Multivariate Analysis Summary Sheet

Shravan Vasishth (vasishth@uni-potsdam.de)

February 24, 2015

Contents

Multivariate Analysis Summary Sheet

Compiled by: Shravan Vasishth (vasishth@uni-potsdam.de)

Version dated: February 24, 2015

Preliminaries

We will treat X' as the data matrix:

$$X' = \begin{pmatrix} x_{11} & \dots & x_{1p} \\ \vdots & \vdots & \vdots \\ x_{n1} & \dots & x_{np} \end{pmatrix} \tag{1}$$

Sample mean

The **sample mean vector** is:

$$\bar{x} = \begin{pmatrix} \bar{x}_1 \\ \vdots \\ \bar{x}_p \end{pmatrix} = \frac{1}{n} \begin{pmatrix} x_{11} + \dots + x_{n1} \\ \vdots \\ x_{1p} + \dots + x_{np} \end{pmatrix} \tag{2}$$

More compactly: $\bar{x} = \frac{1}{n} X 1 \Leftrightarrow \bar{x}' = \frac{1}{n} 1' X'$.

Sample variance

Sample variance (variance-covariance) matrix:

$$S_{p\times p} = var(X') = \frac{1}{n-1}(X - \bar{X})(X - \bar{X})' \tag{3}$$

$\bar{X}_{p\times n} = [\bar{x} \dots \bar{x}]$

Properties of S:

S is symmetric, S_{ii} is sample variance, and S_{ij} is sample covariance.

If columns of S are linearly independent (i.e., if none of the variables is a linear combination of the other), S is non-singular, and positive definite.

The **sample correlation matrix** R is the same as the vcov matrix, but has entries scaled:

$$r_{ij} = \frac{s_{ij}}{\sqrt{s_{ii} s_{jj}}} \tag{4}$$

If L be the diagonal matrix:

$\begin{pmatrix} s_{11} & \dots & 0 \\ 0 & \dots & 0 \\ 0 & \dots & s_{nn} \end{pmatrix}$ and $L^{1/2}$ has sds along the diagonal, then

$$S = L^{1/2} R L^{1/2} \tag{5}$$

1. R is symmetric, $p \times p$
2. $r_{ii} = 1$ for all i
3. $-1 \leq r_{ij} 1$.
4. Geometrically, $r_i j$ is the cosine of the angle between the vectors of deviations of observations of the ith and jth variables from the mean.
   (a) If angle is 0 deg (0 rad), $\rho = 1$.
   (b) If angle is 90 deg ($\pi/2$ rad), $\rho = 0$.

   to-do example

Some useful results

1. Let w be a vector. Then $var(X'w) = w' var(X') w = w' S w$.
2. If A is any $p \times q$ matrix, then $Var(X'A) = A' Var(X') A = A' S A$.

Useful properties of eigenvalues and eigenvectors

Let $A_{p\times p}$, and let eigenvalues be λ_i.

1. $\sum \lambda_i = trace(A)$
2. $\prod \lambda_i = det(A)$
3. If λ_i is an eigenvalue of A, then there is at least one vector x_i, called an eigenvector of A, such that $Ax_i = \lambda_i x_i$.
4. If A is realand symmetric $p \times p$, then there are always p linearly independent eigenvectors.
5. If C is any non-singular square matrix, then A and CAC^{-1} have the same eigenvalues. If x_i is an eigenvector of A with eigenvalue λ_i, then Cx_i is an eigenvector of CAC^{-1} with eigenvalue λ_i. [Prove it]
6. If A is real symmetric, i.e., $A = A'$. Then the eigenvalues of A are real. [Prove it]
7. If A is real symmetric, and λ and μ are eigenvalues, and x and y are corresponding eigenvectors, then x and y are orthogonal. [Prove it]
8. If A is real symmetric, and if X denotes the matrix whose columns are normalized eigenvectors of A, then X is an orthogonal matrix ($XX' = I$, or $X' = X^{-1}$). [Prove it]

   Also, $X^{-1} A X = \Lambda$, where Λ is the diagonal matrix containing the eigenvalues of A along the diagonal. This is because $AX = X\Lambda$.

