What makes multivariate analysis different?

What makes multivariate analysis different?

First, Dr Pavel Krivitsky will discuss some general aspects of multivariate analysis in the video below.

Transcript

Usually, when studying complex phenomena, **many** variables are required. Besides, the process of studying is usually an iterative one with many variables often added or deleted from the study. Multivariate analysis deals with developing methods for better understanding the relationships between the many variables included in the analysis of such complex phenomena.

In previous studies, you may have learned about a variety of methods for analysing many variables. For example, you have probably learned about the *multiple regression* linear model:

$$Y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \cdots + \beta_p x_{i,p} + \epsilon_i,\ i = 1, \ldots, n$$

where Y_i is the ith observation of the response variable, $x_{i,k}$ ith observation of the kth predictor variable, and ϵ_i the ith error. However, in this regression, we designate the p predictors as fixed (conditioned on) and only *one* variable per observation is random. Typically, we assume that the ϵ_is and therefore Y_is are independent (conditional on the xs) or at least uncorrelated. Contrast this with

a *multivariate* linear model,

$$Y_{i,1} = \beta_{0,1} + \beta_{1,1} x_{i,1} + \beta_{2,1} x_{i,2} + \cdots + \beta_{p,1} x_{i,p} + \epsilon_{i,1},$$
$$Y_{i,2} = \beta_{0,2} + \beta_{1,2} x_{i,1} + \beta_{2,2} x_{i,2} + \cdots + \beta_{p,2} x_{i,p} + \epsilon_{i,2},$$

where $Y_{i,1}$ and $Y_{i,2}$ are the ith observations of two distinct response variables, and $\epsilon_{i,1}$ and $\epsilon_{i,2}$ may be correlated. The multivariate linear model can be used when multiple observations are taken on each individual in the sample, and it can allow us to model the relationships among these measurements.

Difficulties in such a process include:

- There is more data to analyse.
- More involved mathematics are necessary.
- Computer intensive methods are involved in the process.

Objectives of multivariate methods are the following:

- **Data reduction:** presenting the phenomenon as simply as possible **but** without sacrificing valuable information. Typical *representative method*: Principal components analysis. Sometimes, this reduction
  is achieved by introducing a small number of unobservable (latent) variables when trying to explain a large number of observable output variables. Representative methods: *factor analysis* and *covariance*
  structure analysis.
- **Sorting or grouping:** creating groups of "similar" objects or variables that in a sense are more closer to each other than to objects outside the group; and finding reasonable explanation for the existing grouping. *Representative methods*: Factor Analysis, Cluster Analysis, Discriminant Analysis.
- **Investigation of dependence among variables:** finding which sets of variables can be considered as independent and which are "more dependent"; and "measuring" the dependence. *Representative*
  Methods: Correlation Analysis, Partial Correlations, Canonical Correlations.
- **Prediction:** predicting values of one or more variables on the basis of observations of other variables that have been found to influence the former variables: a basic but important goal. *Representative*:
  Multivariate Regression.
- **Hypothesis testing:** either validating assumptions (e.g., normality) on the basis of which certain analysis is being done or to reinforce some prior modelling convictions (e.g., equality of parameters). Hypothesis testing is relevant to the applications of all multivariate methods we will be dealing with.

As a basic **mathematical model** for our analyses in this course the **multivariate normal distribution** will be used. The reasons for this are our limited time and the complexity of other approaches. Although in practice also other distributions are relevant, modelling based on the multivariate normal distribution can still be a very good approximation.

Optional video: Watch the below video for further review of multivariate functions.

Watch: Copula stock return

Watch the copula stock example below by Dr Pavel Krivitsky.

Revision: Matrix algebra

Vectors and matrices

As a shorthand notation, we shall be using $X \in \mathcal{M}_{p,n}$ to indicate that X is a matrix with p rows and n columns. A notation $\boldsymbol{x} \in \mathbb{R}^n$ will be used to indicate that $\boldsymbol{x}$ is a n-dimensional *column* vector. Of course, if $\boldsymbol{x} \in \mathbb{R}^n$, it also means that $\boldsymbol{x} \in \mathcal{M}_{n,1}$. *Transposition* will be denoted by $^\top$. After a transposition, from a matrix $X \in \mathcal{M}_{p,n}$ we get a new matrix $X^\top \in \mathcal{M}_{n,p}$. In particular, from a *column* vector $\boldsymbol{x} \in \mathbb{R}^n$ we arrive, after a transposition, to a *row* a vector $\boldsymbol{x}^\top \in \mathcal{M}_{1,n}$. It is well known that multiplication of a matrix (vector) with a scalar means multiplication of each of the elements of the matrix (vector) with that scalar. Also, two matrices (vectors) of the same dimension can be added (subtracted) and the result is a new matrix (vector) of the same dimension and elements which are the element wise sum (difference) of the elements of the matrices (vectors) to be added (subtracted).

The *Euclidean norm* of a vector $\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{pmatrix} \in \mathbb{R}^p$ is denoted by $\|\boldsymbol{x}\|$ and is defined as $\|\boldsymbol{x}\| = \sqrt{\sum_{i=1}^p x_i^2}$.

The *inner product* or, equivalently, the *scalar product* of two p-dimensional vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ is denoted and defined in the following way:

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^\top \boldsymbol{y} = \sum_{i=1}^p x_i y_i \tag{0.1}$$

Obviously, the relation $\|\boldsymbol{x}|^2 = \langle \boldsymbol{x}, \boldsymbol{x} \rangle$ holds. It is well known that if θ is the angle between two p-dimensional vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ then it also holds

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \|\boldsymbol{x}\| |\boldsymbol{y}| \cos(\theta) \tag{0.2}$$

Since $|\cos(\theta)| \leq 1$, we have the inequality

$$|\langle \boldsymbol{x}, \boldsymbol{y} \rangle| \leq \|\boldsymbol{x}\| |\boldsymbol{y}|$$

which is one variant of the *Cauchy–Bunyakovsky–Schwartz* Inequality. Further, if we want to *orthogonally project* the vector $\boldsymbol{x} \in \mathbb{R}^p$ on the vector $\boldsymbol{y} \in \mathbb{R}^p$ then (having in mind the geometric

interpretation of orthogonal projection) the result will be: $\frac{\boldsymbol{x}^\top \boldsymbol{y}}{\boldsymbol{y}^\top \boldsymbol{y}} \boldsymbol{y}$.

Finally, the rules for matrix multiplication are recalled: if $X \in \mathcal{M}_{p,k}$ and $Y \in \mathcal{M}_{k,n}$ (i.e. the number of columns in X is equal to the number of rows in Y) then the multiplication XY is possible and the result is a matrix $Z = XY \in \mathcal{M}_{p,n}$ with elements

$$z_{i,j},\ i = 1, 2, \ldots, p,\ j = 1, 2, \ldots, n:\ z_{i,j} = \sum_{m=1}^{k} x_{i,m} y_{m,j} \tag{0.3}$$

i.e. the element in the ith row and jth column of Z is a scalar product of the ith row of X and the j th column of Y. Note that the multiplication of matrices is **not commutative** and in general, it is not necessary for YX to even exist when XY exists. When the matrices are both square (quadratic) of the same dimension p (i.e. both $X \in \mathcal{M}_{p,p}$ and $Y \in \mathcal{M}_{p,p}$) then both XY and YX will be defined but would in general **not** give rise to the same result. The following transposition rule is important to be mentioned (and easy to check): if $X \in \mathcal{M}_{p,k}$ and $Y \in \mathcal{M}_{k,n}$ then the product XY exists and it holds:

$$(XY)^\top = Y^\top X^\top \tag{0.4}$$

One should be very careful with transposition though in order to avoid silly mistakes. If $X \in \mathbb{R}^p$, for example, both $X^\top X$ and $X X^\top$ exist. While the former is a scalar, the latter belongs to $\mathcal{M}_{p,p}$!

A square matrix $X \in \mathcal{M}_{p,p}$ is called *symmetric* if $x_{i,j} = x_{j,i}$ for $i = 1, 2, \ldots, p$ and $j = 1, 2, \ldots, p$ holds. For such a matrix, we have $X^\top = X$. The square matrix $\overset{p \times p}{\boldsymbol{I}} = \delta_{ij}$ for $i = 1, 2, \ldots, p$ and $j = 1, 2, \ldots, p$ holds (i.e., ones on the diagonal and zeros outside the diagonal) is called the *identity matrix* (of dimension p). Obviously, when the multiplication is possible then always $X\boldsymbol{I} = X$ and $\boldsymbol{I}X = X$ holds.

The trace of a square matrix $X \in \mathcal{M}_{p,p}$ is denoted by $\mathrm{tr}(X) = \sum_{i=1}^{p} x_{ii}$. The following properties of traces are easy to obtain:

1. $\mathrm{tr}(X + Y) = \mathrm{tr}(X) + \mathrm{tr}(Y)$
2. $\mathrm{tr}(XY) = \mathrm{tr}(YX)$
3. $\mathrm{tr}(X^{-1} Y X) = \mathrm{tr}(Y)$
4. If $\boldsymbol{a} \in \mathbb{R}^p$ and $X \in \mathcal{M}_{p,p}$ then $\boldsymbol{a}^\top X \boldsymbol{a} = \mathrm{tr}(X \boldsymbol{a}\boldsymbol{a}^\top)$

Dr Pavel Krivitsky provides further detail regarding these concepts in the video below.

Transcript

Inverse matrices

Watch the below video of an introduction of inverse matrices by Dr Pavel Krivitsky.

Transcript

To any **square** matrix $X \in \mathcal{M}_{p,p}$ one can attach a number $|X| \equiv \det(X)$ called a *determinant* of the matrix. It is defined as

$$|X| = \sum \pm x_{1i} x_{2j} \dots x_{pm}$$

where the summation is over **all** permutations $(i, j, \dots, m)$ of the numbers $(1, 2, \dots, p)$ by taking into account the **sign rule**: summands with an even permutation get a $(+)$ whereas the ones with an odd permutation get a $(-)$ sign.

It can be seen that this is equivalent to another recursive definition, namely:

- when $p = 1$ (scalar case) $X = a$ is just a number and $|X| = a$ in this case
- when $p = 2$ then $\begin{vmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{vmatrix} = x_{11}x_{22} - x_{12}x_{21}$
- when $p = 3$ then the following rule applies:

$$\begin{vmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{vmatrix} = x_{11}x_{22}x_{33} + x_{12}x_{23}x_{31} + x_{21}x_{32}x_{13} - x_{31}x_{22}x_{13} - x_{11}x_{23}x_{32} - x_{12}x_{21}x_{33} \tag{0.5}$$

- recursively, for $X \in \mathcal{M}_{(p,p)}$ we can define

$$|X| = \sum_i (-1)^{i+j} x_{ij} |X_{ij}| = \sum_j (-1)^{i+j} x_{ij} |X_{ij}|$$

  where X_{ij} denotes the matrix we get by deleting the ith row and jth column of X.

Here we list some elementary properties of determinants that follow directly from the definition:

1. If one row or one column of the matrix contains zeros only then the value of the determinant is zero.
2. $|X^\top| = |X|$
3. If one row (or one column) of the matrix is modified by multiplying with a scalar c then so is the value of the determinant.
4. $|cX| = c^p |X|$
5. If $X, Y \in \mathcal{M}_{p,p}$ then $|XY| = |X||Y|$
6. If the matrix X is *diagonal* (i.e. all non-diagonal elements are zero) then $|X| = \prod_{i=1}^p x_{ii}$. In particular, *the determinant of the identity matrix is always equal to one*.

Given that $|X| \neq 0$ (or equivalently, if the matrix $X \in \mathcal{M}_{p,p}$ is *nonsingular* then an **inverse** matrix $X^{-1} \in \mathcal{M}_{p,p}$ can be defined that has to satisfy $XX^{-1} = \boldsymbol{I}_{p,p}$. It is easy to check that the inverse X^{-1} has as its (j,i)th entry $\frac{|X_{ij}|}{|X|}(-1)^{i+j}$.

Some elementary properties of inverses follow:

1. $XX^{-1} = X^{-1}X = \boldsymbol{I}$
2. $(X^{-1})^\top = (X^\top)^{-1}$
3. $(XY)^{-1} = Y^{-1}X^{-1}$ when both X and Y are nonsingular square matrices of the same dimension.
4. $|X^{-1}| = |X|^{-1}$
5. If X is diagonal and nonsingular then all its diagonal elements are nonzero and X^{-1} is again diagonal with diagonal elements equal to $\frac{1}{x_{ii}}, i = 1, 2, \ldots, p$.

Rank & orthogonal matrices

A set of vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k \in \mathbb{R}^n$ is *linearly dependent* if there exist k numbers $a_1, a_2, \ldots, a_k$ **not all zero** such that

$$a_1\boldsymbol{x}_1 + a_2\boldsymbol{x}_2 + \cdots + a_k\boldsymbol{x}_k = \boldsymbol{0} \tag{0.6}$$

holds. Otherwise the vectors are *linearly independent*. In particular, for k linearly independent vectors the equality (0.6) would only be possible if **all** numbers $a_1, a_2, \ldots, a_k$ were zero.

The *row rank* of a matrix is the maximum number of linearly independent row vectors. The *column rank* is the rank of its set of column vectors. It turns out that the row rank and the column rank of a matrix are always equal. Thus the rank of a matrix X (denoted $\mathrm{rk}(X)$) is either the row or the column rank. If $X \in \mathcal{M}_{p,n}$ and $\mathrm{rk}(X) = \min(p, n)$ we say that the matrix is of full rank. In particular, a square matrix $A \in \mathcal{M}_{p,p}$ is of full rank if $\mathrm{rk}(A) = p$. As is well known from the basic theorem of linear algebra *Kronecker–Capelli* or *Rouché–Capelli Theorem* this means also that $|A| \neq 0$ when A is of full rank. Then the inverse of A will also exist. Let $\boldsymbol{b} \in \mathbb{R}^p$ be a given vector. Then the linear equation system $A\boldsymbol{x} = \boldsymbol{b}$ has a unique solution $\boldsymbol{x} = A^{-1}\boldsymbol{b} \in \mathbb{R}^p$.

Orthogonal matrices

A square matrix $X \in \mathcal{M}_{p,p}$ is *orthogonal* if $XX^\top = X^\top X = \boldsymbol{I}_{p,p}$ holds. The following properties of orthogonal matrices are obvious:

1. X is of full rank ($\mathrm{rk}(X) = p$) and $X^{-1} = X^\top$
2. The name *orthogonal* of the matrix originates from the fact that the scalar product of each two different column vectors equals zero. The same holds for the scalar product of each two different row vectors of the matrix. The norm of each column vector (or each row vector) is equal to one. These properties are equivalent to the definition.
3. $|X| = \pm 1$

Dr Pavel Krivitsky provides further detail regarding these concepts in the video below.

Transcript

Eigenvalues and eigenvectors

For **any** square matrix $X \in \mathcal{M}_{p,p}$ we can define the *characteristic polynomial* equation of degree p,

$$f(\lambda) = |X - \lambda \boldsymbol{I}| = 0. \tag{0.7}$$

Equation (0.7) is a polynomial equation of power p so it has exactly p roots. In general, some of them may be complex and some may coincide. Since the coefficients are real, if there is a complex root of (0.7) then also its complex conjugate must be a root of the same equation. Denote **any** such eigenvalue by λ^*. In addition, $\mathrm{tr}(X) = \sum_{i=1}^{p} \lambda_i$ and $|X| = \prod_{i=1}^{p} \lambda_i$.

Obviously, the matrix $X - \lambda^* \boldsymbol{I}$ is singular (its determinant is zero). Then, according to the Kronecker theorem, there exists a non-zero vector $\boldsymbol{y} \in \mathbb{R}^p$ such that $(X - \lambda^* \boldsymbol{I})\boldsymbol{y} = \boldsymbol{0}, \boldsymbol{0} \in \mathbb{R}^p$. We call $\boldsymbol{y}$ an *eigenvector* of X that corresponds to the eigenvalue λ^*. Note that the eigenvector is not uniquely defined: $\mu\boldsymbol{y}$ for any real non-zero μ would also be an eigenvector corresponding to the same eigenvalue.

Sparing some details of the derivation, we shall formulate the following basic result:

Theorem 0.1. *When the matrix X is real symmetric then **all** of its p eigenvalues are **real**. If the eigenvalues are all different then all the p eigenvectors that correspond to them, are orthogonal (and hence form a basis in $\mathbb{R}^p$). These eigenvectors are also unique (up to the norming constant μ above). If some of the eigenvalues coincide then the eigenvectors corresponding to them are not necessarily unique but even in this case they can be chosen to be mutually orthogonal.*

For each of the p eigenvalues λ_i, $i = 1, 2, \ldots, p$, of X, denote its corresponding set of mutually orthogonal eigenvectors of *unit length* by $\boldsymbol{e}_i$, $i = 1, 2, \ldots, p$, i.e.

$$X\boldsymbol{e}_i = \lambda_i \boldsymbol{e}_i,\ i = 1, 2, \ldots, p,\ \|\boldsymbol{e}_i\| = 1,\ \boldsymbol{e}_i^\top \boldsymbol{e}_j = 0,\ i \neq j$$

holds. Then is can be shown that the following decomposition *(spectral decomposition)* of any symmetric matrix $X \in \mathcal{M}_{p,p}$ holds:

$$X = \lambda_1 \boldsymbol{e}_1 \boldsymbol{e}_1^\top + \lambda_2 \boldsymbol{e}_2 \boldsymbol{e}_2^\top + \ldots \lambda_p \boldsymbol{e}_p \boldsymbol{e}_p^\top. \tag{0.8}$$

Equivalently, $X = P\Lambda P^\top$ where $\Lambda = \begin{pmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_p \end{pmatrix}$ is diagonal and $P \in \mathcal{M}_{p,p}$ is an *orthogonal matrix* containing the p orthogonal eigenvectors $\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_p$.

The above decomposition is a very important analytical tool. One of its most widely used applications is for defining a square root of a symmetric positive definite matrix. A symmetric matrix $X \in \mathcal{M}_{p,p}$

is *positive definite* if all of its eigenvalues are positive (it is called *non-negative definite* if all eigenvalues are ≥ 0). For a symmetric positive definite matrix we have all λ_i, $i = 1, 2, \ldots, p$, to be positive in the spectral decomposition (0.8).

But then

$$X^{-1} = (P^\top)^{-1}\Lambda^{-1}P^{-1} = P\Lambda^{-1}P^\top = \sum_{i=1}^{p} \frac{1}{\lambda_i}\boldsymbol{e}_i\boldsymbol{e}_i^\top \tag{0.9}$$

(i.e. inverting X is very easy if the spectral decomposition of X is known).

Moreover we can define the *square root* of the symmetric non-negative definite matrix X in a natural way:

$$X^{\frac{1}{2}} = \sum_{i=1}^{p} \sqrt{\lambda_i}\boldsymbol{e}_i\boldsymbol{e}_i^\top \tag{0.10}$$

The definition (0.10) makes sense since $X^{\frac{1}{2}}X^{\frac{1}{2}} = X$ holds. Note that $X^{\frac{1}{2}}$ is also symmetric and non-negative definite. Also $X^{-\frac{1}{2}} = \sum_{i=1}^{p} \lambda_i^{-\frac{1}{2}} e_i e_i^\top = P\Lambda^{-\frac{1}{2}}P^\top$ can be defined where $\Lambda^{-\frac{1}{2}}$ is a diagonal matrix with $\lambda_i^{-1/2}, i = 1, 2, \ldots, p$ being its diagonal elements. These facts will be used essentially in the subsequent sections.

As an illustration of the usefulness of the spectral decomposition approach we shall show the following statement:

Example 0.1. Let $X \in \mathcal{M}_{p,p}$ be symmetric *positive definite* matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p > 0$ and associated eigenvectors of unit length $\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots \boldsymbol{e}_p$. Show that

- $\max_{\boldsymbol{y} \neq \boldsymbol{0}} \frac{\boldsymbol{y}^\top X \boldsymbol{y}}{\boldsymbol{y}^\top \boldsymbol{y}} = \lambda_1$ attained when $\boldsymbol{y} = \boldsymbol{e}_1$.
- $\min_{\boldsymbol{y} \neq \boldsymbol{0}} \frac{\boldsymbol{y}^\top X \boldsymbol{y}}{\boldsymbol{y}^\top \boldsymbol{y}} = \lambda_p$ attained when $\boldsymbol{y} = \boldsymbol{e}_p$.

Let $X = P\Lambda P^\top$ be the decomposition (0.8) for X. Denote $\boldsymbol{z} = P^\top \boldsymbol{y}$. Note that $\boldsymbol{y} \neq \boldsymbol{0}$ implies $\boldsymbol{z} \neq \boldsymbol{0}$. Thus

$$\frac{\boldsymbol{y}^\top X \boldsymbol{y}}{\boldsymbol{y}^\top \boldsymbol{y}} = \frac{\boldsymbol{y}^\top P\Lambda P^\top \boldsymbol{y}}{\boldsymbol{y}^\top \boldsymbol{y}} = \frac{\boldsymbol{z}^\top \Lambda \boldsymbol{z}}{\boldsymbol{z}^\top \boldsymbol{z}} = \frac{\sum_{i=1}^{p} \lambda_i z_i^2}{\sum_{i=1}^{p} z_i^2} \leq \lambda_1 \frac{\sum_{i=1}^{p} z_i^2}{\sum_{i=1}^{p} z_i^2} = \lambda_1$$

If we take $\boldsymbol{y} = \boldsymbol{e}_1$ then having in mind the structure of the matrix P we have $\boldsymbol{z} = P^\top \boldsymbol{e}_1 = \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix}^\top$ and for this choice of $\boldsymbol{y}$ also $\frac{\boldsymbol{z}^\top \Lambda \boldsymbol{z}}{\boldsymbol{z}^\top \boldsymbol{z}} = \frac{\lambda_1}{1} = \lambda_1$. The first part of the exercise is shown. Similar arguments (just changing the sign of the inequality) apply to show the second part.

In addition, you can try to show that $\max_{\boldsymbol{y} \neq 0, \boldsymbol{y} \perp \boldsymbol{e}_1} \frac{\boldsymbol{y}^\top X \boldsymbol{y}}{\boldsymbol{y}^\top \boldsymbol{y}} = \lambda_2$ holds. How?

Watch the below example provided by Dr Pavel Krivitsky.

Transcript

Numerical stability and Cholesky decomposition

Computers perform arithmetic to a finite precision, typically around 16 decimal significant figures. Furthermore, the numbers are expressed internally in scientific notation, and so the absolute magnitude of the number typically has little effect on precision, but certain operations on numbers with very different magnitudes can sometimes produce severe rounding errors. For example, to a computer $1 \times 10^{18} + 1 \times 10^{0} = 1{,}000{,}000{,}000{,}000{,}000{,}000 + 1 = 1{,}000{,}000{,}000{,}000{,}000{,}000$: the 1 gets lost to a rounding error. When it comes to matrix inversion in particular, the key number is the *condition number*, $|\lambda_1/\lambda_p|$ of a positive definite matrix X, where λ_1 is the largest eigenvalue of X and λ_p is the smallest. (The definition for non-positive-definite matrices can be different.) The higher this number is, the less numerically stable the inversion is likely to be. (Notice that if the matrix is singular, this number is infinite.) We generally try to avoid asking the computer to invert matrices in ways that lose precision.

An alternative, more numerically stable definition of a "matrix square root" is the *Cholesky decomposition*. For a symmetric positive definite matrix $X \in \mathcal{M}_{p,p}$, there exists a unique upper-triangular matrix $U \in \mathcal{M}_{p,p}$ such that $U^\top U = X$ holds. Note that many sources use a lower-triangular matrix L such that $LL^\top = X$ instead. It is easy to see that $L \equiv U^\top$, and which definition is used is arbitrary, provided it is used consistently, since $UU^\top \neq X$ and neither do $L^\top L$. For example, the Wikipedia article uses L, whereas the R builtin function is `chol()` returns U. This decomposition is particularly useful for generating correlated variables.

Transcript

Standard facts about multivariate distributions

Random samples in multivariate analysis

In order to study the sampling variability of statistics like $\bar{x}$ and S_n that we introduced in Lecture 1, with the ultimate goal of making inferences, one needs to make some assumptions about the random variables whose values constitute the dataset $X \in \mathcal{M}_{p,n}$ in (1.1). Suppose the data has not been observed yet but we *intend* to collect n sets of measurements on p variables. Since the actual observations can not be predicted before the measurements are made, we treat them as random variables. Each set of p measurements can be considered as a realisation of p-dimensional *random vector* and we have n independent realisations of such random vectors $\boldsymbol{X}_i, i = 1, 2, \ldots, n$, so we have the *random matrix* $\boldsymbol{X} \in \mathcal{M}_{p,n}$:

$$\boldsymbol{X} = \begin{pmatrix} X_{11} & X_{12} & \cdots & X_{1j} & \cdots & X_{1n} \\ X_{21} & X_{22} & \cdots & X_{2j} & \cdots & X_{2n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ X_{i1} & X_{i2} & \cdots & X_{ij} & \cdots & X_{in} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ X_{p1} & X_{p2} & \cdots & X_{pj} & \cdots & X_{pn} \end{pmatrix} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n] \qquad (0.11)$$

The vectors $\boldsymbol{X}_i, i = 1, 2, \ldots, n$ are considered as independent observations of a p-dimensional random vector. We start discussing the distribution of such a vector.

Revision: Multivariate probability

Joint, marginal, conditional distributions

Before we begin this section, watch the below video explaining the concepts we will discuss.

Transcript

Optional video: Watch the below video for a brief review of multivariate functions.

A random vector $\boldsymbol{X} = \begin{pmatrix} X_1 & X_2 & \cdots & X_p \end{pmatrix}^\top \in \mathbb{R}^p$, $p \geq 2$ has a *joint cdf*

$$F_{\boldsymbol{X}}(\boldsymbol{x}) = P(X_1 \leq x_1, X_2 \leq x_2, \ldots, X_p \leq x_p) = F_{\boldsymbol{X}}(x_1, x_2, \ldots, x_p).$$

In case of a *discrete* vector of observations $\boldsymbol{X}$ the *probability mass function* is defined as

$$P_{\boldsymbol{X}}(\boldsymbol{x}) = P(X_1 = x_1, X_2 = x_2, \ldots, X_p = x_p).$$

If a *density* $f_{\boldsymbol{X}}(\boldsymbol{x}) = f_{\boldsymbol{X}}(x_1, x_2, \ldots, x_p)$ exists such that

$$F_{\boldsymbol{X}}(\boldsymbol{x}) = \int_{-\infty}^{x_1} \cdots \int_{-\infty}^{x_p} f_{\boldsymbol{X}}(\boldsymbol{t}) dt_1 \ldots dt_p \tag{0.12}$$

then $\boldsymbol{X}$ is a *continuous* random vector with a joint density function of p arguments $f_{\boldsymbol{X}}(\boldsymbol{x})$. From (0.12) we see that in this case $f_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{\partial^p F_{\boldsymbol{X}}(\boldsymbol{x})}{\partial x_1 \partial x_2 .. \partial x_p}$ holds.

The *marginal cdf of the first* $k < p$ components of the vector $\boldsymbol{X}$ is defined in a natural way as follows:

$$P(X_1 \leq x_1, X_2 \leq x_2, \ldots, X_k \leq x_k) = P(X_1 \leq x_1, X_2 \leq x_2, \ldots, X_k \leq x_k, X_{k+1} \leq \infty, ..., X_$$

$$= F_{\boldsymbol{X}}(x_1, x_2, \ldots, x_k, \infty, \infty, \ldots, \infty) \tag{0.13}$$

The *marginal density* of the first k components can be obtained by partial differentiation in (0.13) and we arrive at

$$\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f_{\boldsymbol{X}}(x_1, x_2, \ldots, x_p) dx_{k+1} \ldots dx_p$$

For **any** other subset of $k < p$ components of the vector $\boldsymbol{X}$, their marginal cdf and density can be obtained along the same lines. X_i has marginal cdf $F_{X_i}(x_i), i = 1, 2, \ldots, p$.

The *conditional density* $\boldsymbol{X}$ **when** $X_{r+1} = x_{r+1}, \ldots, X_p = x_p$ is defined by

$$f_{(X_1,\ldots,X_r|X_{r+1},\ldots,X_p)}(x_1, \ldots, x_r | x_{r+1}, \ldots, x_p) = \frac{f_{\boldsymbol{X}}(\boldsymbol{x})}{f_{X_{r+1},\ldots,X_p}(x_{r+1}, \ldots, x_p)} \tag{0.14}$$

The above conditional density is interpreted as the joint density of $X_1, \ldots, X_r$ when $X_{r+1} = x_{r+1}, \ldots, X_p = x_p$ and is only defined when $f_{X_{r+1},\ldots,X_p}(x_{r+1}, \ldots, x_p) \neq 0$.

In case $\boldsymbol{X}$ has p independent components then

$$F_{\boldsymbol{X}}(\boldsymbol{x}) = F_{X_1}(x_1) F_{X_2}(x_2) \cdots F_{X_p}(x_p) \tag{0.15}$$

holds and, equivalently, also

$$P_{\boldsymbol{X}}(\boldsymbol{x}) = P_{X_1}(x_1) P_{X_2}(x_2) \cdots P_{X_p}(x_p), \; f_{\boldsymbol{X}}(\boldsymbol{x}) = f_{X_1}(x_1) f_{X_2}(x_2) \cdots f_{X_p}(x_p) \tag{0.16}$$

holds. We note that in case of mutual independence the p components, all conditional distributions do **not** depend on the conditions and the factorisations

$$F_{\boldsymbol{X}}(\boldsymbol{x}) = \prod_{i=1}^{p} F_{X_i}(x_i), \; f_{\boldsymbol{X}}(\boldsymbol{x}) = \prod_{i=1}^{p} f_{X_i}(x_i)$$

hold.

Moments & density transformation formula

Before we begin this section, watch the below video explaining moments & density transformation formula.

Transcript

Given the density $f_{\boldsymbol{X}}(\boldsymbol{x})$ of the random vector $\boldsymbol{X}$ the *joint moments of order* $s_1, s_2 \ldots, s_p$ are defined, in analogy to the univariate case, as

$$\mathrm{E}\left(X_1^{s_1} \cdots X_p^{s_p}\right) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} x_1^{s_1} \cdots x_p^{s_p} f_{\boldsymbol{X}}\left(x_1, \ldots, x_p\right) dx_1 \ldots dx_p \tag{0.17}$$

Note that if some of the s_i in (0.17) are equal to zero then in effect we are calculating the joint moment of a subset of the p random variables.

Density transformation formula

Assume, the p existing random variables $X_1, X_2, \ldots, X_p$ with given density $f_{\boldsymbol{X}}(\boldsymbol{x})$ have been transformed by a smooth (i.e. differentiable) one-to-one transformation into p new random variables $Y_1, Y_2 \ldots, Y_p$, i.e. a new random vector $\boldsymbol{Y} \in \mathbb{R}^p$ has been created by calculating

$$Y_i = y_i(X_1, X_2 \ldots, X_p), i = 1, 2, \ldots, p \tag{0.18}$$

The question is how to calculate the density $g_{\boldsymbol{Y}}(\boldsymbol{y})$ of $\boldsymbol{Y}$ by knowing the transformation functions $y_i(X_1, X_2 \ldots, X_p), i = 1, 2, \ldots, p$ and the density $f_{\boldsymbol{X}}(\boldsymbol{x})$ of the original random vector. Naturally, since the transformation (0.18) is assumed to be one-to-one, its inverse transformation $X_i = x_i(Y_1, Y_2 \ldots, Y_p), i = 1, 2, \ldots, p$ also exists and then the following density transformation formula applies:

$$f_{\boldsymbol{Y}}(y_1, \ldots, y_p) = f_{\boldsymbol{X}}[x_1(y_1, \ldots, y_p), \ldots, x_p(y_1, \ldots, y_p)]|J(y_1, \ldots, y_p)| \tag{0.19}$$

where $J(y_1, \ldots, y_p)$ is the *Jacobian* of the transformation:

$$J(y_1, \ldots, y_p) = \left|\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{y}}\right| \equiv \left|\begin{pmatrix} \frac{\partial x_1}{\partial y_1} & \frac{\partial x_1}{\partial y_2} & \cdots & \frac{\partial x_1}{\partial y_p} \\ \frac{\partial x_2}{\partial y_1} & \frac{\partial x_2}{\partial y_2} & \cdots & \frac{\partial x_2}{\partial y_p} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial x_p}{\partial y_1} & \frac{\partial x_p}{\partial y_2} & \cdots & \frac{\partial x_p}{\partial y_p} \end{pmatrix}\right| \equiv \left|\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}}\right|^{-1} \tag{0.20}$$

Note that in (0.19) the *absolute value* of the Jacobian is substituted.

Check your understanding

Complete the below exercises to check your understanding of concepts presented so far.

1. In an ecological experiment, colonies of 2 different species of insect are confined to the same habitat. The survival times of the two species (in days) are random variables X_1 and X_2 respectively. It is thought that X_1 and X_2 have a joint density of the form

$$f_{\boldsymbol{X}}(x_1, x_2) = \theta x_1 \mathrm{e}^{-x_1(\theta + x_2)} \quad (0 < x_1, x_2)$$

for some constant $\theta > 0$.

a) Show that $f_{\boldsymbol{X}}(x_1, x_2)$ is a valid density.

b) Find the probability that both species die out within t days of the start of the experiment.

c) Derive the marginal density of X_1. Identify this distribution and write down $\mathrm{E}(X_1)$ and $\mathrm{Var}(X_1)$.

d) Derive the marginal density of X_2, and the conditional density of X_2 **given** $X_1 = x_1$.

e) What evidence do you now have that X_1 and X_2 are not independent?

Solutions to these exercises will be provided in the 'Quizzes' discussion forum.

Demonstration: Matrices

This demonstration can be completed using the provided RStudio or your own RStudio.

To complete this task select the 'Matrix_Examples.demo.Rmd' in the 'Files' section of RStudio. Follow the demonstration contained within the RMD file.

If you choose to complete the example in your own RStudio, upload the following file:

Matrix_Examples.demo.Rmd

After working through the demonstration, complete the associated 'Challenge' on the next slide.

