambda u$.
Thus

$$Su = \lambda Vu = \lambda LL^T u,$$

say, where L is a real non-singular matrix. Hence

$$L^{-1}Su = \lambda L^T u$$

and so

$$L^{-1}S(L^{-1})^T)(L^T u) = \lambda(L^T u).$$

Thus, λ is an eigen-value of the symmetric positive-definite matrix $L^{-1}S(L^{-1})^T$, and hence λ is real and positive. Let $\lambda_1, \ldots, \lambda_p$ be the eigen-values of $L^{-1}S(L^{-1})^T$.
Further we may write

$$|V^{-1}S| = |(L^T)^{-1}L^{-1}S| = |L^{-1}S(L^{-1})^T| = \Pi\lambda_i,$$

similarly,

$$tr(V^{-1}S) = tr((L^T)^{-1}L^{-1}S) = tr(L^{-1}S(L^{-1})^T) = \Sigma\lambda_i.$$

Thus, our problem reduces to the problem,
find $\lambda_1, \ldots, \lambda_p \geq 0$ to minimise

$$l(V) = -n \log \Pi\lambda_i + n\Sigma\lambda_i.$$

Thus,

$$l(V) = n\Sigma_i(\lambda_i - \log \lambda_i).$$

Now find

$$\frac{\partial l(\lambda)}{\partial \lambda_i}, \text{ and } \frac{\partial^2 l(\lambda)}{\partial \lambda_i^2}$$

in order to show that $l(V)$ is minimised with respect to $\lambda_1, \ldots, \lambda_p > 0$ by $\lambda_i = 1$ for $i = 1, \ldots, p$, thus

$$V^{-1}S = I_p, \text{ and so } V = S,$$

as we would expect, from the 1-dimensional case.
We write $\hat{V} = S$, we shall show later that this is not an unbiased estimator of V.

b) The brute force method.
Put $\Psi = V^{-1}$; our problem is to choose Ψ to minimise

$$f(\Psi) = -n \log |\Psi| + \Sigma_k z_k^T \Psi z_k,$$

where we have written $z_k = x_k - \bar{x}$ for $k = 1, \ldots, n$. Heroic feats (well, perhaps mildly heroic) of calculus (to be demonstrated by your lecturer) enable us to write down the equations

$$\frac{\partial f(\Psi)}{\partial \Psi_{ij}} = 0$$

and hence find the solution

$$\Psi^{-1} = S,$$

as above. Here is how to get started.
Note that we consider the more general problem: choose a matrix Ψ to minimise $f(\Psi)$. We do not include the constraint thatΨ is a symmetric positive definite matrix.

$$\frac{\partial f(\Psi)}{\partial \Psi_{ij}} = (-n/det(\Psi))\frac{\partial det\Psi}{\partial \Psi_{ij}} + \Sigma z_{ki} z_{kj}$$

Hence

$$\frac{\partial f(\Psi)}{\partial \Psi_{ij}} = 0$$

is equivalent to

$$(1/det(\Psi))\frac{\partial det\Psi}{\partial \Psi_{ij}} = S_{ij}$$

using the fact that $S_{ij} = \Sigma z_{ki} z_{kj}$.

2.2 The distribution of the mle's

Notation, for reminder. Take $X_1, \ldots, X_n$ a random sample from $N_p(\mu, V)$, and write $\hat{\mu} = \bar{X}$, $\hat{V} = S$. Recall, if $p = 1$,
$\bar{X} \sim N(\mu, V/n)$ and $nS/(\sigma^2) \sim \chi^2_{n-1}$, independently, (where $V = \sigma^2$). We seek the multivariate version of this result, which we will then apply.
Here is the easy part of the result: clearly $X_1, \ldots, X_n \sim NID(\mu, V)$ implies that $\bar{X} \sim N(\mu, V/n)$.
For the rest, we first need a multivariate version of the χ^2 distribution: this is the **Wishart** distribution, defined as follows.
Take $Z_\alpha,\ 1 \leq \alpha \leq n-1$ as $NID(0, V)$ where of course V is a $p \times p$ matrix. Then we say that

$$\Sigma_1^{n-1} Z_\alpha Z_\alpha^T$$

has the Wishart distribution, parameters $n-1, V$.
Fact

$$nS = \Sigma_1^n (X_i - \bar{X})(X_i - \bar{X})^T$$

which is of course a random matrix, is distributed as

$$\Sigma_1^{n-1} Z_\alpha Z_\alpha^T,$$

independently of $\bar{X}$.
Proof is omitted (but see eg Seber 1984).
Exercise. Note that $nS = \Sigma X_i X_i^T - n\bar{X}\bar{X}^T$. Hence show that $E(nS) = (n-1)V$.
Testing hypotheses for μ when V is known, for example, to test $H_0 : \mu = \mu_0$.
Now since $\sqrt{n}(\bar{X} - \mu_o) \sim N(0, V)$ if H_0 is true, we refer $n(\bar{X} - \mu_0)^T V^{-1} (\bar{X} - \mu_0)$ to χ^2_p to test H_0.
Similarly, we can contruct a 95% confidence region for μ from $\bar{X}$.
Testing hypotheses for μ when V is unknown:
We know that $\bar{X} \sim N(\mu, V/n)$, and hence $(\bar{X} - \mu)\sqrt{n} \sim N(0, V)$, independently of our estimate S of V; we know that nS has the Wishart distribution given above.
It therefore seems obvious that our test statistic, eg of $H_0 : \mu = \mu_0$, must be

$$\text{constant } (\bar{X} - \mu)^T S^{-1} (\bar{X} - \mu)$$

having distribution (on H_0) which is free of the unknown V, and is the multivariate version of t_{n-1}.
This is actually true, but is surprisingly lengthy to prove: see standard texts, eg Seber, for proof.
For the present, we merely note that **Hotelling's** T^2 is defined as

$$T^2 = n(\bar{x} - \mu_0)^T (nS/(n-1))^{-1} (\bar{x} - \mu_0)$$

and the exact distribution of T^2 on H_0 is known,
it is obviously $F_{1,n-1}$ if $p = 1$,
and for general p,

$$\frac{n-p}{p} \frac{T^2}{n-1} \sim F_{p,n-p}.$$

Again, we omit the proof.
But, we note that this result can be **used** to find the multivariate version of the 2-sample t-test:
eg, take $X_1, \ldots, X_m \sim NID(\mu_1, V)$
and $Y_1, \ldots, Y_n \sim NID(\mu_2, V)$ and assume that the X's are independent of the Y's.
Then

$$\bar{X} - \bar{Y} \sim N(\mu_1 - \mu_2, (1/m + 1/n)V)$$

independently of

$$(m+n)S = \Sigma_i (X_i - \bar{X})(X_i - \bar{X})^T + \Sigma_j (Y_j - \bar{Y})(Y_j - \bar{Y})^T$$

from which we could construct a Hotelling's T^2 statistic, of known distribution, to test $H_0 : \mu_1 = \mu_2$.

Exercise.
Let $X_1, \ldots, X_n$ be a random sample from $N(\mu, V)$, where μ, V are unknown. Show that T^2, as defined above, is equivalent to the generalized likelihood ratio test of $H_0 : \mu = \mu_0$.
Hint: you will need the matrix identity

$$|A + uu^T| = |A|(1 + u^T A^{-1} u)$$

where A is a positive-definite $p \times p$ matrix, and u is a $p-$dimensional vector.
Here's how the matrix identity is proved:
Suppose $A = LL^T$ where L is a real and non-singular matrix. Then we see that

$$A + uu^T = LL^T + uu^T = L(I + vv^T)L^T$$

where we have defined the vector v to be $L^{-1}u$. Thus we see that

$$det(A+uu^T) = |L|det(I+vv^T)|L^T| = det(LL^T)det(I+vv^T) = |A|det(I+vv^T).$$

But, $det(I + vv^T)$ is the product of the eigen values of this matrix. Now,

$$(I + vv^T)v = (1 + v^T v)v$$

hence $(1 + v^T v)$ is an eigen-value of this matrix, and it corresponds to eigen vector v. Take x any vector orthogonal to v, then

$$(I + vv^T)x = 1x$$

hence *every other* eigen value of $(I + vv^T)$ is 1. Thus the product of the eigen values is $(1 + v^T v)$, and so we see that

$$det(A + uu^T) = |A|(1 + u^T A^{-1} u)$$

as required.

Discussion of the derivation of a test criterion via the generalized likelihood ratio method leads naturally to the next topic:

2.3 The multivariate analysis of variance

`manova()`

in R.
First an example, from

```
library(MASS)
```

of the famous 'painters' data, for which we may wish to apply

```
manova(cbind(Composition,Drawing, Colour, Expression) ~ School, painters)
```

An 18th century art critic called de Piles rated each of 54 painters, on a scale of 0 to 20, on each of the following 4 variables, Composition, Drawing, Colour and Expression. The painters are also classified according to their 'School': these are
A= Renaissance, B= Mannerist, C= Seicento, D= Venetian, E= Lombard, F= 16thC, G= 17thC, and finally H= French. Here are the data

	Composition	Drawing	Colour	Expression	School
Da Udine	10	8	16	3	A
Da Vinci	15	16	4	14	A
Del Piombo	8	13	16	7	A
Del Sarto	12	16	9	8	A
Fr. Penni	0	15	8	0	A
Guilio Romano	15	16	4	14	A
Michelangelo	8	17	4	8	A
Perino del Vaga	15	16	7	6	A
Perugino	4	12	10	4	A
Raphael	17	18	12	18	A
F. Zucarro	10	13	8	8	B
Fr. Salviata	13	15	8	8	B
Parmigiano	10	15	6	6	B
Primaticcio	15	14	7	10	B
T. Zucarro	13	14	10	9	B
Volterra	12	15	5	8	B
Barocci	14	15	6	10	C
Cortona	16	14	12	6	C
Josepin	10	10	6	2	C
L. Jordaens	13	12	9	6	C
Testa	11	15	0	6	C
Vanius	15	15	12	13	C
Bassano	6	8	17	0	D
Bellini	4	6	14	0	D
Giorgione	8	9	18	4	D
Murillo	6	8	15	4	D
Palma Giovane	12	9	14	6	D
Palma Vecchio	5	6	16	0	D