The decomposition $X^{-1}AX = \Lambda$ is called the **spectral decomposition**, and is equivalent to

$$A = \lambda_1 x_1 x_1' + \cdots + \lambda_p x_p x_p' \tag{6}$$

where x_i is an eigenvector of A with eigenvalue λ_i.

9. If A is any positive definite real symmetric matrix, it will have positive definite real symmetric square roots. [Prove it]
10. The eigenvalues of a variance-covariance matrix are non-negative.

Vector calculus review

Some useful results:

1. If S is a symmetric $p \times p$ vector, and if $f(x) = x'Sx$. Then $\frac{\delta f}{\delta x} = 2Sx$. [Derive this.]
2. If S is not symmetric, $\frac{\delta f(x)}{\delta x} = (S + S')x$
3. If S is I, then $\frac{\delta f(x)}{\delta x} = \frac{\delta x'x}{\delta x} = 2x$.
4. If a is a constant p-vector, and $f(x) = a'x$, then $\frac{\delta f}{\delta x} = a$.

Constrained optimization (Lagrange Multipliers)

Suppose $t_1, \ldots, t_n$ are unbiased estimates of θ, and variance of t_i, $i = 1, \ldots, n$, is σ_i^2.

```
> ## t_i, equal variance
> t<-rnorm(10)
```

Find **B**est **L**inear **U**nbiased **E**stimator of θ.
Solution:
The BLUE of θ will be a weighted sum of the t_i. Let this weighted sum be $\tau = \sum a_i t_i$, such that $E[\tau] = \theta$.
We need to minimize the variance: $Var(\tau) = Var(\sum a_i t_i) = \sum a_i^2 Var(t_i) = \sum a_i^2 \sigma_i^2$.
Since $E[t_i] = E[\tau] = \theta$, the weights a_i must sum to 1. So the constraint is that $\sum a_i = 1$.
So, we minimize this function:

$$\Omega = \sum a_i^2 \sigma_i^2 + \lambda(\sum a_i - 1) \tag{7}$$

Differentiating with respect to each a_i, we get: $2a_i\sigma_i^2 + \lambda = 0$, which implies that

$$a_i = -\frac{\lambda}{2\sigma_i^2} = -\frac{1}{2}\frac{\lambda}{\sigma_i^2} \tag{8}$$

Differentiating with respect to the Lagrangian, we get

$$\sum a_i - 1 = 0 \tag{9}$$

Replacing a_i in the above with the RHS in equation 8,

$$\sum -\frac{1}{2}\frac{\lambda}{\sigma_i^2} - 1 = 0 \tag{10}$$

Adding -1 to both sides:

$$\sum -\frac{1}{2}\frac{\lambda}{\sigma_i^2} = 1 \tag{11}$$

Multiplying both sides by -1:

$$\sum \frac{1}{2}\frac{\lambda}{\sigma_i^2} = -1 \tag{12}$$

Solve for λ (change index i to k):

$$\lambda = \frac{-1}{\frac{1}{2}}\frac{1}{\frac{1}{\sum \sigma_k^2}} = \frac{-2}{\frac{1}{\sum \sigma_k^2}} \tag{13}$$

Now we can figure out each a_i by plugging in λ into

$$a_i = -\frac{1}{2}\frac{\lambda}{\sigma_i^2} = -\frac{1}{2}\frac{\frac{-2}{\frac{1}{\sum \sigma_k^2}}}{\sigma_i^2} = \frac{1}{\sigma_i^2}\left[\frac{1}{\sum \sigma_k^2}\right]^{-1} \tag{14}$$

Finally, we need to plug in the definition of a_i into $\tau = \sum a_i t_i$. At this stage it makes sense to use the index i again (instead of k):

$$\tau = \sum a_i t_i = \sum \left[\frac{1}{\sigma_i^2}\left[\frac{1}{\sum \sigma_k^2}\right]^{-1}\right] t_i = \sum \frac{t_i}{\sigma_i^2}\left[\frac{1}{\sum \sigma_k^2}\right]^{-1} \tag{15}$$

Multivariate distributions

Definition: If μ is a p-vector and Σ is a positive definite symmetric $p \times p$ matrix, then MVN distribution $N_p(\mu, \Sigma)$ is:

$$f_x(x) = \frac{1}{(2\pi)^{p/2} \mid \Sigma \mid^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)'\Sigma^{-1}(x-\mu)\right) \tag{16}$$