The contents of the RMD file are also displayed below:

Constructing matrices in R

R provides a number of tools for constructing and manipulating matrices. The following is a quick overview, particularly where they are distinguished from data frames:

- `diag(n)` (`n` scalar integer) is a `n`×`n` Identity matrix. Note that the function behaves differently depending on whether `n` has length 1 or not.
- `diag(x)` (`x` a vector of length `n`) is a `n`×`n` diagonal matrix whose diagonal elements come from vector `x`; can lead to unpredictable behaviour when the length of `x` is not known in advance and could potentially be 1.
- `diag(x, n)` is a `n`×`n` diagonal matrix whose column elements come from vector `x`; works consistently unlike the previous case.
- `matrix(1,n,n)` is a `n`×`n` matrix of ones
- `matrix(1,n,p)` is a `n`×`p` matrix of ones
- `matrix(x,n,p)` (`x` a scalar or a vector) is a `n`×`p` filled with values of `x` *by columns* with `x` recycled as needed to length `n*p`
- `matrix(1,n)` is an `n`×`1` column vector of ones
- `matrix(1,ncol=n)` is an 1×n row vector of ones
- `cbind(x1,x2,x3)` bind dimensionless or column vectors or matrices `x1`, `x2`, and `x3` into a matrix as columns
- `rbind(x1,x2,x3)` bind dimensionless or row vectors or matrices `x1`, `x2`, and `x3` into a matrix as rows

Matrix arithmetic and operations

- Arithmetic operations `+`, `-`, `*`, `/`, etc. perform operations elementwise, as do many functions such as `exp()` and `sin()`.
- Matrix product is `%*%`: not `*`!

Matrix functions

- `dim(X)`, `nrow(X)`, `ncol(X)` obtain matrix dimensions
- `t(X)` transpose of `X`
- `c(X)` convert a matrix (or a vector with dimension) into a dimensionless vector; for a matrix, this is equivalent to stacking the columns.
- `solve(X)` inverse of `X`: not `X^-1`!
- `crossprod(X,Y)` is equivalent to `t(X)%*%Y` but much faster; if `Y` is omitted, `t(X)%*%X` is computed.
- `chol(X)` Cholesky decomposition: if `U <- chol(X)`, then `t(U)%*%U` reconstructs `X`
- `eigen(X)` Eigendecomposition: if `e <- eigen(X)`, then `e$vectors%*%diag(e$values,length(e$values))%*%t(e$vectors)` reconstructs `X`

Rowwise and columnwise computation

- `apply(X, MARGIN, FUN)` for each row (if `MARGIN=1`) or column (if `MARGIN=2`), evaluate function `FUN` and return the resulting vector. (If `FUN` itself returns a vector, `apply()` returns a matrix.)
- `sweep(X, MARGIN, STATS, FUN)` for each row (if `MARGIN=1`) or column (if `MARGIN=2`) `x`, evaluate `FUN(x, STATS[i])` and replace the original row.
- `colMeans`, `rowMeans`, `colSums`, `rowSums`: exactly as it sounds.
- `scale(X)`: centre the matrix so that its column means are 0 and its column mean-squared value is 1; further arguments can be used to do one or the other only.

Challenge: Matrices

If you choose to complete this task in your own RStudio, upload the following file:

Matrix_Examples.challenge.Rmd

Click on the 'Matrix_Examples.challenge.Rmd' in the 'Files' section to begin.

This activity and the solution will be discussed at the Collaborate session this week. In the meantime, share and discuss your results in the 'Tutorials' discussion forum.

The solution will also be available here on Friday of this week by clicking on the 'Solution' tab in the top right corner.

The contents of the RMD file are also displayed below:

Challenges

Task 1: Explain why the empirical covariance matrix (with a factor (1/(n-1) in front) of the 5 variables X1, X2, X3, X4, X5 can be calculated within R by using the following commands:

```
X <- cbind(x1,x2,x3,x4,x5)
n <- nrow(X)
S <- 1/(n-1) * t(X)%*%(diag(n)-1/n*matrix(1,n,n))%*%X
```

Task 2: Using matrices to calculate linear regression

Now you try the following exercise: using R, enter the following matrix $X =$

Enter the output $\boldsymbol{y}$ which is supposed to be a column-vector containing the 5 output observations 1, 5, 9, 23, 36. Calculate the least-squares estimate $\boldsymbol{b} = (X'X)^{-1}X'\boldsymbol{y}$. Next, using it, calculate the predictions $\hat{\boldsymbol{y}} = X\boldsymbol{b}$, the residuals $\boldsymbol{r} = \boldsymbol{y} - \hat{\boldsymbol{y}}$, the sum of squares of residuals (SSE), the degress of freedom $df = \mathrm{nrow}(X) - \mathrm{ncol}(X)$ and the mean-squared error error estimate $\mathrm{MSE} = \mathrm{SSE} \,/\, \mathrm{df}$ for the regression equation

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \varepsilon.$$

Print out the data and all the calculated quantities.

Now, create within R a function called `regress(X,y)` which uses as entries a matrix `X` and a vector `y`, calculates the above quantities and prints them out. Test the function by using the values of `X` and `y` as above (you are supposed to get the same results as before). Then extend the matrix `X` by adding one more row of values: (1,6,36) to the existing 5 rows, add a sixth output value (42) to the `y` vector and recalculate the quantities by just calling the subroutine again with the new entries for `X` and `y`.

Hint: To extend the matrix `X` by one more row `Z1` of suitable dimension you can write `X <- rbind(X,Z1)`, and to extend it by one more column `Z2` of suitable dimension you could write `X <- cbind(X,Z2)`.

Task 3: Matrix power

Implement a function `matpow(X, p)` that takes a symmetric, positive definite matrix `X` and a scalar `p` and returns `X` taken to the power `p`, including for negative and fractional values of `p`.

Use it to evaluate the square, the square root, the inverse, and the square root of the inverse of the following matrix:

```
2  2  1
2  3  2
1  2  4
```

Then, check your answers using basic matrix operations. (For example, compare `matpow(X,2)` with `X` multiplied by itself, check the inverse against `solve()`, and check that the square root of the matrix multiplied by itself recovers the original matrix.)

Hint: See the example for the `eigen()` function above and the lecture notes on the relationship between eigendecomposition and matrix powers.

Topic 1: Exploratory data analysis

Welcome to Week 1

Dr Pavel Krivitsky gives you a brief overview of topics and concepts we'll be covering in this week.

Transcript

Weekly learning outcomes

- Describe the basic properties of multivariate normal distribution.
- Diagnose deviations from normality in real-world datasets.
- Obtain and plot point estimates and confidence intervals for vector means and differences of vector means.
- Test linear hypotheses about vector means and differences of vector means.
- Explain the assumptions underlying the above inferential procedures.
- Use R packages to perform analyses.

Topics we will cover are:

- Topic 1: Exploratory data analysis of multivariate data

- Topic 2: The multivariate normal distribution
- Topic 3: Estimation of vector mean and of variance matrices: point estimates
- Topic 4: Confidence intervals and hypothesis tests for the mean vector

Optional readings

An alternative presentation of the concepts for this week can be found in:

Johnson, R. A., & Wichern, D. (2008). *Applied Multivariate Statistical Analysis* (6th ed.). Pearson Prentice Hall.

- 4.1–4.2, 4.6
- 5.1–5.5 and 6.1–6.3

All readings are available from the course Leganto reading list. Please keep in mind that you will need to be logged into Moodle to access the Leganto reading list.

Questions about this week's topics?

This week's topics were prepared by Dr P. Krivitsky. If you have any questions or comments, please post them under Discussion or email directly: p.krivitsky@unsw.edu.au

Exploratory data analysis of multivariate data

Introduction

We begin by taking a look at how to summarise multivariate data—with a focus on quantitative data—and to visualise it. Watch the following by Dr Pavel Krivitsky video exploring the concepts discussed in this section.

Transcript

Data organisation

Assume, we are dealing with $p \geq 1$ *variables*. The values of these variables are all recorded for each distinct *item*, *individual*, or *experimental trial*. Each of these three words will be substituted sometimes by the word "case". We will use the notation x_{ij} to indicate a particular value of the ith variable that is observed on the jth case. Consequently, n measurements on p variables can be represented in a form of a matrix

$$\overset{p\times n}{X} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1j} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2j} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i1} & x_{i2} & \cdots & x_{ij} & \cdots & x_{in} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{p1} & x_{p2} & \cdots & x_{pj} & \cdots & x_{pn} \end{pmatrix} \tag{1.1}$$

The matrix X above contains the data consisting of all the observations on all the variables. This way of representing the data allows easy manipulations to be performed in order to obtain some easy descriptive statistics for each of the variables.

Basic summaries

For example, the *sample mean* of the second variable is just $\bar{x}_2 = \frac{1}{n}\sum_{j=1}^{n} x_{2j}$, the *sample variance* of the second variable is just $s_2^2 = \frac{1}{n}\sum_{j=1}^{n}(x_{2j} - \bar{x}_2)^2$ (Note that for the sample variance we shall sometimes use the divisor of $n-1$ rather than n and each time this will be differentiated by displaying the appropriate expression).

The *sample covariance* (the simple measure of linear association between variables 1 and 2) is given by $s_{12} = \frac{1}{n}\sum_{j=1}^{n}(x_{1j} - \bar{x}_1)(x_{2j} - \bar{x}_2)$ and one can understand easily how $s_{ik}, i = 1, 2, \ldots, p$, $k = 1, 2, \ldots, p$ can be defined. Finally, the *sample correlation coefficient* (the measure of linear association between two variables that does not depend on the units of measurement) can be defined. The sample correlation coefficient of the ith and kth variables is defined by $r_{ik} = \frac{s_{ik}}{\sqrt{s_{ii}}\sqrt{s_{kk}}}$. Because of the well-known Cauchy–Bunyakovsky–Schwartz Inequality, $|r_{ik}| \leq 1$ holds. Note also that $r_{ik} = r_{ki}$ for all $i = 1, 2, \ldots, p$ and $k = 1, 2, \ldots, p$ holds.

It should be repeatedly noted that the sample correlations and covariance are useful only when trying to measure the *linear* association between two variables. Their value is less informative and is misleading in cases of *nonlinear* association. In this case one needs to invoke the *correlation quotient* instead.

Since covariance and correlation coefficients are routinely calculated and analysed they are very widely used and provide useful numerical summaries of association when the data do not exhibit obvious nonlinear patterns of association.

The descriptive statistics that we discussed until now are usually organised into arrays, namely:

- **Vector of sample means** $\bar{x} = \begin{pmatrix} \bar{x}_1 & \bar{x}_2 & \cdots & \bar{x}_p \end{pmatrix}^\top$
- **Matrix of sample variances and covariances**

$$\overset{n\times n}{S} = \begin{pmatrix} s_{11} & s_{12} & \cdots & s_{1p} \\ s_{21} & s_{22} & \cdots & s_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ s_{p1} & s_{p2} & \cdots & s_{pp} \end{pmatrix} \tag{1.2}$$

- **Matrix of sample correlations**

$$\underset{n\times n}{R} = \begin{pmatrix} 1 & r_{12} & \cdots & r_{1p} \\ r_{21} & 1 & \cdots & r_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ r_{p1} & r_{p2} & \cdots & 1 \end{pmatrix} \tag{1.3}$$

Visualisation

Dr Pavel Krivitsky introduces this section in the video below.

Transcript

Initial graphical representation of the data

Some simple characteristics of the data are worth studying before the actual multivariate analysis would begin:

- drawing a scatterplot of the data
- calculating simple univariate descriptive statistics for each variable
- calculating sample correlation and covariance coefficients
- linking multiple two-dimensional scatterplots

R

In R, these are implemented in `base::rowMeans`, `base::colMeans`, `stats::cor`, `graphics::plot`, `graphics::pairs`, `GGally::ggpairs`. Here, the format is *`PACKAGE`*`::`*`FUNCTION`*, and you can learn more by running

```
library(PACKAGE)
```

```
? FUNCTION
```

Next, work through the basic multivariate summaries and visualisation example in the next section.

Example: R activity on basic multivariate summaries and visualisation

The first practice example will demonstrate basic multivariate summary statistics and graphics. You can use the RStudio Console here or complete the exercise in your own RStudio.

To complete this task select the 'Multivariate_exploration_visualisation.Rmd' in the 'Files' section of RStudio. Follow the example contained within the RMD file.

If you choose to complete the example in your own RStudio, upload the following file:

 Multivariate_exploration_visualisation.Rmd

The contents of the RMD file are also displayed below:

The following example demonstrates basic multivariate summary statistics and graphics.

The "Iris" dataset is built into the `stats` package, so we can acquire it via the `data()` function:

```
data(iris)
iris
```

Basic data summaries

The following illustrates both built-in summary functions and `dplyr`.

To obtain basic 5-number summaries,

```
summary(iris)
```

More generally, we can use the `apply` function to obtain univariate summaries for each variable:

```
# Column means:
colMeans(iris[,-5])
# Column standard deviations:
apply(iris[-5], 2, sd)
# Column 95th %iles:
apply(iris[-5], 2, quantile, 0.95) # This evaluates: quantile(x, 0.95) for x each
column in turn.
```

Similarly, we can calculate multivariate summaries:

```
# Variance-covariance matrix
cov(iris[,-5])
```

```
# Correlation matrix
cor(iris[,-5])
```

We can also apply to subgroups. For this, we use the "pipe" operator `%>%` from package `magrittr`, which evaluates each function in turn, passing its results as the first argument of the next one (or `.`). We also use `map` function from the `purrr` package.

```
library(magrittr)
library(purrr)
iris %>% # Start with dataset iris
  split(iris$Species) %>% # Split it according to its species column
  map(~.[-5]) %>% # For each species, throw away the last column (Species)
  map(cov)
```

Visualisation

The most common visualisation for multivariate data is a pairwise plot: plot every variable against every other variable. A function that does it well---and automatically takes care of categorical variables---is `GGally`'s `ggpairs` function.

```
library(GGally)
# aes() here is optional, telling it to colour-code in accodance with species, and
using transparency (alpha).
ggpairs(iris, aes(colour=Species, alpha = 0.7))
```

Topic 2: The multivariate normal distribution

The multivariate normal distribution

This course will make heavy use of multivariate normal distribution. This distribution generalises the familiar normal distribution to multiple dimensions and provides a convenient way to represent many variables and their (pairwise) dependence on each other.

Before we begin examing multivariate normal distribution, watch the below video by Dr Pavel Krivitsky exploring the concepts discussed in this topic.

Transcript

The *multivariate normal* (*MVN*) density is a generalisation of the univariate normal for $p \geq 2$ dimensions. Looking at the term $(\frac{x-\mu}{\sigma})^2 = (x-\mu)(\sigma^2)^{-1}(x-\mu)$ in the exponent of the well known formula

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \mathrm{e}^{-[(x-\mu)/\sigma]^2/2}, -\infty < x < \infty \tag{1.4}$$

for the univariate density function, a natural way to generalise this term in higher dimensions is to *replace* it by $(\boldsymbol{x}-\boldsymbol{\mu})^\top \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})$. Here $\boldsymbol{\mu} = \mathrm{E}\,\boldsymbol{X} \in \mathbb{R}^p$ is the expected value of the random vector $\boldsymbol{X} \in \mathbb{R}^p$ and the matrix

$$\Sigma = \mathrm{E}(\boldsymbol{X}-\boldsymbol{\mu})(\boldsymbol{X}-\boldsymbol{\mu})^\top = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1p} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{p1} & \sigma_{p2} & \cdots & \sigma_{pp} \end{pmatrix} \in \mathcal{M}_{p,p}$$

is the *covariance matrix*. Note that on the diagonals of Σ we get the *variances* of each of the p random variables whereas $\sigma_{ij} = \mathrm{E}[(X_i - \mathrm{E}(X_i))(X_j - \mathrm{E}(X_j))], i \neq j$ are the *covariances* between the ith and jth random variables. Sometimes, we will also denote σ_{ii} by σ_i^2.

Of course, the above replacement would only make sense if Σ were positive definite. In general, however, we can only claim that Σ is (as any covariance matrix) non-negative definite.

If Σ were positive definite, then the density of the random vector $\boldsymbol{X}$ can be written as

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{\frac{1}{2}}}\, \mathrm{e}^{-(\boldsymbol{x}-\boldsymbol{\mu})^\top \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})/2},\ -\infty < x_i < \infty,\ i = 1, 2 \ldots, p. \tag{1.5}$$

It can be directly checked that the random vector $\boldsymbol{X} \in \mathbb{R}^p$ has $\mathrm{E}\,\boldsymbol{X} = \boldsymbol{\mu}$ and

$$\mathrm{E}[(\boldsymbol{X}-\boldsymbol{\mu})(\boldsymbol{X}-\boldsymbol{\mu})^\top] = \Sigma.$$

Since the density is uniquely defined by the *mean vector and the covariance matrix* we will denote it by $N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

However, we will introduce the multivariate normal distribution not through its density formula but through more general reasoning that also allows to cover the case of singular Σ. We will utilise the famous **Cramer–Wold argument** according to which the distribution of a p-dimensional random vector $\boldsymbol{X}$ is completely characterised by the one-dimensional distributions of **all** linear transformations $\boldsymbol{t}^\top \boldsymbol{X}, \boldsymbol{t} \in \mathbb{R}^p$. Hence the following definition will be adopted here:

Definition 1.1. The random vector $\boldsymbol{X} \in \mathbb{R}^p$ has a multivariate normal distribution if and only if (iff) any linear transformation $\boldsymbol{t}^\top \boldsymbol{X}, \boldsymbol{t} \in \mathbb{R}^p$ has a univariate normal distribution.

Theorem 1.1. *Suppose that for a random vector $\boldsymbol{X} \in \mathbb{R}^p$ with a normal distribution according to Definition 1.1 we have $\mathrm{E}(\boldsymbol{X}) = \boldsymbol{\mu}$ and $D(\boldsymbol{X}) = \mathrm{E}[(\boldsymbol{X}-\boldsymbol{\mu})(\boldsymbol{X}-\boldsymbol{\mu})^\top] = \Sigma$. Then for any fixed $\boldsymbol{t} \in \mathbb{R}^p$, $\boldsymbol{t}^\top X \sim N(\boldsymbol{t}^\top \boldsymbol{\mu}, \boldsymbol{t}^\top \Sigma \boldsymbol{t})$ i.e. $\boldsymbol{t}^\top \boldsymbol{X}$ has an one dimensional normal distribution with expected value $\boldsymbol{t}^\top \boldsymbol{\mu}$ and variance $\boldsymbol{t}^\top \Sigma \boldsymbol{t}$.*

As an upshot, we see that given the expected value vector $\boldsymbol{\mu}$ and the covariance matrix Σ we can still use its properties in cases of singular (i.e. non-invertible) Σ.

Properties of multivariate normal

The following *properties* of multivariate normal are useful in the rest of the course.

Property 1. If $\Sigma = D(\boldsymbol{X}) = \Lambda$ is a diagonal matrix then the p components of $\boldsymbol{X}$ are independent.

The above property can be paraphrased as "for a multivariate normal, if its components are uncorrelated they are also independent". On the other hand, it is well known that *always, i.e. not only for normal* from the fact that certain components are independent we can conclude that they are also uncorrelated. Therefore, for the **multivariate normal distribution** we can conclude that its components are **independent if and only if they are uncorrelated!**

Property 2. If $\boldsymbol{X} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $C \in \mathcal{M}_{q,p}$ is an arbitrary matrix of real numbers then

$$Y = CX \sim N_q(C\boldsymbol{\mu}, C\Sigma C^\top).$$

Note also that if it happens that the rank of C is full and if $\mathrm{rk}(\boldsymbol{\Sigma}) = p$ then the rank of $C\Sigma C^\top$ is also full, i.e. the distribution of $\boldsymbol{Y}$ would not be degenerate in this case.

Property 3. (This is a finer version of Property 1). Assume the vector $\boldsymbol{X} \in \mathbb{R}^p$ is divided into subvectors $\boldsymbol{X} = \begin{pmatrix} \boldsymbol{X}_{(1)} \\ \boldsymbol{X}_{(2)} \end{pmatrix}$ and according to this subdivision the vector means are $\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_{(1)} \\ \boldsymbol{\mu}_{(2)} \end{pmatrix}$ and the covariance matrix Σ has been subdivided into $\Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}$. Then the vectors $\boldsymbol{X}_{(1)}$ and $\boldsymbol{X}_{(2)}$ are independent iff $\Sigma_{12} = 0$.

Property 4. Let the vector $\boldsymbol{X} \in \mathbb{R}^p$ be divided into subvectors $\boldsymbol{X} = \begin{pmatrix} \boldsymbol{X}_{(1)} \\ \boldsymbol{X}_{(2)} \end{pmatrix}$, $\boldsymbol{X}_{(1)} \in \mathbb{R}^r, r < p$, $\boldsymbol{X}_{(2)} \in \mathbb{R}^{p-r}$ and according to this subdivision the vector means are $\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_{(1)} \\ \boldsymbol{\mu}_{(2)} \end{pmatrix}$ and the covariance matrix Σ has been subdivided into $\Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}$. Assume for simplicity that the rank of Σ_{22} is full. Then the conditional density of $\boldsymbol{X}_{(1)}$ given that $\boldsymbol{X}_{(2)} = \boldsymbol{x}_{(2)}$ is

$$N_r(\boldsymbol{\mu}_{(1)} + \Sigma_{12}\Sigma_{22}^{-1}(\boldsymbol{x}_{(2)} - \boldsymbol{\mu}_{(2)}), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21})$$

Example 1.1. As an immediate consequence of Property 4 we see that if $p = 2, r = 1$ then for a two-dimensional normal vector $\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim N\left\{ \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{pmatrix} \right\}$, its conditional density $f(x_1|x_2)$ is $N(\mu_1 + \frac{\sigma_{12}}{\sigma_2^2}(x_2 - \mu_2), \sigma_1^2 - \frac{\sigma_{12}}{\sigma_2^2})$.

Optional Exercise: Try to derive the above result by direct calculations starting from the joint density $f(x_1, x_2)$, going over to the marginal $f(x_2)$ by integration and finally getting $f(x_1|x_2) = \frac{f(x_1,x_2)}{f(x_2)}$.

Property 5. If $\boldsymbol{X} \sim N_p(\boldsymbol{\mu}, \Sigma)$ and Σ is nonsingular then $(\boldsymbol{X} - \boldsymbol{\mu})^\top \Sigma^{-1} (\boldsymbol{X} - \boldsymbol{\mu}) \sim \chi^2_p$ where χ^2_p denotes the chi-square distribution with p degrees of freedom.

Remark 1.1. This stems from the fact that the vector $\boldsymbol{Y} \in \mathbb{R}^p : \boldsymbol{Y} = \Sigma^{-\frac{1}{2}} (\boldsymbol{X} - \boldsymbol{\mu}) \sim N(0, I_p)$ i.e. it has p independent standard normal components, where $\Sigma^{-\frac{1}{2}}$ is defined either via the spectral decomposition (0.8) or the Cholesky Decomposition on slide "Numerical stability and Cholesky decomposition" Then

$$(\boldsymbol{x} - \boldsymbol{\mu})^\top \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) = \boldsymbol{Y}^\top \boldsymbol{Y} = \sum_{i=1}^{p} Y_i^2 \sim \chi^2_p$$

according to the definition of χ^2_p as a distribution of the sum of squares of p independent standard normals. However, this multivariate version of standardising the normal distribution is useful in other situations, such as multivariate ANOVA and regression diagnostics.

Finally, one more interpretation of the result in Property 4 will be given. Assume we want, as is a typical situation in statistics, to predict a random variable Y that is correlated with some p random variables (predictors) $\boldsymbol{X} = \begin{pmatrix} X_1 & X_2 & \cdots & X_p \end{pmatrix}$. Trying to find the *best predictor* of Y we would like to minimise the expected value $\mathrm{E}_Y (Y - g(\boldsymbol{X}) | \boldsymbol{X} = \boldsymbol{x})^2$ over all possible choices of the function g such that $\mathrm{E}\, g(\boldsymbol{X})^2 < \infty$. A little careful work and use of basic properties of conditional expectations leads us (see lecture) to the conclusion that the optimal solution to the above minimisation problem is $g^*(\boldsymbol{x}) = \mathrm{E}(Y | \boldsymbol{X} = \boldsymbol{x})$. This optimal solution is also called the *regression function*. Thus given a particular realisation $\boldsymbol{x}$ of the random vector $\boldsymbol{X}$ the regression function is just the conditional expected value of Y given $\boldsymbol{X} = \boldsymbol{x}$.

In general, the conditional expected value may be a complicated nonlinear function of the predictors. However, if we assume *in addition* that the joint $(p+1)$-dimensional distribution of Y and $\boldsymbol{X}$ is **normal** then by applying Property 4 we see that given the realisation $\boldsymbol{x}$ of $\boldsymbol{X}$, the best prediction of the Y value is given by $b + \sigma_0^\top C^{-1} \boldsymbol{x}$ where $b = \mathrm{E}(Y) - \sigma_0^\top C^{-1} \mathrm{E}(\boldsymbol{X})$, C is the covariance matrix of the vector $\boldsymbol{X}$, σ_0 is the vector of Covariances of Y with $X_i, i = 1, \ldots, p$.

Indeed, we know that when the joint $(p+1)$-dimensional distribution of Y and $\boldsymbol{X}$ is **normal** the regression function is given by

$$\mathrm{E}(Y) + \sigma_0^\top C^{-1} (\boldsymbol{x} - \mathrm{E}(\boldsymbol{X})).$$

By introducing the notation $b = \mathrm{E}(Y) - \sigma_0^\top C^{-1} \mathrm{E}(\boldsymbol{X})$ we can write this as $b + \sigma_0^\top C^{-1} \boldsymbol{x}$.

That is, **in case of normality, the optimal predictor of Y in the least squares sense turns out to be a very simple linear function of the predictors**. The vector $C^{-1}\sigma_0 \in \mathbb{R}^p$ is the *vector of the regression coefficients*. Substituting the optimal values we get the minimal value of the sum of squares which is equal to $V(Y) - \sigma_0^\top C^{-1} \sigma_0$.

Tests for multivariate normality

We have seen that the assumption of multivariate normality may bring essential simplifications in analysing data. But applying inference methods based on the multivariate normality assumption in cases where it is grossly violated may introduce serious defects in the quality of the analysis. It is therefore important to be able to check the multivariate normality assumption. Based on the properties of normal distributions discussed in this topic, we know that all linear combinations of normal variables are normal and the contours of the multivariate normal density are ellipsoids.

Before exploring this further, watch the below video by Dr Pavel Krivitsky.

Transcript

Therefore we can (to some extent) check the multivariate normality hypothesis by:

1. checking if the marginal distributions of each component appear to be normal (by using Q–Q plots, for example);
2. checking if the scatterplots of pairs of observations give the elliptical appearance expected from normal populations;
3. are there any outlying observations that should be checked for accuracy.

All this can be done by applying univariate techniques and by drawing scatterplots, which are well developed in R. To some extent, however, there is a price to be paid for concentrating on univariate and bivariate examinations of normality.

There is a need to construct a "good" overall test of multivariate normality. One of the simple and tractable ways to verify the multivariate normality assumption is by using tests based on **Mardia's multivariate skewness and kurtosis measures**. For any general multivariate distribution we define these respectively as

$$\beta_{1,p} = \mathrm{E}[(\boldsymbol{Y} - \boldsymbol{\mu})^\top \Sigma^{-1} (\boldsymbol{X} - \boldsymbol{\mu})]^3 \tag{1.6}$$

provided that $\boldsymbol{X}$ is independent of $\boldsymbol{Y}$ but has the same distribution and

$$\beta_{2,p} = \mathrm{E}[(\boldsymbol{X} - \boldsymbol{\mu})^\top \Sigma^{-1} (\boldsymbol{X} - \boldsymbol{\mu})]^2 \tag{1.7}$$

(if the expectations in (1.6) and (1.7) exist). For the $N_p(\boldsymbol{\mu}, \Sigma)$ distribution: $\beta_{1,p} = 0$ and $\beta_{2,p} = p(p+2)$.

(Note that when $p = 1$, the quantity $\beta_{1,1}$ is the square of the skewness coefficient $\frac{\mathrm{E}(X-\mu)^3}{\sigma^3}$ whereas $\beta_{2,1}$ coincides with the kurtosis coefficient $\frac{\mathrm{E}(X-\mu)^4}{\sigma^4}$.)

For a sample of size n *consistent estimates* of $\beta_{1,p}$ and $\beta_{2,p}$ can be obtained as

$$\hat{\beta}_{1,p} = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} g_{ij}^3$$

$$\hat{\beta}_{2,p} = \frac{1}{n} \sum_{i=1}^{n} g_{ii}^2$$

where $g_{ij} = (\boldsymbol{x}_i - \bar{\boldsymbol{x}})^\top \boldsymbol{S}_n{}^{-1} (\boldsymbol{x}_j - \bar{\boldsymbol{x}})$. Notice that for $\hat{\beta}_{1,p}$, we take advantage of our sample being independent and use observations $\boldsymbol{x}_j$ for $j \neq i$ as the "$\boldsymbol{Y}$" values for $\boldsymbol{x}_i$.

Both quantities $\hat{\beta}_{1,p}$ and $\hat{\beta}_{2,p}$ are nonnegative and for multivariate data, one would expect them to be around zero and $p(p+2)$, respectively. Both quantities can be utilised to detect departures from multivariate normality.

Mardia has shown that asymptotically, $k_1 = n\hat{\beta}_{1,p}/6 \sim \chi^2_{p(p+1)(p+2)/6}$, and $k_2 = [\hat{\beta}_{2,p} - p(p+2)]/[8p(p+2)/n]^{\frac{1}{2}}$ is standard normal. Thus we can use k_1 and k_2 to test the null hypothesis of multivariate normality. If neither hypothesis is rejected, the multivariate normality assumption is in reasonable agreement with the data. It also has been observed that Mardia's multivariate kurtosis can be used as a measure to detect outliers from the data that are supposedly distributed as multivariate normal. Mardia's multivariate kurtosis can also be used to detect outliers.

***Remark* 1.2** (Overreliance on hypothesis tests). Shapiro–Wilk, Mardia, and other distribution tests have, as their null hypothesis, that the true population distribution is (multivariate) normal. This means that if the population distribution deviates from normality even a little, then as the sample size increases, the power of the test (the probability of rejecting the null hypothesis of normality) approaches 1.

At the same time, as the sample size increases, the Central Limit Theorem tells us that many statistics,

including sample means and (much more slowly) sample variances and covariances, approach normality—and multivariate statistics generally approach multivariate normality. This means that regardless of the underlying distribution, the statistical procedures depending on the normality assumption become valid—even as the chances that a statistical hypothesis test will detect what non-normality there is approaches 1.

This means that we must not rely on hypothesis testing blindly but consider the situation on a case-by-case basis, particularly when dealing with large datasets. For a decent sample size, the "symmetric, bell-shaped" heuristic may indicate an adequate distribution, even if a hypothesis test reports a small p-value.

Check your understanding

Complete the below exercises to check your understanding of concepts presented so far.

1. Let $\boldsymbol{X} = [X_1, X_2]^\top$ a random vector with $\mathrm{E}(\boldsymbol{X}) = \boldsymbol{\mu}$ and

$$\mathrm{Var}(\boldsymbol{X}) = \Sigma = \sigma^2 \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$$

a) Find $\mathrm{Cov}(X_1 - X_2, X_1 + X_2)$.

b) Find $\mathrm{Cov}(X_1, X_2 - \rho X_1)$.

c) Choose b to minimise $\mathrm{Var}(X_2 - bX_1)$.

2. Let X_1 and X_2 denote i.i.d. $N(0, 1)$ r.v.'s.

a) Show that the r.v.'s $Y_1 = X_1 - X_2$ and $Y_2 = X_1 + X_2$ are **independent**, and find their marginal densities.

b) Find $P\left(X_1^2 + X_2^2 < 2.41\right)$.

3. Let $\boldsymbol{X} \sim N_3(\boldsymbol{\mu}, \Sigma)$ where

$$\boldsymbol{\mu} = \begin{pmatrix} 3 \\ -1 \\ 2 \end{pmatrix} \quad \Sigma = \begin{pmatrix} 3 & 2 & 1 \\ 2 & 3 & 1 \\ 1 & 1 & 2 \end{pmatrix}$$

a) For $A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \end{pmatrix}$ find the distribution of $\boldsymbol{Z} = A\boldsymbol{X}$ and find the correlation between the two components of $\boldsymbol{Z}$.

b) Find the conditional distribution of $[X_1, X_3]^\top$ given $X_2 = 0$.

Solutions to these exercises will be provided in the 'Quizzes' discussion forum.

Example: The multivariate normal distribution

Before working through the demonstration and challenge in the following sections, watch the below example by Dr Pavel Krivitsky.

Transcript

Demonstration: Multivariate normal distribution

This demonstration can be completed using the provided RStudio or your own RStudio.

To complete this task select the 'MVN_Examples.demo.Rmd' in the 'Files' section of RStudio. Follow the demonstration contained within the RMD file.

If you choose to complete the example in your own RStudio, upload the following file:

MVN_Examples.demo.Rmd

After working through the demonstration, complete the associated 'Challenge' on the next slide.

The contents of the RMD file are also displayed below:

Packages

```
library(here)
library(mvtnorm)
library(GGally)
library(MVN)
library(dplyr)
library(readr)
```

Multivariate normal random variables

Generating

The `*mvnorm` family of functions from the `mvtnorm` can be used to calculate densities and generate values from the multivariate normal. We will use the `ggpairs` function found in package `GGally`:

```
(mu <- c(2,3,4))
(Sigma <- diag(1:3) + 1)
x <- rmvnorm(1000, mu, Sigma)
dim(x)
```

we see that `x` has samples in rows and variables in columns.

Let's plot these variables:

```
ggpairs(as.data.frame(x))
```

Marginally normal vs. jointly normal

Consider two variables $Z_1 = (2W-1)Y$ and $Z_2 = Y$, where $Y \sim N(0,1)$ and, independently, $W \sim \text{Binomial}(1, 1/2)$ (so $2W-1$ takes -1 and $+1$ with equal probabilities).

```
y <- rnorm(1000)
w <- rbinom(1000,1,1/2)
z <- cbind((2*w-1)*y, y)
ggpairs(as.data.frame(z))
```

We see that they are normal marginally yet clearly not jointly.

What happens when we use Mardia's diagnostics, using the `mvn` function in the `MVN` package?

```
mvn(z)
```

We see that normality tests for individual dimensions (i.e., Shapiro--Wilk) pass, but Mardia's kurtosis does not.