Pordenone	8	14	17	5	D
Tintoretto	15	14	16	4	D
Titian	12	15	18	6	D
Veronese	15	10	16	3	D
Albani	14	14	10	6	E
Caravaggio	6	6	16	0	E
Corregio	13	13	15	12	E
Domenichino	15	17	9	17	E
Guercino	18	10	10	4	E
Lanfranco	14	13	10	5	E
The Carraci	15	17	13	13	E
Durer	8	10	10	8	F
Holbein	9	10	16	13	F
Pourbus	4	15	6	6	F
Van Leyden	8	6	6	4	F
Diepenbeck	11	10	14	6	G
J. Jordaens	10	8	16	6	G
Otho Venius	13	14	10	10	G
Rembrandt	15	6	17	12	G

	Composition	Drawing	Colour	Expression	School
Rubens	18	13	17	17	G
Teniers	15	12	13	6	G
Van Dyck	15	10	17	13	G
Bourdon	10	8	8	4	H
Le Brun	16	16	8	16	H
Le Suer	15	15	4	15	H
Poussin	15	17	6	15	H

You could argue that the format of this dataset is typical of a **pattern recognition** problem: suppose that a newly discovered old master, say Patriziani, has a a set of scores (Composition=15, Drawing= 19, Colour=17, Expression=3), can we use the above 'Training Set' to assign this new painter to the correct School?
You could also reasonably argue that any analysis of the above data-set must **start** by suitable plots: this is what we will do in the practical classes. Try

```
attach(painters)
plot(Composition,Drawing, type="n")
text(Composition,Drawing, c("A","B","C","D","E","F","G","H")[School])
```

But here we will restrict ourselves to a much more specific problem (and this will turn out to give the generalization of the well-known 1-way anova).
The model: assume that we have independent observations from g different groups,

$$x_j^{(\nu)} \sim NID(\mu^{(\nu)}, V), \text{ for } j = 1, \ldots, n_\nu, \nu = 1, \ldots, g$$

where $\Sigma n_\nu = n$. We wish to test

$$H_0 : \mu^{(1)} = \ldots = \mu^{(g)} = \mu \text{ say, where } \mu, V \text{unknown}$$

against

$$H_1 : \mu^{(1)}, \ldots, \mu^{(g)}, V \text{ all unknown.}$$

We maximise the log-likelihood under each of H_0, H_1 respectively, in order to find the likelihood ratio criterion (which of course must turn out to be the matrix version of (ss between groups)/(ss within groups)).
Let S_ν be the sample covariance matrix for the νth group.
Now -2 loglikelihood for all n observations=

$$-2l(\mu^{(1)}, \ldots, \mu^{(g)}, V) \text{ say } = \Sigma_\nu n_\nu (\log|V| + tr(V^{-1}S_\nu) + (\bar{x}^{(\nu)} - \mu^{(\nu)})^T V^{-1} (\bar{x}^{(\nu)} - \mu^{(\nu)})$$

$$= n \log|V| + \Sigma_\nu n_\nu tr(V^{-1}S_\nu) + \Sigma_\nu n_\nu (\bar{x}^{(\nu)} - \mu^{(\nu)})^T V^{-1} (\bar{x}^{(\nu)} - \mu^{(\nu)}).$$

We have already done the hard work that enables us to minimise this expression.
Verify that this is minimised under H_1 by

$$\hat{\mu}^{(\nu)} = \bar{x}^{(\nu)}, \text{ for each } \nu, \text{ and } \hat{V} = (1/n)\Sigma n_\nu S_\nu = (1/n)W \text{ say,}$$

$$min_{H_1} - 2l(\mu^{(1)}, \ldots, \mu^{(g)}, V) = n \log|\hat{V}| + n \; tr(\hat{V}^{-1}\hat{V})) = n \log(\hat{V}| + np.$$

Now, under H_0, -2 loglikelihood $= -2l(\mu, V)$ say

$$= n \log|V| + \Sigma_\nu n_\nu tr(V^{-1}S_\nu) + \Sigma_\nu n_\nu (\bar{x}^{(\nu)} - \mu)^T V^{-1} (\bar{x}^{(\nu)} - \mu).$$

We may write the second Σ term as

$$\Sigma n_\nu (\bar{x}^{(\nu)} - \bar{x} + \bar{x} - \mu)^T V^{-1} (\bar{x}^{(\nu)} - \bar{x} + \bar{x} - \mu)$$

$$= \Sigma n_\nu (\bar{x}^{(\nu)} - \bar{x})^T V^{-1} (\bar{x}^{(\nu)} - \bar{x}) + \Sigma n_\nu (\bar{x} - \mu)^T V^{-1} (\bar{x} - \mu)$$

where we have defined $\bar{x} = \Sigma n_\nu \bar{x}^{(\nu)}/n$, the mean of all the observations.
Hence, under H_0, -2 loglikelihood is minimised with respect to μ, V by $\hat{\mu}^* = \bar{x}$, and

$$\hat{V}^* = \frac{1}{n}(\Sigma n_\nu S_\nu + \Sigma (\bar{x}^{(\nu)} - \bar{x})(\bar{x}^{(\nu)} - \bar{x})^T n_\nu)$$

$$= \frac{1}{n}(W + B) \text{say,}$$

where $W =$ 'within-groups ss', $B =$ 'between-groups ss', and

$$min_{H_0} - 2l(\mu, V) = n \log(1/n)|W + B| + np.$$

So we see that the l.r. test of H_0 against H_1 is to reject H_0 in favour of H_1 iff

$$\log|W + B|/|W| > \text{constant.}$$

(Compare this with the traditional 1-way anova.) Define $\Lambda = |W|/|W + B|$.
We simplify this expression, as follows.

Now W, B are symmetric matrices, with $W > 0, B \geq 0$. Put $W = LL^T$ say. Suppose

$$W^{-1}Bv_j = \lambda_j v_j.$$

Then

$$Bv_j = \lambda_j W v_j = \lambda_j L(L^T)v_j,$$

and so

$$L^{-1}B(L^{-1})^T(L^T v_j) = \lambda_j L^T v_j,$$

put $u_j = L^T v_j$, thus

$$L^{-1}B(L^{-1})^T u_j = \lambda_j u_j.$$

Now $L^{-1}B(L^{-1})^T$ is a symmetric matrix, ≥ 0, hence λ_j is real, and ≥ 0. Further,

$$\frac{|W+B|}{|W|} = \frac{|LL^T+B|}{|W|} = \frac{|L||L^T|det(I+L^{-1}B(L^T)^{-1})}{|W|}$$

$$= det(I+L^{-1}B(L^T)^{-1}) = \Pi(1+\lambda_j).$$

This final quantity has known distribution, under H_0.

2.4 Applications to Linear Discriminant Analysis.

```
 discrim()
lda() # in library(MASS)
```

Here is the problem: given data $(x_j^{(\nu)}, 1 \leq j \leq n_\nu, 1 \leq \nu \leq g)$ from our g different groups, choose a p-dimensional vector a such that the between-groups ss for $a^T x$ is as large as possible for a given value of the within-groups ss for $a^T x$,
ie maximise $a^T Ba$ subject to $a^T Wa = 1$. Clearly, the corresponding Lagrangian is

$$L(a, \lambda) = a^T Ba + \lambda(1 - a^T WA)$$

giving (by differentiating wrt a)

$$Ba = \lambda Wa,$$

ie we choose a as the eigen vector corresponding to the largest eigen value of $W^{-1}B$, equivalently, $L^T a$ is the eigen vector corresponding to the largest eigen value of $L^{-1}B(L^{-1})^T$.

An alternative approach is to use a Bayesian decision rule: this is what lies behind the R function lda(). Our explanation below follows Venables and Ripley, 2nd edition, p397 (with some minor modifications).

Suppose we have observations from a set of g classes, which correspond respectively to observations from pdfs $f_1(x), \ldots, f_c(x), \ldots, f_g(x)$. Initially we assume that all these pdf's are known. We take a new observation, x say, and we wish to assign it to the correct class out of $C_1, \ldots, C_g$, which have known prior probabilities $\pi_1, \ldots, \pi_g$, adding to 1. Clearly, by Bayes' theorem, we see that
$P(\text{new obsn belongs to class } c|x) \propto \pi_c f_c(x)$
and if we have a symmetric loss-function, then we should assign the new observation to class c if $\pi_c f_c(x)$ is the largest over the g classes, ie we assign to class c if Q_c is smallest, where

$$Q_c = -2log(f_c(x)) - 2log\pi_c.$$

Take the special case of $f_c(x)$ as the pdf of $N(\mu_c, V_c)$. Then it is easily checked that

$$Q_c = (x-\mu_c)^T {V_c}^{-1}(x-\mu_c) + logdet(V_c) - 2log\pi_c.$$

The quantity $(x-\mu_c)^T {V_c}^{-1}(x-\mu_c)$ is called the *Mahalanobis distance* of x from the class centre of C_c. Since the difference between any pair Q_c, Q_d say is a quadratic function of x, this method will give quadratic boundaries of the decision region for x, and the method is known as *quadratic discriminant analysis*.
Take the special case $V_c = V$ for all c. Then you can easily check that the quadratic rule given above simplifies to a *linear discriminant* rule: we assign x to class c if L_c is the largest, where we define

$$L_j = x^T V^{-1}\mu_j - \mu_j^T V^{-1}\mu_j/2 + log(\pi_j).$$

In real life, of course, we do not know $\mu_1, \ldots, \mu_g$ and we do not know V: so in practice we use the 'obvious' estimates for them, namely the within-class means to estimate the means, and we use W to estimate V.