1. The quadratic form $(x-\mu)'\Sigma^{-1}(x-\mu)$ in the kernel is a statistical distance measure; for any value of x, the quadratic form gives the squared statistical distance of x from μ, called squared Mahalanobis distance.
2. Note that the MVN density is constant on surfaces of contours where $(x-\mu)'\Sigma^{-1}(x-\mu) = c^2$

   "The axes of each ellipsoid of constant density are in the direction of the eigen- vectors of Σ^-1 (recall that these are the same as the eigenvectors of Σ, but if $\Sigma x = \lambda x$, then $\Sigma^{-1}x = \lambda^{-1}x$), and their lengths are proportional to the reciprocals of the square roots of the eigenvalues of Σ^{-1}." (p. 95)
3. If $x \sim N_p(\mu, \Sigma)$, then
   (a) $(x-\mu)'\Sigma^{-1}(x-\mu) \sim \chi^2_p$.
   (b) The solid ellipsoid $\{x \mid (x-\mu)'\Sigma^{-1}(x-\mu) \leq \chi^2_p(\alpha)\}$ has probability $1-\alpha$.

   This follows from the fact that if $x \sim N_p(\mu, \Sigma)$ then $y = \Sigma^{1/2}(x-\mu) \sim N_p(0, I_p)$ and therefore:

$$y'y = (x-\mu)'\Sigma^{-1}(x-\mu) = \sum_{i=1}^{2} Y_i^2 \sim \chi^2_p \tag{17}$$

   "One of the consequences of the properties is that the marginal distributions of the individual variables of a multivariate normal distribution is a univariate normal distribu- tion." (p. 96)
4. If $X \sim N_p(\mu, \Sigma)$ and w is a p-vector, then the linear combination $w'X \sim N(w'\mu, w'\Sigma w)$.
5. If $X \sim N_p(\mu, \Sigma)$ and A is a $q \times p$ matrix, then the linear combination $AX \sim N(A\mu, A\Sigma A')$.
6. If $X \sim N_p(\mu_X, \Sigma_X)$ and $Y \sim N_q(\mu_Y, \Sigma_Y)$, then the p+q vector $\begin{pmatrix} X \\ Y \end{pmatrix} \sim N_{p+q}\left(\begin{pmatrix} \mu_X \\ \mu_Y \end{pmatrix}, \begin{pmatrix} \Sigma_X & 0 \\ 0 & \Sigma_Y \end{pmatrix}\right)$
   as long as X and Y are independent.
7. If $\begin{pmatrix} X \\ Y \end{pmatrix} \sim N_{p+q}\left(\begin{pmatrix} \mu_X \\ \mu_Y \end{pmatrix}, \begin{pmatrix} \Sigma_X & \Sigma \\ \Sigma' & \Sigma_Y \end{pmatrix}\right)$
   then X and Y are independent iff $\Sigma = 0$.

Principal Components Analysis

Useful mostly when we have continuous data.
[Sources: Tutorial on PCA by Shlens,]
Let $X_{n\times p}$ be our data, where we have p different variables, and n measurements on each variables. Example:

```
> n<-5
> x1<-c(1,3,5,7,9)
> x2<-c(4,7,8,11,15)
> ## n=5, p=2:
> X<-data.frame(x1=x1,x2=x2)
> ## centered variables:
> x1<-scale(x1,scale=F)
> x2<-scale(x2,scale=F)
> X<-data.frame(x1,x2)
> X<-as.matrix(X)
```

The matrix $\frac{1}{n-1}X'X$ is the real symmetric variance-covariance matrix, and represents the relationships between the variables:

```
> var1<-(1/(n-1))*t(X)%*%X

   x1   x2
x1 10 13.0
x2 13 17.5

> var2<-var(X)

   x1   x2
x1 10 13.0
x2 13 17.5
```

The total amount of dispersion on the data is the sum of the variances (the trace):

```
> ## sum of the variances=the total dispersion:
> sum(diag(var2))

[1] 27.5
```

Fact 1 *Any symmetric matrix X is diagonalized by an orthogonal matrix of its eigenvectors (see Theorems 1 and 2 in MatrixAlgebraSummary.pdf). For a symmetric matrix X, $X = EDE^T$, where D is a diagonal matrix and E is a matrix of eigenvectors of X arranged as columns.*