Lastly, let us try viewing some *projections* of this distribution. For a true multivariate normal distribution, all projections are also normal. Here,

```
zp1 <- z%*%c(1,-1)
plot(density(zp1))
```

Contrasting with a true MVN, even correlated,

```
xp1 <- x%*%c(1,-1,0)
plot(density(xp1))
xp2 <- x%*%rnorm(3) # A random projection.
plot(density(xp2))
```

Diagnosing multivariate normality

These data, taken from Johnson and Wichern (2007), consist of measurements of radiation emitted from 42 randomly chosen microwave ovens, with doors open and closed. Load the data and plot:

```
ovens <- read_csv(here("datasets","ovens.csv"))
ggpairs(ovens)
```

Clearly, the data are not even close to MVN. Both variables are very strongly right-skewed.

Let's see the hypothesis tests:

```
mvn(ovens)
```

We see that the multivariate tests for skewness and kurtosis reject the null hypotheses at any reasonable threshold.

Now, let's try transforming the data. It should be noted that transformations can change the interpretation of the results.

```
ovens4 <- ovens^(1/4) # Take 4th root.
ggpairs(ovens4)
```

We now see far more normal-like shapes; though the `open` variable looks like it might be slightly bimodal or right-skewed.

Let's try the tests:

```
mvn(ovens4)
```

We see that both skewness and kurtosis are not inconsistent with a multivariate normal distribution.

Challenge: Multivariate normal distribution

If you choose to complete this task in your own RStudio, upload the following file:

Click on the 'MVN_Examples.challenge.Rmd' in the 'Files' section to begin. Enter your response to the tasks in the 'Enter your code here' section.

MVN_Examples.challenge.Rmd

This activity and the solution will be discussed at the Collaborate session this week. In the meantime, share and discuss your results in the 'Tutorials' discussion forum.

The solution will also be available here on Friday of this week by clicking on the 'Solution' tab in the top right corner.

The contents of the RMD file are also displayed below:

Packages

```
library(here)
library(mvtnorm)
library(GGally)
library(MVN)
library(dplyr)
library(readr)
```

Challenge

Now, consider an extract from a dataset containing a number of chemical properties of Portuguese Vinho Verde wines and their quality, taken from the UCI Machine Learning Repository. Eleven properties of a wine were measured, along with a subjective quality score. This was done for red and for white wines. For the purposes of this exercise, we will focus on the following 7 variables:

- `fixed acidity`
- `volatile acidity`
- `free sulfur dioxide`
- `total sulfur dioxide`
- `density`
- `pH`
- `sulphates`

Let's explore the data and see if any variables would need to be transformed for multivariate analysis:

```
wine <- read_csv(here("datasets","winequality-red.extract7.csv"))
```

Task 1: Make a pairwise plot of this dataset and identify which variables may deviate from normality. Perform the hypothesis tests. Do the results agree?

Task 2: Find an appropriate transformation for these variables, and check that they are now normal.

Topic 3: Estimation of vector mean and of variance matrices: point estimates

Estimation of the mean vector and covariance matrix of multivariate normal distribution

In the previous topic, we have derived the multivariate normal distribution and its basic properties. In order to make use of them, we must now estimate this distribution from observed data. This topic derives the estimators for the parameters of this distribution and some properties of these estimators.

Maximum Likelihood Estimation

Optional viewing: Maximum Likelihood estimation - An introduction

Likelihood function

Transcript

Suppose we have observed n independent realisations of p-dimensional random vectors from $N_p(\boldsymbol{\mu}, \Sigma)$. Suppose for simplicity that Σ is non-singular. The data matrix has the form

$$X = \begin{pmatrix} X_{11} & X_{12} & \cdots & X_{1j} & \cdots & X_{1n} \\ X_{21} & X_{22} & \cdots & X_{2j} & \cdots & X_{2n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ X_{i1} & X_{i2} & \cdots & X_{ij} & \cdots & X_{in} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ X_{p1} & X_{p2} & \cdots & X_{pj} & \cdots & X_{pn} \end{pmatrix} = [X_1, X_2, \ldots, X_n] \tag{1.8}$$

The goal to estimate the unknown mean vector and the covariance matrix of the multivariate normal distribution by the Maximum Likelihood Estimation (MLE) method.

Based on our knowledge from the previous topic we can write down the *likelihood function*

$$L(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma) = (2\pi)^{-\frac{np}{2}} |\Sigma|^{-\frac{n}{2}} \mathrm{e}^{-\frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{x}_i - \boldsymbol{\mu})^\top \Sigma^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu})} \tag{1.9}$$

(Note that we have substituted the observations above and consider L as a function of the unknown parameters $\boldsymbol{\mu}, \Sigma$ only.) Correspondingly, we get the *log-likelihood function* in the form

$$\log L(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma) = -\frac{np}{2}\log(2\pi) - \frac{n}{2}\log(|\Sigma|) - \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{x}_i - \boldsymbol{\mu})^\top \Sigma^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}) \tag{1.10}$$

It is well known that maximising either (1.9) or (1.10) will give the same solution for the MLE.

We start deriving the MLE by trying to maximise (1.10). To this end, first note that by utilising properties of traces from slide "Vectors and matrices", we can transform:

$$\sum_{i=1}^{n} (\boldsymbol{x}_i - \boldsymbol{\mu})^\top \Sigma^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}) = \sum_{i=1}^{n} \operatorname{tr}\left[\Sigma^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}) (\boldsymbol{x}_i - \boldsymbol{\mu})^\top\right] =$$

$$\operatorname{tr}\left[\Sigma^{-1} \left(\sum_{i=1}^{n} (\boldsymbol{x}_i - \boldsymbol{\mu}) (\boldsymbol{x}_i - \boldsymbol{\mu})^\top\right)\right] =$$

(by adding $\pm\overline{\boldsymbol{x}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i$ to each term $(\boldsymbol{x}_i - \boldsymbol{\mu})$ in $\sum_{i=1}^{n} (\boldsymbol{x}_i - \boldsymbol{\mu}) (\boldsymbol{x}_i - \boldsymbol{\mu})^\top\Big)$

$$\operatorname{tr}\left[\Sigma^{-1} \left(\sum_{i=1}^{n} (\boldsymbol{x}_i - \bar{\boldsymbol{x}}) (\boldsymbol{x}_i - \bar{\boldsymbol{x}})^\top + n(\bar{\boldsymbol{x}} - \boldsymbol{\mu})(\bar{\boldsymbol{x}} - \boldsymbol{\mu})^\top\right)\right]$$

$$= \operatorname{tr}\left[\Sigma^{-1} \left(\sum_{i=1}^{n} (\boldsymbol{x}_i - \bar{\boldsymbol{x}}) (\boldsymbol{x}_i - \bar{\boldsymbol{x}})^\top\right)\right] + n(\bar{\boldsymbol{x}} - \boldsymbol{\mu})^\top \Sigma^{-1} (\bar{\boldsymbol{x}} - \boldsymbol{\mu})$$

Thus

$$\begin{aligned} \log L(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma) \;=\; & -\tfrac{np}{2}\log(2\pi) - \tfrac{n}{2}\log(|\Sigma|) - \tfrac{1}{2}\operatorname{tr}\left[\Sigma^{-1}\left(\textstyle\sum_{i=1}^{n} (\boldsymbol{x}_i - \bar{\boldsymbol{x}}) (\boldsymbol{x}_i - \bar{\boldsymbol{x}})^\top\right)\right] \\ & \tfrac{1}{2} n (\bar{\boldsymbol{x}} - \boldsymbol{\mu})^\top \Sigma^{-1} (\bar{\boldsymbol{x}} - \boldsymbol{\mu}) \end{aligned} \tag{1.11}$$

Maximum Likelihood Estimators

Transcript

The MLE are the ones that maximise (1.11). Looking at (1.11) we realise that (since Σ is non-

negative definite) the minimal value for $\frac{1}{2}n(\bar{\boldsymbol{x}}-\boldsymbol{\mu})^{\top}\Sigma^{-1}(\bar{\boldsymbol{x}}-\boldsymbol{\mu})$ is zero and is attained when $\boldsymbol{\mu}=\bar{\boldsymbol{x}}$. It remains to find the optimal value for Σ. We will use the following:

Theorem 1.2 (Anderson's lemma)**.** If $A\in\mathcal{M}_{p,p}$ *is symmetric positive definite, then the maximum of the function* $(G)=-n\log(|G|)-\operatorname{tr}\left(G^{-1}A\right)$*(defined over the set of symmetric positive definite matrices* $G\in\mathcal{M}_{p,p}$*) exists, occurs at* $G=\frac{1}{n}A$ *and has the maximal value of* $np\log(n)-n\log(|A|)-np$*.*

Using the structure of the log-likelihood function in (1.11) and Theorem (1.2) (applied for the case $A=\sum_{i=1}^{n}(\boldsymbol{x}_i-\bar{\boldsymbol{x}})(\boldsymbol{x}_i-\bar{\boldsymbol{x}})^{\top}$ (!)) it is now easy to formulate following:

Theorem 1.3. *Suppose* $\boldsymbol{X}_1,\boldsymbol{X}_2,\ldots,\boldsymbol{X}_n$ *is a random sample from* $N_p(\boldsymbol{\mu},\Sigma),p<n$*. Then*$\hat{\boldsymbol{\mu}}=\bar{\boldsymbol{X}}$ *and* $\hat{\boldsymbol{\Sigma}}=\frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i-\bar{\boldsymbol{x}})(\boldsymbol{x}_i-\bar{\boldsymbol{x}})^{\top}$ *are the* maximum likelihood estimators *of* $\boldsymbol{\mu}$ *and* Σ*, respectively.*

Application in correlation matrix estimation

The correlation matrix can be defined in terms of the elements of the covariance matrix Σ. The correlation coefficients $\rho_{ij},i=1,\ldots,p,j=1,\ldots,p$ are defined as $\rho_{ij}=\frac{\sigma_{ij}}{\sqrt{\sigma_{ii}}\sqrt{\sigma_{jj}}}$ where $\Sigma=(\sigma_{ij},i=1\ldots,p;j=1,\ldots,p)$ is the covariance matrix. Note that $\rho_{ii}=1,i=1,\ldots,p$. To derive the *MLE* of $\rho_{ij},i=1,\ldots,p,j=1,\ldots,p$ we note that these are continuous transformations of the covariances whose maximum likelihood estimators have already been derived. Then we can claim (*according to the transformation invariance properties of MLE*) that

$$\hat{\rho}_{ij}=\frac{\hat{\sigma}_{ij}}{\sqrt{\hat{\sigma}_{ii}}\sqrt{\hat{\sigma}_{jj}}},i=1,\ldots,p,j=1,\ldots,p. \tag{1.12}$$

Transcript

Distributions of MLE of mean vector and covariance matrix of multivariate normal distribution

Inference is not restricted to only finding point estimators but also to construct confidence regions, test hypotheses etc. To this end we need the distribution of the estimators (or of suitably chosen functions of them).

Sampling distribution of $\bar{\boldsymbol{X}}$

In the univariate case ($p = 1$) it is well known that for a sample of n observations from *normal distribution* $N(\mu, \sigma^2)$ the sample mean is normally distributed: $N(\mu, \frac{\sigma^2}{n})$. Moreover, the sample mean and the sample variance are *independent* in the case of sampling from a univariate normal population (Basu's Lemma). This fact was very useful in developing t-statistics for testing the mean vector. Do we have similar statements about the sample mean and sample variance in the multivariate ($p > 1$) case?

Let the random vector $\bar{\boldsymbol{X}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{X}_i \in \mathbb{R}^p$. For any $\boldsymbol{l} \in \mathbb{R}^p$: $\boldsymbol{l}^\top \bar{\boldsymbol{X}}$ is a linear combination of normals and hence is normal (see Definition 1.1). Since taking expected value is a linear operation, we have $\mathrm{E}\, \bar{\boldsymbol{X}} = \frac{1}{n} n\boldsymbol{\mu} = \boldsymbol{\mu}$; In analogy with the univariate case we could formally write $\operatorname{Cov} \bar{\boldsymbol{X}} = \frac{1}{n^2} n \operatorname{Cov} \boldsymbol{X}_1 = \frac{1}{n}\Sigma$ and hence $\bar{\boldsymbol{X}} \sim N_p(\boldsymbol{\mu}, \frac{1}{n}\Sigma)$.

If you are interested, see the next (optional) slide for a detailed proof.

Independence of $\bar{\boldsymbol{X}}$ and $\hat{\boldsymbol{\Sigma}}$

How can we show that $\bar{\boldsymbol{X}}$ and $\hat{\boldsymbol{\Sigma}}$ are independent? If you are interested, see the next (optional) slide, where we prove the following theorem:

Theorem 1.4. *For a sample of size n from $N_p(\boldsymbol{\mu}, \Sigma), p < n$ the sample average $\bar{\boldsymbol{X}} \sim N_p(\mu, \frac{1}{n}\Sigma)$. Moreover, the MLE $\hat{\boldsymbol{\mu}} = \bar{\boldsymbol{X}}$ and $\hat{\boldsymbol{\Sigma}}$ are independent.*

Sampling distribution of the MLE of Σ

Definition 1.2. A random matrix $\boldsymbol{U} \in \mathcal{M}_{p,p}$ has a **Wishart distribution** with parameters Σ, p, n (denoting this by $\boldsymbol{U} \sim W_p(\Sigma, n)$) if there exist n independent random vectors $\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_n$ each with $N_p(0, \Sigma)$ distribution such that the distribution of $\sum_{i=1}^{n} \boldsymbol{Y}_i \boldsymbol{Y}_i^\top$ coincides with the distribution of $\boldsymbol{U}$.

Note that we *require* that $p < n$ and that $\boldsymbol{U}$ be non-negative definite.

Having in mind the proof of Theorem 1.4 we can claim that the distribution of the matrix $n\hat{\boldsymbol{\Sigma}} = \sum_{i=1}^{n} (\boldsymbol{X}_i - \bar{\boldsymbol{X}})(\boldsymbol{X}_i - \bar{\boldsymbol{X}})^\top$ is the same as that of $\sum_{i=2}^{n} \boldsymbol{Z}_i \boldsymbol{Z}_i^\top$ and therefore is Wishart with

parameters $\Sigma, p, n-1$. That is, we can denote:

$$n\hat{\boldsymbol{\Sigma}} \sim W_p(\Sigma, n-1).$$

The density formula for the Wishart distribution is given in several sources but we will not deal with it in this course. Some properties of Wishart distribution will be mentioned though since we will make use of them later in the course:

1. If $p = 1$ and if we denote the "matrix" Σ by σ^2 (as usual) then $W_1(\Sigma, n)/\sigma^2 = \chi^2_n$. In particular, when $\sigma^2 = 1$ we see that $W_1(1, n)$ is exactly the χ^2_n random variable. In that sense we can state that the Wishart distribution is a generalisation (with respect to the dimension p) of the χ^2 distribution.
2. For an arbitrary fixed matrix $H \in \mathcal{M}_{k,p}, k \leq p$ one has:

$$nH\hat{\boldsymbol{\Sigma}}H^\top \sim W_k(H\Sigma H^\top, n-1).$$

3. Refer to the previous case for the particular value of $k = 1$. The matrix $H \in \mathcal{M}_{1,p}$ is just a p-dimensional row vector that we could denote by $\boldsymbol{c}^\top$. Then:
   (i) $n\frac{\boldsymbol{c}^\top\hat{\boldsymbol{\Sigma}}\boldsymbol{c}}{\boldsymbol{c}^\top\Sigma\boldsymbol{c}} \sim \chi^2_{n-1}$
   (ii) $n\frac{\boldsymbol{c}^\top\Sigma^{-1}\boldsymbol{c}}{\boldsymbol{c}^\top\hat{\boldsymbol{\Sigma}}^{-1}\boldsymbol{c}} \sim \chi^2_{n-p}$
4. Let us partition $\boldsymbol{S} = \frac{1}{n-1}\sum_{i=1}^n(\boldsymbol{X}_i - \bar{\boldsymbol{X}})(\boldsymbol{X}_i - \bar{\boldsymbol{X}})^\top \in \mathcal{M}_{p,p}$ into

$$\boldsymbol{S} = \begin{pmatrix} \boldsymbol{S}_{11} & \boldsymbol{S}_{12} \\ \boldsymbol{S}_{21} & \boldsymbol{S}_{22} \end{pmatrix}, \boldsymbol{S}_{11} \in \mathcal{M}_{r,r}, r < p$$

$$\Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}, \Sigma_{11} \in \mathcal{M}_{r,r}, r < p.$$

   Further, denote

$$\boldsymbol{S}_{1|2} = \boldsymbol{S}_{11} - \boldsymbol{S}_{12}\boldsymbol{S}_{22}^{-1}\boldsymbol{S}_{21},\ \Sigma_{1|2} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}.$$

   Then it holds

$$(n-1)\boldsymbol{S}_{11} \sim W_r(\Sigma_{11}, n-1)$$

$$(n-1)\boldsymbol{S}_{1|2} \sim W_r(\Sigma_{1|2}, n-p+r-1)$$

For a walkthorugh of the above example, watch the follwing video by Dr Pavel Krivitsky.

Transcript

Distributions of MLE of mean vector and covariance matrix of multivariate normal distribution: Detailed derivations

Sampling distribution of $\bar{\boldsymbol{X}}$

Here, we prove the claims about the sampling distribution of $\bar{\boldsymbol{X}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{X}_i \in \mathbb{R}^p$ from the previous slide.

Watch the below video for the exposition:

Transcript

Let the random vector $\bar{\boldsymbol{X}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{X}_i \in \mathbb{R}^p$. For any $\boldsymbol{l} \in \mathbb{R}^p$: $\boldsymbol{l}^\top \bar{\boldsymbol{X}}$ is a linear combination of normals and hence is normal (see Definition 1.1). Since taking expected value is a linear operation, we have $\mathrm{E}\,\bar{\boldsymbol{X}} = \frac{1}{n} n\boldsymbol{\mu} = \boldsymbol{\mu}$; In analogy with the univariate case we could formally write $\mathrm{Cov}\,\bar{\boldsymbol{X}} = \frac{1}{n^2} n\,\mathrm{Cov}\,\boldsymbol{X}_1 = \frac{1}{n}\Sigma$ and hence $\bar{\boldsymbol{X}} \sim N_p(\boldsymbol{\mu}, \frac{1}{n}\Sigma)$. But we would like to develop a more appropriate machinery for the multivariate case that would help us to more rigorously prove statements like the last one. It is based on operations with *Kronecker products*.

Kronecker product of two matrices $A \in \mathcal{M}_{m,n}$ *and* $B \in \mathcal{M}_{p,q}$ is denoted by $A \otimes B$ and is defined (in block matrix notation) as

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{pmatrix} \tag{1.13}$$

The following basic properties of Kronecker products will be used:

$$(A \otimes B) \otimes C = A \otimes (B \otimes C)$$

$$(A + B) \otimes C = A \otimes C + B \otimes C$$

$$(A \otimes B)^\top = A^\top \otimes B^\top$$

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$$

$$(A \otimes B)(C \otimes D) = AC \otimes BD$$

(whenever the corresponding matrix products and inverses exist)

$$\operatorname{tr}(A \otimes B) = \operatorname{tr}(A) \operatorname{tr}(B)$$

$$|A \otimes B| = |A|^p |B|^m$$

(in case $A \in \mathcal{M}_{m,m}, B \in \mathcal{M}_{p,p}$).

In addition, the $\vec{\square}$ operation on a matrix $A \in \mathcal{M}_{m,n}$ will be defined. This operation creates a vector $\vec{A} \in \mathbb{R}^{mn}$ which is composed by stacking the n columns of the matrix $A \in \mathcal{M}_{m,n}$ under each other (the second below the first etc). For matrices A, B and C (of suitable dimensions) it holds:

$$\overrightarrow{ABC} = (C^\top \otimes A)\vec{B}$$

Let us see how we could utilise the above to derive the distribution of $\bar{\boldsymbol{X}}$. Denote by $\mathbf{1}_n$ the vector of n ones. Note that if $\boldsymbol{X}$ is the random data matrix (see (0.11) in slide "Standard facts about multivariate distributions") then $\vec{\boldsymbol{X}} \sim N(\mathbf{1}_n \otimes \boldsymbol{\mu}, I_n \otimes \Sigma)$ and $\bar{\boldsymbol{X}} = \frac{1}{n}(\mathbf{1}_n{}^\top \otimes I_p)\vec{\boldsymbol{X}}$. Hence $\bar{\boldsymbol{X}}$ is multivariate normal with

$$\operatorname{E} \bar{\boldsymbol{X}} = \frac{1}{n}(\mathbf{1}_n{}^\top \otimes I_p)(\mathbf{1}_n \otimes \boldsymbol{\mu}) = \frac{1}{n}(\mathbf{1}_n{}^\top \mathbf{1}_n \otimes \boldsymbol{\mu}) = \frac{1}{n} n \boldsymbol{\mu} = \boldsymbol{\mu},$$

$$\operatorname{Cov} \bar{\boldsymbol{X}} = n^{-2}(\mathbf{1}_n{}^\top \otimes I_p)(I_n \otimes \Sigma)(\mathbf{1}_n \otimes I_p) = n^{-2}(\mathbf{1}_n{}^\top \mathbf{1}_n \otimes \Sigma) = n^{-1}\Sigma.$$

Independence of $\bar{\boldsymbol{X}}$ and $\hat{\boldsymbol{\Sigma}}$

Here, we prove Theorem 1.4 from the previous slide. Recall the likelihood function

$$L(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma) = \frac{1}{(2\pi)^{\frac{np}{2}} |\Sigma|^{\frac{n}{2}}} \, \mathrm{e}^{-\frac{1}{2} \operatorname{tr}[\Sigma^{-1}(\sum_{i=1}^n (\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^\top + n(\bar{\boldsymbol{x}} - \boldsymbol{\mu})(\bar{\boldsymbol{x}} - \boldsymbol{\mu})^\top)]}$$

We have two summands in the exponent from which one is a function of the observations through $n\hat{\boldsymbol{\Sigma}} = \sum_{i=1}^{n}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^\top$ only and the other one depends on the observations through $\bar{\boldsymbol{x}}$ only. The idea is now to transform the original data matrix $\boldsymbol{X} \in \mathcal{M}_{p,n}$ into a new matrix $\boldsymbol{Z} \in \mathcal{M}_{p,n}$ of n independent $N(\mathbf{0}, \Sigma)$ vectors in such a way that $\bar{\boldsymbol{X}}$ would only be a function of $\boldsymbol{Z}_1$ whereas $\sum_{i=1}^{n}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^\top$ would only be a function of $\boldsymbol{Z}_2, \ldots, \boldsymbol{Z}_n$. If we succeed then clearly $\bar{\boldsymbol{X}}$ and $\sum_{i=1}^{n}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^\top = n\hat{\boldsymbol{\Sigma}}$ would be independent.

Now the claim is that the sought after transformation is given by $\boldsymbol{Z} = \boldsymbol{X}A$ with $A \in \mathcal{M}_{n,n}$ *being an orthogonal matrix with a first column equal to* $\frac{1}{\sqrt{n}}\mathbf{1}_n$. Note that the first column of $\boldsymbol{Z}$ would be then $\sqrt{n}\bar{\boldsymbol{X}}$. (An explicit form of the matrix A will be discussed at the lecture.) Since $\vec{\boldsymbol{Z}} = \overrightarrow{I_p\boldsymbol{X}A} = (A^\top \otimes I_p)\vec{\boldsymbol{X}}$, the Jacobian of the transformation ($\vec{\boldsymbol{X}}$ into $\vec{\boldsymbol{Z}}$) is $|A^\top \otimes I_p| = |A|^p = \pm 1$ (note that A is orthogonal). Therefore, the absolute value of the Jacobian is equal to one. For $\vec{\boldsymbol{Z}}$ we have:

$$\mathrm{E}(\vec{\boldsymbol{Z}}) = (A^\top \otimes I_p)(\mathbf{1}_n \otimes \boldsymbol{\mu}) = A^\top\mathbf{1}_n \otimes \boldsymbol{\mu} = \begin{pmatrix} \sqrt{n} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \otimes \boldsymbol{\mu}$$

Further,

$$\mathrm{Cov}(\vec{\boldsymbol{Z}}) = (A^\top \otimes I_p)(I_n \otimes \Sigma)(A \otimes I_p) = A^\top A \otimes I_p\Sigma I_p = I_n \otimes \Sigma$$

which means that the $\boldsymbol{Z}_i, i = 1, \ldots, n$ are *independent*. Note $\boldsymbol{Z}_1 = \sqrt{n}\bar{\boldsymbol{X}}$ holds (because of the choice of the first column of the orthogonal matrix A). Further

$$\sum_{i=1}^{n}(\boldsymbol{X}_i - \bar{\boldsymbol{X}})(\boldsymbol{X}_i - \bar{\boldsymbol{X}})^\top = \sum i = 1^n \boldsymbol{X}_i{\boldsymbol{X}_i}^\top - \frac{1}{n}(\sum_{i=1}^{n}\boldsymbol{X}i)(\sum_{i=1}^{n}\boldsymbol{X}i^\top) =$$

$$\boldsymbol{Z}A^\top A\boldsymbol{Z}^\top - \boldsymbol{Z}_1{\boldsymbol{Z}_1}^\top = \sum_{i=1}^{n}\boldsymbol{Z}_i\boldsymbol{Z}_i^\top - \boldsymbol{Z}_1\boldsymbol{Z}_1^\top = \sum_{i=2}^{n}\boldsymbol{Z}_i\boldsymbol{Z}_i^\top$$

Hence we proved the following

Theorem 1.4. *For a sample of size n from $N_p(\boldsymbol{\mu}, \Sigma), p < n$ the sample average $\bar{\boldsymbol{X}} \sim N_p(\mu, \frac{1}{n}\Sigma)$. Moreover, the MLE $\hat{\boldsymbol{\mu}} = \bar{\boldsymbol{X}}$ and $\hat{\boldsymbol{\Sigma}}$ are independent.*

For a further explanation of these concepts, watch the below video.

Transcript

Topic 4: Confidence intervals and hypothesis tests for the mean vector

Hypothesis tests for the multivariate normal mean

In this topic, we will begin to apply the multivariate normal distribution to answer questions based on the population mean. Here, we will consider confidence intervals and hypothesis tests about μ, as well as how to compare means of several populations.

The concepts for this section are explored in the video below by Dr Pavel Krivitsky.

Hotelling's T^2

Suppose again that, Like in Topic 3 (slide "Estimation of the Mean Vector and Covariance Matrix of Multivariate Normal Distribution"), we have observed n independent realisations of p-dimensional random vectors from $N_p(\boldsymbol{\mu}, \Sigma)$. Suppose for simplicity that Σ is non-singular. The data matrix has the form

$$\boldsymbol{X} = \begin{pmatrix} X_{11} & X_{12} & \cdots & X_{1j} & \cdots & X_{1n} \\ X_{21} & X_{22} & \cdots & X_{2j} & \cdots & X_{2n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ X_{i1} & X_{i2} & \cdots & X_{ij} & \cdots & X_{in} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ X_{p1} & X_{p2} & \cdots & X_{pj} & \cdots & X_{pn} \end{pmatrix} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n]$$

Based on our knowledge from Topic 3 of this week (slide "Distributions of MLE of mean vector and covariance matrix of multivariate normal distribution"), we can claim that $\bar{\boldsymbol{X}} \sim N_p(\boldsymbol{\mu}, \frac{1}{n}\Sigma)$ and $n\hat{\boldsymbol{\Sigma}} \sim W_p(\Sigma, n-1)$.

Consequently, any linear combination $\boldsymbol{c}^\top \bar{\boldsymbol{X}}, \boldsymbol{c} \neq 0 \in \mathbb{R}^p$ follows $N(\boldsymbol{c}^\top \boldsymbol{\mu}, \frac{1}{n}\boldsymbol{c}^\top \Sigma \boldsymbol{c})$ and the quadratic form $n\boldsymbol{c}^\top \hat{\boldsymbol{\Sigma}}\boldsymbol{c} / \boldsymbol{c}^\top \Sigma \boldsymbol{c} \sim \chi^2_{n-1}$. Further, we have shown that $\bar{\boldsymbol{X}}$ and $\hat{\boldsymbol{\Sigma}}$ are independently distributed and hence

$$T = \sqrt{n}\boldsymbol{c}^\top(\bar{\boldsymbol{X}} - \boldsymbol{\mu}) / \sqrt{\boldsymbol{c}^\top \frac{n}{n-1}\hat{\boldsymbol{\Sigma}}\boldsymbol{c}} \sim t_{n-1},$$

i.e. follows the t distribution with $n-1$ degrees of freedom. This result has useful applications in testing for contrasts.

Indeed, if we would like to test $H_0 : \boldsymbol{c}^\top \boldsymbol{\mu} = \sum_{i=1}^p c_i \mu_i = 0$, we note that under H_0, T becomes simply

$$T = \sqrt{n}\boldsymbol{c}^\top \bar{\boldsymbol{X}} / \sqrt{\boldsymbol{c}^\top \boldsymbol{S}\boldsymbol{c}},$$

it does not involve the unknown $\boldsymbol{\mu}$ and can be used as a test-statistic whose distribution under H_0 is known. If $|T| > t_{1-\alpha/2, n-1}$ we should reject H_0 in favour of $H_1 : \boldsymbol{c}^\top \mu = \sum_{i=1}^p c_i \mu_i \neq 0$.

The formulation of the test for other (one-sided) alternatives is left for you as an exercise.

More often we are interested in testing the mean vector of a multivariate normal. First consider the case of *known* covariance matrix Σ (variance σ^2 in the univariate case). The standard univariate ($p = 1$) test for this purpose is the following: to test $H_0 : \mu = \mu_0$ versus $H_1 : \mu \neq \mu_0$ at level of significance α, we look at $U = \sqrt{n}\frac{\bar{X} - \mu_0}{\sigma}$ and reject H_0 if $|U|$ exceeds the upper $\frac{\alpha}{2} \cdot 100\%$ point of the standard normal distribution. Checking if $|U|$ is large enough is equivalent to checking if $U^2 = n(\bar{X} - \mu_0)(\sigma^2)^{-1}(\bar{X} - \mu_0)$ is large enough. We can now easily generalise the above test statistic in a natural way for the multivariate ($p > 1$) case: calculate $U^2 = n(\bar{\boldsymbol{X}} - \boldsymbol{\mu}_0)^\top \Sigma^{-1}(\bar{\boldsymbol{X}} - \boldsymbol{\mu}_0)$ and reject the null hypothesis of $\boldsymbol{\mu} = \boldsymbol{\mu}_0$ when U^2 is large enough. Similarly to the proof of *Property 5* of the multivariate normal distribution (Topic 2, slide "Properties of multivariate normal") and by using Theorem 1.4 of Topic 3 (slide "Distributions of MLE of mean vector and covariance matrix of multivariate normal distribution"), you can convince yourself that $U^2 \sim \chi^2_p$ under the null hypothesis. Hence, tables of the χ^2-distribution will suffice to perform the above test in the multivariate case.

Now let us turn to the (practically more relevant) case of unknown covariance matrix Σ. The standard univariate ($p = 1$) test for this purpose is the t-test. Let us recall it: to test $H_0 : \mu = \mu_0$ versus $H_1 : \mu \neq \mu_0$ at level of significance α, we look at

$$T = \sqrt{n}\frac{\bar{X} - \mu_0}{S},\ S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2$$

and reject H_0 if $|T|$ exceeds the upper $\frac{\alpha}{2} \cdot 100\%$ point of the t-distribution with $n-1$ degrees of freedom. We note that checking if $|T|$ is large enough is equivalent to checking if $T^2 = n(\bar{X} - \mu_0)(s^2)^{-1}(\bar{X} - \mu_0)$ is large enough. Of course, under H_0, the statistic T^2 is F-distributed: $T^2 \sim F_{1,n-1}$ which means that H_0 would be rejected at level α when $T^2 > F_{\alpha;1,n-1}$. We can now easily generalise the above test statistic in a natural way for the multivariate ($p > 1$) case:

Definition 1.3 (Hotelling's T^2)**.** The statistic

$$T^2 = n(\bar{\boldsymbol{X}} - \boldsymbol{\mu}_0)^\top \boldsymbol{S}^{-1}(\bar{\boldsymbol{X}} - \boldsymbol{\mu}_0) \tag{1.14}$$

where $\bar{\boldsymbol{X}} = \frac{1}{n}\sum_{j=1}^{n}\boldsymbol{X}_i$, $\boldsymbol{S} = \frac{1}{n-1}\sum_{i=1}^{n}(\boldsymbol{X}_i - \bar{\boldsymbol{X}})(\boldsymbol{X}_i - \bar{\boldsymbol{X}})^\top$, $\boldsymbol{\mu}_0 \in \mathbb{R}^p$, $\boldsymbol{X}_i \in \mathbb{R}^p, i = 1, \ldots, n$ is named after Harold Hotelling.

Sampling distribution of T^2

Obviously, the test procedure based on Hotelling's statistic will reject the null hypothesis $H_0 : \boldsymbol{\mu} = \boldsymbol{\mu}_0$ if the value of T^2 is sufficiently high. It turns out we do not need special tables for the distribution of T^2 under the null hypothesis because of the following basic result (that represents a true generalisation of the univariate ($p = 1$) case:

Theorem 1.5. *Under the null hypothesis $H_0 : \boldsymbol{\mu} = \boldsymbol{\mu}_0$, Hotelling's T^2 is distributed as $\frac{(n-1)p}{n-p}F_{p,n-p}$ where $F_{p,n-p}$ denotes the F-distribution with p and $n-p$ degrees of freedom.*

Noncentral Wishart

It is possible to extend the definition of the Wishart distribution (see Definition 1.2) by allowing the random vectors $\boldsymbol{Y}_i, i = 1, \ldots, n$ there to be independent with $N_p(\boldsymbol{\mu}_i, \Sigma)$ (instead of just having all $\boldsymbol{\mu}_i = \boldsymbol{0}$). One arrives at the *noncentral* Wishart distribution with parameters $\Sigma, p, n-1, \Gamma$ in that way (denoted also as $W_p(\Sigma, n-1, \Gamma)$. Here $\Gamma = MM^\top \in \mathcal{M}_{p,p}$, $M = [\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_n]$ is called a *noncentrality parameter*. When all columns of $M \in \mathcal{M}_{p,n}$ are zero, this is the usual (*central*) Wishart distribution. Theorem 1.5 can be extended to derive the distribution of the T^2 statistic under alternatives, i.e. the distribution of $T^2 = n(\bar{\boldsymbol{X}} - \boldsymbol{\mu})^\top \boldsymbol{S}^{-1}(\bar{\boldsymbol{X}} - \boldsymbol{\mu})$ for $\boldsymbol{\mu} \neq \boldsymbol{\mu}_0$. This distribution turns out to be related to *noncentral F-distribution*. It is helpful in studying power of the test of $H_0 : \boldsymbol{\mu} = \boldsymbol{\mu}_0$ versus $H_1 : \boldsymbol{\mu} \neq \boldsymbol{\mu}_0$.