Chapter 3

Principal components analysis.

(pca)‘

```
princomp()
```

Suppose our observation vector $X \sim N(\mu, V)$, so that for example, for $p = 2$, $\mu = (0, 0)^T$,

$$V = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$$

Then for the special case $\rho = .9$, the contours of the density function will be very elongated ellipses, and a random sample of 500 observations from this density function has the scatter plot given here as Figure 3.1. Using y_1, y_2 as the **principal axes** of the ellipse, we see that most of the variation in the data can be expected to be in the y_1 direction. Indeed, the 2-dimensional picture Figure 3.1 could be quite satisfactorily ‘collapsed’ into a 1-dimensional picture.
Clearly, if we had n points in p dimensions as our original data, it would be a great saving if we could adequately represent these n points in say just 2 or 3 dimensions.
Formally: with $X \sim N(\mu, V)$, our problem is to choose a direction a_1 say to maximise $var(a_1^T X)$, subject to $a_1^T a_1 = 1$,
ie to choose a_1 to maximise $a_1^T V a_1$ subject to $a_1^T a_1 = 1$.
This gives the Lagrangian

$$L(a_1, \lambda_1) = a_1^T V a_1 + \lambda_1(1 - a_1^T a_1).$$

$$\frac{\partial L}{\partial a_1} = 0 \text{ implies } V a_1 = \lambda_1 a_1$$

and hence a_1 is an eigenvector of V, corresponding to eigenvalue λ_1. Further

$$a_1^T V a_1 = \lambda_1 a_1^T a_1 = \lambda_1,$$

hence we should take λ_1 as the largest eigen value of V. Then the first principal component is said to be $a_1^T X$: it has variance λ_1.
Our next problem is to choose a_2 to maximise $var(a_2^T X)$ subject to $cov(a_1^T X, a_2^T X) = 0$ and $a_2^T a_2 = 1$.

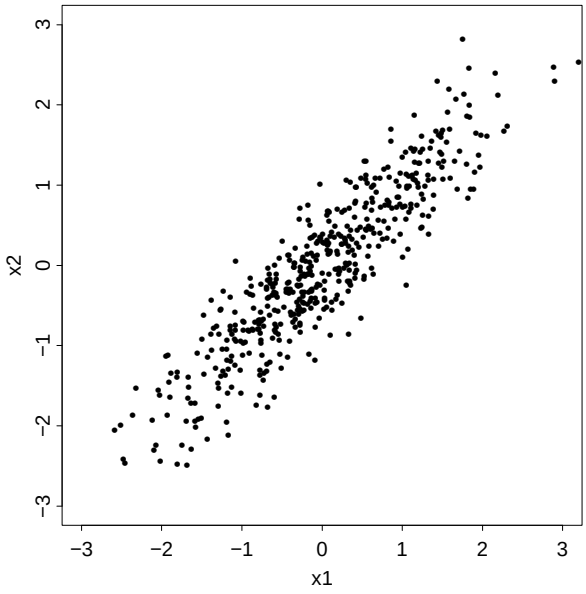

Figure 3.1: A random sample from a bivariate normal distribution

Now, using the fact that $Va_1 = \lambda_1 a_1$, we see that $cov(a_1^T X, a_2^T X) = 0$ is equivalent to the condition $a_2^T a_1 = 0$. Hence we take as the Lagrangian

$$L(a_2, \mu, \lambda_2) = a_2^T V a_2 + 2\mu a_2^T a_1 + \lambda_2(1 - a_2^T a_2).$$

Now find $\frac{\partial L}{\partial a_2}$ and apply the constraints to show that a_2 is the eigen vector of V corresponding to its second largest eigen value, λ_2.
And so on. Let us denote

$$\lambda_1 \geq \lambda_2 \geq \ldots \lambda_p (> 0)$$

as the eigen values of V, and let

$$a_1, \ldots, a_p$$

be the corresponding eigen vectors.
(Check that for the given example, with $p = 2$, the eigen -values are $1+\rho, 1-\rho$.)
Define $Y_i = a_i^T X$, these are the p principal components, and you can see that

$$Y_i \sim NID(a_i^T \mu, \lambda_i), \text{ for } i = 1, \ldots, p.$$

The practical application Of course, in practice the matrix V is unknown, so we must replace it by say S, its sample estimate. Suppose our original $n \times p$ data matrix is X, so that

$$X = \begin{pmatrix} x_1^T \\ \vdots \\ x_n^T \end{pmatrix}$$

corresponding to n points in p dimensions. Then

$$nS = (X^T X - n\bar{x}^T \bar{x})$$

(correcting apparent error in Venables and Ripley) where $\bar{x} = 1^T X/n$ is the row vector of the means of the p variables: we have already proved that S is is the mle of V. (Warning: you may find that $(n-1)$ is used as the divisor, in some software.) Here, for example, what the R function

```
princomp()
```

does is to choose a_1 to maximise $a_1^T S a_1$ subject to $a_1^T a_1 = 1$, obtaining

$$Sa_1 = \lambda_1 a_1,$$

λ_1 being the largest eigenvalue of S. Then, for example, $a_1^T x_1, \ldots, a_1^T x_n$ show the first principal component for each of the original n data points.
Interpretation of the principal components (where possible) is very important in practice.
The scree-plot
Plot $(\lambda_1 + \cdots + \lambda_i)/(\lambda_1 + \cdots + \lambda_p)$ against i. This gives us an informal guide as to how many components are needed for an acceptable representation of the original data. Clearly $(\lambda_1 + \cdots + \lambda_i)/(\lambda_1 + \cdots + \lambda_p)$ increases with i, and is 1 for $p = i$, but we may hope that 3 dimensions represents the overall picture satisfactorily, ie
$(\lambda_1 + \lambda_2 + \lambda_3)/(\lambda_1 + \cdots + \lambda_p) > 3/4$, say.
The difficulty of scaling
Suppose, eg x_i has 2 components, namely
x_{1i} a length, measured in feet, and x_{2i} a weight, measured in pounds.
Suppose that these variables have covariance matrix S.
We consider the effect of rescaling these variables, to inches and ounces respectively. (1 ft= 12 inches, 1 pound = 16 ounces)
The covariance matrix for these new units is say SS, and

$$SS = \begin{pmatrix} 12^2 S_{11} & 12 \times 16 S_{12} \\ 12 \times 16 S_{21} & 16^2 S_{22} \end{pmatrix}.$$

There is no simple relationship between the eigenvalues/vectors of S and those of SS.
So the 'picture' we get of the data by applying pca to SS might be quite different from what we found by pca on S.
Note, if one column of the original data matrix

$$\begin{pmatrix} x_1^T \\ \vdots \\ x_n^T \end{pmatrix}$$

has a particularly large sample variance, this column will tend to dominate the pca picture, although it may be scientifically uninteresting. For example

in a biological dataset, this column may simply correspond to the **size** of the animal.
For example, suppose our observation vector is X, and

$$X = lZ + \epsilon$$

where Z, ϵ are independent, with $Z \sim N_1(0, v)$ and $\epsilon_i \sim NID(0, \sigma_i^2)$, for $i = 1, \ldots, p$. Then

$$V = var(X) = ll^T v + diag(\sigma_1^2, \ldots, \sigma_p^2).$$

Suppose $v >> \sigma_1^2, \ldots, \sigma_p^2$. Then you will find that the first principal component of V is approximately $l^T X$; in other words Z (which may correspond to the **size** of the animal) dominates the pca 'picture'.
Further, if $\sigma_1^2 = \ldots = \sigma_p^2 = \sigma^2$, then

$$V = ll^T v + \sigma^2 I$$

and hence

$$Vl = l(l^T vl) + \sigma^2 l.$$

Suppose (wlog) that $l^T l = 1$. Then we see that l is an eigen-vector of V corresponding to eigen value $v + \sigma^2$. If u is any vector orthogonal to l, then

$$Vu = l(l^T u)v + \sigma^2 u,$$

hence $Vu = \sigma^2 u$, and so u is an eigen vector of V corresponding to eigenvalue σ^2. This is clearly an extreme example: we have constructed V to have as its eigenvalues $v + \sigma^2$ (once) and σ^2, repeated $p - 1$ times.
The standard fixup, if there is no 'natural scale' for each of the columns on the data matrix X, is to standardise so that each of the columns of this matrix has sample variance 1. Thus, instead of finding the eigen values of S, the sample covariance matrix, we find the eigen values of R, the sample correlation matrix. This has the following practical consequences:
i) $\Sigma\lambda_i = tr(R) = p$ since $R_{ii} = 1$ for each i.
ii) the original measurements (eg height, weight, price) are all given equal **weight** in the pca.
So this is quite a draconian 'correction'.
Factor analysis, which is a different technique, with a different model, attempts to remedy this problem, but at the same time introduces a whole new raft of difficulties.

`factanal()`

For this reason, we discuss factor analysis only very briefly, below. It is well covered by R, and includes some thoughtful warnings.
Formal definition of **Factor analysis**.
We follow the notation of Venables and Ripley, who give further details and an example of the technique.