PCA using eigenvalue decomposition:
The goal is to find some orthonormal matrix P in $Y = PX$ such that the covariance matrix $C_Y = \frac{1}{n}YY^T$ is a diagonal matrix. The rows of P are the *principal components* of X.
First, write C_Y in terms of the unknown Y:

$$\begin{aligned} C_Y &= \frac{1}{n}YY^T \\ &= \frac{1}{n}(PX)(PX)^T \\ &= \frac{1}{n}PXX^TP^T \\ &= P(\frac{1}{n}XX^T)P^T \\ C_Y &= PC_XP^T \end{aligned} \tag{18}$$

Choose P to be a matrix such that each row p_i is an eigenvector of Cov_X. So $P = E^T$ where E has the eigenvectors of X in each column. Also note that $P^{-1} = P^T$.
Next, we show that this choice of P diagonalizes C_Y—that's the goal of PCA.

$$\begin{aligned} C_Y &= PC_XP^T \\ &= P(E^TDE)P^T \\ &= P(P^TDP)P^T \\ &= (PP^T)D(PP^T) \\ &= (PP^{-1})D(PP^{-1}) \\ C_Y &= D \end{aligned} \tag{19}$$

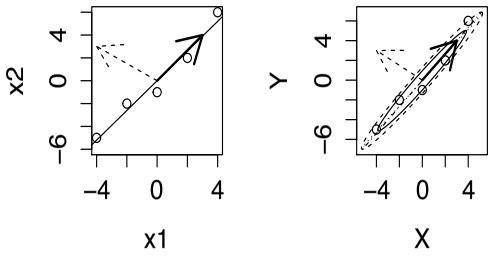
Key point: the i-th diagonal value C_Y is the variance of X along the principal component (an eigenvector) p_i.
The real symmetric var-cov matrix var2 can be decomposed into $U\lambda U'$, where

1. U is the orthonormal matrix containing the eigenvectors of var2
2. λ is a diagonal matrix containing the eigenvalues of var2

```
> lambda<-diag(eigen(var2)$values)
> U<-eigen(var2)$vectors
> U%*%lambda%*%t(U)
```

```
     [,1] [,2]
[1,]   10 13.0
[2,]   13 17.5
```

We want to linearly transform the vectors x1 and x2:



In this figure, the largest direction of variance lies along the best fit line, not perpendicular to the best fit line. Thus, by assumption, the dynamics of interest lie along the direction with largest variance. Maximizing the variance corresponds to finding the appropriate rotation of the naive basis.

By transforming X' to $Y' = aX'$, we have projected X' onto a, a one-dimensional space, a single line. The values of Y' give the co-ordinates of each observation along the vector a.

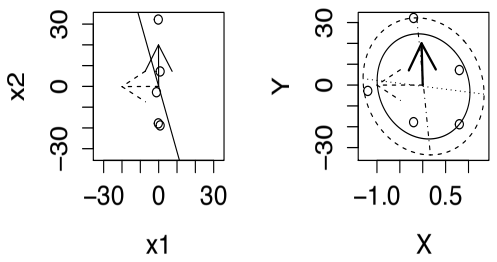
For example, if we have $x_1 = \begin{pmatrix}1\\2\end{pmatrix}, x_2 = \begin{pmatrix}2\\1\end{pmatrix}, x_3 = \begin{pmatrix}-1\\1\end{pmatrix}$, and $a = \begin{pmatrix}\frac{3}{5}\\\frac{-4}{5}\end{pmatrix}$, then

$$X' = \begin{pmatrix}1 & 2\\2 & 1\\-1 & 1\end{pmatrix}.$$

Therefore $Y' = X'a = \begin{pmatrix}-1\\\frac{2}{5}\\-\frac{7}{5}\end{pmatrix}$.

So, $x_1 = -1.a + z_1$ where z_1 is orthogonal to a. And so on.

If the maximum variance were *not* along the line of best fit, but in a perpendicular direction, the first PC would point in that direction:



Assumption (sometimes incorrect): **Large variances have important structure**. Therefore maximize variance to find the most important PCs.
The reasoning is that if variance is small, then all the observations have similar values, so there is little information in the data. If variance is large, then we have more "information".