Transcript

This section is continued on the next slide.

Hypothesis tests for the multivariate normal mean continued

T^2 as a likelihood ratio statistic

Before we continue, watch the below video regarding the concepts discussed in this section.

Transcript

It is worth mentioning that Hotelling's T^2 that we introduced by *analogy* with the univariate squared t statistic can in fact also be derived as the *likelihood ratio test statistic* for testing $H_0: \boldsymbol{\mu} = \boldsymbol{\mu}_0$ versus $H_1: \boldsymbol{\mu} \neq \boldsymbol{\mu}_0$. This safeguards the asymptotic optimality of the test suggested above. To see this, first recall the likelihood function (1.9)

$$L(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma) = (2\pi)^{-\frac{np}{2}} |\Sigma|^{-\frac{n}{2}} \mathrm{e}^{-\frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{x}i-\boldsymbol{\mu})^\top \Sigma^{-1}(\boldsymbol{x}i-\boldsymbol{\mu})}$$

Its unconstrained maximisation gives as a maximum value:

$$L(\boldsymbol{x}; \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}) = \frac{1}{(2\pi)^{\frac{np}{2}} |\hat{\boldsymbol{\Sigma}}|^{\frac{n}{2}}} \mathrm{e}^{-\frac{np}{2}}$$

On the other hand, under H_0:

$$\max_{\Sigma} L(x; \boldsymbol{\mu}_0, \Sigma) = \max_{\Sigma} \frac{1}{(2\pi)^{\frac{np}{2}} |\Sigma|^{\frac{n}{2}}} \mathrm{e}^{-\frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{x}_i-\boldsymbol{\mu}_0)^\top \Sigma^{-1}(\boldsymbol{x}_i-\boldsymbol{\mu}_0)}$$

Since $\log L(\boldsymbol{x};\boldsymbol{\mu}_0,\Sigma) = -\frac{np}{2}\log(2\pi) - \frac{n}{2}\log|\Sigma| - \frac{1}{2}\operatorname{tr}[\Sigma^{-1}(\sum_{i=1}^{n}(\boldsymbol{x}_i-\boldsymbol{\mu}_0)(\boldsymbol{x}_i-\boldsymbol{\mu}_0)^\top)]$, on applying Anderson's lemma (see Theorem 1.2) we find that maximum of $\log L(\boldsymbol{x};\boldsymbol{\mu}_0,\Sigma)$ (whence also of $L(\boldsymbol{x};\boldsymbol{\mu}_0,\Sigma)$) is obtained when $\hat{\boldsymbol{\Sigma}}_0 = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i-\boldsymbol{\mu}_0)(\boldsymbol{x}_i-\boldsymbol{\mu}_0)^\top$ and the maximal value is

$$\frac{1}{(2\pi)^{\frac{np}{2}}|\hat{\boldsymbol{\Sigma}}0|^{\frac{n}{2}}}\,\mathrm{e}^{-\frac{np}{2}}\,.$$

Hence the likelihood ratio is:

$$\Lambda = \frac{\max_{\Sigma} L(x;\boldsymbol{\mu}_0,\Sigma)}{\max_{\boldsymbol{\mu},\Sigma} L(x;\boldsymbol{\mu},\Sigma)} = \left(\frac{|\hat{\boldsymbol{\Sigma}}|}{|\hat{\boldsymbol{\Sigma}}_0|}\right)^{\frac{n}{2}} \tag{1.15}$$

The equivalent statistic $\Lambda^{\frac{2}{n}} = \frac{|\hat{\boldsymbol{\Sigma}}|}{|\hat{\boldsymbol{\Sigma}}_0|}$ is called *Wilks' lambda*. Small values of Wilks' lambda lead to rejecting $H_0 : \boldsymbol{\mu} = \boldsymbol{\mu}_0$.

Wilks' lambda and T^2

The following theorem shows the relation between Wilks' lambda and T^2:

Theorem 1.6. *The likelihood ratio test is equivalent to the test based on T^2 since $\Lambda^{\frac{2}{n}} = (1+\frac{T^2}{n-1})^{-1}$ holds.*

Numerical calculation of T^2

Hence H_0 is rejected for small values of $\Lambda^{\frac{2}{n}}$ or equivalently, for large values of T^2. The critical values for T^2 are determined from Theorem 1.5. Relation in Theorem 1.6 can be used to calculate T^2 from $\Lambda^{\frac{2}{n}} = \frac{|\hat{\boldsymbol{\Sigma}}|}{|\hat{\boldsymbol{\Sigma}}_0|}$ thus avoiding the need to invert the matrix $\boldsymbol{S}$ when calculating T^2!

Watch the below example by Dr Pavel Krivitsky explaining this concept.

Transcript

Asymptotic distribution of T^2

Since $\boldsymbol{S}^{-1}$ is a consistent estimator of Σ^{-1}, the limiting distribution of T^2 will coincide with the one of $n(\bar{\boldsymbol{x}} - \boldsymbol{\mu})^\top \Sigma^{-1}(\bar{\boldsymbol{x}} - \boldsymbol{\mu})$ which, as we know already, is χ^2_p. This coincides with a general claim of asymptotic theory which states that $-2\log\Lambda$ is asymptotically distributed as χ^2_p. Indeed:

$$-2\log\Lambda = n\log(1 + \frac{T^2}{n-1}) \approx \frac{n}{n-1}T^2 \approx T^2$$

(by using the fact that $\log(1+x) \approx x$ for small x).

Transcript

Confidence regions for the mean vector and for its components

Confidence region for the mean vector

Transcript

For a given confidence level $(1-\alpha)$ it can be constructed in the form

$$\left\{\boldsymbol{\mu} | n(\bar{\boldsymbol{x}}-\boldsymbol{\mu})^{\top} \boldsymbol{S}^{-1}(\bar{\boldsymbol{x}}-\boldsymbol{\mu}) \leq F_{1-\alpha, p, n-p} \frac{p}{n-p}(n-1)\right\}$$

where $F_{1-\alpha,p,n-p}$ is the upper $\alpha \cdot 100\%$ percentage point of the F distribution with $(p, n-p)$ df. This confidence region has the form of an *ellipsoid* in $\mathbb{R}^p$ centred at $\bar{\boldsymbol{x}}$. The axes of this *confidence ellipsoid* are directed along the eigenvectors $\boldsymbol{e}_i$ of the matrix $S = \frac{1}{n-1}\sum_{i=1}^{n}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^{\top}$. The half-lengths of the axes are given by the expression $\sqrt{\lambda_i}\sqrt{\frac{p(n-1)F_{1-\alpha,p,n-p}}{n(n-p)}}$, with $\lambda_i, i = 1, \ldots, p$ being the corresponding eigenvalues, i.e.

$$S\boldsymbol{e}_i = \lambda_i \boldsymbol{e}_i, i = 1, \ldots, p$$

For illustration, see numerical example 5.3, pages 221–223, Johnson and Wichern.

Simultaneous confidence statements

For a given confidence level $(1-\alpha)$ the confidence ellipsoids in the section above correctly reflect

the joint (multivariate) knowledge about plausible values of $\boldsymbol{\mu} \in \mathbb{R}^p$ but nevertheless one is often interested in confidence intervals for means of each individual component. We would like to formulate these statements in such a form that *all of the separate confidence statements should hold* simultaneously with a prespecified probability. This is why we are speaking about *simultaneous confidence intervals*.

First, note that if the vector $\boldsymbol{X} \sim N_p(\boldsymbol{\mu}, \Sigma)$ then for any $\boldsymbol{l} \in \mathbb{R}^p : \boldsymbol{l}^\top \boldsymbol{X} \sim N_1(\boldsymbol{l}^\top \boldsymbol{\mu}, \boldsymbol{l}^\top \Sigma \boldsymbol{l})$ and, hence, for any fixed $\boldsymbol{l}$ we can construct an $(1-\alpha)\cdot 100\%$ confidence interval of $\boldsymbol{l}^\top \boldsymbol{\mu}$ in the following simple way:

$$\left(\boldsymbol{l}^\top \bar{\boldsymbol{x}} - t_{1-\alpha/2,n-1} \frac{\sqrt{\boldsymbol{l}^\top \boldsymbol{S} \boldsymbol{l}}}{\sqrt{n}}, \boldsymbol{l}^\top \bar{\boldsymbol{x}} + t_{1-\alpha/2,n-1} \frac{\sqrt{\boldsymbol{l}^\top \boldsymbol{S} \boldsymbol{l}}}{\sqrt{n}} \right) \tag{1.16}$$

By taking $\boldsymbol{l}^\top = [1, 0, \ldots, 0]$ or $\boldsymbol{l}^\top = [0, 1, 0, \ldots, 0]$ etc. we obtain from (1.16) the usual confidence interval for each separate component of the mean. Note however that the confidence level for all these statements taken together is not $(1-\alpha)$. To make it $(1-\alpha)$ for all possible choices simultaneously we need to take a larger constant than $t_{1-\alpha/2,n-1}$ in the right hand side of the inequality $\left| \frac{\sqrt{n}(\boldsymbol{l}^\top \bar{\boldsymbol{x}} - \boldsymbol{l}^\top \bar{\boldsymbol{\mu}})}{\sqrt{\boldsymbol{l}^\top \boldsymbol{S} \boldsymbol{l}}} \right| \leq t_{1-\alpha/2,n-1}$ (or equivalently $\frac{n(\boldsymbol{l}^\top \bar{\boldsymbol{x}} - \boldsymbol{l}^\top \bar{\boldsymbol{\mu}})^2}{\boldsymbol{l}^\top \boldsymbol{S} \boldsymbol{l}} \leq t^2_{1-\alpha/2,n-1}$).

Simultaneous confidence ellipsoid

Theorem 1.7. *Simultaneously for all $\boldsymbol{l} \in \mathbb{R}^p$, the interval*

$$\left(\boldsymbol{l}^\top \bar{\boldsymbol{x}} - \sqrt{\frac{p(n-1)}{n(n-p)} F_{1-\alpha,p,n-p} \boldsymbol{l}^\top \boldsymbol{S} \boldsymbol{l}}, \boldsymbol{l}^\top \bar{\boldsymbol{x}} + \sqrt{\frac{p(n-1)}{n(n-p)} F_{1-\alpha,p,n-p} \boldsymbol{l}^\top \boldsymbol{S} \boldsymbol{l}} \right)$$

will contain $\boldsymbol{l}^\top \bar{\boldsymbol{\mu}}$ with a probability at least $(1-\alpha)$.

For Illustration, see example 5.4, p. 226 in Johnson and Wichern.

Bonferroni Method

The simultaneous confidence intervals when applied for the vectors $\boldsymbol{l}^\top = [1, 0, \ldots, 0], \boldsymbol{l}^\top = [0, 1, 0, \ldots, 0]$ etc. are much more reliable at a given confidence level than the one-at-a-time intervals. Note that the former also utilise the covariance structure of all p variables in their construction. However, sometimes we can do better in cases where one is interested in a small number of individual confidence statements.

In this latter case, the simultaneous confidence intervals may give too large a region and the Bonferroni method may prove more efficient instead. The idea of the Bonferroni approach is based on a simple probabilistic inequality. Assume that simultaneous confidence statements about m linear combinations $\boldsymbol{l}_1^\top \boldsymbol{\mu}, \boldsymbol{l}_2^\top \boldsymbol{\mu}, \ldots, \boldsymbol{l}_m^\top \boldsymbol{\mu}$ are required. If $C_i, i = 1, 2, \ldots, m$ denotes the ith confidence statement and $P(C_i \text{ true}) = 1 - \alpha_i$ then

$$P\left(\text{ all } C_i \text{ true }\right) = 1 - P\left(\text{ at least one } C_i \text{ false }\right) \geq$$

$$1 - \sum_{i=1}^{m} P\left(C_i \text{ false }\right) = 1 - \sum_{i=1}^{m} \left(1 - P\left(C_i \text{ true }\right)\right) = 1 - \left(\alpha_1 + \alpha_2 + \cdots + \alpha_m\right)$$

Hence, if we choose $\alpha_i = \frac{\alpha}{m}, i = 1, 2, \ldots, m$ (that is, if calculate each statement at confidence level $\left(1 - \frac{\alpha}{m}\right) \cdot 100\%$ instead of $\left(1 - \alpha\right) \cdot 100\%$) then the probability of any statement being false will not exceed α.

The Bonferroni method is further explained with an example in the below video.

Transcript

Comparison of two or more mean vectors

Finally, let us note that comparison of the mean vectors of two or more than two different multivariate populations when there are independent observations from each of the populations is an important, practically relevant problem. For the purposes of this section, suppose that we observe two samples, $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_{n\boldsymbol{X}} \in \mathbb{R}^p$ and $\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_{n\boldsymbol{Y}} \in \mathbb{R}^p$, with means $\boldsymbol{\mu}_{\boldsymbol{X}} \in \mathbb{R}^p$ and $\boldsymbol{\mu}_{\boldsymbol{Y}} \in \mathbb{R}^p$ respectively and variances $\Sigma_{\boldsymbol{X}} \in \mathcal{M}_{p,p}$ and $\Sigma_{\boldsymbol{Y}} \in \mathcal{M}_{p,p}$, respectively. Typically, we wish to test $H_0 : \boldsymbol{\mu}_{\boldsymbol{X}} - \boldsymbol{\mu}_{\boldsymbol{Y}} = \boldsymbol{\delta}_0$.

Multivariate ANOVA for comparing more than two populations is discussed in Topic 2 of Week 3.

Transcript

Reducing to a single population

As with the univariate t-test, under some scenarios the test of a difference between two populations in fact reduces to a one-sample test. For example, if the samples are paired and $n_{\boldsymbol{X}} = n_{\boldsymbol{Y}} = n$, we may proceed analogously to the paired t-test: we take $\boldsymbol{D}_i = \boldsymbol{X}_i - \boldsymbol{Y}_i$ for $i = 1, \ldots, n$ and proceed as if with a 1-sample T^2 test:

$$T^2 = n(\bar{\boldsymbol{D}} - \boldsymbol{\delta}_0)^\top \boldsymbol{S}_{\boldsymbol{D}}^{-1} (\bar{\boldsymbol{D}} - \boldsymbol{\delta}_0) \sim \frac{(n-1)p}{n-p} F_{p,n-p}, \tag{1.17}$$

where $\bar{\boldsymbol{D}} \in \mathbb{R}^p$ and $\boldsymbol{S}_{\boldsymbol{D}} \in \mathcal{M}_{p,p}$ are the sample mean and variance of $\boldsymbol{D}_1, \ldots, \boldsymbol{D}_n$, respectively,

assuming $\boldsymbol{D}_i$ are normally distributed. (It is important to note that any diagnostics for this test should be performed on the differences, not on the original values.)

We can also formulate this is in a "multivariate" form: let the *contrast matrix* $C \in \mathcal{M}_{p,p+p}$ be

$$C = \begin{pmatrix} +1 & & & -1 & & \\ & +1 & & & -1 & \\ & & +1 & & & -1 \end{pmatrix}.$$

Then, we can express $\boldsymbol{D}_i = C \begin{pmatrix} \boldsymbol{X}_i \\ \boldsymbol{Y}_i \end{pmatrix}$ and the test as $H_0 : C \begin{pmatrix} \boldsymbol{\mu_X} \\ \boldsymbol{\mu_Y} \end{pmatrix} = \boldsymbol{\delta}_0$. It is easy to show that the test statistic reduces to (1.17).

C can have more complex forms. For example, in a *repeated measures design*, we may measure the results of a series of p treatment outcomes on each sampling unit. If we then collect each individual i 's measurements into a vector $\boldsymbol{X}_i$, we may test whether all outcomes are the same by forming

$$C = \begin{pmatrix} 1 & -1 & & \\ \vdots & & \ddots & \\ 1 & & & -1 \end{pmatrix} \in \mathcal{M}_{p-1,p}$$

and testing $H_0 : C\boldsymbol{\mu_X} = \mathbf{0}_{p-1}$. It is easy to show that $C\boldsymbol{\mu_X} = \mathbf{0}_{p-1}$ holds if and only if all elements of $\boldsymbol{\mu_X}$ are equal.

The two-sample T^2-test

We now turn to the scenario where $\boldsymbol{X}$ and $\boldsymbol{Y}$ are, in fact, independent samples. As with the univariate test, we must decide whether we are prepared to assume that $\Sigma_{\boldsymbol{X}} = \Sigma_{\boldsymbol{Y}} = \Sigma$ in the population and therefore use the pooled test. If so—and necessarily if the sample sizes are small—we evaluate

$$\boldsymbol{S}_{\text{pooled}} = \frac{(n_{\boldsymbol{X}} - 1)\boldsymbol{S}_{\boldsymbol{X}} + (n_{\boldsymbol{Y}} - 1)\boldsymbol{S}_{\boldsymbol{Y}}}{n_{\boldsymbol{X}} + n_{\boldsymbol{Y}} - 2}.$$

Since $\boldsymbol{S}_{\text{pooled}}$ estimates Σ,

$$\text{Var}(\bar{\boldsymbol{X}} - \bar{\boldsymbol{Y}}) = \frac{\Sigma}{n_{\boldsymbol{X}}} + \frac{\Sigma}{n_{\boldsymbol{Y}}} \approx \frac{\boldsymbol{S}_{\text{pooled}}}{n_{\boldsymbol{X}}} + \frac{\boldsymbol{S}_{\text{pooled}}}{n_{\boldsymbol{Y}}} = \boldsymbol{S}_{\text{pooled}} \left(\frac{1}{n_{\boldsymbol{X}}} + \frac{1}{n_{\boldsymbol{Y}}} \right).$$

And, since $\bar{\boldsymbol{X}} - \bar{\boldsymbol{Y}} \sim N_p(\boldsymbol{\mu_X} - \boldsymbol{\mu_Y}, \Sigma(n_{\boldsymbol{X}}^{-1} + n_{\boldsymbol{Y}}^{-1}))$, we write

$$\begin{aligned} T^2 &= (\bar{\boldsymbol{X}} - \bar{\boldsymbol{Y}} - \boldsymbol{\delta}_0)^\top \left\{ \boldsymbol{S}_{\text{pooled}} \left(\tfrac{1}{n_{\boldsymbol{X}}} + \tfrac{1}{n_{\boldsymbol{Y}}} \right) \right\}^{-1} (\bar{\boldsymbol{X}} - \bar{\boldsymbol{Y}} - \boldsymbol{\delta}_0) \\ &\sim \frac{(n_{\boldsymbol{X}} + n_{\boldsymbol{Y}} - 2)p}{n_{\boldsymbol{X}} + n_{\boldsymbol{Y}} - p - 1} F_{p, n_{\boldsymbol{X}} + n_{\boldsymbol{Y}} - p - 1}. \end{aligned} \tag{1.18}$$

We would thus reject H_0 if T^2 falls above the F critical value in (1.18), construct a confidence region based on

$$\left\{\boldsymbol{\delta}|(\bar{\boldsymbol{x}}-\bar{\boldsymbol{y}}-\boldsymbol{\delta})^\top\left\{S_{\text{pooled}}\left(\frac{1}{n_{\boldsymbol{X}}}+\frac{1}{n_{\boldsymbol{Y}}}\right)\right\}^{-1}(\bar{\boldsymbol{x}}-\bar{\boldsymbol{y}}-\boldsymbol{\delta})\leq\frac{(n_{\boldsymbol{X}}+n_{\boldsymbol{Y}}-2)p}{n_{\boldsymbol{X}}+n_{\boldsymbol{Y}}-p-1}F_{1-\alpha,p,n_{\boldsymbol{X}}+n_{\boldsymbol{Y}}-}\right.$$

and simultaneous contrast confidence intervals

$$\boldsymbol{l}^\top(\bar{\boldsymbol{x}}-\bar{\boldsymbol{y}})\pm\sqrt{\frac{(n_{\boldsymbol{X}}+n_{\boldsymbol{Y}}-2)p}{n_{\boldsymbol{X}}+n_{\boldsymbol{Y}}-p-1}F_{1-\alpha,p,n_{\boldsymbol{X}}+n_{\boldsymbol{Y}}-p-1}\boldsymbol{l}^\top S_{\text{pooled}}\left(\frac{1}{n_{\boldsymbol{X}}}+\frac{1}{n_{\boldsymbol{Y}}}\right)\boldsymbol{l}}.$$

If we are not prepared to make the pooling assumption, our test statistic is instead

$$T^2=(\bar{\boldsymbol{X}}-\bar{\boldsymbol{Y}}-\boldsymbol{\delta}_0)^\top\left(\frac{\boldsymbol{S}_{\boldsymbol{X}}}{n_{\boldsymbol{X}}}+\frac{\boldsymbol{S}_{\boldsymbol{Y}}}{n_{\boldsymbol{Y}}}\right)^{-1}(\bar{\boldsymbol{X}}-\bar{\boldsymbol{Y}}-\boldsymbol{\delta}_0).$$

Even for modest sample sizes, under multivariate normality, the distribution of this T^2 is reasonably well approximated by $\frac{\nu p}{\nu-p+1}F_{p,\nu-p+1}$, where

$$\nu=\frac{p+p^2}{\sum_{i=1}^2\frac{1}{n_i}\left(\text{tr}\left[\left\{\frac{1}{n_i}\boldsymbol{S}_i\left(\frac{1}{n_1}\boldsymbol{S}_1+\frac{1}{n_2}\boldsymbol{S}_2\right)^{-1}\right\}^2\right]+\left[\text{tr}\left\{\frac{1}{n_i}\boldsymbol{S}_i\left(\frac{1}{n_1}\boldsymbol{S}_1+\frac{1}{n_2}\boldsymbol{S}_2\right)^{-1}\right\}\right]^2\right)}.$$

The confidence regions are then produced by

$$\left\{\boldsymbol{\delta}|(\bar{\boldsymbol{x}}-\bar{\boldsymbol{y}}-\boldsymbol{\delta})^\top\left(\frac{\boldsymbol{S}_{\boldsymbol{X}}}{n_{\boldsymbol{X}}}+\frac{\boldsymbol{S}_{\boldsymbol{Y}}}{n_{\boldsymbol{Y}}}\right)^{-1}(\bar{\boldsymbol{x}}-\bar{\boldsymbol{y}}-\boldsymbol{\delta})\leq\frac{\nu p}{\nu-p+1}F_{p,\nu-p+1}\right\}$$

and simultaneous contrast confidence intervals

$$\boldsymbol{l}^\top(\bar{\boldsymbol{x}}-\bar{\boldsymbol{y}})\pm\sqrt{\frac{\nu p}{\nu-p+1}F_{p,\nu-p+1}\boldsymbol{l}^\top\left(\frac{\boldsymbol{S}_{\boldsymbol{X}}}{n_{\boldsymbol{X}}}+\frac{\boldsymbol{S}_{\boldsymbol{Y}}}{n_{\boldsymbol{Y}}}\right)\boldsymbol{l}}.$$

Check your understanding

Complete the below exercises to check your understanding of concepts presented so far.

1. Suppose $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n$ are independent $N_p(\boldsymbol{\mu}, \Sigma)$ random vectors with sample mean vector $\bar{\boldsymbol{X}}$, S and sample covariance matrix S. We wish to test the hypothesis

$$H_0 : \mu_2 - \mu_1 = \mu_3 - \mu_2 = \cdots = \mu_p - \mu_{p-1} = 1$$

where $\mu_1, \mu_2, \ldots, \mu_p$ are the elements of μ.

a) Determine a $(p-1) \times p$ matrix C so that H_0 may be written equivalently as $H_0 : C\mu = \mathbf{1}$ where $\mathbf{1}$ is a $(p-1) \times 1$ vector of ones.

b) Make an appropriate transformation of the vectors $\boldsymbol{X}_i, i = 1, 2, \ldots, n$ and hence find the rejection region of a size α *test of* H_0 in terms of $\bar{\boldsymbol{X}}$, S and C.

2. A sample of 50 vector observations , each containing three components, is drawn from a normal distribution having covariance matrix

$$\Sigma = \begin{pmatrix} 3 & 1 & 1 \\ 1 & 4 & 1 \\ 1 & 1 & 2 \end{pmatrix}$$

The components of the sample mean are 0.8, 1.1 and 0.6. Can you reject the null hypothesis of zero distribution mean against a general alternative?

3. Evaluate Hotelling's statistic T^2 for testing hypothesis $H_0 : \mu = \begin{pmatrix} 7 \\ 11 \end{pmatrix}$ using the data matrix $\boldsymbol{X} = \begin{pmatrix} 2 & 8 & 6 & 8 \\ 12 & 9 & 9 & 10 \end{pmatrix}$. Test the hypothesis H_0 at level $\alpha = 0.05$. What conclusion is reached?

Demonstration: Hotelling's

This demonstration can be completed using the provided RStudio or your own RStudio.

To complete the demonstration select the 'Hotelling_CI_Examples.demo.Rmd' in the 'Files' section of RStudio. Follow the demonstration contained within the RMD file.

If you choose to complete the example in your own RStudio, upload the following file:

Hotelling_CI_Examples.demo.Rmd

The contents of the RMD file are also displayed below:

Packages

```
library(here)
library(car) # For drawing ellipsoids.
library(GGally)
library(readr)
library(dplyr)
library(expm) # For %^% matrix power.
```

The Microwave Ovens Example

We will use the data in Johnson and Wichern (2007), Examples 5.3--5.4 to illustrate Hotelling's T^2 confidence regions and to develop some intuition for their construction.

Data

We begin by loading the data and transforming (See the Multivariate Normal exercises.):

```
ovens4 <- read_csv(here("datasets","ovens.csv"))^(1/4)
ggpairs(ovens4)
```

Summaries

Let's compute the basic data summaries:

```
(n <- nrow(ovens4)) # sample size
(p <- ncol(ovens4)) # number of variables
(m <- colMeans(ovens4)) # sample mean
(S <- cov(ovens4)) # S
```

```
(Sinv <- solve(S)) # S^-1
```

T^2 confidence ellipsoid

The following demonstration is intended to provide intuition for how a confidence ellipsoid is constructed.

We first define a function testing if a value is in the confidence ellipsoid. This is essentially a direct implementation of Section 4.2.1:

```
in_ellipsoid <- function(x, n, m, Sinv, CL=0.95){
    p <- length(m)
    n*(m-x)%*%Sinv%*%(m-x) <= p*(n-1)/(n-p) * qf(CL, p, n-p)
}
```

We then draw a confidence ellipsoid in a very inefficient manner by taking a large number of points and testing them for being inside it. A more efficient way is to take the eigendecomposition of $\hat{\Sigma}$ and construct a parametric equation for its outline, then plot that. Let's draw both on the same plot.

```
# Brute-force:
closeds <- seq(.50,.65,length.out=200)
opens <- seq(.55,.66,length.out=200)
xy <- expand.grid(closeds,opens)
z <- apply(xy, 1, in_ellipsoid, n, m, Sinv)
# Note: For meanings of asp=1 see ? par .
plot(xy[,1],xy[,2], col=z, xlab="closed",ylab="open", pch=".", asp=1)
points(m[1],m[2], col=2)

# Eigendecomposition:
(S.e <- eigen(S))

scl <- p*(n-1)/(n-p) * qf(0.95, p, n-p) / n
lines(rbind(m-S.e$vectors[,1]*sqrt(S.e$values[1]*scl),
m+S.e$vectors[,1]*sqrt(S.e$values[1]*scl))) # Major axis
lines(rbind(m-S.e$vectors[,2]*sqrt(S.e$values[2]*scl),
m+S.e$vectors[,2]*sqrt(S.e$values[2]*scl))) # Minor axis
```

The popular R package `car` provides a tool for drawing a confidence ellipsoid. Note the intermediate step of fitting a multivariate linear model with variables on the LHS:

```
plot(open~closed, data=ovens4)
# I.e. fit a multivariate linear model (more on that later) with just the intercept:
confidenceEllipse(lm(cbind(closed,open)~1, data=ovens4), add=TRUE) # add=TRUE means
don't start a new plot.
```

Any point in this ellipsoid is a plausible value for the joint population mean microwave radiation

releases.

Simultaneous intervals

In practice, we may also want to express the intervals numerically. We shall discuss two ways of doing so.

Projection of the ellipsoid

Firstly, we can use "shadows" (projections) of the confidence ellipsoid onto the axes (or, technically, onto any other linear combination). The following function implements Theorem 4.3:

```
contrastCI <- function(l, n, m, S, CL=0.95){
    p <- length(m)
    # Note the use of "*c(-1,+1)" as plus-or-minus.
    c(crossprod(l,m)) + c(-1,+1)* c(sqrt(p*(n-1)/(n-p)*qf(CL, p, n-p) *
t(l)%*%S%*%l/n))
}
```

We can then evaluate these for unit vectors to get contrast CIs:

```
l <- c(1,0) # Closed
(closedCI <- contrastCI(l, n, m, S))
l <- c(0,1) # Open
(openCI <- contrastCI(l, n, m, S))
```

Observe their relationship to the ellipsoid:

```
confidenceEllipse(lm(cbind(closed,open)~1, data=ovens4), xlab="closed",ylab="open")
abline(v=closedCI)
abline(h=openCI)
```

Bonferroni

Finally, Bonferroni: this is just a pair of one-sample t-intervals with at a higher individual CL:

```
(closedBCI <- t.test(ovens4$closed,conf.level=.975)$conf.int)
(openBCI <- t.test(ovens4$open,conf.level=.975)$conf.int)
```

Observe that these are slightly narrower. This is because the ellipsoid intervals are simultaneously true for *every* linear combination, so we could, say, compute an interval for `open-closed` contrast using the ellipsoid projection without any additional "miss" probability.

Challenge: Hotelling's

If you choose to complete this task in your own RStudio, upload the following file:

Hotelling_CI_Examples.challenge.Rmd

Click on the 'Hotelling_CI_Examples.challenge.Rmd' in the 'Files' section to begin. Enter your response to the tasks in the 'Enter your code here' section.

This activity and the solution will be discussed at the Collaborate session this week. In the meantime, share and discuss your results in the 'Tutorials' discussion forum.

The contents of the RMD file are also displayed below:

Packages

```
library(here)
library(car) # For drawing ellipsoids.
library(GGally)
library(readr)
library(dplyr)
library(expm) # For %^% matrix power.
```

Challenge

One-sample intervals

We continue our analysis of of Portuguese red wines. We begin with the transformation suggested in the Multivariate Normality exercises:

```
red.t <- read_csv(here("datasets","winequality-red.extract7.csv")) %>% mutate(`log sulphates`=log(`sulphates`), `root4 free sulfur dioxide` = (`free sulfur dioxide`)^(1/4), `root4 total sulfur dioxide` = (`total sulfur dioxide`)^(1/4), `log fixed acidity`=log(`fixed acidity`), `log volatile acidity`=log(`volatile acidity`), .keep="unused")
ggpairs(red.t)
```

Task 1: Plot the data and the 95% confidence ellipse for the mean `log fixed acidity` and `log volatile acidity`.

Task 2: Report the simultaneous confidence intervals for the mean `log fixed acidity` and `log volatile acidity` using the projections of the ellipsoid and the Bonferroni method. Which are wider? Why?

Task 3: Now, report the simultaneous confidence intervals for all 7 variables in the dataset using the projections of the ellipsoid and the Bonferroni method. Which are wider? Why?

Two-sample intervals and tests

Now, consider a related dataset with the same measurements of Portuguese *white* wines. We will apply the same transformation to those as for the red wines:

```
white.t <- read_csv(here("datasets","winequality-white.extract7.csv")) %>% mutate(`log sulphates`=log(`sulphates`), `root4 free sulfur dioxide` = (`free sulfur dioxide`)^(1/4), `root4 total sulfur dioxide` = (`total sulfur dioxide`)^(1/4), `log fixed acidity`=log(`fixed acidity`), `log volatile acidity`=log(`volatile acidity`), .keep="unused")
ggpairs(white.t)
```

Task 4: Produce a pairwise plot of the two datasets together, colour-coded by wine type. Comment on the apparent differences.

Task 5: Assuming equal variances, implement the 2-Sample Hotelling's T^2 test described in Section 4.3.2 and use it to test the null hypothesis that all of the features of red and white wines are equal in expectation. Calculate the test statistic (T^2), the $\alpha = 0.05$ critical value (i.e., rejection threshold) for T^2, and the p-value.

Task 6: Repeat Task 5 assuming unequal variances. Also report the effective sample size for variance estimation (ν).

Topic 1: Partial correlations

Welcome to Week 2

Dr Pavel Krivitsky gives you a brief overview of topics and concepts we'll be covering in this week.

Transcript

Weekly learning outcomes

- Calculate and interpret a partial correlation between two variables controlling for one or more variables.
- Perform a hypothesis test for a partial correlation and interpret the conclusion in the context of the problem.
- Explain the relationship between multiple correlation coefficient and R^2 in multiple regression.
- Explain the linear algebra foundations of a principal component analysis.
- Interpret the results of a principal component analysis, relating the principal components to underlying variables.
- Create and interpret a principal component biplot.
- Select the optimal number of principal components according to several techniques.
- Utilise principal components as a data reduction technique.

Topics we will cover are:

- Topic 1: Partial correlations
- Topic 2: Testing for partial correlations
- Topic 3: Multiple correlation coefficients
- Topic 4: Principal component analysis

Questions about this week's topics?

This week's topics were prepared by Dr P. Krivitsky. If you have any questions or comments, please post them under Discussion or email directly: p.krivitsky@unsw.edu.au

Partial correlation

Introduction

To begin Week 2, we will make some general comments on similarities and differences between correlations and dependencies.

Very often we are interested in correlations (dependencies) between a number of random variables and are trying to describe the “strength” of the (mutual) dependencies. For example, we would like to know if there is a correlation (mutual non-directed dependence) between the length of the arm and of the leg. But, if we would like to get information about (or to predict) the length of the arm by measuring the length of the leg, we are dealing with the dependence of the arm’s length on the leg’s length. Both problems described in this example make sense.

On the other hand, there are other examples/situations in which only one of the problems is interesting or makes sense. If we study the dependence between rain and crops, this makes a perfect sense but there is no sense at all to study the (directed) influence of crops on rain.