The model, for a single underlying factor: Suppose that the observation vector X is given by

$$X = \mu + \lambda f + u$$

where μ is fixed, and λ is a fixed vector of 'loadings' and $f \sim N_1(0,1)$ and $u \sim N_p(0, \Psi)$ are independent, and Ψ is an unknown diagonal matrix. Our random sample consists of

$$X_i = \mu + \lambda f_i + u_i$$

for $i = 1, \ldots, n$. This gives $X_1, \ldots, X_n$ a rs from $N(\mu, V)$ where

$$V = \lambda\lambda^T + \Psi.$$

For $k < p$ common factors, we have

$$X_i = \mu + \Lambda f_i + u_i,$$

for $i = 1, \ldots, n$ with $f \sim N_k(0, I_k)$ and $u \sim N_p(0, \Psi)$ independent, with Λ a fixed, unknown matrix of loadings, and with Ψ is an unknown diagonal matrix as before, so that $X_1, \ldots, X_n$ is a rs from $N(\mu, V)$ where

$$V = \Lambda\Lambda^T + \Psi.$$

We have a basic problem of unidentifiability, since

$$X_i = \mu + \Lambda f_i + u_i$$

is the same model as

$$X_i = \mu + (\Lambda G)(G^T f_i) + u_i$$

for any orthogonal matrix G.
Choosing an 'appropriate' G is known as choosing a **rotation**. In the ml fit of the factor analysis model above, we choose Λ, Ψ (subject to a suitable constraint) to maximise

$$-tr(V^{-1}S) + \log|(V^{-1}S)|.$$

Note that the number of parameters in V is $p + p(p-1)/2 = p(p+1)/2$.
Define

$$s = p(p+1)/2 - [p(k+1) - k(k-1)/2] = (p-k)^2/2 + (p+k)/2$$

as the degrees of freedom of this ml problem. Then
s = number of parameters in V − (number of parameters in Λ, Ψ) (taking account of the rotational freedom in Λ, since only $\Lambda\Lambda^T$ is determined).
We assume $s \geq 0$ for a solution (otherwise we have non-identifiability).
If $s > 0$, then in effect, factor analysis 'chooses the scaling of the variables via $\hat{\Psi}$', whereas in pca, the user must choose the scaling.

Lastly, here is **another view of pca** on S, the sample covariance matrix, as the solution to a minimisation problem.
This is a 'data-analytic' rather than an mle approach.
Suppose we have observations $y_1, \ldots, y_n \epsilon R^p$, take $\Sigma y_i = 0$ for simplicity, so that $\bar{y} = 0$.
Take $k < p$ and consider the problem of finding the best-fitting k-dim linear subspace for $(y_1, \ldots, y_n)$, in the following sense:
take k orthonormal vectors $a_1, \ldots, a_k$ (ie such that $a_i^T a_j = \delta_{ij}$ for $i, j = 1, \ldots, k$) to minimise

$$\Sigma (y_i - Py_i)^T (y_i - Py_i)$$

where Py_i is the **projection** of y_i onto

$$\Omega = \ell(a_1, \ldots, a_k)$$

the linear subspace spanned by $a_1, \ldots, a_k$.
Solution.
Any orthonormal set $a_1, \ldots, a_k \epsilon R^p$ may be extended to

$$a_1, \ldots, a_k, \ldots, a_p$$

an orthonormal basis of R^P. Furthermore, any $y \epsilon R^p$ may then be rewritten as

$$y = \Sigma_1^p (y^T a_i) a_i$$

and then

$$Py = \Sigma_1^k (y^T a_i) a_i,$$

and so

$$y - Py = \Sigma_{k+1}^p (y^T a_i) a_i;$$

and

$$(y - Py)^T (y - Py) = \Sigma_{k+1}^p (y^T a_i)^2.$$

Hence, our problem is to choose $a_1, \ldots, a_k$ to minimise

$$\Sigma_{j=1}^n \Sigma_{i=k+1}^p (y_j^T a_i)^2.$$

But

$$\Sigma_{j=1}^n y_j^T y_j = \Sigma_{j=1}^n \Sigma_{i=1}^p (y_j^T a_i)^2$$

is fixed, so our problem is therefore to **maximise**

$$\Sigma_{j=1}^n \Sigma_{i=1}^k (y_j^T a_i)^2,$$

and this last term is

$$\Sigma_{i=1}^k a_i^T (\Sigma_{j=1}^n y_j y_j^T) a_i.$$

Thus our problem is to choose $a_1, \ldots, a_k$ subject to $a_i^T a_j = \delta_{ij}$ to maximise $\Sigma_1^k a_i^T S a_i$, where

$$S \propto \Sigma_1^n y_j y_j^T.$$

We have already shown how to solve this problem, at the beginning of this Chapter. The solution is to take a_1 as the eigenvector of S corresponding to the largest eigen value λ_1, and so on.

Note added September 2013. I have now come across the term
'multinomial PCA'
mainly in the context of **text-mining analysis** and **bioinformatics**. What has this to do with the pca described above?
One answer is that both techniques are concerned with 'reducing' a large data matrix. Non-negative Matrix Factorisation (NMF) decomposes a *positive* matrix, say V, into a product of non-negative factors, thus

$$V \approxeq W.H$$

where V, W, H are of orders $m \times n, m \times p, p \times n$, respectively, and for a useful representation $p << (m, n)$.
(Every element of V, W, H is non-negative.)
This is related to the problem of fitting a probability mixture distribution. Written in the context of what is rather grandly called 'Probabilistic Latent Semantic Analysis' (PLSA) this model is

$$P(w_i, d_j) = \sum_c P(c)P(w_i|c)P(d_j|c)$$

where w_i corresponds to word i, d_j corresponds to document j, and the summation runs from $c = 1, \ldots, K$. This will clearly give quite a hard maximum likelihood problem, but I believe at least one R package does exist.

Chapter 4

Cluster Analysis

Here we seek to put n objects into (disjoint) groups, using data on d variables.
The data consist of points $x_1, ..., x_n$ say, giving rise to an $n \times d$ data matrix.
x_i may consist of continuous, or discrete variables or a mixture of the two, eg "red hair, blue eyes, 1.78 m, 140 kg, 132 iq " and so on.
There are NO probability models involved in cluster analysis: in this sense the method is said to be 'data-analytic'.
We start by constructing from $x_1, ..., x_n$ the DISSIMILARITY matrix (d_{rs}) between all the individuals, or equivalently the SIMILARITY matrix, say (c_{rs}).
For example, we might take $d_{rs} = |x_r - x_s|$, Euclidean distance.
There are 3 types of clustering available
i) hierarchical clustering

`hclust()`

in which we form a dendrogram of clusters (like a tree, upside down) by grouping points into clusters, and then grouping the clusters themselves into bigger clusters, and so on, until all n points form one big cluster.
See Swiss cantons data-set as an example.
ii) we could partition the n points into a given number, eg 4, of non-overlapping clusters

`kmeans()`

iii) we could partition the n points into overlapping clusters.

How to construct the dissimilarity matrix
Take any 3 objects, say A, B, C.
Let $d(A, B)$ be the dissimilarity between A and B.
It is reasonable to require the following of the function $d(,)$.

$$d(A, B) \geq 0, d(A, B) = 0 \text{ iff } \; A = B, d(A, C) \leq d(A, B) + d(B, C).$$

The R function

`dist()`

produces, for n points, the $n(n-1)/2$ distances, eg with
Euclidean, $|x_r - x_s|$ as default
or 'manhattan' (also called city-block), ie $\Sigma_i |x_{ri} - x_{si}|$
or 'maximum', ie $max_i |x_{ri} - x_{si}|$.
It also allows for binary variables, for example if
$x_r = (1, 0, 0, 1, 1, 0, 0)$ and
$x_s = (1, 0, 0, 0, 0, 0, 0)$ then the simple matching coefficient gives
$d(x_r, x_s) = 1 - 5/7$ (this counts all matches)
but the Jaccard coefficient gives
$d(x_r, x_s) = 1 - 1/3$ (just one 'positive' match in the 3 places where matching matters).

What do we do with the distance matrix when we've got it ?
The R function *hclust*() has 3 possible methods, our problem being that we have to decide how to define the distance or dissimilarity between any 2 **clusters**.
We now describe the **agglomerative** technique used in hierarchical clustering.
Begin with n clusters, each consisting of just 1 point, and $(d_{rs}) = D$ say, a dissimilarity matrix, and a measure of dissimilarity between any 2 clusters, say $d(C_1, C_2)$ for clusters C_1, C_2.
Fuse the 2 nearest points into a single cluster.
Now we have $n - 1$ clusters.
Fuse the 2 nearest such clusters into a single cluster.
Now we have $n - 2$ clusters.
And so on. Finally we have one cluster containing all n objects.
This now gives us a hierarchical clustering, and for example we could sort the n objects into 3 groups by drawing a horizontal line across the resulting dendrogram.

The possible definitions for $d(C_1, C_2)$.
i) 'compact' (complete linkage) $d(C_1, C_2) = max\ d(i, j)$
ii) 'average' $d(C_1, C_2) = average\ d(i, j)$
iii) 'connected' (single linkage) $d(C_1, C_2) = min\ d(i, j)$.
(this one tends to produce long straggly clusters)
In all of i),ii),iii) we take i in C_1, j in C_2.
Warning: we repeat, this method is entirely 'data-analytic'. We can't reasonably ask for significance tests: eg, 'do 3 clusters explain the data significantly better than 2?'
(Simulation, or using random subsets of the whole data-set, may go some way towards answering this last type of question.)
Here is a small scale example, from a subset of recent MPhil students. Each student was asked to reply Yes or No (coded here as $1, 0$ respectively) to each of 10 (rather boring, but not embarrassing personal) questions. These were

Do you eat eggs? Do you eat meat? Do you drink coffee? Do you drink beer? Are you a UK resident? Are you a Cambridge Graduate? Are you Female? Do you play sports? Are you a car driver? Are you Left-handed? The answers for this subset of students form the dataset given below.