Geometric interpretation

Basically, the PCA method just rotates the data so that the ellipsoid's axes are the principal components.

Presentation from lecture notes

Goal: reduce $X_{n\times p}$ to $Y_{n\times q}$
"A linear transformation $X' \to Y'$ is given by $Y' = X'A$ where A is a $p \times p$-matrix; it makes statistical sense to restrict attention to non-singular matrices A. If A happens to be an orthogonal matrix, i.e., $A'A = I_p$,then the transformation $X' \to Y'$ is an orthogonal transformation (i.e., just a rotation and/or a reflection of the n points in p-dimensional space)".
We want to choose A such that variance X'A is maximized; this is because "maximizing the variance corresponds to finding the appropriate rotation of the naive basis" (Shlens tutorial).
How to choose A such that variance of $X'A$ is maximized? If we just choose some A:

```
> m<-matrix(rnorm(1000),byrow=TRUE,ncol=10)
> diag(var(m))

 [1] 1.0372989 0.9729787 1.0643479 1.0336312 1.0152501
 [6] 1.2064683 0.9213986 0.7930977 0.8738666 0.8945069

> a<-matrix(rnorm(100),byrow=TRUE,ncol=10)
```

You can increase variance arbitrarily by multiplying a with some number, here 2:

```
> diag(m1<-var(m%*%a))

 [1]  9.056444  4.519197  3.353855 11.664507 11.406446
 [6] 13.361439 10.428704  5.050008 13.424693 18.017853

> diag(m2<-var(m%*%(2*a)))

 [1] 36.22578 18.07679 13.41542 46.65803 45.62579 53.44576
 [7] 41.71482 20.20003 53.69877 72.07141
```

Notice that variances have increased by a factor of 2^2:

```
> diag(m2)/diag(m1)

 [1] 4 4 4 4 4 4 4 4 4 4
```

The point here: The "maximum variance" here is unbounded. We can increase it arbitrarily. To maximize variance, we have to constrain it somehow, and that's why a constraint is imposed (below), that $w_i'w_i$ sums to 1 and the eigenvectors w_i, w_j $(i \neq j)$ are orthogonal.

Theorem 1 *The p principal components of data X' are the p eigenvectors $a_1, \ldots, a_p$ corresponding to the p ordered eigenvalues $\lambda_1 \geq \cdots \geq \lambda_p$ of S, the variance of X'. [Theorem 3.3 in lecture notes]*

Definition 1 *The first principal component is the vector a_1 such that the projection of the data X' onto a_1, i.e., $X'a_1$, has maximal variance, subject to the normalising constraint that $a_1'a_1 = 1$ (i.e., a_1 has length 1).*

PCA example by hand

To find first PC, we maximize $w_1 S w_1$ subject to $w_1'w_1 = 1$, where $S = X'X$. w_1 will be the eigenvector representing the direction of the maximum variance.

$$w_1' S w_1 - \lambda_1(w_1'w_1 - 1) \tag{20}$$

Differentiating with respect to w_1 gives:

$$Sw_1 - \lambda_1 w_1 = 0 \Leftrightarrow Sw_1 = \lambda_1 w_1 \tag{21}$$

But this means that λ_1 is an eigenvalue of S, and that w_1 is the corresponding eigenvector.
When finding the second principal component, we want to ensure that w_2 is uncorrelated to w_1. This means that $w_2'Sw_1 = 0$, but since $Sw_1 = \lambda_1 w_1$, we have $w_2'w_1 = 0$.

So the second PC w_2 should have the property that the projection of X' onto w_2 should have maximum variance subject to $w_2'w_2 = 1$ and $w_2'w_1 = 0$. Since we have two constraints we will use two Lagrangian multipliers:

$$\Omega_2 = w_2'Sw_2 - \mu w_2'w_1 - \lambda_2(w_2'w_2 - 1) \tag{22}$$

Differentiate with respect to w_2:

$$2Sw_2 - \mu w_1 - 2\lambda_2 w_2 = 0 \tag{23}$$

Pre-multiplying each side with w_1':

$$2w_1'Sw_2 - \mu w_1'w_1 - 2\lambda_2 w_1'w_2 = 0 \tag{24}$$

Pre-multiplying by w_2':

$$2w_2'Sw_2 - \mu w_2'w_1 - 2\lambda_2 w_2'w_2 = 0 \tag{25}$$

Because $\lambda_2 w_1'w_2 = 0$, we can see from equation 24 that

$$\mu = 2w_1'Sw_2 = 2(Sw_a)'w_2 = 2(\lambda_1 w_1)'w_2 = 2\lambda_2 w_1'w_2 = 0 \tag{26}$$