In a nutshell, we can say that when studying the mutual (linear) dependence, we are dealing with correlation theory whereas when studying directed influence of one (input) variable on another (output) variable, we are dealing with regression theory.

It should be clearly pointed out though that correlation alone, no matter how strong, can not help us identify the direction of influence and can not help us in regression modelling. Our reasoning about direction of influence should come outside of statistical theory, from another theory.

Another important point to always bear in mind is that, as already discussed in The Multivariate Normal Distribution, uncorrelated does not necessarily mean independent if the multivariate data happens to fail the multivariate normality test. Nonetheless, for multivariate normal data, the notions of "uncorrelated" and "independent" coincide.

In general, there are 3 types of correlation coefficients:

- The usual correlation coefficient between 2 variables
- *Partial correlation* coefficient between 2 variables after adjusting for the effect (regression, association) of a set of other variables
- *Multiple correlation* between a single random variable and a set of p other variables.

For further explanation, watch the below video by Dr Pavel Krivitsky.

Transcript

Partial correlation

For $\boldsymbol{X} \sim N_p(\boldsymbol{\mu}, \Sigma)$ we defined the correlation coefficient $\rho_{ij} = \frac{\sigma_{ij}}{\sqrt{\sigma_{ii}}\sqrt{\sigma_{jj}}}, i, j = 1, 2, \ldots, p$ and discussed the MLE $\hat{\rho}_{ij}$ in (1.12). It turned out that they coincide with the sample correlations r_{ij} we introduced in the Exploratory Data Analysis of Multivariate Data slide (formula (1.3)).

To define *partial correlation coefficients*, recall the Property 4 of the multivariate normal distribution from Week 1 (Properties of multivariate normal slide):

If vector $\boldsymbol{X} \in \mathbb{R}^p$ is divided into $X = \begin{pmatrix} X_{(1)} \\ X_{(2)} \end{pmatrix}, X_{(1)} \in R^r, r < p, \boldsymbol{X}_{(2)} \in R^{p-r}$ and according to this subdivision the vector means are $\boldsymbol{u} = \begin{pmatrix} \boldsymbol{\mu}_{(1)} \\ \boldsymbol{\mu}_{(2)} \end{pmatrix}$ and the covariance matrix Σ has been subdivided into $\Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}$ and the rank of Σ_{22} is full then the conditional density of $\boldsymbol{X}_{(1)}$ given that $\boldsymbol{X}_{(2)} = \boldsymbol{x}_{(2)}$ is

$$N_r\left(\boldsymbol{\mu}_{(1)} + \Sigma_{12}\Sigma_{22}^{-1}\left(\boldsymbol{x}_{(2)} - \boldsymbol{\mu}_{(2)}\right), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\right)$$

We **define** the partial correlations of $\boldsymbol{X}_{(1)}$ given $\boldsymbol{X}_{(2)} = \boldsymbol{x}_{(2)}$ as the usual correlation coefficients calculated from the elements $\sigma_{ij.(r+1),(r+2)\ldots,p}$ of the matrix $\Sigma_{1|2} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$, i.e.

$$\rho_{ij.(r+1),(r+2),\ldots,p} = \frac{\sigma_{ij.(r+1),(r+2),\ldots,p}}{\sqrt{\sigma_{ii.(r+1),(r+2),\ldots,p}}\sqrt{\sigma_{jj.(r+1),(r+2),\ldots,p}}} \tag{2.1}$$

We call $\rho_{ij.(r+1),(r+2),\ldots,p}$ the correlation of the ith and jth component when the components $(r+1),(r+2)$, etc. up to the pth (i.e. the last $p-r$ components) have been held fixed. The interpretation is that we are looking for the association (correlation) between the ith and jth component after eliminating the effect that the last $p-r$ components might have had on this association.

To find ML estimates for these, we use the translation invariance property of the MLE to claim that if $\hat{\mathbf{\Sigma}} = \begin{pmatrix} \hat{\mathbf{\Sigma}}11 & \hat{\mathbf{\Sigma}}12 \\ \hat{\mathbf{\Sigma}}21 & \hat{\mathbf{\Sigma}}22 \end{pmatrix}$ is the usual MLE of the covariance matrix then $\hat{\mathbf{\Sigma}}_{1|2} = \hat{\mathbf{\Sigma}}_{11} - \hat{\mathbf{\Sigma}}_{12}\hat{\mathbf{\Sigma}}_{22}^{-1}\hat{\mathbf{\Sigma}}_{21}$ with elements $\hat{\sigma}_{ij.(r+1),(r+2),\ldots,p}, i,j = 1,2,\ldots,r$ is the MLE of $\Sigma_{1|2}$ and correspondingly,

$$\hat{\rho}_{ij.(r+1),(r+2),\ldots,p} = \frac{\hat{\sigma}_{ij.(r+1),(r+2),\ldots,p}}{\sqrt{\hat{\sigma}_{ii.(r+1),(r+2),\ldots,p}}\sqrt{\hat{\sigma}_{jj.(r+1),(r+2),\ldots,p}}}, i,j = 1,2,\ldots,r$$

will be the ML estimators of $\rho_{ij.(r+1),(r+2)\ldots,p}, i,j = 1,2,\ldots,r$.

Transcript

Simple formulae

For situations when p is not large, as a partial case of the above general result, simple plug-in formulae are derived that express the partial correlation coefficients by the usual correlation coefficients. We shall discuss such formulae now. The formulae are given below:

1. Partial correlation between first and second variable by adjusting for the effect of the third:

$$\rho_{12.3} = \frac{\rho_{12} - \rho_{13}\rho_{23}}{\sqrt{(1-\rho_{13}^2)\,(1-\rho_{23}^2)}}$$

2. Partial correlation between first and second variable by adjusting for the effects of third and fourth variable:

$$\rho_{12.3,4} = \frac{\rho_{12.4} - \rho_{13.4}\rho_{23.4}}{\sqrt{(1-\rho_{13.4}^2)\,(1-\rho_{23.4}^2)}}.$$

Transcript

For higher dimensional cases computers need to be utilised:

```
R: ggm::pcor, ggm::parcor
```

Optional viewing: Partial Correlation Practice Problem

Statistics at Nevada State College. (2014). Partial Correlation Practice Problem. Retrieved from: https://youtu.be/8i0h98chSHU.

Example: Variables

Three variables have been measured for a set of schoolchildren:

1. X_1: Intelligence
2. X_2: Weight
3. X_3: Age

The number of observations was large enough so that one can assume the empirical correlation matrix $\hat{\rho} \in \mathcal{M}_{3,3}$ to be the true correlation matrix: $\begin{pmatrix} 1 & 0.6162 & 0.8267 \\ 0.6162 & 1 & 0.7321 \\ 0.8267 & 0.7321 & 1 \end{pmatrix}$.

This suggests there is a high degree of positive dependence between weight and intelligence. But $\hat{\rho}_{12.3} = 0.0286$ so that, after the effect of age is adjusted for, there is virtually no correlation between weight and intelligence, i.e. weight obviously plays little part in explaining intelligence.

This example is explained in the following video by Dr Pavel Krivitsky. Watch this before completing the demonstration and challenge activities in the next sections.

Transcript

Demonstration: Partial correlations

This demonstration can be completed using the provided RStudio or your own RStudio.

To complete this task select the 'Parcor_Example.demo.Rmd' in the 'Files' section of RStudio. Follow the demonstration contained within the RMD file.

If you choose to complete the example in your own RStudio, upload the following file:

Parcor_Example.demo.Rmd

The contents of the RMD file are also displayed below:

Packages

```
library(here)
library(readr)
library(dplyr)
library(GGally)
library(ggm)
```

Intelligence, weight and age

Suppose that three variables have been measured for a set of schoolchildren:

- X_1: Intelligence
- X_2: Weight
- X_3: Age

The number of observations was large enough so that one can assume the empirical correlation matrix $\hat{\rho} \in \mathcal{M}3, 3$ *to be the true correlation matrix:*

$$\hat{\rho} = \begin{pmatrix} 1 & 0.6162 & 0.8267 \\ 0.6162 & 1 & 0.7321 \\ 0.8267 & 0.7321 & 1 \end{pmatrix}.$$

This suggests there is a high degree of positive dependence between weight and intelligence. But let's compute the partial correlation adjusting for age:

$$\hat{\rho}_{12.3} = \frac{\hat{\rho}_{12} - \hat{\rho}_{13}\hat{\rho}_{23}}{\sqrt{(1-\hat{\rho}_{13}^2)(1-\hat{\rho}_{23}^2)}} = \frac{0.6162 - 0.8267 \times 0.7321}{\sqrt{(1-0.8267^2)(1-0.7321^2)}} = 0.0286.$$

Alternatively, R can do it for us:

```
R <- matrix(c(1     , 0.6162, 0.8267,
              0.6162, 1     , 0.7321,
              0.8267, 0.7321, 1     ),
            3,3)
# Calculate the partial correlations for each pair of variables given the rest, and
take the one between X1 and X2:
parcor(R)[1,2]

# Calculate the partial correlation between variables 1 and 2 given variable 3 in R:
pcor(c(1:2,3), R)
```

Either way, after the effect of age is adjusted for, there is virtually no correlation between weight and intelligence, i.e. weight obviously plays little part in explaining intelligence.

We will also use these data to study *testing* correlations.

Challenge: Partial correlations

If you choose to complete this task in your own RStudio, upload the following file:

Parcor_Example.challenge.Rmd

Click on the 'Parcor_Example.challenge.Rmd' in the 'Files' section to begin. Enter your response to the tasks in the 'Enter your code here' section.

This activity and the solution will be discussed at the Collaborate session this week. In the meantime, share and discuss your results in the 'Tutorials' discussion forum.

The solution will also be available here on Friday of this week by clicking on the 'Solution' tab in the top right corner.

The contents of the RMD file are also displayed below:

Packages

```
library(here)
library(readr)
library(dplyr)
library(GGally)
library(ggm)
```

Challenge

Recall the Portuguese red wines data:

```
red.t <- read_csv(here("datasets","winequality-red.extract7.csv")) %>% mutate(`log sulphates`=log(`sulphates`), `root4 free sulfur dioxide` = (`free sulfur dioxide`)^(1/4), `root4 total sulfur dioxide` = (`total sulfur dioxide`)^(1/4), `log fixed acidity`=log(`fixed acidity`), `log volatile acidity`=log(`volatile acidity`), .keep="unused")
ggpairs(red.t)
```

Consider the variables `pH`, `log fixed acidity` and `log volatile acidity`.

Task 1: Calculate the correlation between pH and log volatile acidity and their partial correlation given log fixed acidity. How are they different? Interpret the partial correlation and the difference between correlation and partial correlation in this case.

Task 2: Calculate the partial correlation between density and log sulphates content, given free and total

sulphur dioxide.

Topic 2: Multiple correlation coefficients

Multiple correlation

Multiple correlations is explained by Dr Pavel Krivitsky in the following video.

Transcript

Recall our discussion at the end of slide "Properties of multivariate normal", for the best prediction in mean squares sense in case of multivariate normality: If we want to predict a random variable Y that is correlated with p random variables (predictors) $X = (X_1 \quad X_2 \quad \cdots \quad X_p)^T$ by trying to minimise the expected value $\mathrm{E}(Y - g(\boldsymbol{X})|\boldsymbol{X} = \boldsymbol{x})^2$ the optimal solution (i.e. the regression function) was $g^*(\boldsymbol{X}) = \mathrm{E}(Y \mid \boldsymbol{X})$.

When the joint $(p+1)$-dimensional distribution of Y and $\boldsymbol{X}$ is **normal** this function was **linear** in $\boldsymbol{X}$. Given a specific realisation $\boldsymbol{x}$ of $\boldsymbol{X}$ it was given by $b + \boldsymbol{\sigma}_0^\top C^{-1}\boldsymbol{x}$ where $b = \mathrm{E}(Y) - \boldsymbol{\sigma}_0^\top C^{-1}\mathrm{E}(\boldsymbol{X}), C$ is the covariance matrix of the vector $\boldsymbol{X}$, $\boldsymbol{\sigma}_0$ is the vector of Covariances of Y with $X_i, i = 1, \ldots, p$. The vector $C^{-1}\boldsymbol{\sigma}_0 \in \mathbb{R}^p$ was the *vector of the regression coefficients.*

Now, let us **define** the multiple correlation coefficient between the random variable Y and the random vector $\boldsymbol{X} \in \mathbb{R}^p$ to be the maximum correlation between Y and *any linear combination* $\boldsymbol{\alpha}^\top \boldsymbol{X}, \boldsymbol{\alpha} \in \mathbb{R}^p$. This makes sense to look at the maximal correlation that we can get by trying to

predict Y as a linear function of the predictors. The solution to this which also gives us an algorithm to calculate (and estimate) the multiple correlation coefficient is given in the next lemma.

Multiple correlation coefficient as ordinary correlation coefficient of transformed data

Lemma 2.1. *The multiple correlation coefficient is the ordinary correlation coefficient between* Y *and* $\boldsymbol{\sigma}_0^\top C^{-1} X \equiv \beta^\top X$. $\left(\text{I.e., } \beta \equiv C^{-1}\boldsymbol{\sigma}_0.\right)$

Coefficient of Determination From Lemma 2.1 the maximum correlation between Y and any linear combination $\boldsymbol{\alpha}^\top \boldsymbol{X}, \boldsymbol{\alpha} \in \mathbb{R}^p$ is $R = \sqrt{\frac{\boldsymbol{\beta}^{*\top} C \boldsymbol{\beta}^*}{\sigma_Y^2}}$. This is the multiple correlation coefficient. Its square R^2 is called *coefficient of determination.* Having in mind that $\boldsymbol{\beta}^* = C^{-1}\boldsymbol{\sigma}_0$ we see that $R = \sqrt{\frac{\boldsymbol{\sigma}_0^\top C^{-1}\boldsymbol{\sigma}_0}{\sigma_Y^2}}$.

If $\Sigma = \begin{pmatrix} \sigma_Y^2 & \boldsymbol{\sigma}_0^\top \\ \boldsymbol{\sigma}_0 & C \end{pmatrix} = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}$ is the partitioned covariance matrix of the $(p+1)$-dimensional vector $(Y, \boldsymbol{X})^\top$ then we know how to calculate the MLE of Σ by $\hat{\boldsymbol{\Sigma}} = \begin{pmatrix} \hat{\boldsymbol{\Sigma}}_{11} & \hat{\boldsymbol{\Sigma}}_{12} \\ \hat{\boldsymbol{\Sigma}}_{21} & \hat{\boldsymbol{\Sigma}}_{22} \end{pmatrix}$ so the MLE of R would be $\hat{R} = \sqrt{\frac{\hat{\Sigma}_{12}\hat{\Sigma}_{22}^{-1}\hat{\Sigma}_{21}}{\hat{\Sigma}_{11}}}$.

Interpretation of R

At the end of slide "Properties of multivariate normal", we derived the minimal value of the mean squared error when trying to predict Y by a linear function of the vector $\boldsymbol{X}$. It is achieved when

using the regression function and the value itself was $\sigma_Y^2 - \boldsymbol{\sigma}_0^\top C^{-1} \boldsymbol{\sigma} 0$. The latter value can also be expressed by using the value of R. It is equal to $\sigma_Y^2 (1 - R^2)$.

Thus, our conclusion is that when $R^2 = 0$ there is no predictive power at all. In the opposite extreme case, if $R^2 = 1$, it turns out that Y can be predicted without any error at all (it is a true linear function of $\boldsymbol{X}$).

Activity: Numerical example

Let $\mu = \begin{pmatrix} \mu_Y \\ \mu_{X_1} \\ \mu_{X_2} \end{pmatrix} = \begin{pmatrix} 5 \\ 2 \\ 0 \end{pmatrix}$ and $\Sigma = \begin{pmatrix} 10 & 1 & -1 \\ 1 & 7 & 3 \\ -1 & 3 & 2 \end{pmatrix} = \begin{pmatrix} \sigma_{YY} & \boldsymbol{\sigma}_0^\top \\ \boldsymbol{\sigma}_0 & \Sigma_{XX} \end{pmatrix}$.

Question

Calculate:

1. The best linear prediction of Y using X_1 and X_2
2. The multiple correlation coefficient $R^2_{Y.(X_1,X_2)}$
3. The mean squared error of the best linear predictor.

No response

Calculation of the coefficient of determination

Remark about the calculation of R^2

Sometimes, the *correlation matrix only* may be available. It can be shown that in that case the relation

$$1 - R^2 = \frac{1}{\rho^{YY}} \tag{2.2}$$

is the upper left-hand corner of the inverse of the *correlation matrix* $\boldsymbol{\rho} \in \mathcal{M}_{p+1,p+1}$ determined from Σ.

The following proof is not examinable.

We note that the relation $\boldsymbol{\rho} = V^{-\frac{1}{2}} \Sigma V^{-\frac{1}{2}}$ holds with

$$V = \begin{pmatrix} \sigma_y^2 & 0 & 0 & \cdots & 0 \\ 0 & c_{11} & 0 & \cdots & 0 \\ 0 & 0 & c_{22} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & c_{pp} \end{pmatrix}$$

One can use 2.2 to calculate R^2 by first calculating the right hand side in 2.2. To show Equality 2.2 we note that

$$1 - R^2 = \frac{\sigma_Y^2 - \boldsymbol{\sigma}_0^\top C^{-1} \boldsymbol{\sigma}_0}{\sigma_Y^2} = \frac{|C|}{|C|} \frac{\sigma_Y^2 - \boldsymbol{\sigma}_0^\top C^{-1} \boldsymbol{\sigma}_0}{\sigma_Y^2} = \frac{|\Sigma|}{|C| \sigma_Y^2}$$

But $\frac{|C|}{|\Sigma|} = \sigma^{YY}$, the entry in the first row and column of Σ^{-1}. (Recall from slide "Inverse matrices": $(X^{-1})_{ji} = \frac{|X_{ij}|}{|X|}(-1)^{i+j}$.) Since $\boldsymbol{\rho}^{-1} = V^{\frac{1}{2}} \Sigma^{-1} V^{\frac{1}{2}}$, we see that $\rho^{YY} = \sigma^{YY} \sigma_Y^2$ holds. Therefore $1 - R^2 = \frac{1}{\rho^{YY}}$.

Demonstration: Total correlations

This demonstration can be completed using the provided RStudio or your own RStudio.

To complete this task select the 'Totcor_Example.demo.Rmd' in the 'Files' section of RStudio. Follow the demonstration contained within the RMD file.

If you choose to complete the example in your own RStudio, upload the following file:

Totcor_Example.demo.Rmd

The contents of the RMD file are also displayed below:

Packages

```
library(GGally)
```

Equivalence between total correlation and regression

This example demonstrates that total correlation is equivalent to the square root of the coefficient of determination of a linear regression.

```
x1 <- rnorm(100)
x2 <- rnorm(100)+x1
y <- x1*2+x2+rnorm(100)

ggpairs(data.frame(y,x1,x2))
```

Now, consider the following equivalent ways to compute it:

```
# R^2 reported by summary.lm():
summary(lmfit <- lm(y~x1+x2))

# Correlation between the response and the fitted values:
cor(predict(lmfit), y)
cor(predict(lmfit), y)^2

# From the correlation matrix:
1-1/solve(cor(cbind(y,x1,x2)))[1,1]
```

Topic 3: Testing correlation coefficients

Usual correlation coefficients

When considering the distribution of a particular correlation coefficient $\hat{\rho}_{ij} = r_{ij}$ the problem becomes bivariate because only the variables X_i and X_j are involved. Direct transformations with the bivariate normal can be utilised to derive the **exact** distribution of r_{ij} under the hypothesis $H_0 : \rho_{ij} = 0$. It turns out that in this case the statistic $T = r_{ij}\sqrt{\frac{n-2}{1-r_{ij}^2}} \sim t_{n-2}$ and tests can be performed by using the tables of the t-distribution. For other hypothetical values the derivations are more painful. There is one most frequently used **approximation** that holds no matter what the true value of ρ_{ij} is. We shall discuss it here. Consider **Fisher's Z transformation** $Z = \frac{1}{2}\log\left[\frac{1+r_{ij}}{1-r_{ij}}\right]$. Under the hypothesis $H_0 : \rho_{ij} = \rho_0$ it holds:

$$Z \approx N\left(\frac{1}{2}\log\left[\frac{1+\rho_0}{1-\rho_0}\right], \frac{1}{n-3}\right)$$

In particular, in the most common situation, when one would like to test $H_0 : \rho_{ij} = 0$ versus $H_1 : \rho_{ij} \neq 0$ one would reject H_0 at 5% significance level if $|Z|\sqrt{n-3} \geq 1.96$.

Partial correlation coefficients

Coming over to testing *partial correlations* not much has to be changed. Fisher's Z approximation can be used again in the following way: to test $H_0 : \rho_{ij.r+1,r+2,\ldots,p} = \rho_0$ versus $H_1 : \rho_{ij.r+1,r+2,\ldots,p} \neq \rho_0$ we construct $Z = \frac{1}{2} \log \left[\frac{1+r_{ij.r+1,r+2,\ldots,p}}{1-r_{ij.r+1,r+2,\ldots,p}} \right]$ and $a = \frac{1}{2} \log[\frac{1+\rho_0}{1-\rho_0}]$. Asymptotically $Z \sim N(a, \frac{1}{n-(p-r)-3})$ holds. Hence, test statistic to be compared with significance points of the standard normal is now : $\sqrt{n-(p-r)-3}|Z-a|$. (Notice that $p-r$ is the number of variables being conditioned on.)

For $H_0 : \rho_{ij} = 0$, we can also use the (more exact) t-test, $T = r_{ij}\sqrt{\frac{n-2}{1-r_{ij}^2}} \sim t_{n-2-(p-r)}$.

Multiple correlation coefficients

It turns out that under the hypothesis $H_0 : R = 0$ the statistic $F = \frac{\hat{R}^2}{1-\hat{R}^2} \times \frac{n-p-1}{p} \sim F_{p,n-p-1}$. Hence, when testing significance of the multiple correlation, the rejection region would be $\left\{ \frac{\hat{R}^2}{1-\hat{R}^2} \times \frac{n-p-1}{p} > F_{1-\alpha,p,n-p-1} \right\}$ for a given significance level α.

Remark 2.1. This expression should be familiar in the context of the ANOVA F-test; there, p would be the number of predictors, and the denominator degrees of freedom would be $n - p - 1$, with the -1 being for the intercept.

Check your understanding

Complete the below exercises to check your understanding of concepts presented so far.

1. Suppose $\boldsymbol{X} \sim N_4(\mu, \Sigma)$ where $\mu = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix}$ and $\Sigma = \begin{pmatrix} 3 & 1 & 0 & 1 \\ 1 & 4 & 0 & 0 \\ 0 & 0 & 1 & 4 \\ 1 & 0 & 4 & 20 \end{pmatrix}$

Determine:

a) the distribution of $\begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_1 + X_2 + X_4 \end{pmatrix}$

b) the conditional mean and variance of X_1 given x_2, x_3, and x_4;

c) the partial correlation coefficients $\rho_{12.3}, \rho_{12.4}$;

d) the multiple correlation between X_1 and (X_2, X_3, X_4). Compare it to ρ_{12} and comment;

e) Justify that $\begin{pmatrix} X_2 \\ X_3 \\ X_4 \end{pmatrix}$ is independent of $X_1 - \begin{pmatrix} 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 4 & 0 & 0 \\ 0 & 1 & 4 \\ 0 & 4 & 20 \end{pmatrix}^{-1} \begin{pmatrix} X_2 \\ X_3 \\ X_4 \end{pmatrix}$.

2. Consider a random vector $\boldsymbol{X} \sim N_3(\boldsymbol{\mu}, \Sigma)$ with $\boldsymbol{\mu} = \begin{pmatrix} 2 \\ -3 \\ 1 \end{pmatrix}$ and $\Sigma = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 3 & 2 \\ 1 & 2 & 2 \end{pmatrix}$.

a) Find the distribution of $3X_1 - 2X_2 + X_3$.

b) Find a vector $\boldsymbol{a} \in \mathbb{R}^2$ *such that* X_2 and $X_2 - \boldsymbol{a}^\top \begin{pmatrix} X_1 \\ X_3 \end{pmatrix}$ are independent.

3. Suppose that you observe two independent bivariate samples: (X_{1i}, X_{2i}) for $i = 1, \ldots, n_X$ and, independently, (Y_{1i}, Y_{2i}) for $i = 1, \ldots, n_Y$. Describe how you would use the Fisher's Z transformation to test the null hypothesis $H_0 : \rho_X = \rho_Y$, where ρ_X is the correlation between X_1 and X_2 and similarly for Y.

Demonstration: Hypothesis tests of correlations

This demonstration can be completed using the provided RStudio or your own RStudio.

To complete this task select the 'Cor_Test_Example.demo.Rmd' in the 'Files' section of RStudio. Follow the demonstration contained within the RMD file.

If you choose to complete the example in your own RStudio, upload the following file:

Cor_Test_Example.demo.Rmd

The contents of the RMD file are also displayed below:

Packages

```
library(here)
library(readr)
library(dplyr)
library(GGally)
library(ggm)
```

Intelligence, weight and age

Data

Recall the 3-variable example from before:

- X_1: Intelligence
- X_2: Weight
- X_3: Age

Suppose that these correlations are actually based on sample size $n = 20$:

$$\hat{\rho} = \begin{pmatrix} 1 & 0.6162 & 0.8267 \\ 0.6162 & 1 & 0.7321 \\ 0.8267 & 0.7321 & 1 \end{pmatrix}.$$

```
R <- matrix(c(1     , 0.6162, 0.8267,
              0.6162, 1     , 0.7321,
              0.8267, 0.7321, 1     ),
            3,3)
```

Testing correlations

We wish to test if there is sufficient evidence that weight and intelligence are correlated.

Ordinary correlation

We can begin by using the t-test:

$$T = \hat{\rho}ij\sqrt{\frac{n-2}{1-\hat{\rho}ij^2}} = 0.6162\sqrt{\frac{n-2}{1-0.6162^2}} = 3.3194.$$

The t distribution has $n-2$ degrees of freedom:

```
R <- matrix(c(1     , 0.6162, 0.8267,
              0.6162, 1     , 0.7321,
              0.8267, 0.7321, 1     ),
            3,3)

2*pt(abs(R[1,2]*sqrt((20-2)/(1-R[1,2]^2))), 20-2, lower.tail=FALSE)
```

We can also use the `pcor.test()` function in `ggm`:

```
pcor.test(R[1,2], 0, 20)
```

Alternatively, we can use Fisher's Z approximation:

$$Z = \frac{1}{2}\log\frac{1+\hat{\rho}12}{1-\hat{\rho}12} = \frac{1}{2}\log\frac{1+0.6162}{1-0.6162} = 0.7189.$$

Under the null hypothesis, this statistic will have mean 0 and variance $1/(n-3)$, so we compute a two-sided p-value as

```
2*pnorm(abs(log((1+R[1,2])/(1-R[1,2]))/2), 0, sqrt(1/(20-3)), lower.tail=FALSE)
```

We conclude that the correlation is highly significant.

Partial correlation

To test partial correlation given age, we compute the sample value,

```
(r12.3 <- pcor(c(1,2,3), R))
```

Then,

$$Z = \frac{1}{2}\log\frac{1+\hat{\rho}_{12.3}}{1-\hat{\rho}_{12.3}} = \frac{1}{2}\log\frac{1+0.02863}{1-0.02863} = 0.02864,$$

and the variance is $1/(n-(p-r)-3)$. Here $p-r$ is the number of conditioned-on variables, which is in this case 1.

```
2*pnorm(abs(log((1+r12.3)/(1-r12.3))/2), 0, sqrt(1/(20-1-3)), lower.tail=FALSE)
```

Similarly, we can use the t-test, using `pcor.test()`:

```
pcor.test(r12.3, 1, 20)
```

We conclude that there is not sufficient evidence of correlation.

Challenge: Hypothesis tests of correlations

If you choose to complete this task in your own RStudio, upload the following file:

Cor_Test_Example.challenge.Rmd

Click on the 'Cor_Test_Example.challenge.Rmd' in the 'Files' section to begin. Enter your response to the tasks in the 'Enter your code here' section.

This activity and the solution will be discussed at the Collaborate session this week. In the meantime, share and discuss your results in the 'Tutorials' discussion forum.

The solution will also be available here on Friday of this week by clicking on the 'Solution' tab in the top right corner.

The contents of the RMD file are also displayed below:

Packages

```
library(here)
library(readr)
library(dplyr)
library(GGally)
library(ggm)
```

Challenge

Recall the Portuguese red wines data:

```
red.t <- read_csv(here("datasets","winequality-red.extract7.csv")) %>% mutate(`log sulphates`=log(`sulphates`), `root4 free sulfur dioxide` = (`free sulfur dioxide`)^(1/4), `root4 total sulfur dioxide` = (`total sulfur dioxide`)^(1/4), `log fixed acidity`=log(`fixed acidity`), `log volatile acidity`=log(`volatile acidity`), .keep="unused")
ggpairs(red.t)
```

Consider the variables `pH`, `log fixed acidity` and `log volatile acidity`.

Task 1: Test the presence of correlation between pH and log volatile acidity, and interpret the results.

Task 2: Test the presence of partial correlation between pH and log volatile acidity given log fixed acidity, and interpret the results.

Topic 4: Principal component analysis

Introduction

Principal component analysis is applied mainly as a **variable reduction procedure**. It is usually applied in cases when data is obtained from a possibly **large number** of variables which are possibly **highly correlated**. The goal is to try to "condense" the information.

This is done by summarising the data in a (small) number of transformations of the original variables. Our motivation to do that is that we believe there is some redundancy in the presentation of the information by the original set of variables since e.g. many of these variables are measuring the same construct. In that case we try to reduce the observed variables into a smaller number of **principal components** (artificial variables) that would account for most of the variability in the observed variables.

For simplicity, these artificial new variables are presented as a **linear combinations** of the (**optimally weighted**) observed variables. If one linear combination is not enough, we can choose to construct two, three, etc. such combinations. Note also that principal components analysis may be just an intermediate step in much larger investigations. The principal components obtained can be used for example as inputs in a regression analysis or in a cluster analysis procedure. They are also a basic method in extracting factors in factor analysis.

For a further introduction to principal component analysis, watch the below video by Dr Pavel Krivitsky.

Precise mathematical formulation

Before we begin, let's examine the concepts presented in this section by watching the below video.

Let $\boldsymbol{X} \sim N_p(\boldsymbol{\mu}, \Sigma)$ where p is assumed to be relatively large. To perform a reduction, we are looking for a linear combination $\boldsymbol{\alpha}_1^\top \boldsymbol{X}$ with $\boldsymbol{\alpha}_1 \in \mathbb{R}^p$ suitably chosen such that it maximises the variance of $\boldsymbol{\alpha}_1^\top \boldsymbol{X}$ subject to the reasonable normalising constraint $\|\boldsymbol{\alpha}_1\|^2 =$. Since $\mathrm{Var}(\boldsymbol{\alpha}_1^\top \boldsymbol{X}) = \boldsymbol{\alpha}_1^\top \Sigma \boldsymbol{\alpha}_1$ we need to choose $\boldsymbol{\alpha}_1$ to maximise $\boldsymbol{\alpha}_1^\top \Sigma \boldsymbol{\alpha}_1$ subject to $\alpha_1^\top \alpha_1 = 1$.

This requires us to apply Lagrange's optimisation under constraint procedure. You will not be examined on the derivation itself, but it is helpful to examine it to understand the relationship between the principal components and the eigenvalues and the eigenvectors of Σ.

1. construct the Lagrangian function

$$\mathrm{Lag}\left(\boldsymbol{\alpha}_1, \lambda\right) = \boldsymbol{\alpha}_1^\top \Sigma \boldsymbol{\alpha}_1 + \lambda \left(1 - \boldsymbol{\alpha}_1^\top \boldsymbol{\alpha}_1\right)$$

   where $\lambda \in \mathbb{R}^1$ is the Lagrange multiplier;
2. take the partial derivative with respect to $\boldsymbol{\alpha}_1$ and equate it to zero:

$$2\Sigma\boldsymbol{\alpha}_1 - 2\lambda\boldsymbol{\alpha}_1 = \mathbf{0} \Longrightarrow \left(\Sigma - \lambda I_p\right)\boldsymbol{\alpha}_1 = 0 \tag{2.3}$$

From (2.3) we see that $\boldsymbol{\alpha}_1$ must be an eigenvector of Σ and since we know from the Example 0.1 what the maximal value of $\frac{\boldsymbol{\alpha}^\top \Sigma \boldsymbol{\alpha}}{\boldsymbol{\alpha}^\top \boldsymbol{\alpha}}$ is, we conclude that $\boldsymbol{\alpha}_1$ should be the **eigenvector that**

corresponds to the largest eigenvalue $\bar{\lambda}_1$ of Σ. The random variable $\boldsymbol{\alpha}_1^\top \boldsymbol{X}$ is called the **first principal component**.