	eggs	meat	coffee	beer	UKres	Cantab	Fem	sports	driver	Left.h
Philip	1	1	1	0	1	1	0	0	1	1
Chad	1	1	1	0	0	0	0	1	1	0
Graham	1	1	1	1	1	1	0	1	1	0
Tim	1	1	1	1	1	1	0	1	0	0
Mark	1	1	0	1	1	1	0	0	0	1
Juliet	0	1	1	0	1	0	1	0	0	0
Garfield	0	1	1	1	0	0	0	1	0	0
Nicolas	1	1	1	1	0	0	0	1	1	0
Frederic	1	1	0	1	0	0	0	1	1	0
John	1	1	1	1	0	0	0	0	1	0
Sauli	1	1	0	0	1	0	0	1	1	0
Fred	1	1	1	0	0	0	0	1	0	0
Gbenga	1	1	1	0	0	0	0	1	0	0

```
# taking a as the data-matrix above, we compute d, the appropriate
# set of 14*13/2 interpoint distances, and present the corresponding
# 14 by 14 distance matrix
> d = dist(a, metric="binary") ; round(dist2full(d), 2)
```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]	[,11]	[,12]	[,13]
[1,]	0.00	0.50	0.33	0.44	0.38	0.62	0.78	0.56	0.67	0.50	0.50	0.62	0.62
[2,]	0.50	0.00	0.38	0.50	0.78	0.71	0.50	0.17	0.33	0.33	0.33	0.20	0.20
[3,]	0.33	0.38	0.00	0.12	0.44	0.67	0.50	0.25	0.38	0.38	0.38	0.50	0.50
[4,]	0.44	0.50	0.12	0.00	0.38	0.62	0.43	0.38	0.50	0.50	0.50	0.43	0.43
[5,]	0.38	0.78	0.44	0.38	0.00	0.75	0.75	0.67	0.62	0.62	0.62	0.75	0.75
[6,]	0.62	0.71	0.67	0.62	0.75	0.00	0.67	0.75	0.88	0.71	0.71	0.67	0.67
[7,]	0.78	0.50	0.50	0.43	0.75	0.67	0.00	0.33	0.50	0.50	0.71	0.40	0.40
[8,]	0.56	0.17	0.25	0.38	0.67	0.75	0.33	0.00	0.17	0.17	0.43	0.33	0.33
[9,]	0.67	0.33	0.38	0.50	0.62	0.88	0.50	0.17	0.00	0.33	0.33	0.50	0.50
[10,]	0.50	0.33	0.38	0.50	0.62	0.71	0.50	0.17	0.33	0.00	0.57	0.50	0.50
[11,]	0.50	0.33	0.38	0.50	0.62	0.71	0.71	0.43	0.33	0.57	0.00	0.50	0.50
[12,]	0.62	0.20	0.50	0.43	0.75	0.67	0.40	0.33	0.50	0.50	0.50	0.00	0.00
[13,]	0.62	0.20	0.50	0.43	0.75	0.67	0.40	0.33	0.50	0.50	0.50	0.00	0.00

```
> h =  hclust(d, method="compact"); h
```

and here is a resulting dendrogram (obtained using the method "Compact") as Figure 4.1.

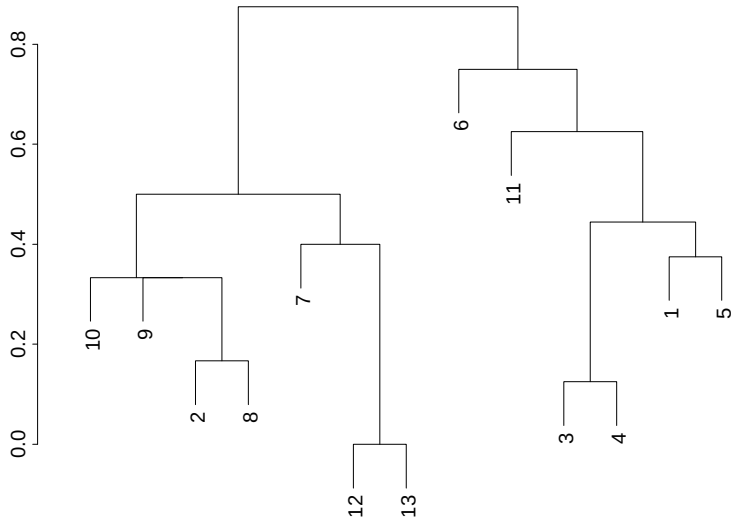

Figure 4.1: Example of a dendrogram

Chapter 5

Tree-based methods, ie decision trees/ classification trees

(Regression trees are also a possibility, but we do not discuss them here.)
We take as our example the autolander data (These data are taken from D.Michie, 1989, and are discussed in Venables and Ripley).
For the shuttle autolander data, we seek to base our decision about the desired level of a particular **factor**, here "use", which has possible values "auto" and "noauto", on the levels of certain other "explanatory" variables, here "stability", "error", "wind", "visibility", ... (As it happens, all the variables in this problem are factors, with two or more levels, but this need not be the case for this method to work.)
We show how to construct a *decision tree*, or classification tree, using the R library

```
rpart()
```

or the previously favoured method

```
tree()
post.tree(shuttle.tree, file="pretty")
```

Here's how it works. The decision tree provides a probability model: at each **node** of the classification tree, we have a probability distribution (p_{ik}) over the classes k, here k has values just $1, 2$, corresponding to "auto", "noauto" respectively. (Note that R works alphabetically, by default) and

$$\Sigma_k p_{ik} = 1,$$

for each node i.
The partition is given by the **leaves** of the tree, which are also known as the **terminal nodes**, denoted as * in the R output. (Confusingly, the convention is that the tree grows upside-down, ie with its root at the top of the page.)
In the current example, each of the total of $n = 256$ cases in the **training set** is assigned to a leaf, and so at each leaf we have a sample, say (n_{ik}), from a multinomial distribution, say $Mn(n_i, (p_{ik}))$ (in the shuttle example these are actually binomial distributions.)

Condition on the observed variables (x_i) in the training set (ie the observed 'covariate' values).
Then we know the numbers (n_i) of cases assigned to each node of the tree, and in particular to each of the leaves. The conditional likelihood is thus

$$\propto \Pi_{cases\ j}\ \ p_{[j]y_j}$$

where $[j]$ is the leaf assigned to case j. Hence this conditional likelihood is

$$\propto \Pi_{leaves\ i}\ \ \Pi_{classes\ k}\ \ p_{ik}^{n_{ik}}$$

and we can define a **deviance** for the tree as

$$D = \Sigma_{leaves\ i} D_i,$$

where

$$D_i = -2\Sigma_k n_{ik} \log p_{ik}.$$

Note that a perfect classification would result in each (p_{i1}, p_{i2}) as a $(1, 0)$ or $(0, 1)$ (zero-entropy) distribution, and in this case each $D_i = 0$ (recall that $0 \log 0 = 0$) and so $D = 0$.
Our general aim is to construct a tree with D as small as possible, but without too many leaves.
How does the algorithm decide how to "split" a given node?
Consider splitting node s into nodes t, u say.

$$t\ \ s\ \ u$$

This will change the probability distribution/model within node s. The total reduction in deviance for the tree will be

$$D_s - D_t - D_u = 2\Sigma(n_{tk} \log(p_{tk}/p_{sk}) + n_{uk} \log(p_{uk}/p_{sk})).$$

We do not actually **know** the probabilities (p_{tk}) etc, so the best we can do is to estimate them from the sample proportions in the split node, thus obtaining

$$\hat{p}_{tk} = n_{tk}/n_t, \quad \hat{p}_{uk} = n_{uk}/n_u,$$

and

$$\hat{p}_{sk} = (n_t \hat{p}_{tk} + n_u \hat{p}_{uk})/n_s = n_{sk}/n_s,$$

and correspondingly, the estimated reduction in deviance is

$$\hat{D}_s - \hat{D}_t - \hat{D}_u = 2\Sigma(n_{tk} \log(\hat{p}_{tk}/\hat{p}_{sk}) + n_{uk} \log(\hat{p}_{uk}/\hat{p}_{sk}).$$

This gives us a measure of the **value** of a split at node s.
NB: since this depends on n_s, n_t, n_u, there will be more value in splitting leaves with a larger number of cases.
The algorithm for the tree construction is designed to take the MAXIMUM reduction in deviance over all allowed splits of all leaves, to choose the next split.
Usually, tree construction continues until

either, the number of cases reaching each leaf is small enough (default, < 10 in tree())
or, a leaf is homogeneous enough (eg, its deviance is $< (1/100)$ of deviance of root node).

Remarks

1. In classifying new 'cases', missing values of some $x_1, x_2, \ldots$ values can easily be handled (unlike, say, in logistic regression, where we may need to infer the missing covariate values in order to find a corresponding 'fitted value').
(For example, in the final tree construction, it may turn out that x_1, eg wind speed, is not even used.)
2. 'Cutting trees down to size'
The function

```
tree()
```

may produce some 'useless' nodes. The function

```
prune.tree()
```

will prune the tree to something more useful, eg by reducing
deviance $+\alpha$ size .
thus, we have a tradeoff between the overall **cost** and the complexity (ie number of terminal nodes). (The idea is similar to that of the use of the AIC in regression models.) This is not explained further here, but is best understood by experimenting

```
> library(MASS) ; library(rpart)
>
>shuttle[120:130,]
    stability error sign wind   magn vis    use
120      stab    XL   nn tail    Out  no   auto
121      stab    MM   pp head    Out  no   auto
122      stab    MM   pp tail    Out  no   auto
123      stab    MM   nn head    Out  no   auto
124      stab    MM   nn tail    Out  no   auto
125      stab    SS   pp head    Out  no   auto
126      stab    SS   pp tail    Out  no   auto
127      stab    SS   nn head    Out  no   auto
128      stab    SS   nn tail    Out  no   auto
129     xstab    LX   pp head  Light yes noauto
130     xstab    LX   pp head Medium yes noauto
> table(use)
   auto noauto
    145  111

>fgl.rp = rpart(use ~ ., shuttle, cp = 0.001)
> fgl.rp
```

```
n= 256

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 256 111 auto (0.5664062 0.4335938)
   2) vis=no 128   0 auto (1.0000000 0.0000000) *
   3) vis=yes 128  17 noauto (0.1328125 0.8671875)
     6) error=SS 32  12 noauto (0.3750000 0.6250000)
      12) stability=stab 16   4 auto (0.7500000 0.2500000) *
      13) stability=xstab 16   0 noauto (0.0000000 1.0000000) *
     7) error=LX,MM,XL 96   5 noauto (0.0520833 0.9479167) *
node), split, n, deviance, yval, (yprob)
      * denotes terminal node
> plot(fgl.rp, uniform=T)
> text(fgl.rp, use.n = T)
 # see graph attached

 # here's another way,
>shuttle.tree = tree(use ~ ., shuttle); shuttle.tree
```