From equation 23, we can then conclude that

$$2Sw_2 = \lambda_2 w_2 \tag{27}$$

Therefore, w_2 is an eigenvector of S, with eigenvalue λ_2.
Recall that $var(X'w_2) = w_2'var(X')w_2 = w_2'Sw_2$. Equation 25 is that:

$$2w_2'Sw_2 - \mu w_2'w_1 - \lambda_2 w_2'w_2 = 0 \tag{28}$$

Replacing $w_2'Sw_2$ with $var(X'w_2)$, we get that:

$$2var(X'w_2) - \underset{\uparrow}{\mu w_2'w_1}_{=0} - 2\lambda_2 w_2'w_2 = var(X'w_2) - 2\lambda_2 = 0 \tag{29}$$

Therefore, $var(X'w_2) = 2\lambda_2$. This entails that λ_2 must be the second largest eigenvalue with eigenvector w_2. (Question: why is this entailed?)
All Principal Components can be found using the diagonalization of S. The eigenvalue decomposition of S is also called the **spectral decomposition** of S. The set of eigenvalues is called the **spectrum** of S.

Computing PCs using eigen()

```
> x<-c(1,3,5,7,9)
> y<-c(4,7,8,11,15)
> plot(x,y)
> xy<-data.frame(x=x,y=y)
> var_matrix<-var(xy)
> lambda<-eigen(var_matrix)$values
> eigenvectors<-eigen(var_matrix)$vectors
> ## first principle component contains 99.2% of the information:
> lambda[1]/(lambda[1]+lambda[2])

[1] 0.9920022

> ## using prebuilt function:
> xy_pc<-princomp(xy)
> plot(xy_pc)
> #source("code/scree.R")
```

How to map the points onto the PC

Take the first example above:

```
> n<-5
> x1<-c(1,3,5,7,9)
> x2<-c(4,7,8,11,15)
> ## n=5, p=2:
> ## centered variables:
> x1<-scale(x1,scale=F)
> x2<-scale(x2,scale=F)
> X<-data.frame(x1,x2)
> X<-as.matrix(X)
> S<-var(X)
> lambda<-eigen(S)$values
> U<-eigen(S)$vectors
```

The first set of points $x_1 = (1, 4)$, and e_1 is the first PC.

$$x_1 = \bar{x} + te_1 + ue_2 \tag{30}$$

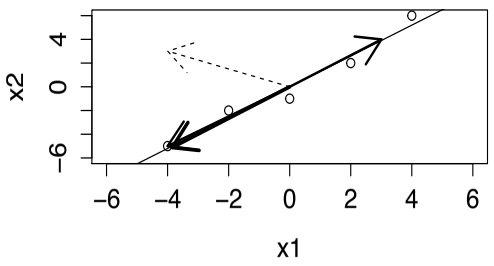
$\bar{x} = 0$ as data are centered, so we have:

$$x_1 = te_1 + ue_2 \tag{31}$$

Premultiplying by e_1:

$$e_1'x_1 = e_1'te_1 + e_1'ue_2 = t \tag{32}$$

So, $t = e_1'x_1$ (**or**: $t = x_1'e_1$), so $x_1 = te_1$. I.e., te_1 is the point where x_1 is mapped on to on the first PC.