For the **second** principal component $\boldsymbol{\alpha}_2^\top \boldsymbol{X}$ we want it to be normalised according to $\boldsymbol{\alpha}_2^\top \boldsymbol{\alpha}_2 = 1$, uncorrelated with the first component and to give maximal variance of a linear combination of the components of $\boldsymbol{X}$ under these constraints. To find it, we construct the Lagrange function:

$$\mathrm{Lag}_1\left(\boldsymbol{\alpha}_2, \lambda_1, \lambda_2\right) = \boldsymbol{\alpha}_2^\top \Sigma \boldsymbol{\alpha}_2 + \lambda_1\left(1 - \boldsymbol{\alpha}_2^\top \boldsymbol{\alpha}_2\right) + \lambda_2 \boldsymbol{\alpha}_1^\top \Sigma \boldsymbol{\alpha}_2$$

Its partial derivative w.r.t. $\boldsymbol{\alpha}_2$ gives

$$2\Sigma\boldsymbol{\alpha}_2 - 2\lambda_1\boldsymbol{\alpha}_2 + \lambda_2\Sigma\boldsymbol{\alpha}_1 = \mathbf{0} \tag{2.4}$$

Multiplying (2.4) by $\boldsymbol{\alpha}_1^\top$ from left and using the two constraints $\boldsymbol{\alpha}_2^\top\boldsymbol{\alpha}_2 = 1$ and $\boldsymbol{\alpha}_2^\top\Sigma\boldsymbol{\alpha}_1 = 0$ gives:

$$-2\lambda_1\boldsymbol{\alpha}_1^\top\boldsymbol{\alpha}_2 + \lambda_2\boldsymbol{\alpha}_1^\top\Sigma\boldsymbol{\alpha}_1 = 0 \Longrightarrow \lambda_2 = 0$$

Have in mind that $\boldsymbol{\alpha}_1$ is an eigenvector of Σ. Note:

$$\Sigma\boldsymbol{\alpha}_1 = \bar{\lambda}_1\boldsymbol{\alpha}_1 \Longrightarrow \boldsymbol{\alpha}_1^\top\boldsymbol{\alpha}_2 = \frac{1}{\lambda_1}\boldsymbol{\alpha}_1^\top\Sigma\boldsymbol{\alpha}_2 = 0$$

But then (2.4) also implies that $\boldsymbol{\alpha}_2 \in \mathbb{R}^p$ must be an eigenvector of Σ (has to satisfy $(\Sigma - \lambda_1 I_p)\boldsymbol{\alpha}2 = \mathbf{0}$). Since it has to be different from $\boldsymbol{\alpha}_1$, having in mind that we aim at variance maximisation, we see that $\boldsymbol{\alpha}_2$ has to be the normalised eigenvector that corresponds to the second largest eigenvalue $\bar{\lambda}_2$ of Σ. The process can be continued further. The third principal component should be uncorrelated with the first two, should be normalised and should give maximal variance of a linear combination of the components of $\boldsymbol{X}$ under these constraints. One can easily realise then that the vector $\boldsymbol{\alpha}_3 \in \mathbb{R}^p$ in the formula $\boldsymbol{\alpha}_3^\top\boldsymbol{X}$ should be the normalised eigenvector that corresponds to the third largest eigenvalue $\bar{\lambda}_3$ of the matrix Σ etc.

Note that if we extract **all possible** p principal components then $\sum_{i=1}^p \mathrm{Var}(\boldsymbol{\alpha} i^\top \boldsymbol{X})$ will just equal the sum of all eigenvalues of Σ and hence

$$\sum_{i=1}^p \mathrm{Var}\left(\boldsymbol{\alpha}_i^\top\boldsymbol{X}\right) = \mathrm{tr}(\Sigma) = \Sigma_{11} + \cdots + \Sigma_{pp}$$

Therefore, if we only take a small number of k principal components instead of the total possible number p we can interpret their inclusion as one that explains a $\frac{\mathrm{Var}\left(\boldsymbol{\alpha}_1^\top\boldsymbol{X}\right) + \cdots + \mathrm{Var}\left(\boldsymbol{\alpha}_k^\top\boldsymbol{X}\right)}{\Sigma_{11} + \cdots + \Sigma_{pp}} \times 100\%$ $= \frac{\bar{\lambda}_1 + \cdots + \bar{\lambda}_k}{\sum_{11} + \cdots + \Sigma_{pp}} \times 100\%$ of the total population variance $\Sigma_{11} + \cdots + \Sigma_{pp}$.

Estimation of the principal components

In practice, Σ is unknown and has to be estimated. The principal components are derived from the normalised eigenvectors of the estimated covariance matrix.

Note also that extracting principal components from the (estimated) covariance matrix has the drawback that it is influenced by the scale of measurement of each variable X_i, $i = 1, \ldots, p$. A variable with large variance will necessarily be a large component in the first principal component (note the goal of explaining **the bulk** of variability by using the first principal component). Yet the large variance of the variable may be just an artefact of the measurement scale used for this variable. Therefore, an alternative practice is adopted sometimes to extract principal components from the correlation matrix $\boldsymbol{\rho}$ instead of the covariance matrix Σ.

Example 2.1 (Eigenvalues obtained from Covariance and Correlation Matrices: see page 437 Johnston and Wichern)**.** It demonstrates the great effect standardisation may have on the principal components. The relative magnitudes of the weights after standardisation (i.e. from $\boldsymbol{\rho}$ may become in direct opposition to the weights attached to the same variables in the principal component obtained from Σ.

For the reasons mentioned above, variables are often **standardised** before sample principal components are extracted. Standardisation is accomplished by calculating the vectors $\boldsymbol{Z}_i = \begin{pmatrix} \frac{X_{1i}-\bar{X}_1}{\sqrt{s_{11}}} & \frac{X_{2i}-\bar{X}_2}{\sqrt{s_{22}}} & \ldots & \frac{X_{pi}-\bar{X}_p}{\sqrt{s_{pp}}} \end{pmatrix}^\top, i = 1, \ldots, n$. The standardised observations matrix $\boldsymbol{Z} = [\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots, \boldsymbol{Z}_n] \in \mathcal{M}_{p,n}$ gives the sample mean vector $\bar{\boldsymbol{Z}} = \frac{1}{n}\boldsymbol{Z}\boldsymbol{1}_n = 0$ and a sample covariance matrix $\boldsymbol{S}_{\boldsymbol{Z}} = \frac{1}{n-1}\boldsymbol{Z}\boldsymbol{Z}^\top = \boldsymbol{R}$ (the correlation matrix of the original observations). The principal components are extracted in the usual way from $\boldsymbol{R}$ now.

Watch the below video by Dr Pavel Krivitsky for a more in depth explanation of these concepts.

Deciding how many principal components to include

To reduce the dimensionality (which is the motivating goal), we should restrict attention to the first k principal components and ideally, k should be kept much less than p but there is a trade-off to be made here since we would also like the proportion $\psi_k = \frac{\bar{\lambda}_1 + \ldots \bar{\lambda}_k}{\lambda_1 + \ldots \lambda_p}$ be close to one. How could a reasonable trade-off be made? Three methods are most widely used:

- The scree plot": basically, it is a graphical method of plotting the ordered $\bar{\lambda}_k$ against k and deciding visually when the plot has flattened out. Typically, the initial part of the plot is like the side of the mountain, while the flat portion where each $\bar{\lambda}_k$ is just slightly smaller than $\bar{\lambda}_{k-1}$, is like the rough scree at the bottom. This motivates the name of the plot. The task here is to find where "the scree begins".
- Choose an arbitrary constant $c \in (0, 1)$ and choose k to be the smallest one with the property $\psi_k \geq c$. Usually, $c = 0.9$ is used but please, note the arbitrariness of the choice here.
- **Kaiser's rule:** it suggests that from all p principal components only the ones should be retained whose variances (after standardisation) are greater than unity, or, equivalently, only those components which, individually, explain at least $\frac{1}{p}100\%$ of the total variance. (This is the same as excluding all principal components with eigenvalues less than the overall average). This criterion has a number of positive features that have contributed to its popularity but can not be defended on a safe theoretical ground.
- Formal tests of significance. Note that it actually **does not make sense** to test whether $\bar{\lambda}_{k+1} = \cdots = \bar{\lambda}_p = 0$ since if such a hypothesis were true then the population distribution would be contained **entirely** within a k-dimensional subspace and the same would be true for any **sample** from this distribution, hence we would have the **estimated** $\bar{\lambda}$ values for indices $k+1, \ldots, p$ being also equal to zero with probability one! What seems to be reasonable to do instead, is to test $H_0 : \bar{\lambda}_{k+1} = \cdots = \bar{\lambda}_p$ (without asking the common value to be zero). This is a more quantitative variant of the scree test. A test for this hypothesis is to form the arithmetic and geometric means $a_0 =$ arithmetic mean of the last $p-k$ estimated eigenvalues; $g_0 =$ geometric mean of the last $p-k$ estimated eigenvalues, and then construct $-2\log\lambda = n(p-k)\log\frac{a_0}{g_0}$. The asymptotic distribution of this statistic under the null hypothesis is χ^2_ν where $\nu = \frac{(p-k+2)(p-k-1)}{2}$. The interested student can find more details about this test in the monograph of Mardia, Kent and Bibby. We should note, however, that the last result holds under multivariate normality assumption and is only valid as stated for the **covariance-based** (**not** the correlation-based) version of the principal component analysis. In practice, many data analysts are reluctant to make a multivariate normality assumption at the early stage of the descriptive data analysis and hence distrust the above quantitative test but prefer the simple Kaiser criterion.

Implementations

Principal components analysis in R using `stats::prcomp, stats::princomp,` or about half-dozen other implementations.

Optional viewing: StatQuest: PCA in R

StatQuest with Josh Starmer. (2016). StatQuest: PCA in R. Retrieved from https://youtu.be/0Jp4gsfOLMs

Demonstration: Crime rates in the USA

This demonstration can be completed using the provided RStudio or your own RStudio.

To complete this task select the 'PCA_Examples.demo.Rmd' in the 'Files' section of RStudio. Follow the demonstration contained within the RMD file.

If you choose to complete the example in your own RStudio, upload the following file:

PCA_Examples.demo.Rmd

The contents of the RMD file are also displayed below:

Packages

```
library(here)
library(GGally)
library(dplyr)
library(readr)
library(purrr)
```

Example: Crime rates in US states

Data

The following are rates of certain crimes in US states, with their two-letter abbreviations.

```
'murder rape robbery assault burglary larceny auto
"AL" 14.2 25.2  96.8 278.3 1135.5 1881.9  280.7
"AK" 10.8 51.6  96.8 284.0 1331.7 3369.8  753.3
"AZ"  9.5 34.2 138.2 312.3 2346.1 4467.4  439.5
"AR"  8.8 27.6  83.2 203.4  972.6 1862.1  183.4
"CA" 11.5 49.4 287.0 358.0 2139.4 3499.8  663.5
"CO"  6.3 42.0 170.7 292.9 1935.2 3903.2  477.1
"CT"  4.2 16.8 129.5 131.8 1346.0 2620.7  593.2
"DE"  6.0 24.9 157.0 194.2 1682.6 3678.4  467.0
"FL" 10.2 39.6 187.9 449.1 1859.9 3840.5  351.4
"GA" 11.7 31.1 140.5 256.5 1351.1 2170.2  297.9
"HI"  7.2 25.5 128.0  64.1 1911.5 3920.4  489.4
"ID"  5.5 19.4  39.6 172.5 1050.8 2599.6  237.6
"IL"  9.9 21.8 211.3 209.0 1085.0 2828.5  528.6
"IN"  7.4 26.5 123.2 153.5 1086.2 2498.7  377.4
"IA"  2.3 10.6  41.2  89.8  812.5 2685.1  219.9
```

```
"KS"  6.6 22.0 100.7 180.5 1270.4 2739.3  244.3
"KY" 10.1 19.1  81.1 123.3  872.2 1662.1  245.4
"LA" 15.5 30.9 142.9 335.5 1165.5 2469.9  337.7
"ME"  2.4 13.5  38.7 170.0 1253.1 2350.7  246.9
"MD"  8.0 34.8 292.1 358.9 1400.0 3177.7  428.5
"MA"  3.1 20.8 169.1 231.6 1532.2 2311.3 1140.1
"MI"  9.3 38.9 261.9 274.6 1522.7 3159.0  545.5
"MN"  2.7 19.5  85.9  85.8 1134.7 2559.3  343.1
"MS" 14.3 19.6  65.7 189.1  915.6 1239.9  144.4
"MO"  9.6 28.3 189.0 233.5 1318.3 2424.2  378.4
"MT"  5.4 16.7  39.2 156.8  804.9 2773.2  309.2
"NE"  3.9 18.1  64.7 112.7  760.0 2316.1  249.1
"NV" 15.8 49.1 323.1 355.0 2453.1 4212.6  559.2
"NH"  3.2 10.7  23.2  76.0 1041.7 2343.9  293.4
"NJ"  5.6 21.0 180.4 185.1 1435.8 2774.5  511.5
"NM"  8.8 39.1 109.6 343.4 1418.7 3008.6  259.5
"NY" 10.7 29.4 472.6 319.1 1728.0 2782.0  745.8
"NC" 10.6 17.0  61.3 318.3 1154.1 2037.8  192.1
"ND"  0.9  9.0  13.3  43.8  446.1 1843.0  144.7
"OH"  7.8 27.3 190.5 181.1 1216.0 2696.8  400.4
"OK"  8.6 29.2  73.8 205.0 1288.2 2228.1  326.8
"OR"  4.9 39.9 124.1 286.9 1636.4 3506.1  388.9
"PA"  5.6 19.0 130.3 128.0  877.5 1624.1  333.2
"RI"  3.6 10.5  86.5 201.0 1489.5 2844.1  791.4
"SC" 11.9 33.0 105.9 485.3 1613.6 2342.4  245.1
"SD"  2.0 13.5  17.9 155.7  570.5 1704.4  147.5
"TN" 10.1 29.7 145.8 203.9 1259.7 1776.5  314.0
"TX" 13.3 33.8 152.4 208.2 1603.1 2988.7  397.6
"UT"  3.5 20.3  68.8 147.3 1171.6 3004.6  334.5
"VT"  1.4 15.9  30.8 101.2 1348.2 2201.0  265.2
"VI"  9.0 23.3  92.1 165.7  986.2 2521.2  226.7
"WA"  4.3 39.6 106.2 224.8 1605.6 3386.9  360.3
"WV"  6.0 13.2  42.2  90.9  597.4 1341.7  163.3
"WI"  2.8 12.9  52.2  63.7  846.9 2614.2  220.7
"WY"  5.4 21.9  39.7 173.9  811.6 2772.2  282.0' %>%
textConnection() %>% read.table(header=TRUE) -> crime

crime
```

Exploratory data analysis

```
summary(crime)
ggpairs(crime)
```

PCA

Fitting

Fitting PCA is straightforward:

```
(crime.pc <- prcomp(crime, scale=TRUE))
```

Selecting the number of principal components

Similarly, we can quickly obtain the variances explained plot to begin the process of selecting the appropriate number of PCs to use:

```
summary(crime.pc)
screeplot(crime.pc)
```

The following produces a convenient visualisation as well, juxtaposing individual and cumulative variance proportions and indicating cutoffs for the 90% rule and the Kaiser's rule:

```
(pcvars <- crime.pc$sdev^2) # Eigenvalues
(var.explained <- cumsum(pcvars)/sum(pcvars))

plot(seq_along(pcvars), pcvars/sum(pcvars), type="o", ylim=c(0,1), xlab="k",
ylab="Proportion of variance explained")
points(seq_along(pcvars), var.explained, lty=2, type="o")
legend("right", lty=c(1,2), legend=c("Individual","Cumulative"))
abline(h=c(1/ncol(crime), 0.9), lty=3)
```

We can also implement these rules as follows:

```
# c=90% of variance explained
min(which(var.explained>=0.9))
# Kaiser's rule:
max(which(pcvars>=1))
```

Kaiser's rule is more conservative in this case.

Interpreting

Lastly, consider the interpretations of the first two components. A biplot plots the significance of each variable for the first two components; and the location of the data points with respect to those components.

```
biplot(crime.pc)
```

Note that principle components are invariant to reflection (negation), and so the highest-crime states

may well end up with the lowest values for their PC1.

PC1 clearly incorporates the overall propensity of a state for crime: all variables "point" in the same direction. On PC2, observe that the components are arranged on a spectrum from violent crimes and crimes against person (starting with murder) to crimes against property (culminating in crimes in which the perpetrator and the victim do not interact).

Some observations:

- As the saying goes, "What happens in Vegas, stays in Vegas"---but it counts towards Nevada's crime statistics nonetheless.
- States with the strongest crimes-against-person values of PC2 tend to be in the American Southeast, the "Deep South".
- Conversely, states with the strongest crimes-against-property values of PC2 tend to be in the Northeast: the "New England".

It may also be helpful to sort the data by component to see which states have the highest/lowest value for overall crime and for the crime type.

```
## First two components for each state:
## Note that PCs are invariant up to negation, so different implementations will put
different observations first and last.
# Ordered by PC1:
round(cbind(crime.pc$x[,1:2], crime)[order(crime.pc$x[,1]),],2)
# Ordered by PC2:
round(cbind(crime.pc$x[,1:2], crime)[order(crime.pc$x[,2]),],2)
```

We can also make biplots of other components. For example, here are the pairwise biplots of the first 3:

```
biplot(crime.pc, c(1,2))
biplot(crime.pc, c(1,3))
biplot(crime.pc, c(2,3))
```

It is not clear what PC3 represents, however.

Challenge: PCA

If you choose to complete this task in your own RStudio, upload the following file:

PCA_Examples.challenge.Rmd

Click on the 'PCA_Examples.challenge.Rmd' in the 'Files' section to begin. Enter your response to the tasks in the 'Enter your code here' section.

This activity and the solution will be discussed at the Collaborate session this week. In the meantime, share and discuss your results in the 'Tutorials' discussion forum.

The solution will also be available here on Friday of this week by clicking on the 'Solution' tab in the top right corner.

The contents of the RMD file are also displayed below:

Packages

```
library(here)
library(GGally)
library(dplyr)
library(readr)
library(purrr)
```

Challenge

Consider the dataset `pizza.csv`, containing nutritional data from a variety of pizza brands:

`brand` : Pizza brand (class label)

`id` : Sample analysed

`prot` : Amount of protein per 100 grams in the sample

`fat` : Amount of fat per 100 grams in the sample

`ash` : Amount of ash per 100 grams in the sample

`sodium` : Amount of sodium per 100 grams in the sample

`carb` : Amount of carbohydrates per 100 grams in the sample

`cal` : Amount of calories per 100 grams in the sample

We will focus on brand A:

```
pizzaA <- read_csv(here("datasets","pizza.csv")) %>% filter(brand=="A") %>% select(-brand)
```

Task 1: Perform the PCA on these data, and identify a good number of principal components to use. Which variables covary, and how?

Check your understanding

Complete the below exercises to check your understanding of concepts presented so far.

1. A random vector $\boldsymbol{Y} = \begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \end{pmatrix}$ is normally distributed with zero mean vector $\Sigma = \begin{pmatrix} 1 & \rho/2 & 0 \\ \rho/2 & 1 & \rho \\ 0 & \rho & 1 \end{pmatrix}$ where ρ is positive.

a) Find the coefficients of the first principal component and the variance of that component. What percentage of the overall variability does it explain?

b) Find the joint distribution of Y_1, Y_2 and Y_1, Y_2, Y_3

c) Find the conditional distribution of Y_1, Y_2 given $Y_3 = y_3$.

d) Find the multiple correlation of Y_3 with Y_1, Y_2.

Topic 1: Canonical correlations

Welcome to Week 3

Dr Pavel Krivitsky gives you a brief overview of topics and concepts we'll be covering in this week.

Transcript

Weekly learning outcomes

- Provide the mathematical definition of a canonical correlation.
- Evaluate and interpret the canonical correlations between two datasets.
- Test for the significance of a canonical correlation.
- Determine the optimal number of canonical correlations.
- Perform the regular linear model inference on multivariate linear models.
- Explain the assumptions underlying the above inferential procedures and check them.

Topics we will cover are:

- Topic 1: Canonical correlations
- Topic 2: Linear models with multivariate response

- Topic 3: Multivariate ANOVA: testing canonical correlations and multivariate linear models

Optional reading

Härdle W.K., Simar L. (2012) Regression Models. In: Applied Multivariate Statistical Analysis. Springer, Berlin, Heidelberg

- Section 8.1.2

Questions about this week's topics?

This week's topics were prepared by Dr P. Krivitsky. If you have any questions or comments, please post them under Discussion or email directly: p.krivitsky@unsw.edu.au

Canonical correlation analysis

Introduction

Assume we are interested in the association between two **sets** of random variables. Typical examples include: relation between a set of governmental policy variables and a set of economic goal variables; relation between college "performance" variables (like grades in courses in five different subject matter areas) and pre-college "achievement" variables (like high-school grade-point averages for junior and senior years, number of high-school extracurricular activities) etc.

The way the above problem of measuring association is solved in Canonical Correlation Analysis, is to consider the largest possible correlation between a *linear combination of the variables in the first set* and a *linear combination of the variables in the second set*. The pair of linear combinations obtained through this maximisation process is called **first canonical variables** and their correlation is called **first canonical correlation**. The process can be continued (similarly to the principal components procedure) to find a second pair of linear combinations having the largest correlation among all pairs that are uncorrelated with the initially selected pair. This would give us the second set of canonical variables with their second canonical correlation etc. The maximisation process that we are performing at each step reflects our wish (again like in principal components analysis) to concentrate the initially high dimensional relationship between the 2 sets of variables into a few pairs of canonical variables only. Often, even only **one** pair is considered. The rationale in canonical correlation analysis is that when the number of variables is large, interpreting the **whole set** of correlation coefficients between pairs of variables from each set is hopeless and in that case, one should concentrate on a *few* carefully chosen representative correlations. Finally, we should note that the traditional (simple) correlation coefficient and the multiple correlation coefficient (Topic 2 of Week 2) are *special cases* of canonical correlation in which one or both sets contain a single variable.

Application in testing for independence of sets of variables

Besides being interesting in its own right (See Introduction section above), calculating canonical correlations turns out to be important for the sake of **testing independence of sets of random variables**. Let us remember that testing for independence and for uncorrelatedness in the case of multivariate normal are equivalent problems. Assume now that that $\boldsymbol{X} \sim N_p(\boldsymbol{\mu}, \Sigma)$. Furthermore, let $\boldsymbol{X}$ be partitioned into r, q components ($r + q = p$) with $\boldsymbol{X}^{(1)} \in \mathbb{R}^r$, $\boldsymbol{X}^{(2)} \in \mathbb{R}^q$ and correspondingly, the covariance matrix

$$\Sigma = \mathrm{E}(\boldsymbol{X} - \boldsymbol{\mu})(\boldsymbol{X} - \boldsymbol{\mu})^\top = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1p} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{p1} & \sigma_{p2} & \cdots & \sigma_{pp} \end{pmatrix} \in \mathcal{M}_{p,p}$$

has been also partitioned into $\Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}$, accordingly. We shall assume for simplicity that the matrices Σ, Σ_{11}, and Σ_{22} are nonsingular. To test $H_0 : \Sigma_{12} = \mathbf{0}$ against a general alternative, a sensible way to go would be the following: for fixed vectors $\boldsymbol{a} \in \mathbb{R}^r, \boldsymbol{b} \in \mathbb{R}^q$ let $Z_1 = \boldsymbol{a}^\top \boldsymbol{X}^{(1)}$ and $Z_2 = \boldsymbol{b}^\top \boldsymbol{X}^{(2)}$ giving $\rho_{\boldsymbol{a},\boldsymbol{b}} = \mathrm{Cor}(Z_1, Z_2) = \frac{\boldsymbol{a}^\top \Sigma_{12} \boldsymbol{b}}{\sqrt{\boldsymbol{a}^\top \Sigma_{11} \boldsymbol{a} \boldsymbol{b}^\top \Sigma_{22} \boldsymbol{b}}}$. H_0 is equivalent to $H_0 : \rho_{\boldsymbol{a},\boldsymbol{b}} = 0$ for all $\boldsymbol{a} \in \mathbb{R}^r, \boldsymbol{b} \in \mathbb{R}^q$. For a particular pair $\boldsymbol{a}, \boldsymbol{b}$, H_0 would be accepted if $|r_{\boldsymbol{a},\boldsymbol{b}}| = \frac{|\boldsymbol{a}^\top \boldsymbol{S}_{12} \boldsymbol{b}|}{\sqrt{\boldsymbol{a}^\top \boldsymbol{S}_{11} \boldsymbol{a} \boldsymbol{b}^\top \boldsymbol{S}_{22} \boldsymbol{b}}} \leq k$ for certain positive constant k. (Here $\boldsymbol{S}_{ij}$ are the corresponding data based estimators of Σ_{ij}.) Hence an appropriate acceptance region for H_0 would be given in the form $\left\{ \boldsymbol{X} \in \mathcal{M}_{p,n} : \max_{\boldsymbol{a},\boldsymbol{b}} r^2_{\boldsymbol{a},\boldsymbol{b}} \leq k^2 \right\}$. But maximising $r^2_{\boldsymbol{a},\boldsymbol{b}}$ means to find the maximum of $\left(a^\top S_{12} b\right)^2$ under constraints $\boldsymbol{a}^\top \boldsymbol{S}_{11} \boldsymbol{a} = 1$ and $\boldsymbol{b}^\top \boldsymbol{S}_{22} \boldsymbol{b} = 1$, and this is exactly the data-based version of the optimisation problem to be solved in the above introduction.

For the goals in the above to be achieved, we need to solve the type of problems presented in the next slide.

Precise mathematical formulation and solution to the problem

Canonical variables are the variables $Z_1 = \boldsymbol{a}^\top \boldsymbol{X}^{(1)}$ and $Z_2 = \boldsymbol{b}^\top \boldsymbol{X}^{(2)}$ where $\boldsymbol{a} \in \mathbb{R}^r, \boldsymbol{b} \in \mathbb{R}^q$ are obtained by maximising $\left(\boldsymbol{a}^\top \Sigma_{12} \boldsymbol{b}\right)^2$ under the constraints $\boldsymbol{a}^\top \Sigma_{11} \boldsymbol{a} = \boldsymbol{b}^\top \Sigma_{22} \boldsymbol{b} = 1$. To solve the above maximisation problem, we construct

$$\operatorname{Lag}\left(\boldsymbol{a}, \boldsymbol{b}, \lambda_1, \lambda_2\right) = \left(\boldsymbol{a}^\top \Sigma_{12} \boldsymbol{b}\right)^2 + \lambda_1 \left(\boldsymbol{a}^\top \Sigma_{11} \boldsymbol{a} - 1\right) + \lambda_2 \left(\boldsymbol{b}^\top \Sigma_{22} \boldsymbol{b} - 1\right)$$

Partial differentiation with respect to the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ gives:

$$2 \left(\boldsymbol{a}^\top \Sigma_{12} \boldsymbol{b}\right) \Sigma_{12} \boldsymbol{b} + 2 \lambda_1 \Sigma_{11} \boldsymbol{a} = \boldsymbol{0} \in \mathbb{R}^r \tag{3.1}$$

$$2 \left(\boldsymbol{a}^\top \Sigma_{12} \boldsymbol{b}\right) \Sigma_{21} \boldsymbol{a} + 2 \lambda_2 \Sigma_{22} \boldsymbol{b} = \boldsymbol{0} \in \mathbb{R}^q \tag{3.2}$$

We multiply (3.1) by the vector $\boldsymbol{a}^\top$ from left and equation (3.2) by $\boldsymbol{b}^\top$ from left and after subtracting the two equations obtained we get $\lambda_1 = \lambda_2 = -\left(\boldsymbol{a}^\top \Sigma_{12} \boldsymbol{b}\right)^2 = -\mu^2$. Hence:

$$\Sigma_{12} \boldsymbol{b} = \mu \Sigma_{11} \boldsymbol{a} \tag{3.3}$$

and

$$\Sigma_{21} \boldsymbol{a} = \mu \Sigma_{22} \boldsymbol{b} \tag{3.4}$$

Now we first multiply (3.3) by $\Sigma_{21} \Sigma_{11}^{-1}$ from left, then both sides of (3.4) by the scalar μ and after finally adding the two equations we get:

$$\left(\Sigma_{21} \Sigma_{11}^{-1} \Sigma_{12} - \mu^2 \Sigma_{22}\right) \boldsymbol{b} = \boldsymbol{0} \tag{3.5}$$

The homogeneous equation system (3.5) having a non-trivial solution w.r.t. $\boldsymbol{b}$ means that

$$\left|\Sigma_{21} \Sigma_{11}^{-1} \Sigma_{12} - \mu^2 \Sigma_{22}\right| = 0 \tag{3.6}$$

must hold. Then, of course,

$$\left|\Sigma_{22}^{-\frac{1}{2}}\right| \left|\Sigma_{21} \Sigma_{11}^{-1} \Sigma_{12} - \mu^2 \Sigma_{22}\right| \left|\Sigma_{22}^{-\frac{1}{2}}\right| = \left|\Sigma_{22}^{-\frac{1}{2}} \Sigma_{21} \Sigma_{11}^{-1} \Sigma_{12} \Sigma_{22}^{-\frac{1}{2}} - \mu^2 I_q\right| = 0$$

must hold. This means that μ^2 has to be an eigenvalue of the matrix $\Sigma_{22}^{-\frac{1}{2}} \Sigma_{21} \Sigma_{11}^{-1} \Sigma_{12} \Sigma_{22}^{-\frac{1}{2}}$. Also, $\boldsymbol{b} = \Sigma_{22}^{-\frac{1}{2}} \hat{\boldsymbol{b}}$ where $\hat{\boldsymbol{b}}$ is the eigenvector of $\Sigma_{22}^{-\frac{1}{2}} \Sigma_{21} \Sigma_{11}^{-1} \Sigma_{12} \Sigma_{22}^{-\frac{1}{2}}$ corresponding to this eigenvalue.

(Note, however, that this representation is good mainly for theoretical purposes, the main advantage being that one is dealing with eigenvalues of a symmetric matrix. If doing calculations by hand, it is usually easier to calculate $\boldsymbol{b}$ directly as the solution of the linear equation (3.5), i.e., find the largest eigenvalue of the (non-symmetric) matrix $\Sigma_{22}^{-1} \Sigma_{21} \Sigma_{11}^{-1} \Sigma_{12}$ and then find the eigenvector $\boldsymbol{b}$ that

corresponds to it. Besides, we also see from the definition of μ that $\mu^2 = \left(\boldsymbol{a}^\top \Sigma_{12} \boldsymbol{b}\right)^2$ holds.)

Since we wanted to **maximise** the right hand side, it is obvious that μ^2 must be chosen to be the **largest eigenvalue** of the matrix $\Sigma_{22}^{-\frac{1}{2}} \Sigma_{21} \Sigma_{11}^{-1} \Sigma_{12} \Sigma_{22}^{-\frac{1}{2}}$ (or, which is the same thing, the largest eigenvalue of the matrix $\Sigma_{21} \Sigma_{11}^{-1} \Sigma_{12} \Sigma_{22}^{-1}$). Finally, we can obtain the vector $\boldsymbol{a}$ from (3.3): $\boldsymbol{a} = \frac{1}{\mu} \Sigma_{11}^{-1} \Sigma_{12} \boldsymbol{b}$. That way, the **first** canonical variables $Z_1 = \boldsymbol{a}^\top \boldsymbol{X}^{(1)}$ and $Z_2 = \boldsymbol{b}^\top \boldsymbol{X}^{(2)}$ are determined and the value of the first canonical correlation is just μ. The orientation of the vector $\boldsymbol{b}$ is chosen such that the sign of μ should be positive.

Now, it is easy to see that if we want to extract a second pair of canonical variables we need to repeat the same process by starting with the **second largest** eigenvalue μ^2 of the matrix $\Sigma_{22}^{-\frac{1}{2}} \Sigma_{21} \Sigma_{11}^{-1} \Sigma_{12} \Sigma_{22}^{-\frac{1}{2}}$ (or of the matrix $\Sigma_{22}^{-1} \Sigma_{21} \Sigma_{11}^{-1} \Sigma_{12}$). This will automatically ensure that the second pair of canonical variables is uncorrelated with the first pair. The process can theoretically be continued until the number of pairs of canonical variables equals the number of variables in the smaller group. But in practice, much fewer canonical variables will be needed. Each canonical variable is uncorrelated with all the other canonical variables of either set except for the one corresponding canonical variable in the opposite set.

Note. It is important to point out that already by definition the canonical correlation is at least as large as the multiple correlation between any variable and the opposite set of variables. It is in fact possible for the first canonical correlation to be *very large* while all the multiple correlations of each separate variable with the opposite set of canonical variables are small. This once again underlines the importance of Canonical Correlation analysis.

Watch the below example by Dr Pavel Krivitsky.

Estimating and testing canonical correlations

The way to estimate the canonical variables and canonical correlation coefficients is based on the plug-in technique: one follows the steps outlined in the previous section, by each time substituting $\boldsymbol{S}_{ij}$ in place of Σ_{ij}:

1. From the sample variance--covariance matrix $\boldsymbol{S}$, calculate the $\boldsymbol{A} = \boldsymbol{S}_{22}^{-\frac{1}{2}} \boldsymbol{S}_{21} \boldsymbol{S}_{11}^{-1} \boldsymbol{S}_{12} \boldsymbol{S}_{22}^{-\frac{1}{2}}$.
2. Calculate μ^2 as the greatest eigenvalue of $\boldsymbol{A}$ and $\hat{\boldsymbol{b}}$ as the corresponding eigenvector.
3. Calculate $\boldsymbol{b} = \boldsymbol{S}_{22}^{-\frac{1}{2}} \hat{\boldsymbol{b}}$ and $\boldsymbol{a} = \frac{1}{\mu} \boldsymbol{S}_{11}^{-1} \boldsymbol{S}_{12} \boldsymbol{b}$.

Then, $\mu = \sqrt{\mu^2}$ is the first canonical correlation, and $\boldsymbol{a}$ and $\boldsymbol{b}$ represent the linear combinations of the data vectors that produce it.

Let us now discuss the independence testing issue outlined earlier. The acceptance region of the independence test of H_0 would be $\{\boldsymbol{X} \in \mathcal{M}_{p,n} : \text{ largest eigenvalue of } A \leq k_\alpha\}$ where k_α has been worked out and is given in the so called **Hecks charts**. This distribution depends on three parameters: $s = \min(r, q)$, $m = \frac{|r-q|-1}{2}$, and $N = \frac{n-r-q-2}{2}$, n being the sample size. Besides using the charts, one can also use good F-distribution-based approximations for a (transformations of) this distribution like Wilk's lambda, Pillai's trace, Hotelling trace, and Roy's greatest root. *Permutation tests* that are not as sensitive to these assumptions are also possible.

In software

Here we shall only mention that all these statistics and their P-values (using suitable F-distribution-based approximations) are readily available as an output in R: see `stats::cancor` and package `CCA` for computing and visualisation, and package `CCP` for testing canonical correlations.

Some important computational issues

Note that calculating $X^{-\frac{1}{2}}$ and $X^{\frac{1}{2}}$ for a symmetric positive definite matrix X according to the theoretically attractive spectral decomposition method may be numerically unstable. This is especially the case when some of the eigenvalues are close to zero (or, more precisely, when the ratio of the greatest eigenvalue and the least eigenvalue —the *condition number*— is high).

We can use the **Cholesky decomposition** described in slide "Numerical stability and Cholesky decomposition". Looking back at (3.5), we see that if $U^\top U = \Sigma_{22}^{-1}$ gives the Cholesky decomposition of the matrix Σ_{22}^{-1} then μ^2 is an eigenvalue of the matrix $A = U\Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}U^\top$. Indeed, by multiplying from left by U and from right by $U^\top$ in (3.6) we get:

$$\left|A - \mu^2 U\Sigma_{22}U^\top\right| = 0$$

But $U\Sigma_{22}U^\top = U\left(U^\top U\right)^{-1}U^\top = UU^{-1}\left(U^\top\right)^{-1}U^\top = I$ holds.