check that the 'root deviance' is

$$350.4 = -2[145\log(145/256) + 111\log(111/256)]$$

```
  1) root 256 350.400 auto ( 0.5664 0.4336 )
    2) vis:no 128   0.000 auto ( 1.0000 0.0000 ) *
    3) vis:yes 128 100.300 noauto ( 0.1328 0.8672 )
      6) stability:stab 64  74.090 noauto ( 0.2656 0.7344 )
       12) error:MM,SS 32  44.240 auto ( 0.5312 0.4688 )
         24) magn:Out 8   0.000 noauto ( 0.0000 1.0000 ) *
         25) magn:Light,Medium,Strong 24  28.970 auto ( 0.7083 0.2917 )
           50) error:MM 12  16.300 noauto ( 0.4167 0.5833 )
            100) sign:nn 6   0.000 noauto ( 0.0000 1.0000 ) *
            101) sign:pp 6   5.407 auto ( 0.8333 0.1667 ) *
           51) error:SS 12   0.000 auto ( 1.0000 0.0000 ) *
       13) error:LX,XL 32   0.000 noauto ( 0.0000 1.0000 ) *
      7) stability:xstab 64   0.000 noauto ( 0.0000 1.0000 ) *

Classification tree:
tree(formula = use ~ ., data = shuttle)
Variables actually used in tree construction:
[1] "vis"       "stability" "error"     "magn"      "sign"
Number of terminal nodes:  7
Residual mean deviance:  0.02171 = 5.407 / 249
Misclassification error rate: 0.003906 = 1 / 256
tree.rp = rpart(use ~., shuttle)
```

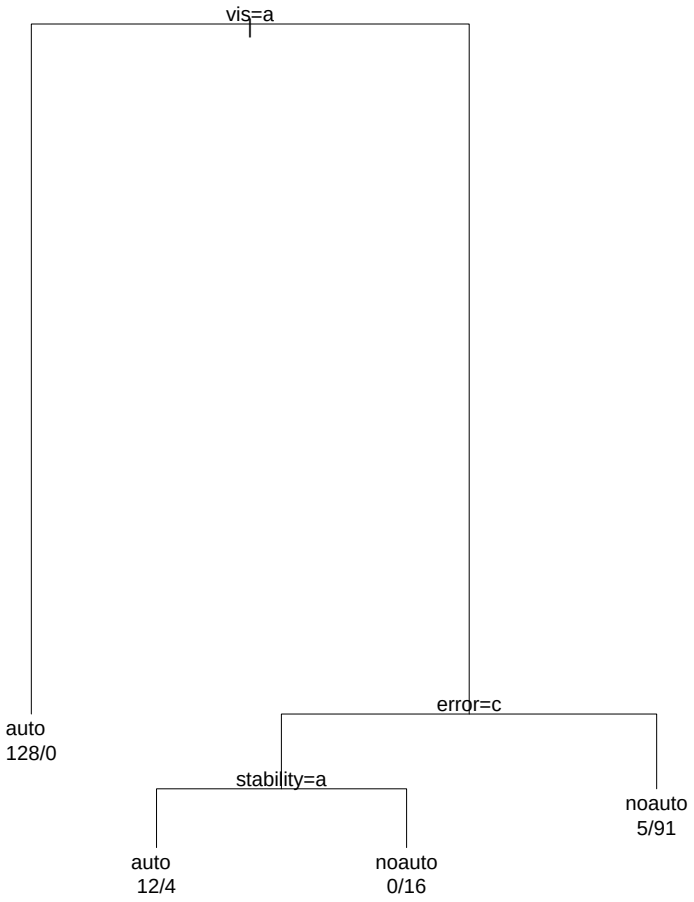

Figure 5.1: Tree for shuttle data drawn by rpart()

```
plot(tree.rp, compress=T) ; text(tree.rp, use.n=T)
```

and here is the graph drawn by rpart()

Chapter 6

Classical Multidimensional Scaling

Let $D = (\delta_{rs})$ be our dissimilarity/distance matrix, computed from given data points $x_1, \ldots, x_n$ in R^d, for example by

```
dist(X,metric="binary")
```

Take p given, assume $p < d$.
When does a given D correspond to a configuration $y_1, \ldots, y_n$ in R^p, in terms of Euclidean distances?
Notation and definitions
Given D, define $A = (a_{rs})$, where $a_{rs} = -(1/2)\delta_{rs}^2$, and define $B = (b_{rs})$, where

$$b_{rs} = a_{rs} - \bar{a}_{r+} - \bar{a}_{+s} + \bar{a}_{++}$$

where $\bar{a}_{r+} = (1/n)\Sigma_s a_{rs}$, etc.
Thus

$$B = (I_n - (1/n)11^T)A(I_n - (1/n)11^T)$$

as you can check.
We say that D is Euclidean if there exists a p-dimensional configuration of points $y_1, \ldots, y_n$ for some p, such that $\delta_{rs} = |y_r - y_s|$ for all r, s.
Theorem.
Given the matrix D of interpoint distances, then D is Euclidean iff B is positive-semidefinite.
(the proof follows Seber, 1984, p236)
Proof
Suppose D corresponds to the configuration of points $y_1, \ldots, y_n$ in R^p and

$$-2a_{rs} = \delta_{rs}^2 = (y_r - y_s)^T(y_r - y_s).$$

Hence, as you may check,

$$b_{rs} = a_{rs} - \bar{a}_{r+} - \bar{a}_{+s} + \bar{a}_{++} = (y_r - \bar{y})^T(y_s - \bar{y}).$$

Define the matrix $\tilde{X}$ by

$$\tilde{X}^T = (y_1 - \bar{y} : y_2 - \bar{y} : .... : y_n - \bar{y}).$$

Hence we see that

$$B = \tilde{X}\tilde{X}^T$$

and clearly $\tilde{X}\tilde{X}^T \geq 0$ (just look at $u^T\tilde{X}\tilde{X}^T u$ for any u).
Hence D Euclidean implies B positive semidefinite.
b) Conversely, given D, with A, B defined as above, suppose that B is positive semidefinite, of rank p. Then B has eigen-values say $\gamma_1 \geq \gamma_2 \geq \ldots \gamma_p > 0$, with all remaining eigen-values $= 0$, and there exists an orthonormal matrix V, constructed as usual from the eigen-vectors of B such that

$$V^T B V = \begin{pmatrix} \Gamma & 0 \\ 0 & 0 \end{pmatrix}$$

where the matrix Γ is diagonal, with diagonal entries $\gamma_1, \ldots, \gamma_p$. Thus,

$$B = V \begin{pmatrix} \Gamma & 0 \\ 0 & 0 \end{pmatrix} V^T.$$

Now partition V as $V = (V_1 : V_2)$ so that the columns of V_1 are the first p eigen vectors of B. Then you can see that

$$B = \begin{pmatrix} V_1 & V_2 \end{pmatrix} \begin{pmatrix} \Gamma & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix}$$

and hence

$$B = V_1 \Gamma V_1^T.$$

This last equation may be rewritten as

$$B = (V_1\Gamma^{1/2})(V_1\Gamma^{1/2})^T = YY^T$$

say, where we have defined Y as $V_1\Gamma^{1/2}$, thus Y is an $n \times p$ matrix.
Define $y_1^T, \ldots, y_n^T$ as the rows of the matrix Y, thus $y_1, \ldots, y_n$ are vectors in R^p, and you may check that since $B = YY^T$, it follows that with $B = (b_{rs})$,

$$b_{rs} = y_r^T y_s.$$

hence

$$\begin{aligned} |y_r - y_s|^2 &= y_r^T y_r - 2y_r^T y_s + y_s^T y_s = b_{rr} - 2b_{rs} + b_{ss} \\ &= a_{rr} + a_{ss} - 2a_{rs} = \delta_{rs}^2 \end{aligned}$$

since $a_{rr} = a_{ss} = 0$. Thus the y_i give the required configuration, and so we see that D is Euclidean.
This concludes the proof. It also indicates to us how we may use, for example, the first 2 eigen vectors of B to **construct** the best 2-dimensional configuration $y_i, i = 1, \ldots, n$ corresponding to a given 'distance' matrix D, as in

```
cmdscale()
```

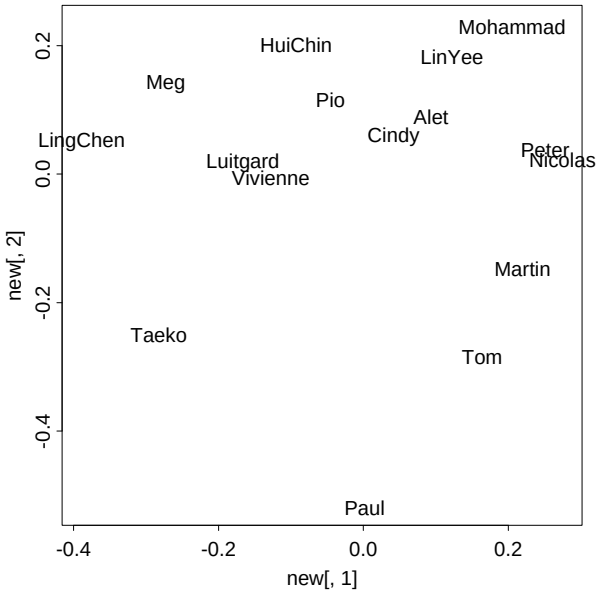

Figure 6.1: Classical Multidimensional Scaling for the students of Lent 2003

An example plot is given as Figure 6.1.