Computing PCs using the correlation matrix

Needed when data are of mixed type, or one variable has very high variance compared to others.
Rule of thumb: "If the largest variance is more than about 4 times the smallest, then use the correlation matrix (otherwise the variables with large variance will dominate the principal component calculation)."
One should use the vcov matrix if the scales of measurement are similar, and the standard deviations of all variables are similar.
Note: The sort of interpretation given in the turtle example is valid only really when the variance matrix is used rather than the correlation matrix.
Interpretation of loadings is only viable for small numbers of variables.
Steps:

1. Center and scale data to get a correlation matrix:

```
> x1<-c(1,2,4,4,5,6,8,10)
> x2<-c(2,3,6,7,8,11,13,14)
> x1<-scale(x1,scale=F)
> x2<-scale(x2,scale=F)
> X<-data.frame(x1,x2)
> X<-as.matrix(X)
> S<-var(X)
> lambda<-eigen(S)$values
> U<-eigen(S)$vectors
> S_corr<-cov2cor(S)
```

Quadratic PCA

With a very small number of variables, one might try to generalise to e.g., quadratic principal components by adding variables for each quadratic combination.

Other things to look up

1. Projection pursuit
2. ICA
3. Factor analysis
4. Kohonen's SOMs
5. Generative topographic mapping

Multidimensional scaling

Goal: visualize a proximity matrix, if possible with a good lower-dimensional approximation (similar to PCA).
The mathematical problem: $n \times n$ proximity matrix (a symmetric matrix of δ_{ij} dissimilarities), find a q-dimensional space such that the calculated distance matrix d_{ij} reasonably matches the given dissimilarity matrix δ_{ij}. We can generally find one for $q = n - 1$. The more interesting case is when q is very small.
Aims of MDS:

1. To learn about the measure of dissimilarity itself
2. To discover underlying structure in the data
3. To see whether the data naturally divides into groups (clustering)

Principal coordinate analysis (classical metric scaling)

PCA = classical MDS

Theorem 2 *D is a matrix of Euclidean distances iff B is positive semidefinite (iff all eigenvalues of B are semidefinite).* *[Theorem 4.2, p. 52]*

Given a distance matrix $D = (\delta_{ij})$, we find the configuration of points:

1. Find B=HAH, where $A = -\frac{1}{2}\delta_{ij}^2$, and $H = I_n - \frac{1}{n}J_n$.
2. Find eigenanalysis of B.
3. Transpose matrix of eigenvectors.
4. The columns of this transposed matrix are the principal coordinates of the points.

Example:

```
> x<-c(1,3,5,7,9)
> y<-c(4,7,8,11,15)
> D<-matrix(rep(NA,25),ncol=5)
> ## compute Eudlidean distance:
> for(i in 1:5){
   for(j in 1:5){
     d<-(x[i]-x[j])^2+(y[i]-y[j])^2
     D[i,j]<-d
   }
 }
> A<- -0.5 * D
> H<-diag(5) - (1/5)*matrix(rep(1,25),ncol=5)
> B<- H%*% A %*% H
> eigen_B<-eigen(B)
> eigen_B$values


[1]  1.091202e+02  8.797635e-01  6.856497e-16 -9.909611e-16
[5] -6.869581e-15


> ## take first eigenvector:
> v1<-eigen_B$vectors[,1]
> f<-v1*sqrt(eigen_B$values[1])
> ## D2 fits with D:
> D2<-matrix(rep(NA,25),ncol=5)
> for(i in 1:5){
   for(j in 1:5){
     d<-(f[i]-f[j])^2
     D2[i,j]<-d
   }
 }
```

Cluster analysis

Multivariate analysis

To test the null hypothesis: $H_0 : \mu_1 = \mu_2$. From the lecture notes (p. 102): obtain the sample Mahalanobis distance D^2 and reject H_0 if

$$\frac{n_1 n_2 (n-p-1)}{n(n-2)p} D^2 > F_{p,n-p-1}(\alpha) \tag{33}$$

where D^2:

$$D^2 = (\bar{x}_1 - \bar{x}_2)' S^{-1} (\bar{x}_1 - \bar{x}_2) \quad S = \frac{(n_1-1)S_1 + (n_2-1)S_2}{n-2} \tag{34}$$

To test the hypothesis

$$H_0 : \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} = \begin{pmatrix} \mu_3 \\ \mu_4 \end{pmatrix} \tag{35}$$

the procedure is identical, except that $\bar{x}_1$ are vectors, not single values.

Assessing normality

One can plot D^2 against the appropriate chi-squared distribution:

```
> library(MVA)
> #chiplot
```

Analysis using the library ICSNP

Example code (From ex 3):

```
m1<-with(shapedata,
         HotellingsT2(cbind(taper,point) ~ batch))
```

Time Series