Check your understanding

Complete the below exercises to check your understanding of concepts presented so far.

1. Let the components of X correspond to scores on tests in arithmetic speed (X_1), arithmetic power (X_2), memory for words (X_3), memory for meaningful symbols (X_4), and memory for meaningless symbols (X_5). The observed correlations in a sample of 140 are

$$\begin{bmatrix} 1.0000 & 0.4248 & 0.0420 & 0.0215 & 0.0573 \\ & 1.0000 & 0.1487 & 0.2489 & 0.2843 \\ & & 1.0000 & 0.6693 & 0.4662 \\ & & & 1.0000 & 0.6915 \\ & & & & 1.0000 \end{bmatrix}$$

Find the canonical correlations and canonical variates between the first two variates and the last three variates. Comment. Write a SAS-IML or R code to implement the required calculations.

2. Students sit 5 different papers, two of which are closed book and the rest open book. For the 88 students who sat these exams the sample covariance matrix is

$$S = \begin{bmatrix} 302.3 & 125.8 & 100.4 & 105.1 & 116.1 \\ & 170.9 & 84.2 & 93.6 & 97.9 \\ & & 111.6 & 110.8 & 120.5 \\ & & & 217.9 & 153.8 \\ & & & & 294.4 \end{bmatrix}$$

Find the canonical correlations and canonical variates between the first two variates (closed book exams) and the last three variates (open book exams). Comment.

3. Consider a random vector $\boldsymbol{X} \sim N_4(\boldsymbol{\mu}, \Sigma)$ with $\mu = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$ and $\begin{pmatrix} 1 & 2\rho & \rho & \rho \\ 2\rho & 1 & \rho & \rho \\ \rho & \rho & 1 & 2\rho \\ \rho & \rho & 2\rho & 1 \end{pmatrix}$

where ρ is a small enough positive constant.

a) Find the two canonical correlations between $\begin{pmatrix} X_1 \\ X_2 \end{pmatrix}$ and $\begin{pmatrix} X_3 \\ X_4 \end{pmatrix}$

Comment.

b) Find the first pair of canonical variables.

4. Consider the following covariance matrix Σ of a four dimensional normal vector:

$\Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix} = \left(\begin{array}{cc|cc} 100 & 0 & 0 & 0 \\ 0 & 1 & 0.95 & 0 \\ \hline 0 & 0.95 & 1 & 0 \\ 0 & 0 & 0 & 100 \end{array} \right)$. Verify that the first pair of canonical variates are just the second and the third component of the vector and the canonical correlation equals 0.95.

Demonstration: Canonical correlation examples

This demonstration can be completed using the provided RStudio or your own RStudio.

To complete this task select the 'Cancor_Examples.demo.Rmd' in the 'Files' section of RStudio. Follow the demonstration contained within the RMD file.

If you choose to complete the example in your own RStudio, upload the following file:

Cancor_Examples.demo.Rmd

The contents of the RMD file are also displayed below:

Packages

```
library(here)
library(GGally)
library(dplyr)
library(readr)
library(purrr)
library(CCA)
library(CCP)
```

Fitness club example

Three physiological and three exercise variables were measured on twenty middle aged men in a fitness club. Canonical correlation is used to determine if the physiological variables are related in any way to the exercise variables.

Data

```
'weight waist pulse chins situps jumps
   191  36  50   5  162   60
   189  37  52   2  110   60
   193  38  58  12  101  101
   162  35  62  12  105   37
   189  35  46  13  155   58
   182  36  56   4  101   42
   211  38  56   8  101   38
   167  34  60   6  125   40
   176  31  74  15  200   40
   154  33  56  17  251  250
```

```
   169  34  50  17  120   38
   166  33  52  13  210  115
   154  34  64  14  215  105
   247  46  50   1   50   50
   193  36  46   6   70   31
   202  37  62  12  210  120
   176  37  54   4   60   25
   157  32  52  11  230   80
   156  33  54  15  225   73
   138  33  68   2  110   43' %>%
textConnection() %>% read.table(header=TRUE) -> fitness

fitness
```

Exploratory data analysis

```
summary(fitness)
ggpairs(fitness)
```

The data do have some outliers.

Canonical Correlation Analysis

Estimation and visualisation

Fitting canonical correlations is straightforward:

```
X <- fitness[1:3]
Y <- fitness[4:6]
print(fitness.cc <- cc(X,Y))
```

What do we see? Firstly,

`cor` : Canonical correlations for the first, second, and third pair of canonical vectors.

`xcoef`, `ycoef` : Canonical coefficients for the first and second set of variables. (Variables are in rows, and vectors are in columns.)

`xscores`, `yscores` : Canonical scores for each observation, produced by taking the inner product of each observation vector and a canonical coefficient vector. These are the scores whose correlations we calculate.

`corr.X.xscores`, `corr.Y.xscores`, etc. : These are the correlations between the canonical scores and the original variables. Note that these may or may not be the same as the coefficients: in particular, if there is correlation within X or within Y, some of the variables may be "redundant".

We can also produce plots. It helps to set `var.label=TRUE` to label which variables have which canonical correlation weights:

```
plt.cc(fitness.cc, var.label=TRUE)
```

Here, the left-hand-side plot plots the `corr.X.xscores` for the first two canonical vectors for each of the variables and similarly for the `corr.Y.xscores`. Then, the horizontal position of a variable reflects its relationship with the first canonical vector and its vertical position the second. In this case, we might conclude that in the most important dimension of correlation waist and weight, as they increase, are negatively correlated with chins and situps.

Standardised canonical correlation coefficients

To compare coefficients between variables *in the same group*, it can be helpful to standardise the variables first. This can be done by the `scale` function. Note that hypothesis testing should probably be done on the unstandardised data.

```
Xs <- scale(X)
Ys <- scale(Y)
print(fitness.ccs <- cc(Xs,Ys))
```

Hypothesis testing (asymptotic)

Hypothesis testing only requires the correlations, sample sizes, and variable counts. A number of tests are possible:

```
n <- nrow(fitness)
p <- ncol(X)
q <- ncol(Y)
p.asym(fitness.cc$cor, n, p, q, tstat = "Wilks")
p.asym(fitness.cc$cor, n, p, q, tstat = "Hotelling")
p.asym(fitness.cc$cor, n, p, q, tstat = "Pillai")
p.asym(fitness.cc$cor, n, p, q, tstat = "Roy")
```

Here, the rows "i to j" should be read as "$H_0 : \rho_i = \cdots = \rho_j = 0$, where ρ_k is the kth canonical correlation."

We thus see that the first canonical correlation does not appear to meet significance at the conventional $\alpha = 0.05$ for most of the tests. The fact that the data deviates from normality and the relatively small sample size probably explain the differences between the different tests.

Where available, the first test tests for significance of the first canonical correlation (i.e., any linear dependence between datasets), the second for the significance of the second (i.e., there are linear relationships between the datasets beyond the first), etc.

Permutation testing

Permutation tests can only test for the first canonical correlation, but they are valid even if the data are not even close to MVN:

```
p.perm(X,Y, type = "Wilks")
p.perm(X,Y, type = "Hotelling")
p.perm(X,Y, type = "Pillai")
p.perm(X,Y, type = "Roy")
```

When only correlations are available: Chicken data

Based on Example 10.4, p. 552 in Johnston and Wichern, this is a study of canonical correlations between leg and head bone measurements: X_1, X_2 are skull length and skull breadth, respectively; X_3, X_4 are leg bone measurements: femur and tibia length, respectively. Observations have been taken on $n = 276$ White Leghorn chicken. The example is chosen to also illustrate how a canonical correlation analysis can be performed when the original data are not given but the empirical correlation matrix (or empirical covariance matrix) is available.

Data

```
" 1.0      .505   .569   .602
   .505   1.0     .422   .467
   .569   .422   1.0     .926
   .602   .467   .926   1.0" %>%
   textConnection() %>% scan() %>%
   matrix(4,4, byrow=TRUE) -> chicken

rownames(chicken) <- colnames(chicken) <- c("head1","head2","leg1","leg2")

n <- 276

chicken
```

Canonical Correlation Analysis

Estimation

Here, we are using the algorithm from slide "Estimating and testing canonical correlations".

```
matpow <- function(A, p){
with(eigen(A), vectors%*%diag(values^p,nrow(A))%*%t(vectors))
}
(ccdecomp <-
eigen(matpow(chicken[3:4,3:4],-1/2)%*%chicken[3:4,1:2]%*%solve(chicken[1:2,1:2])%*%chi
cken[1:2,3:4]%*%matpow(chicken[3:4,3:4],-1/2)))
```

```
(ccors <- sqrt(ccdecomp$values))
(b <- matpow(chicken[3:4,3:4],-1/2) %*% ccdecomp$vectors[,1])
(a <- 1/sqrt(ccdecomp$values[1])*solve(chicken[1:2,1:2])%*%chicken[1:2,3:4]%*%b)
```

Hypothesis testing (asymptotic)

```
library(CCP)
p <- 2
q <- 2
p.asym(ccors, n, p, q, tstat = "Wilks")
p.asym(ccors, n, p, q, tstat = "Hotelling")
p.asym(ccors, n, p, q, tstat = "Pillai")
p.asym(ccors, n, p, q, tstat = "Roy")
```

Challenge: Canonical correlation examples

If you choose to complete this task in your own RStudio, upload the following file:

Cancor_Examples.challenge.Rmd

Click on the 'Cancor_Examples.challenge.Rmd' in the 'Files' section to begin. Enter your response to the tasks in the 'Enter your code here' section.

This activity and the solution will be discussed at the Collaborate session this week. In the meantime, share and discuss your results in the 'Tutorials' discussion forum.

The solution will also be available here on Friday of Week 1 by clicking on the 'Solution' tab.

The contents of the RMD file are also displayed below:

Recall the dataset `pizza.csv`, containing nutritional data from a variety of pizza brands:

`brand` : Pizza brand (class label)

`id` : Sample analysed

`prot` : Amount of protein per 100 grams in the sample

`fat` : Amount of fat per 100 grams in the sample

`ash` : Amount of ash per 100 grams in the sample

`sodium` : Amount of sodium per 100 grams in the sample

`carb` : Amount of carbohydrates per 100 grams in the sample

`cal` : Amount of calories per 100 grams in the sample

We will focus on brand A:

```
pizzaA <- read_csv(here("datasets","pizza.csv")) %>% filter(brand=="A") %>% select(-brand)
```

Task 1: Calculate the canonical correlations between caloric, ash, and sodium content (set 1) and fat, protein, and carbohydrate content (set 2). What do the vectors represent?

Task 2: Test the significance of correlation between the two sets; comment on the results. How many canonical correlations are significant, and what do they represent?

Topic 2: Linear models with multivariate response

Univariate linear models and ANOVA

Recall the univariate linear model: for observations $i = 1, 2, \ldots, n$, let the response variable $Y_i = \boldsymbol{x}_i\boldsymbol{\beta} + \epsilon_i$, for predictor row vector $\boldsymbol{x}_i^\top \in \mathbb{R}^k$ assumed fixed and known, coefficient vector $\boldsymbol{\beta} \in \mathbb{R}^p$ fixed and unknown, and $\epsilon_i \overset{\text{i.i.d.}}{\sim} N(0, \sigma^2)$. In matrix form, $\boldsymbol{Y} = \begin{pmatrix} Y_1 & Y_2 & \cdots & Y_n \end{pmatrix}^\top$ and $X = \begin{pmatrix} x_1^\top & x_2^\top & \cdots & x_n^\top \end{pmatrix}^\top \in \mathcal{M}_{n,k}$. We will assume that X contains an intercept. Then,

$$\boldsymbol{Y} = X\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

where $\boldsymbol{\epsilon} \sim N_n(\boldsymbol{0}, I_n\sigma^2)$. The MLE for $\boldsymbol{\beta}$ requires us to minimise

$$\sum_{i=1}^{n} (Y_i - \boldsymbol{x}_i\boldsymbol{\beta})^2 = |\boldsymbol{Y} - X\boldsymbol{\beta}|^2 = (\boldsymbol{Y} - \boldsymbol{X\beta})^\top(\boldsymbol{Y} - \boldsymbol{X\beta})$$

and, after some vector calculus, we get

$$\hat{\boldsymbol{\beta}} = \left(X^\top X\right)^{-1} X^\top \boldsymbol{Y}$$

with

$$\operatorname{Var}(\hat{\boldsymbol{\beta}}) = \left(X^\top X\right)^{-1} X^\top \operatorname{Var}(\boldsymbol{Y}) X \left(X^\top X\right)^{-1} = \left(X^\top X\right)^{-1} \sigma^2$$

Furthermore, we can consider projection matrices $A = I_n - X\left(X^\top X\right)^{-1} X^\top$ and $B = X\left(X^\top X\right)^{-1} X^\top - \boldsymbol{1}_n\left(\boldsymbol{1}_n^\top \boldsymbol{1}_n\right)^{-1} \boldsymbol{1}_n^\top$ with

$$A\boldsymbol{Y} = \boldsymbol{Y} - X\left\{\left(X^\top X\right)^{-1} X^\top \boldsymbol{Y}\right\} = \boldsymbol{Y} - \hat{\boldsymbol{Y}}$$

the residual vector and

$$B\boldsymbol{Y} = X\left\{(X^\top X)^{-1}X^\top\right\}\boldsymbol{Y} - \boldsymbol{1}_n(\boldsymbol{1}_n^\top\boldsymbol{1}_n)^{-1}\boldsymbol{1}_n^\top\boldsymbol{Y} = \hat{\boldsymbol{Y}} - \boldsymbol{1}_n\bar{\boldsymbol{Y}}$$

the vector of fitted values over and above the mean, and observe that

$$\begin{aligned}
\operatorname{Cov}(A\boldsymbol{Y}, B\boldsymbol{Y}) =& A\operatorname{Var}(\boldsymbol{Y})B^\top = \sigma^2 AB^\top \\
=& X\left(X^\top X\right)^{-1} X^\top - X\left(X^\top X\right)^{-1} X^\top X \left(X^\top X\right)^{-1} X^\top \\
& - \boldsymbol{1}_n\left(\boldsymbol{1}_n^\top\boldsymbol{1}_n\right)^{-1}\boldsymbol{1}_n^\top + X\left(X^\top X\right)^{-1} X^\top \boldsymbol{1}_n\left(\boldsymbol{1}_n^\top\boldsymbol{1}_n\right)^{-1}\boldsymbol{1}_n^\top \\
=& \frac{1}{n}\left(X\left(X^\top X\right)^{-1} X^\top \boldsymbol{1}_n - \boldsymbol{1}_n\right)\boldsymbol{1}_n^\top = \boldsymbol{0}
\end{aligned}$$

if X contains an intercept effect. Then, $\mathrm{SSE} = \boldsymbol{Y}^\top A \boldsymbol{Y} \sim \sigma^2 \chi^2_{n-k}$ and $\mathrm{SSA} = \boldsymbol{Y}^\top B \boldsymbol{Y} \sim \sigma^2 \chi^2_{k-1}$, independent, letting us set up $F = \frac{\mathrm{SSA}/(k-1)}{\mathrm{SSE}/(n-k)} \sim F_{k-1,n-k}$, etc.

Optional reading

For an alternate presentation of these concepts, read the following:

Härdle W.K., Simar L. (2012) Regression Models. In: Applied Multivariate Statistical Analysis. Springer, Berlin, Heidelberg

- Section 8.1.2

All readings are available from the course Leganto reading list. Please keep in mind that you will need to be logged into Moodle to access the Leganto reading list.

Multivariate Linear Model and MANOVA

How do we generalise the previous slide's ideas to a multivariate response? That is, suppose that we observe the following response matrix:

$$\boldsymbol{Y} = \begin{pmatrix} \boldsymbol{Y}_1^\top \\ \boldsymbol{Y}_2^\top \\ \vdots \\ \boldsymbol{Y}_n^\top \end{pmatrix} = \begin{pmatrix} Y_{11} & Y_{12} & \cdots & Y_{1p} \\ Y_{21} & Y_{22} & \cdots & Y_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{n1} & Y_{n2} & \cdots & Y_{np} \end{pmatrix} \in \mathcal{M}_{n,p}$$

with $\boldsymbol{x}_i$ and X as before, and

$$\boldsymbol{Y}_i^\top = \boldsymbol{x}_i\boldsymbol{\beta} + \boldsymbol{\epsilon}_i^\top$$

where $\boldsymbol{\beta} \in \mathcal{M}_{k,p}$, and $\boldsymbol{\epsilon}_i \sim N_p(\boldsymbol{0}, \Sigma), \Sigma \in \mathcal{M}_{p,p}$ symmetric positive definite. In matrix form,

$$\boldsymbol{Y} = X\boldsymbol{\beta} + \boldsymbol{E}$$

where

$$\boldsymbol{E} = \begin{pmatrix} \boldsymbol{\epsilon}_1 & \boldsymbol{\epsilon}_2 & \cdots & \boldsymbol{\epsilon}_n \end{pmatrix}^\top \in \mathcal{M}_{n,p}$$

Then, we can write $\vec{\boldsymbol{E}} \sim N_{np}(\boldsymbol{0}, \Sigma \otimes I_n)$ or $\overrightarrow{\boldsymbol{E}^\top} \sim N_{np}(\boldsymbol{0}, I_n \otimes \Sigma)$, and

$$\vec{\boldsymbol{Y}} \sim N_{np}\left(\left\{\boldsymbol{\beta}^\top \otimes I_n\right\}\vec{X}, \Sigma \otimes I_n\right)$$

or

$$\overrightarrow{\boldsymbol{Y}^\top} \sim N_{np}\left(\left\{I_n \otimes \boldsymbol{\beta}^\top\right\}\overrightarrow{X^\top}, I_n \otimes \Sigma\right)$$

MLE is equivalent to the OLS problem minimising $\sum_{i=1}^{n} \operatorname{tr}\left\{(\boldsymbol{Y}_i - \boldsymbol{x}_i\boldsymbol{\beta})(\boldsymbol{Y}_i - \boldsymbol{x}_i\boldsymbol{\beta})^\top\right\} = \operatorname{tr}\left\{(\boldsymbol{Y} - X\boldsymbol{\beta})^\top(\boldsymbol{Y} - X\boldsymbol{\beta})\right\}$, leading to

$$\hat{\boldsymbol{\beta}} = \left(X^\top X\right)^{-1} X^\top \boldsymbol{Y}$$

again, with

$$\begin{aligned}
\operatorname{Var}\left(\overrightarrow{\hat{\beta}^\top}\right) &= \operatorname{Var}(\overrightarrow{Y^\top X \left(X^\top X\right)^{-1}}) = \operatorname{Var}\left\{\left(\left(X^\top X\right)^{-1} X^\top \otimes I_p\right)\overrightarrow{Y^\top}\right\} \\
&= \left(\left(X^\top X\right)^{-1} X^\top \otimes I_p\right)(I_p \otimes \Sigma)\left(\left(X^\top X\right)^{-1} X^\top \otimes I_p\right)^\top \\
&= \left(\left(X^\top X\right)^{-1} X^\top \otimes I_p\right)\left(\left(X^\top X\right)^{-1} X^\top \otimes \Sigma\right)^\top \\
&= \left(X^\top X\right)^{-1} \otimes \Sigma
\end{aligned}$$

or

$$\operatorname{Var}(\overrightarrow{\boldsymbol{\beta}}) = \Sigma \otimes \left(X^\top X\right)^{-1}$$

Projection matrices A and B still work, and we can write $\mathrm{SSE} = \boldsymbol{Y}^\top A \boldsymbol{Y} \sim W_p(\Sigma, p(n-k-1))$ and $\mathrm{SSA} = \boldsymbol{Y}^\top B \boldsymbol{Y} \sim W_p(\Sigma, p(k-1))$. Notice that they are now matrices.

Topic 3: Multivariate ANOVA: testing canonical correlations and multivariate linear models

Computations used in the MANOVA tests

In standard (univariate) Analysis of Variance, with usual normality assumptions on the errors, testing about effects of the factors involved in the model description is based on the F test. The F tests are derived from the ANOVA decomposition $\mathrm{SST} = \mathrm{SSA} + \mathrm{SSE}$. The argument goes as follows:

1. SSE and SSA are independent, (up to constant factors involving the variance σ^2 of the errors) χ^2 distributed;
2. By proper norming to account for degrees of freedom, from SSE and SSA one gets statistics that have the following behaviour: the normed SSE always delivers an unbiased estimator of σ^2 no matter if the null hypothesis or alternative is true; the normed SSA delivers an unbiased estimator of σ^2 under the null hypothesis but delivers an unbiased estimator of a "larger" quantity under the alternative.

The above observation is crucial and motivates the F-testing: F statistics are (**suitably normed to account for degrees of freedom**) ratios of SSA/SSE. When taking the ratio, the factors involving σ^2 **cancel out** and σ^2 does not play any role in the distribution of the ratio. Under H_0 their distribution is F. When the null hypothesis is violated, then the same statistics will tend to have "larger" values as

compared to the case when H_0 is true. Hence significant (w.r.t. the corresponding F-distribution) values of the statistic lead to rejection of H_0.

Aiming at generalising these ideas to the Multivariate ANOVA (MANOVA) case, we should note that instead of χ^2 distributions we now have to deal with **Wishart** distributions and we need to properly define (a proper functional of) the SSA/SSE ratio which would be a "ratio" of matrices now. Obviously, there are more ways to define suitable statistics in this context! It turns out that such functionals are related to the eigenvalues of the (properly normed) Wishart-distributed matrices that enter the decomposition SST = SSA + SSE in the multivariate case.

Optional viewing: MANOVA

Matthew E. Clapham. (2016). MANOVA. Retrieved from: https://youtu.be/Oec5hwTx-SE

Roots distributions

Let $\boldsymbol{Y}_i, i = 1, 2, \ldots, n \overset{\text{ind.}}{\sim} N_p\left(\boldsymbol{\mu}_i, \Sigma\right)$. Then the following data matrix:

$$\boldsymbol{Y} = \begin{pmatrix} \boldsymbol{Y}_1^\top \\ \boldsymbol{Y}_2^\top \\ \vdots \\ \boldsymbol{Y}_n^\top \end{pmatrix} = \begin{pmatrix} Y_{11} & Y_{12} & \cdots & Y_{1p} \\ Y_{21} & Y_{22} & \cdots & Y_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{n1} & Y_{n2} & \cdots & Y_{np} \end{pmatrix} \in \mathcal{M}_{n,p}$$

is a $n \times p$ matrix containing n p-dimensional (transposed) vectors. Denote: $\mathrm{E}(\boldsymbol{Y}) = M$, $\mathrm{Var}(\overrightarrow{\boldsymbol{Y}}) = \Sigma \otimes I_n$. Let A and B be projectors such that $\boldsymbol{Q}_1 = \boldsymbol{Y}^\top A \boldsymbol{Y}$ and $\boldsymbol{Q}_2 = \boldsymbol{Y}^\top B \boldsymbol{Y}$ are two **independent** $W_p(\Sigma, v)$ and $W_p(\Sigma, q)$ matrices, respectively. Although the theory is general, to keep you on track, you could always think about a multivariate linear model example:

$$\boldsymbol{Y} = X\boldsymbol{\beta} + \boldsymbol{E}, \hat{\boldsymbol{Y}} = X\hat{\boldsymbol{\beta}}$$
$$A = I_n - X\left(X^\top X\right)^- X^\top, B = X\left(X^\top X\right)^- X^\top - \mathbf{1}_n\left(\mathbf{1}_n^\top \mathbf{1}_n\right)^{-1}\mathbf{1}_n^\top$$

and the corresponding decomposition

$$\boldsymbol{Y}\left[I_n - \mathbf{1}_n\left(\mathbf{1}_n^\top \mathbf{1}_n\right)^{-1}\mathbf{1}_n^\top\right]\boldsymbol{Y} = \boldsymbol{Y}^\top B \boldsymbol{Y} + \boldsymbol{Y}^\top A \boldsymbol{Y} = \boldsymbol{Q}_2 + \boldsymbol{Q}_1$$

of $\mathrm{SST} = \mathrm{SSA} + \mathrm{SSE} = \boldsymbol{Q}_2 + \boldsymbol{Q}_1$ where $\boldsymbol{Q}_2$ is the "hypothesis matrix" and $\boldsymbol{Q}_1$ is the "error matrix".

Lemma 3.1. *Let $\boldsymbol{Q}_1, \boldsymbol{Q}_2 \in \mathcal{M}_{p,p}$ be two positive definite symmetric matrices . Then the roots of the determinant equation $|\boldsymbol{Q}_2 - \theta\left(\boldsymbol{Q}_1 + \boldsymbol{Q}_2\right)| = 0$ are related to the roots of the equation $|\boldsymbol{Q}_2 - \lambda \boldsymbol{Q}_1| = 0$ by: $\lambda_i = \frac{\theta_i}{1-\theta_i}$ $\left(\text{or } \ \theta_i = \frac{\lambda_i}{1+\lambda_i}\right)$*

Lemma 3.2. *Let $\boldsymbol{Q}_1, \boldsymbol{Q}_2 \in \mathcal{M}_{p,p}$ be two positive definite symmetric matrices . Then the roots of the determinant equation $|\boldsymbol{Q}_1 - v\left(\boldsymbol{Q}_1 + \boldsymbol{Q}_2\right)| = 0$ are related to the roots of the equation $|\boldsymbol{Q}_2 - \lambda \boldsymbol{Q}_1| = 0$ by: $\lambda_i = \frac{1-v_i}{v_i}$ (or $v_i = \frac{1}{1+\lambda_i}$).*

We can employ the above two lemmas to see that if λ_i, v_i, θ_i are the roots of

$$|\boldsymbol{Q}_2 - \lambda \boldsymbol{Q}_1| = 0, |\boldsymbol{Q}_1 - v\left(\boldsymbol{Q}_1 + \boldsymbol{Q}_2\right)| = 0, |\boldsymbol{Q}_2 - \theta\left(\boldsymbol{Q}_1 + \boldsymbol{Q}_2\right)| = 0$$

Then:

$$\Lambda = \left|\boldsymbol{Q}_1\left(\boldsymbol{Q}_1 + \boldsymbol{Q}_2\right)^{-1}\right| = \prod_{i=1}^{p}\left(1 + \lambda_i\right)^{-1}$$

(Wilks's Criterion statistic) or

$$\left|\boldsymbol{Q}_2\boldsymbol{Q}_1^{-1}\right| = \prod_{i=1}^{p}\lambda_i = \prod_{i=1}^{p}\frac{1-v_i}{v_i} = \prod_{i=1}^{p}\frac{\theta_i}{1-\theta_i}$$

or

$$\left|\boldsymbol{Q}_2\left(\boldsymbol{Q}_1+\boldsymbol{Q}_2\right)^{-1}\right| = \prod_{i=1}^{p}\theta_i = \prod_{i=1}^{p}\frac{\lambda_i}{1+\lambda_i} = \prod_{i=1}^{p}(1-v_i)$$

and other functional transformations of these products of (random) roots would have a distribution that would only depend on p, q, v.

There are various ways to choose such functional transformations (statistics) and many have been suggested like:

- Λ (Wilks's Lambda)
- $\operatorname{tr}\left(\boldsymbol{Q}_2\boldsymbol{Q}_1^{-1}\right) = \operatorname{tr}\left(\boldsymbol{Q}_1^{-1}\boldsymbol{Q}_2\right) = \sum_{i=1}^{p}\lambda_i$ (Lawley-Hotelling trace)
- $\max_i \lambda_i$ (Roy's criterion)
- $V = \operatorname{tr}\left[\boldsymbol{Q}_2\left(\boldsymbol{Q}_1+\boldsymbol{Q}_2\right)^{-1}\right] = \sum_{i=1}^{p}\frac{\lambda_i}{1+\lambda_i}$ (Pillai statistic / Pillai's trace)

Tables and charts for their exact or approximate distributions are available. Also, P-values for these statistics are readily calculated in statistical packages. In these applications, the meaning of $\boldsymbol{Q}_1$ is of the "error matrix" (also denoted by $\boldsymbol{E}$ sometimes) and the meaning of $\boldsymbol{Q}_2$ is that of a "hypothesis matrix" (also denoted by $\boldsymbol{H}$ sometimes).

The distribution of the statistics defined above depends on the following three parameters:

- $p =$ the number of responses
- $q = \nu_h =$ degrees of freedom for the hypothesis
- $v = \nu_e =$ degrees of freedom for the error

Based on these, the following quantities are calculated: $s = \min(p, q)$, $m = 0.5(|p-q|-1)$, $n = 0.5(v-p-1)$, $r = v - 0.5(p-q+1)$, $u = 0.25(pq-2)$. Moreover, we define: $t = \sqrt{\frac{p^2q^2-4}{p^2+q^2-5}}$ if $p^2+q^2-5>0$ and $t=1$ otherwise. Let us order the eigenvalues of $\boldsymbol{E}^{-1}\boldsymbol{H} = \boldsymbol{Q}_1^{-1}\boldsymbol{Q}_2$ according to: $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p$.

Then the following distribution results are true: (Note: the F -statistic quoted below is **exact** if $s = 1$ or 2, otherwise the F-distribution is an **approximation**):

- Wilks's test. The test statistics, Wilks's lambda, is $\Lambda = \frac{|\boldsymbol{E}|}{|\boldsymbol{E}+\boldsymbol{H}|} = \prod_{i=1}^{p}\frac{1}{1+\lambda_i}$. Then it holds: $F = \frac{1-\Lambda^{1/t}}{\Lambda^{1/t}}.\frac{rt-2u}{pq} \sim F_{pq,rt-2u}$ df (**Rao's F**).
- Lawley–Hotelling trace Test. The Lawley–Hotelling statistic is $U = \operatorname{tr}(\boldsymbol{E}^{-1}\boldsymbol{H}) = \lambda_1 + \cdots +$

λ_p, and $F = 2(sn+1)\frac{U}{s^2(2m+s+1)} \sim F_{s(2m+s+1),2(sn+1)}$ df.

- Pillai's test. The test-statistics, Pillai trace, is $V = \operatorname{tr}(\boldsymbol{H}(\boldsymbol{H}+\boldsymbol{E})^{-1}) = \frac{\lambda_1}{1+\lambda_1} + \cdots + \frac{\lambda_p}{1+\lambda_p}$ and $F = \frac{2n+s+1}{2m+s+1} \times \frac{V}{s-V} \sim F_{s(2m+s+1),s(2n+s+1)}$ df.
- Roy's maximum root criterion. The test-statistic is just the largest eigenvalue λ_1.

Finally, we shall mention one historically older and very universal approximation to the distribution of the Λ statistic due to Bartlett (1927):

It holds: level of $-\left[\nu_e - \frac{p-\nu_h+1}{2}\right]\log\Lambda = c\left(p,\nu_h,M\right) \times$ level of $\chi^2_{p\nu_h}$, where the constant $c\left(p,\nu_h,M=\nu_e-p+1\right)$ is given in accompanying tables. Such tables are prepared for levels $\alpha = 0.10, 0.05, 0.025$ etc..

In the context of testing the hypothesis about the significance of the first canonical correlation, we have:

$$\boldsymbol{E} = \boldsymbol{S}_{22} - \boldsymbol{S}_{21}\boldsymbol{S}_{11}^{-1}\boldsymbol{S}_{12}, \boldsymbol{H} = \boldsymbol{S}_{21}\boldsymbol{S}_{11}^{-1}\boldsymbol{S}_{12}$$

The Wilks's statistic becomes $\frac{|\boldsymbol{S}|}{|\boldsymbol{S}_{11}||\boldsymbol{S}_{22}|}$.

We also see that in this case, if μ_i^2 were the squared canonical correlations then μ_1^2 was defined as the maximal eigenvalue to $\boldsymbol{S}_{22}^{-1}\boldsymbol{H}$, that is, it is a solution to $\left|(\boldsymbol{E}+\boldsymbol{H})^{-1}\boldsymbol{H} - \mu_1^2 I\right| = 0$ However, setting $\lambda_1 = \frac{\mu_1^2}{1-\mu_1^2}$ we see that:

$$\begin{aligned}
&\left|(\boldsymbol{E}+\boldsymbol{H})^{-1}\boldsymbol{H} - \mu_1^2 I\right| = 0 \\
\implies &\left|\boldsymbol{H} - \mu_1^2(\boldsymbol{E}+\boldsymbol{H})\right| = 0 \\
\implies &\left|\boldsymbol{H} - \frac{\mu_1^2}{1-\mu_1^2}\boldsymbol{E}\right| = 0 \\
\implies &\left|\boldsymbol{E}^{-1}\boldsymbol{H} - \lambda_1 I\right| = 0
\end{aligned}$$

holds and λ_1 is an eigenvalue of $\boldsymbol{E}^{-1}\boldsymbol{H}$. Similarly you can argue for the remaining $\lambda_i = \frac{\mu_i^2}{1-\mu_i^2}$ values.

Comparisons

From all statistics discussed, Wilks's lambda has been most widely applied. One important reason for this is that this statistic has the virtue of being convenient to use and, more importantly, being related to the Likelihood Ratio Test! Despite the above, the fact that so many different statistics exist for the same hypothesis testing problem, indicates that there is no universally best test. Power comparisons of the above tests are almost lacking since the distribution of the statistic under alternatives is hardly known.

Example: Fitness club

Before working through the Demonstration and Challenge activities in the following sections. Watch an explanation of the example by Dr Pavel Krivitsky.

Demonstration: Multivariate LM and MANOVA

This demonstration can be completed using the provided RStudio or your own RStudio.

To complete this task select the 'MLM_Examples.demo.Rmd' in the 'Files' section of RStudio. Follow the demonstration contained within the RMD file.

If you choose to complete the example in your own RStudio, upload the following file:

MLM_Examples.demo.Rmd

The contents of the RMD file are also displayed below:

Packages

```
library(here)
library(GGally)
library(dplyr)
library(readr)
library(purrr)
```

Fitness club example

Three physiological and three exercise variables were measured on twenty middle aged men in a fitness club. Canonical correlation is used to determine if the physiological variables are related in any way to the exercise variables.