Note the classical solution is not unique: a shift of origin and a rotation or reflection of axes does not change interpoint distances.
Observe (i) Given (δ_{rs}), our original distance/dissimilarity matrix, and $(y_1, \ldots, y_n)$ our derived points (eg forced into R^2) with interpoint distances (d_{rs}) say, then

$$\Sigma_{r \neq s}(d_{rs} - \delta_{rs})^2 / \Sigma_{r \neq s}\delta_{rs}^2 = E(d, \delta)$$

is a measure of the 'stress' involved in forcing our original n points into a 2-dimensional configuration.
(ii) If (δ_{rs}) has actually been derived as an exact Euclidean distance matrix from the original points $x_1, \ldots, x_n$, then

```
cmdscale(D, scale=2)
```

will give us (exactly) these n points, plotted by their first 2 principal component 'scores'.
Thus if (δ_{rs}) is just a Euclidean distance matrix, we do not gain any extra information by using

```
cmdscale()
```

as compared with principal components on the sample covariance matrix.

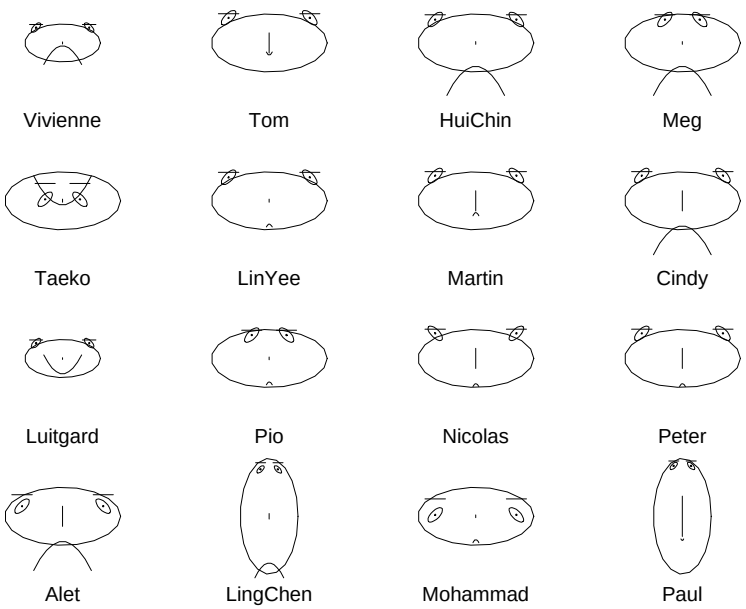

Figure 6.2: Students' Faces

Finally, Chernoff's faces for the class of Lent 2003, or how to insult your students. Chernoff's faces represents the $n \times p$ data matrix ($p \leq 15$), by n faces, of different features, eg size of face, shape of face, length of nose, etc according to the elements of the p columns. Warning: appearances can be deceptive: you will get a very different picture by just changing the *order* of the columns.

Figures 6.1, and 6.2 were obtained from the data from the class of Lent 2003, which is given below.

```
 MPhil/Part III, app mult. analysis, Feb 2003

.....................................................................
          eggs  meat  coffee  beer  UKres  Cantab  Fem  sports driver Left-h
Vivienne  y     n     y       n     y      n       y    y      y      n
Taeko     y     y     y       n     y      y       y    n      n      n
Luitgard  y     n     y       n     n      y       y    y      y      n
Alet      y     y     y       y     n      n       y    n      y      n
Tom       y     y     y       y     y      y       n    y      y      n
LinYee    y     y     y       n     n      n       n    y      y      n
Pio       y     y     y       n     n      n       n    y      n      n
LingChen  y     y     n       n     n      n       y    y      n      n
HuiChin   y     y     y       n     n      n       y    y      y      n
Martin    y     y     y       y     y      n       n    y      y      n
Nicolas   y     y     y       y     n      n       n    y      y      y
Mohammad  y     y     y       n     n      n       n    n      y      n
Meg       y     y     y       n     n      n       y    y      n      n
Cindy     y     y     y       y     n      n       y    y      y      n
```

```
Peter     y    y     y     y     n     n     n     y     y     n
Paul      y    y     n     y     y     y     n     y     n     n
```

Note, the first column turns out to be unhelpful, so you may prefer to omit it before trying, eg
dist() for use with hclust() or cmdscale()
The above data set is of course based on rather trivial questions.
By way of complete contrast, here is a data set from The Independent, Feb 13,2002 on 'Countries with poor human rights records where firms with British links do business'. It occurs under the headline
CORPORATE RISK: COUNTRIES WITH A BRITISH CONNECTION.

```
             1  2  3  4  5  6  7  8  9  10 11 12 13 14
SaudiArabia  1  0  0  0  0  1  0  1  0   0  1  1  0  1
Turkey       1  0  1  0  1  1  0  0  0   1  0  1  0  1
Russia       1  0  1  0  1  1  1  0  0   0  0  1  0  1
China        1  1  1  0  1  1  1  0  0   0  0  1  0  1
Philippines  1  1  1  0  0  0  0  0  1   0  0  1  1  0
Indonesia    1  1  1  0  0  1  1  1  0   0  0  1  0  0
India        1  0  1  0  1  0  0  1  1   0  1  1  0  0
Nigeria      0  0  1  0  0  0  1  0  0   0  0  1  1  0
Brazil       1  0  1  1  1  0  1  0  0   1  0  1  0  0
Colombia     1  1  1  1  1  0  0  0  0   1  0  1  0  0
Mexico       0  1  1  0  0  1  0  0  0   0  0  1  0  1
```

```
   Key to the questions (1 for yes, 0 for no)
Violation types occurring in the countries listed
1 Torture
2 'Disappearance'
3 Extra-judicial killing
4 Hostage taking
5 Harassment of human rights defenders
6 Denial of freedom of assembly & association
7 Forced labour
8 Bonded labour
9 Bonded child labour
10 Forcible relocation
11 Systematic denial of women's rights
12 Arbitrary arrest and detention
13 Forced child labour
14 Denial of freedom of expression
```

What happens when you apply

```
cmdscale()
```

to obtain a 2-dimensional picture of these 11 countries?

Chapter 7

Applied Multivariate Analysis Exercises

by P.M.E.Altham, Statistical Laboratory, University of Cambridge.
See also recent MPhil/Part III examination papers for this course.
The questions below marked with a * are harder, and represent interesting material that I probably will not have time to explain in lectures.

0. (i) Given the $p \times p$ positive-definite symmetric matrix V, with eigen-values $\lambda_1, \ldots, \lambda_p$ say, we may write V as

$$V = \sum \lambda_\nu u_\nu {u_\nu}^T,$$

equivalently

$$V = U \Lambda U^T$$

where U is a $p \times p$ orthonormal matrix, and Λ is the $p \times p$ diagonal matrix with diagonal elements λ_ν. Hence show

$$\text{trace}(V) = \sum \lambda_\nu, \quad \det(V) = \Pi \lambda_\nu \ .$$

0. (ii) Given S, V both $p \times p$ positive-definite symmetric matrices, show that the eigen values of $V^{-1}S$ are real and positive.
0. (iii) Suppose that $Y \sim N(0, V)$ and that A is a symmetric matrix, of the same dimensions as V. Prove that

$$E(Y^T A Y) = tr(AV).$$

1.(i) Given a random sample of 2-dimensional vectors $x_1, \cdots, x_n$ from a bivariate normal distribution with correlation coefficient ρ (and the other 4 parameters unknown) show, elegantly, that the distribution of the sample correlation coefficient depends only on $\rho,\ n$.
1.(ii) Given a random sample of vectors $x_1, \cdots, x_n$ from $N(\mu, V)$, show, elegantly, that the distribution of

$$(\bar{X} - \mu)^T S^{-1} (\bar{X} - \mu)$$

is free of μ, V (with the usual notation for $\bar{X}, S$).
1.(iii) Suppose X is p-variate normal, with $E(X_i) = \mu$ for all i, and $var(X_i) = \sigma^2$ for all i, $cov(X_i, X_j) = \rho\sigma^2$ for all $i \neq j$.
Show that this distribution is what we get if we write

$$X_i = Y + \epsilon_i, \text{ for all i}$$

where $Y, \epsilon_1, \ldots, \epsilon_p$ are NID random variables, with

$$E(Y) = \mu, \ var(Y) = v_1, E(\epsilon_i) = 0, \ var(\epsilon_i) = v_2$$

where v_1, v_2 are suitable functions of σ^2, ρ.
Hence show that $\sum(X_i - \bar{X})^2/\sigma^2(1-\rho)$ has the chi-squared distribution with $p-1$ degrees of freedom.
2. **The distribution of the sample correlation coefficient** r_n**.** You may assume that for $\rho = 0$,

$$\frac{r_n\sqrt{(n-2)}}{\sqrt{(1-r_n^2)}} \sim t_{n-2}.$$

This enables us to do an exact test of $H_0 : \rho = 0$.

Further, for large n

$$r_n \sim N(\rho, v(\rho)/n)$$

where

$$v(\rho) = (1-\rho^2)^2.$$

Show that if we define the transformation $g(\cdot)$ by

$$2g(\rho) = \log(1+\rho)/(1-\rho)$$

then, for large n,

$$g(r_n) \sim N(g(\rho), 1/n)$$

hence having variance free of ρ. Hence show how to construct a large sample confidence-interval for ρ.
Note: $g(\rho) = tanh^{-1}(\rho)$. This is also called Fisher's z-transformation.
3. Let $y_1, \cdots, y_n$ be a random sample from the p-variate normal distribution $N(\mu, V)$. Construct the likelihood ratio test of the null hypothesis

$H_0 : V$ is a diagonal matrix.

Answer:
The l-r criterion is: reject H_0 if $logdet(R) <$ constant, where R is the sample-correlation matrix.
4. Suppose $Y_\nu = U + Z_\nu$ for $\nu = 1, \cdots, p$, where

$$U \sim N(0, v_0) \text{ and } Z_1, \cdots Z_p \sim N(0, v_1)$$

and $U, Z_1, \cdots, Z_p$ are all independent. Show that the covariance matrix of the vector Y is say, V, where

$$V = \sigma^2((1-\rho)I + \rho 11^T)$$

for suitably defined σ^2, ρ. Write

$$V^{-1} = aI + b11^T.$$

Find the eigen values of V^{-1} in terms of $a,\ b$.
5. Given a random sample $y_1, \cdots, y_n$ from $N(\mu, V)$, with V as in question 4, write down the log-likelihood as a function of μ, a and b, and maximise it with respect to these parameters. Hence write down the mle of ρ.
6. Whittaker, 'Graphical Methods in Applied Multivariate Statistics' (1990) discusses a dataset for five different mathematics exams for 88 students, for which the sample covariance matrix is

```
mech  302.29
vect  125.78  170.88
alg   100.43   84.19  111.60
anal  105.07   93.60  110.84  217.88
stat  116.07   97.89  120.49  153.77 294.37
```

Use R to compute the corresponding sample correlation matrix, and discuss.

By inverting the covariance matrix given above, show that

$$var(mech|\ remaining\ 4\ variables) = 0.62 * var(mech),$$

Can you interpret this in terms of prediction? Show also that

$$cor(stat, anal|remaining\ 3\ variables) = 0.25.$$

Note added March 2009: if we use R, we can make use of G.Marchetti's *ggm* library to do the hard work for us, as follows

```
> library(ggm) # for Marchetti's library
> data(marks)
> S = var(marks) # here is the sample covariance matrix
> round(S,2)
           mechanics vectors algebra analysis statistics
mechanics     305.69  127.04  101.47   106.32     117.49
vectors       127.04  172.84   85.16    94.67      99.01
algebra       101.47   85.16  112.89   112.11     121.87
analysis      106.32   94.67  112.11   220.38     155.54
statistics    117.49   99.01  121.87   155.54     297.76


# and here is the sample partial correlation matrix
> round(parcor(S),2)
           mechanics vectors algebra analysis statistics
mechanics       1.00    0.33    0.23     0.00       0.03
vectors         0.33    1.00    0.28     0.08       0.02
algebra         0.23    0.28    1.00     0.43       0.36
analysis        0.00    0.08    0.43     1.00       0.25
statistics      0.03    0.02    0.36     0.25       1.00
```

7. Simulate from the bivariate normal to verify Sheppard's formula

$$Pr(X_1 > 0, X_2 > 0) = \frac{1}{4} + \frac{1}{2\pi} sin^{-1}(\rho)$$

where (X_1, X_2) have the bivariate normal distribution, $E(X_i) = 0$, $var(X_i) = 1$ and $\text{cor}(X_1, X_2) = \rho$.

8. * **Canonical correlations**

Suppose X, Y are vectors of dimension p, q respectively, and $Z^T = (X^T, Y^T)$ has mean vector 0, covariance matrix V. We consider the problem:
choose a, b to maximise the correlation between $a^T X, b^T Y$, equivalently, to maximise

$$\rho = a^T V_{12} b / \sqrt{a^T V_{11} a \; b^T V_{22} b},$$

where V has been partitioned in the obvious way. This is equivalent to maximising $a^T V_{12} b$ subject to $a^T V_{11} a = 1$ and $b^T V_{22} b = 1$. By considering the corresponding Lagrangian, show that a is a solution of the equation

$$V_{12} V_{22}^{-1} V_{21} a - \lambda \mu V_{11} a = 0.$$

We could use this technique to relate one set of variables to another set of variables. Or, as in classification, we could consider one set of variables as defining membership or otherwise of each of g groups, and the other set as our original observations, eg length, height, weight etc. Then our problem is to see what linear function of the original variables best discriminates into groups. This is one way of deriving standard *discriminant analysis*, nowadays seen as part of the *pattern recognition* set of techniques.

9. An example of a classification tree (taken from Venables and Ripley, 1999, p321.)

Fisher's Iris data-set consists of 50 observations on each of 3 species, here labelled "s", "c" and "v". For each of the 150 flowers, the Petal Length, Petal Width, Sepal Length and Sepal Width are measured. Can we predict the Species from these 4 measurements? Experiment with the following:

```
ird <- data.frame(rbind(iris[,  , 1], iris[,  , 2], iris[,  , 3]),
Species <- c(
       rep("s", 50), rep("c", 50), rep("v", 50)))
is.factor(Species) # procedure only makes sense if Species is a factor
ir.tr <- tree(Species ~ .,ird); summary(ir.tr) ; ir.tr
```

The resulting tree has 6 terminal nodes, and an error rate of 0.0267.

10.* This question introduces you to the idea of **Ridge Regression**, following the approach of Hastie, Tibshirani and Friedman 'The elements of Statistical Learning', 2001, p61.
Consider the model

$$y = X\beta + \epsilon.$$

We already know very well how to do ordinary least-squares regression for this problem. But ridge regression 'shrinks' the estimates of β by imposing a penalty on the total size $\beta^T \beta$.

For simplicity we also remove the 'intercept' from the problem, thus assume that $y^T 1 = 0$ and that the matrix X has been centered and scaled, so that, for each j, the vector (x_{ij}) has mean 0 and variance 1.
Now choose β to minimise $(y - X\beta)^T(y - X\beta)$ subject to $\beta^T\beta \leq s$: you should be able to show that this gives

$$\hat{\beta}^{ridge} = (X^T X + \lambda I)^{-1} X^T y,$$

where λ corresponds to the Lagrange multiplier. Thus $\lambda(\geq 0)$ is a complexity parameter that controls the amount of 'shrinkage'.
Assume further that the $n \times p$ matrix X has singular value decomposition

$$X = UDV^T,$$

where U, V are $n \times p$ and $p \times p$ orthogonal matrices, with the columns of U spanning the column space of X, and the columns of V spanning its row space, and D is a $p \times p$ diagonal matrix, with diagonal entries $d_1 \geq d_2 \geq \ldots \geq d_p \geq 0$. Show that, for the ordinary LS estimator,

$$\hat{Y} = X\hat{\beta}^{ls} = UU^T y,$$

while for the ridge regression estimator,

$$\hat{Y}^{ridge} = X\hat{\beta}^{ridge} = \Sigma \frac{d_j^2}{d_j^2 + \lambda} u_j u_j^T y,$$

where the u_j are the columns of U.
Note that, we can use say lm() to obtain the ridge regression estimator for $y = X\beta + \epsilon$ by doing ordinary least-squares regression with the vector y replaced by

$$\begin{pmatrix} y \\ 0_p \end{pmatrix}$$

and the matrix X replaced by

$$\begin{pmatrix} X \\ \sqrt{\lambda} I_p \end{pmatrix}.$$

Choice of the right λ for a given y, X is an art: Hastie et al. use the technique of **cross-validation**: see their book, or Venables and Ripley, for a description of cross-validation.
Interestingly, 'Penalised Discriminant Methods for the Classificaion of Tumors from Gene Expression Data' by D.Ghosh, Biometrics 2003, p992, discusses Ridge regression, and other methods, in the context of gene expression data, where the ith row of the matrix X corresponds to the 'gene expression profile' for the ith sample, and we have G possible tumours, with g_i as the known tumour class of the ith sample: thus g_i takes possible values $1, \ldots, G$.

11. (new for 2006) Many of the techniques used in Multivariate Analysis can be seen as examples in constrained optimization, using appropriate matrix algebra. Here is another such, which goes under the name of **correspondence analysis** for which the R/R function

```
 library(MASS)
corresp()
```

is written.
Consider the $R \times C$ contingency table with known probablities (p_{ij}), thus in matrix notation $u^T pv = 1$, where u, v are unit vectors of dimension R, C respectively. We wish to choose 'scores' $(x_i), (y_j)$ such that the random variables X, Y have joint frequncy function given by

$$P(X = x_i, Y = y_j) = p_{ij}$$

for all i, j, and $(x_i), (y_j)$ **maximise** the covariance between X, Y subject to the following constraints

$$E(X) = 0 = E(Y), var(X) = 1 = var(Y).$$

Show that this problem reduces to that of solving the 2 sets of equations

$$E(Y|X = x_i) = \lambda x_i, \text{ for all } i,$$

$$E(X|y = y_j) = \lambda y_j, \text{ for all } j.$$

In essence this means we focus on the singular value decomposition of the $R \times C$ matrix $(p_{ij}/p_{i+}p_{+j})$. The largest eigen value of this matrix is always 1 (can you see why?) and the next largest eigen value is λ, our desired correlation coefficient.
The reason why I had another look at this technique was its use in the article 'Food buying habits of people who buy wine or beer: cross-sectional study' by Johansen et al, British Medical Journal, March 4, 2006. The authors use correspondence analysis to give a 2-dimensional representation (a biplot) of the data from the 40×4 contingency table obtained from the the 40 categories of food types, and the 4 categories of alcohol buyers. For example, a person who buys both wine (but not beer) and olives contributes to the (olives, wine but not beer) cell. The resulting graph is constructed from the approximation

$$p_{ij} = p_{i+}p_{+j}(1 + \lambda_1 \mu_{i1} \nu_{j1} + \lambda_2 \mu_{i2} \nu_{j2})$$

and you can see it for yourself at `http://bmj.bmjjournals.com`