Data

```
'weight waist pulse chins situps jumps
   191  36  50   5  162   60
   189  37  52   2  110   60
   193  38  58  12  101  101
   162  35  62  12  105   37
   189  35  46  13  155   58
   182  36  56   4  101   42
   211  38  56   8  101   38
   167  34  60   6  125   40
   176  31  74  15  200   40
   154  33  56  17  251  250
   169  34  50  17  120   38
   166  33  52  13  210  115
```

```
   154  34  64  14  215  105
   247  46  50   1   50   50
   193  36  46   6   70   31
   202  37  62  12  210  120
   176  37  54   4   60   25
   157  32  52  11  230   80
   156  33  54  15  225   73
   138  33  68   2  110   43' %>%
textConnection() %>% read.table(header=TRUE) -> fitness

fitness
```

Exploratory data analysis

```
summary(fitness)
ggpairs(fitness)
```

The data do have some outliers.

Multivariate linear model analysis

Estimation

```
class(fitness.mlm <- lm(cbind(chins,situps,jumps)~waist+pulse+weight, data=fitness))
summary(fitness.mlm)
coef(fitness.mlm) # Now a p by k matrix.
vcov(fitness.mlm) # Now a p*k by p*k matrix.
estVar(fitness.mlm) # Now a p by p matrix.
resid(fitness.mlm) # Now a p-column matrix.
fitted(fitness.mlm) # Now a p-column matrix.
```

Hypothesis testing

```
# ? anova.mlm for options
anova(fitness.mlm) # Pillai's trace by default
anova(fitness.mlm, test="Wilks")
anova(fitness.mlm, test="Hotelling-Lawley")
anova(fitness.mlm, test="Roy")
# Sphericity test also possible---see help.

 # Test the effect of weight in the presence of the others:
fitness.mlm0 <- lm(cbind(chins,situps,jumps)~waist+pulse, data=fitness)
anova(fitness.mlm, fitness.mlm0)
```

Diagnostics

```
plot(fitness.mlm) # Error: not implemented.
```

Pairwise plots of all residuals:

```
ggpairs(as_tibble(resid(fitness.mlm))) # Note: type="pearson" doesn't work as of R
3.6!
```

Make our own Pearson residuals:

```
# Residual standard deviations: square root of the diagonal of the estimated
covariance of residuals:
sds <- fitness.mlm %>% estVar %>% diag %>% sqrt
PR <- sweep(resid(fitness.mlm), 2, sds, `/`)
ggpairs(as_tibble(PR))
```

Related: decorrelate the residuals:

```
# Inverse of the Cholesky decomposition (square root) of the estimated variance-
covariance matrix of the residuals
Uinv <- estVar(fitness.mlm) %>% chol %>% solve
(PRI <- resid(fitness.mlm)%*%Uinv) %>% cor %>% zapsmall # Now close to identity
matrix.
ggpairs(as_tibble(PRI))
```

Residuals vs. fitted:

```
# I.e., join the predictors with the residuals, and plot the predictors horizontally
and residuals vertically:
pairs(as_tibble(cbind(PR,fitted(fitness.mlm))), horInd=1:3,verInd=4:6,
panel=function(x,y,...){abline(h=0,col="gray");points(x[abs(y)<2],y[abs(y)
<2]);if(any(abs(y)>=2))
text(x[abs(y)>=2],y[abs(y)>=2],labels=which(abs(y)>=2));lines(lowess(x,y),col="red")})
```

Challenge: Multivariate LM and MANOVA

If you choose to complete this task in your own RStudio, upload the following file:

MLM_Examples.challenge.Rmd

Select the 'MLM_Examples.challenge.Rmd' in the 'Files' section to begin. Enter your response to the tasks in the 'Enter your code here' section.

This activity and the solution will be discussed at the Collaborate session this week. In the meantime, share and discuss your results in the 'Tutorials' discussion forum.

The solution will also be available here on Friday of Week 3 by clicking on the 'Solution' tab.

The contents of the RMD file are also displayed below:

Packages

```
library(here)
library(GGally)
library(dplyr)
library(readr)
library(purrr)
```

Challenge

Recall the dataset `pizza.csv`, containing nutritional data from a variety of pizza brands:

`brand` : Pizza brand (class label)

`id` : Sample analysed

`prot` : Amount of protein per 100 grams in the sample

`fat` : Amount of fat per 100 grams in the sample

`ash` : Amount of ash per 100 grams in the sample

`sodium` : Amount of sodium per 100 grams in the sample

`carb` : Amount of carbohydrates per 100 grams in the sample

`cal` : Amount of calories per 100 grams in the sample

We will focus on brand A:

```
pizzaA <- read_csv(here("datasets","pizza.csv")) %>% filter(brand=="A") %>% select(-brand)
```

Task 1: Fit and diagnose a linear model jointly predicting caloric, ash, and sodium content from fat, protein, and carbohydrate content.

Task 2: Test whether volatile acidity is predictive of the three response variables (jointly) in the presence of other predictors.

Task 3: Test whether fat content is predictive of the three response variables (jointly) in the presence of other predictors.

Task 4: Now, consider the full dataset, with brands. Use MANOVA to test whether different brands have different population mean measured pizza properties.

```
pizza <- read_csv(here("datasets","pizza.csv"))
ggpairs(pizza, mapping=aes(col=brand, alpha=0.3))
```

Topic 1: Deriving likelihood ratio tests for a covariance matrix under the normality assumption

Welcome to Week 4

Dr Pavel Krivitsky gives you a brief overview of topics and concepts we'll be covering in this week.

Transcript

Weekly learning outcomes

- Convert a substantive hypothesis about relationships between variables in a population to a statistical hypothesis about their covariance matrix.
- Derive a test for a given statistical hypothesis about a covariance matrix of a normal population.
- Perform a test for a given statistical hypothesis about a covariance matrix of a normal population and interpret the results in the context of the problem.
- Explain the difference between principal component analysis and factor analysis.
- Interpret results from a factor analysis.

- Identify the optimal number of factors using a variety of techniques.

Topics we will cover are:

- Topic 1: Deriving likelihood ratio tests for a covariance matrix under the normality assumption.
- Topic 2: Factor analysis concepts and interpretation.
- Topic 3: Estimating and testing factor analysis models.
- Topic 4: Overview of structural equation models.

Optional reading

An alternative presentation of the concepts for this week can be found in:

Johnson, R. A., & Wichern, D. (2008). *Applied Multivariate Statistical Analysis* (6th ed.). Pearson Prentice Hall.

- Chapter 9

Muirhead, R. (1982) *Aspects of Multivariate Statistical Theory*. Wiley, New York.

- Chapter 8

All readings are available from the course Leganto reading list. Please keep in mind that you will need to be logged into Moodle to access the Leganto reading list.

Questions about this week's topics?

This week's topics were prepared by Dr P. Krivitsky. If you have any questions or comments, please post them under Discussion or email directly: p.krivitsky@unsw.edu.au

Tests of a covariance matrix

Previously we have developed a number of techniques for decomposing and analysing covariance matrices and their properties. Here, we develop a general family of tests for their structure, which will let you specify almost arbitrary tests for the covariance structure of a multivariate normal population.

Test of $\sum = \sum_0$

We start with this simpler case since ideas are more transparent. The practically more relevant cases are about comparing covariance matrices of two or more multivariate normal populations but the derivations of the latter tests is more subtle. For these we will only formulate the final results.

Assume now that we have the sample $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n$ from a $N_p(\boldsymbol{\mu}, \Sigma)$ distribution and we would like to test $H_0 : \Sigma = \Sigma_0$ against the alternative $H_1 : \Sigma \neq \Sigma_0$. Obviously the problem can be easily transformed into testing $\bar{H}_0 : \Sigma = I_p$ since otherwise we can consider the modified observations $\boldsymbol{Y}_i = \Sigma_0^{-\frac{1}{2}} \boldsymbol{X}_i$ which under H_0 will be multivariate normal with a covariance matrix being equal to I_p. Therefore we can assume that $\boldsymbol{X}_1, \boldsymbol{X}_2, ..., \boldsymbol{X}_n$ is a sample from a $N_p(\boldsymbol{\mu}, \Sigma)$ and we want to test $H_0 : \Sigma = I_p$ versus $H_1 : \Sigma \neq I_p$.

We will derive the likelihood ratio test for this problem. The likelihood function is

$$
\begin{aligned}
L(\boldsymbol{x};\boldsymbol{\mu},\Sigma) &= (2\pi)^{-\frac{np}{2}}\,|\Sigma|^{-\frac{n}{2}}\,\mathrm{e}^{-\frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{x}_i-\boldsymbol{\mu})^{\top}\Sigma^{-1}(\boldsymbol{x}_i-\boldsymbol{\mu})} \\
&= (2\pi)^{-\frac{np}{2}}\,|\Sigma|^{-\frac{n}{2}}\,\mathrm{e}^{-\frac{1}{2}\,\mathrm{tr}\left[\Sigma^{-1}\sum_{i=1}^{n}(\boldsymbol{x}_i-\boldsymbol{\mu})(\boldsymbol{x}_i-\boldsymbol{\mu})^{\top}\right]}
\end{aligned}
$$

Under the hypothesis H_0, the maximum of the likelihood function is obtained when $\bar{\boldsymbol{\mu}} = \bar{\boldsymbol{x}}$. Under the alternative we have to maximise with respect to both $\boldsymbol{\mu}$ and Σ and we know from Section "Maximum Likelihood Estimators" that the maximum of the likelihood function is obtained for $\hat{\boldsymbol{\mu}} = \bar{\boldsymbol{x}}$ and $\hat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^{\tau}$. Then we obtain easily the likelihood ratio

$$
\Lambda = \frac{\max_{\boldsymbol{\mu}} L\left(\boldsymbol{x};\boldsymbol{\mu},I_p\right)}{\max_{\boldsymbol{\mu},\Sigma} L(\boldsymbol{x};\boldsymbol{\mu},\Sigma)} = \frac{\mathrm{e}^{\left[-\frac{1}{2}\,\mathrm{tr}\,\boldsymbol{V}\right]}}{|\boldsymbol{V}|^{-\frac{n}{2}}\,n^{\frac{np}{2}}\,\mathrm{e}^{-\frac{np}{2}}}
$$

where $\boldsymbol{V} = \Sigma_{i=1}^{n}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^{\top}$. Therefore

$$
-2\log\Lambda = np\log n - n\log|\boldsymbol{V}| + \mathrm{tr}\,\boldsymbol{V} - np, \tag{4.1}
$$

and according to the asymptotic theory the quantity in (4.1) is asymptotically distributed as $\chi^2_{p(p+1)/2}$ (the degrees of freedom being the difference of the number of free parameters under the alternative and under the hypothesis). This test would reject H_0 if the value of the $-2\log\Lambda$ statistic is significantly large.

Sphericity test

It is more realistic to assume that the structure of the covariance matrix is only known up to some constant. Having in mind the discussion in the beginning of the previous slide, we can assume without loss of generality that $H_0 : \Sigma = \sigma^2 I_p$ against a general alternative. This test has the name "sphericity test". The likelihood ratio test can be developed in a manner similar to the previous case and the final result is that

$$-2 \log \Lambda = np \log(n\hat{\sigma}^2) - n \log|\boldsymbol{V}|.$$

Here, $\hat{\sigma}^2 = \frac{1}{np} \sum_{i=1}^{n} (\boldsymbol{x}_i - \bar{\boldsymbol{x}})^\tau (\boldsymbol{x}_i - \bar{\boldsymbol{x}})$. The asymptotic distribution of $np \log(n\hat{\sigma}^2) - n \log|\boldsymbol{V}|$ under the null hypothesis will be again χ^2 but the degrees of freedom are this time $\frac{p(p+1)}{2} - 1 = \frac{(p-1)(p+2)}{2}$. Again, the hypothesis will be rejected for large values.

Note that this notion of sphericity is distinct from that of the repeated measures ANOVA tested by Mauchly's Sphericity Test. (You can learn more about it in Longitudinal Data Analysis.) The specific test discussed here (with a scaled identity matrix as the null hypothesis) is called "spherical" because it implies a multivariate normal distribution that is symmetrical in every possible way---like a sphere.

General situation

Testing equality of covariance matrices from k different multivariate normal populations $N_p(\boldsymbol{\mu}_i, \Sigma_i), i = 1, 2, \ldots, k$ is a very important problem especially in discriminant analysis and multivariate analysis of variance. Let,

k be the number of populations;
p the dimension of vector;
n the total sample size $n = n_1 + n_2 + \ldots n_k$,
n_i being the sample size for each population.

The analysis of deviance test statistic that results is

$$-2 \log \frac{\prod_{i=1}^k \left| \hat{\boldsymbol{\Sigma}}_i \right|^{\frac{n_i}{2}}}{\left| \hat{\boldsymbol{\Sigma}}_{\text{pooled}} \right|^{\frac{n}{2}}},$$

with $\hat{\boldsymbol{\Sigma}}_i$ the MLE sample variance (with denominator n) of population i, and $\hat{\boldsymbol{\Sigma}}_{\text{pooled}} = \frac{1}{n} \sum_{i=1}^k n_k \hat{\boldsymbol{\Sigma}}_i$, asymptotically distributed $\chi^2_{(k-1)p(p+1)/2}$.

It has been noticed that this test has the defect that it is (asymptotically) biased: that is, the probability of rejecting H_0 when H_0 is false can be smaller than the probability of rejecting H_0 when H_0 is true (i.e., it may happen that in some points of the parameter space the probability of a correct decision is

smaller than the probability for a wrong decision). Hence it is desirable to modify it to make it asymptotically unbiased.

Further let $N = n - k$ and $N_i = n_i - 1$. Under the null hypothesis of equality of all k covariance matrices, it holds:

$$-2\rho \log \frac{\prod_{i=1}^{k} |\boldsymbol{S}_i|^{\frac{N_i}{2}}}{|\boldsymbol{S}_{\text{pooled}}|^{\frac{N}{2}}} \tag{4.2}$$

for $\rho = 1 - \left[\left(\sum_{i=1}^{k} \frac{1}{N_i} \right) - \frac{1}{N} \right] \frac{2p^2+3p-1}{6(p+1)(k-1)}$, $\boldsymbol{S}_i$ the sample variance (with $n-1$ denominator) of population i, and $S_{\text{pooled}} = \frac{1}{N} \sum_{i=1}^{k} N_k \boldsymbol{S}_i$, is asymptotically distributed as $\chi^2_{\frac{1}{2}(k-1)p(p+1)}$. Large values of the statistic are significant and lead to the rejection of the hypothesis about equality of the k covariance matrices.

In the following, we will avoid the subtle details and refer to Chapter 8 of the monograph:

- Muirhead, R. (1982) *Aspects of Multivariate Statistical Theory*. Wiley, New York.

The *modified* LR is achieved by replacing n_i and n by N_i and N (that is, by the correct degrees of freedom). We note that indeed $\rho = 1 - [(\Sigma_{i=1}^{k} \frac{1}{N_i}) - \frac{1}{N}] \frac{2p^2+3p-1}{6(p+1)(k-1)}$ is close to 1 anyway if all sample sizes n_i were very large. Finally, the scaling of the test statistic by $\rho = 1 - [(\sum_{i=1}^{k} \frac{1}{N_i}) - \frac{1}{N}] \frac{2p^2+3p-1}{6(p+1)(k-1)}$ that is made in (4.2) serves to improve the quality of the asymptotic approximation of the statistic by the limiting $\chi^2_{\frac{1}{2}(k-1)p(p+1)}$ distribution. Such (asymptotically negligible) scalar transformations of the LR statistic that yield improved test statistic with a chi-squared null distribution of order $O(1/n)$ instead of the ordinary $O(1)$ for the standard LR, are known in the literature under the common name **Bartlett corrections**. Thus, (4.2) is a Bartlett corrected version of the modified LR statistic.

Software

R: `heplots::boxM, MVTests::BoxM`

The statistic (4.2) is the one that is implemented in software packages.

Challenge: Multivariate LM and MANOVA

If you choose to complete this task in your own RStudio, upload the following file:

Cov_Test_Example.challenge.Rmd

Click on the 'Cov_Test_Example.challenge.Rmd' in the 'Files' section to begin. Enter your response to the tasks in the 'Enter your code here' section.

The solution will also be available here on Friday of this week by clicking on the 'Solution' tab in the top right corner.

The contents of the RMD file are also displayed below:

Packages

```
library(here)
library(GGally)
library(dplyr)
library(readr)
library(purrr)
```

Challenge

Recall the dataset `pizza.csv`, containing nutritional data from a variety of pizza brands:

`brand` : Pizza brand (class label)

`id` : Sample analysed

`prot` : Amount of protein per 100 grams in the sample

`fat` : Amount of fat per 100 grams in the sample

`ash` : Amount of ash per 100 grams in the sample

`sodium` : Amount of sodium per 100 grams in the sample

`carb` : Amount of carbohydrates per 100 grams in the sample

`cal` : Amount of calories per 100 grams in the sample

```
pizza <- read_csv(here("datasets","pizza.csv"))
ggpairs(pizza, mapping=aes(col=brand, alpha=0.3))
```

Task 1: We had conducted a MANOVA test to determine if different brands have different measurements. That test assumes equal variances. Are they equal?

Hint: Use `boxM` from the `heplots` package.

Check your understanding

Complete the below exercises to check your understanding of concepts presented so far.

1.

a) Follow the discussion about the sphericity test. Argue that if $\hat{\lambda}_i, i = 1, 2, \ldots, p$ denote the eigenvalues of the empirical covariance matrix S then

$$-2 \log \Lambda = np \log \frac{\text{arithm. mean } \hat{\lambda}_i}{\text{geom. mean } \hat{\lambda}_i}$$

Of course, the above statistic is asymptotically $\chi^2_{(p+2)(p-1)/2}$ distributed under H_0 since it only represents the sphericity test in a different form.

b) Show that the likelihood ratio test of

$$H_0 : \Sigma \text{ is the diagonal matrix}$$

rejects H_0 when $-n \log |R|$ is larger than $\chi^2_{\alpha, p(p-1)/2}$. (Here R is the empirical correlation matrix, p is the dimension of the multivariate normal and n is the sample size.)

Topic 2: Factor analysis concepts and interpretation

Factor analysis

Let $\boldsymbol{Y}_i, i = 1, 2, .., n$ be independent $N_p(\boldsymbol{\mu}, \Sigma)$ variables (think of the $\boldsymbol{Y}_i$s as a results of a battery of p tests applied to the ith individual). Fundamental assumption in factor analysis:

$$\boldsymbol{Y}_i = \Lambda \boldsymbol{f}_i + \boldsymbol{e}_i \tag{4.3}$$

$\Lambda \in M_{p,k}$ factor loading matrix (full rank);

$\boldsymbol{f}_i \in \mathbb{R}^k (k < p)$ factor variable. The components of $\boldsymbol{f}_i$ are thought to be the (latent) factors. Usually $\boldsymbol{f}_i$ are taken to be independent $N(\boldsymbol{\alpha}, I_k)$ (i.e., "orthogonal") but also "oblique" factors are considered sometimes with a covariance matrix $\neq I_k$.

$\boldsymbol{e}_i$ independent $N(\boldsymbol{\theta}, \Sigma_e)$ withΣ_e **diagonal**, i.e., $\Sigma_e = \mathrm{diag}(\sigma_1^2, \sigma_2^2, .., \sigma_p^2)$.

Also, the $\boldsymbol{e}$s are independent from the $\boldsymbol{f}$s.

Then,

$$\boldsymbol{\mu} = \Lambda \boldsymbol{\alpha} + \boldsymbol{\theta}; \Sigma = \Lambda \Lambda^\top + \Sigma_e,$$

or, componentwise:

$$\mathrm{Var}(Y_{ir}) = \sum_{j=1}^{k} \lambda_{rj}^2 + \sigma_r^2 = \mathrm{communality} + \mathrm{uniqueness}.$$

$$\mathrm{Cov}(Y_{ir}, Y_{is}) = \sum_{j=1}^{k} \lambda_{rj} \lambda_{sj}.$$

The fundamental idea of factor analysis is to describe the **covariance relationships** among **many** variables (p "large") in terms of few (k "small") underlying, not observable (latent) random quantities (the **factors**). The model is motivated by the following argument: suppose variables can be grouped by their correlations. That is, all variables in a particular group are highly correlated among themselves but have relatively small correlations with variables in a different group. It is then quite reasonable to assume that each group of variables represents a single underlying construct (**factor**) that is "responsible" for the observed correlations.

Optional viewing: Factor Analysis - an introduction

Ben Lambert. (2014). Factor Analysis - an introduction. Retrieved from: https://youtu.be/WV_jcaDBZ2I

Important notes

- The model (4.3) is similar to a linear regression model but the key differences are that $\boldsymbol{f}_i$ are **random and are not observable**.
- If we knew the Λ (or have found estimates of them), then using properties of orthogonal projections on the linear space spanned by the columns of Λ, we would get:

$$\hat{\boldsymbol{\alpha}} = (\Lambda^\top \Lambda)^{-1} \Lambda^\top \bar{\boldsymbol{Y}}; \hat{\boldsymbol{\theta}} = \bar{\boldsymbol{Y}} - \Lambda \hat{\boldsymbol{\alpha}}.$$

  Because of the above observation, we can consider only $\boldsymbol{\mu}$, Λ, and σ_i^2, $i = 1, 2, \ldots, p$ as unknown parameters when parameterising the factor analysis model. Note also that primary interest in factor analysis is focused on estimating Λ.
- **There is a fundamental indeterminacy** in this model even when we require that $\mathrm{Var}(\boldsymbol{f}) = I_k$ since, if $P \in \mathcal{M}_{k,k}$ **is any** orthogonal matrix then obviously

$$\Lambda\Lambda^\top = \Lambda P(\Lambda P)^\top;\ \Lambda \boldsymbol{f}_i = (\Lambda P)(P^\top \boldsymbol{f}_i).$$

Hence replacing Λ by ΛP and $\boldsymbol{f}_i$ by $P^\top \boldsymbol{f}_i$ leads to the same equations.

Maximum Likelihood Estimation

The likelihood function for the n observations $\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_n \in \mathbb{R}^p$ is

$$L(\boldsymbol{Y}; \boldsymbol{\mu}, \Lambda, \sigma_1^2, \sigma_2^2, .., \sigma_p^2) = (2\pi)^{-np/2} |\Sigma|^{-n/2} \exp\left[-\frac{1}{2} \sum_{i=1}^{n} (\boldsymbol{Y}_i - \boldsymbol{\mu})^\top \Sigma^{-1} (\boldsymbol{Y}_i - \boldsymbol{\mu}) \right]$$
$$= (2\pi)^{-np/2} |\Sigma|^{-n/2} \exp\left[-\frac{n}{2} \left(\operatorname{tr}(\Sigma^{-1} \boldsymbol{S}) + (\bar{\boldsymbol{Y}} - \boldsymbol{\mu})^\top \Sigma^{-1} (\bar{\boldsymbol{Y}} - \boldsymbol{\mu}) \right) \right]$$

with $\boldsymbol{S} = \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{Y}_i - \bar{\boldsymbol{Y}})(\boldsymbol{Y}_i - \bar{\boldsymbol{Y}})^\top$, and keeping in mind that Σ is a function of Λ and Σ_e (and therefore of $\sigma_1^2, \ldots, \sigma_p^2$). Taking $\log L$, we get:

$$\log L(\boldsymbol{Y}; \boldsymbol{\mu}, \Lambda, \sigma_1^2, \sigma_2^2, .., \sigma_p^2) = -\frac{np}{2} \log(2\pi) - \frac{n}{2} \log(|\Sigma|) - \frac{n}{2} \left[\operatorname{tr}(\Sigma^{-1} \boldsymbol{S}) + (\bar{\boldsymbol{Y}} - \boldsymbol{\mu})^\top \Sigma^{-1} (\bar{\boldsymbol{Y}}$$

After some vector calculus and matrix algebra, we find that,

$$(\Sigma_e^{-1/2} \boldsymbol{S} \Sigma_e^{-1/2}) \Sigma_e^{-1/2} \Lambda = \Sigma_e^{-1/2} \Lambda (I + \Lambda^\top \Sigma_e^{-1} \Lambda). \tag{4.4}$$

Recall the note about indeterminacy of Λ. This can be a blessing in disguise, in particular (at least one) solution is one for which $\Lambda^\top \Sigma^{-1} e \Lambda$ **is diagonal**. Then (4.4) implies that the matrix $\Sigma e^{-1/2} \Lambda$ has as its columns k eigenvectors that correspond to the k eigenvalues of $\Sigma_e^{-1/2} \boldsymbol{S} \Sigma_e^{-1/2}$. More subtle analysis shows that to obtain the maximum likelihood estimator, these have to be the eigenvectors that correspond to the **largest** eigenvalues of $\Sigma_e^{-1/2} \boldsymbol{S} \Sigma_e^{-1/2}$.

Based on this fact, the following iterative solution (due to Lawley) has been proposed that can be described algorithmically as follows:

1. With an initial guess $\tilde{\Sigma}_e$, calculate $\tilde{\Sigma}_e^{-1/2} \tilde{\Lambda}$ by using the eigenvectors of the k largest eigenvalues of $\tilde{\Sigma}_e^{-1/2} \boldsymbol{S} \tilde{\Sigma}_e^{-1/2}$.
2. Then from $\tilde{\Sigma}_e^{-1/2} \tilde{\Lambda}$, get a (first iteration) value for $\tilde{\Lambda}$.
3. With this value of $\tilde{\Lambda}$ we can calculate the value of $\tilde{Q}(\tilde{\Sigma}_e) = \frac{1}{2} \log|\tilde{\Lambda}\tilde{\Lambda}^\top + \tilde{\Sigma}_e| + \frac{1}{2} \operatorname{tr}(\tilde{\Lambda}\tilde{\Lambda}^\top + \tilde{\Sigma}_e)^{-1} \boldsymbol{S}$ (which is the value of the functional). This functional only depends on the p nonzero values of $\tilde{\Sigma}_e$ and there are several powerful numerical procedures to find its minimum.
4. If it is achieved at Σ_e^*, then update $\tilde{\Sigma}_e$ with the new guess Σ_e^* and repeat from Step 1 to convergence.

Topic 3: Estimating and testing factor analysis models

Hypothesis testing under multivariate normality assumption

The most interesting hypothesis is H_0 : k factors against H_1 : $\neq k$ factors.

$$\log L_1 = -\frac{np}{2}\log(2\pi) - \frac{n}{2}\log|\boldsymbol{S}| - \frac{np}{2}$$

$$\log L_0 = -\frac{np}{2}\log(2\pi) - \frac{n}{2}\log|\hat{\Sigma}| - \frac{n}{2}\operatorname{tr}(\hat{\Sigma}^{-1}\boldsymbol{S})$$

(where $\hat{\Sigma} = \hat{\Lambda}\hat{\Lambda}^\top + \hat{\Sigma}_e$). Hence $-2\log\frac{L_0}{L_1} = n[\log|\hat{\Sigma}| - \log|\boldsymbol{S}| + \operatorname{tr}(\hat{\Sigma}^{-1}\boldsymbol{S}) - p]$. The asymptotic distribution of this statistic is χ^2 with $\mathrm{df} = \frac{p(p+1)}{2} - [pk + p - \frac{k(k-1)}{2}] = \frac{1}{2}[(p-k)^2 - p - k]$.

Varimax method of rotating the factors

If $\hat{\Lambda}_0$ is the estimated factor loading matrix obtained by the ML method, we know that $\hat{\Lambda} = \hat{\Lambda}_0 P$ with any orthogonal $P \in \mathcal{M}_{k,k}$ can be used instead. How to choose a particular P such that $\hat{\Lambda}$ has some desirable properties?

Let $d_r = \sum_{i=1}^p \lambda_{ir}^2$, then the **varimax method of rotating the factors** consists in choosing P to maximise

$$\boldsymbol{S}_d = \sum_{r=1}^k \left\{ \sum_{i=1}^p (\lambda_{ir}^2 - \frac{d_r}{p})^2 \right\} = \sum_{r=1}^k \left\{ \sum_{i=1}^p \lambda_{ir}^4 - \frac{\left(\sum_{i=1}^p \lambda_{ir}^2\right)^2}{p} \right\}.$$

This corresponds to the wish to make, for each column of factor loadings, some of the coordinates to be "very large" and the rest to be "very small" (in absolute value). Iterative solution to the above rotation problem exists.

Note: Rotation of factor loadings is **particularly recommended** for loadings obtained by ML method since the initial values of $\hat{\Lambda}_0$ are constrained to satisfy the condition that $\hat{\Lambda}_0^\top \Sigma_e^{-1} \hat{\Lambda}_0$ be diagonal. This is convenient for computational purposes but may not lead to easily interpretable factors.

Relationship to principal component analysis

There are different ways in which you can relate factor analysis to principal component analysis. We will discuss two of them here.

The principal component solution of the factor model

Starting with the matrix:

$$\boldsymbol{S} = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{Y}_i - \bar{\boldsymbol{Y}})(\boldsymbol{Y}_i - \bar{\boldsymbol{Y}})^\top$$

we can write down its spectral decomposition by using **all** of its p eigenvalues and eigenvectors. In such a way we would derive a perfect reconstruction of $\boldsymbol{S}$ but since it has been achieved by using p factors, it does not deliver any dimension reduction and is useless. We would prefer to employ a smaller number k of eigenvalues and eigenvectors of $\boldsymbol{S}$ and to get only an approximate reconstruction of $\boldsymbol{S}$

$$S \approx \sum_{i=1}^{k} \tau_i \vec{\boldsymbol{a}}_i \vec{\boldsymbol{a}}_i^\top = \Lambda\Lambda^\top$$

whereby τ_i are the characteristic roots of $\boldsymbol{S}$, taking the k biggest ones (w.l.o.g. $\tau_1, \tau_2, .., \tau_k$) and $\boldsymbol{a}_i$ being their corresponding eigenvectors. Since the understanding is that (if k is the right number of factors) all communalities have been taken into account then $s_{ii} - \sum_{j=1}^{k} \lambda_{ij}^2$ would be the estimators of the uniquenesses. This approach shows the k factors have been extracted from $\boldsymbol{S}$ in the same way like the principal components are calculated. The method is called **the principal component solution of the factor model**.

The principal factor solution

This is yet another method that uses similar ideas from principal components analysis. It is similar to the principal component solution, but the factor extraction is not performed directly on $\boldsymbol{S}$. To describe it, let us assume for a moment that the uniquenesses are known (or can be estimated reasonably well) and we can decompose

$$\boldsymbol{S} = \boldsymbol{S}_r + \Sigma_e$$

whereby the number k of factors is known and Σ_e is the diagonal matrix containing the uniquenesses. Then the factor analysis model states that (an estimate of) Λ should satisfy

$$\boldsymbol{S}_r = \boldsymbol{S} - \Sigma_e = \Lambda\Lambda^\top$$

Hence Λ estimate can be found by performing principal component analysis on $\boldsymbol{S}_r$:

If $\boldsymbol{S}_r = \sum_{i=1}^{p} t_i \vec{\boldsymbol{b}}_i \vec{\boldsymbol{b}}_i^{\top}$, t_i being the characteristic roots of $\boldsymbol{S}_r$, take the k biggest ones (w.l.o.g. $t_1, t_2, .., t_k$). Denote

$$B = \begin{pmatrix} \boldsymbol{b}_1 & \boldsymbol{b}_2 & \cdots & \boldsymbol{b}_k \end{pmatrix}; \ \Delta = \operatorname{diag}(t_1, t_2, \ldots, t_k).$$

Then $\hat{\Lambda} = B\Delta^{1/2}$. Can do it also iteratively!

This approach has some problems:

1. There is no reliable estimate of Σ_e available. (The most commonly used one in the case where $\boldsymbol{S}$ is the **correlation** matrix $\boldsymbol{R}$ is $\sigma_{ei}^2 = 1/r^{ii}$ where r^{ii} is the ith diagonal element of $\boldsymbol{R}^{-1}$.)
2. How to select k?

Note: The methods in this section are not efficient as compared to the ML method and in general, the ML method is the preferred one. However, for the ML method one has to assume normality and the alternative approaches described here are used in cases where multivariate normality is in a serious doubt. Most often in practice the choice of k is done by combining subject matter knowledge, "reasonableness" of results and by looking at proportion variance explained.

R implementations

- `stats::factanal()` is the built-in implementation.
- Package `psych` contains additional functions and utilities, as well as its own implementation, `psych::fa()`.
- Package `nFactors` contains utilities for determining the number of factors (e.g., scree plots).

Optional viewing: Principal component analysis and factor analysis in R

Demonstration: Factor analysis

This demonstration can be completed using the provided RStudio or your own RStudio.

To complete this task select the 'Factor_Analysis_Examples.demo.Rmd' in the 'Files' section of RStudio. Follow the demonstration contained within the RMD file.

If you choose to complete the example in your own RStudio, upload the following file:

Factor_Analysis_Examples.demo.Rmd

The contents of the RMD file are also displayed below:

Packages

```
library(here)
library(dplyr)
library(readr)
# stats is loaded by default
library(psych)
library(GGally)
```

Socioeconomic data

Data were collected about five socioeconomic variables for 12 census data in the Los Angeles area. The five variables represent total population, median school years, total unemployment, miscellaneous professional services, and median house value. We will perform a factor analysis on these data and demonstrate maximum likelihood and varimax rotation.

Data

This dataset comprises measurements of five socioeconomic variables for 12 census tracts in the Los Angeles area:

`pop` : total population

`school` : median school years

`employ` : total unemployment

`services` : miscellaneous professional services

`house` : median household value

```
socecon <- read.table(here("datasets","factor.dat"),
```

```
col.names=c("pop","school","employ","services","house"))
```

Exploratory data analysis

```
summary(socecon)
ggpairs(socecon)
```

We do see some deviation from normality, though with a small sample size, it's hard to tell.

Factor analysis

```
(socf1stats <- factanal(socecon, 1))
```

With a small p-value, the χ^2 test tells us that there **is** sufficient evidence to believe that additional factors are needed to explain the correlations among these variables.

```
(socf2stats <- factanal(socecon, 2))
```

We see two factors, the first one having positive loadings on house, school, and services (with modest weight on unemployment), and the second factor on population and unemployment (with modest weight on services). This suggests that the first factor is the wealth of the community and the second factor is its population.

The χ^2 test tells us that there **is not** sufficient evidence to believe that additional factors are needed to explain the correlations among these variables.

We can't fit a model with 3 factors:

```
(socf3stats <- factanal(socecon, 3)) # Error: too many factors for the number of
variables.
```

This is not surprising as the number of parameters in the model will exceed the number of elements in $\Sigma(\frac{1}{2}[(p-k)^2 - p - k] = -2)$.

Using `psych`, we fit a model with one factor: (Note that it has a different default method, so we specify the MLE/VariMax combination explicitly.)

```
(socf1 <- fa(socecon, 1, fm="ml", rotate="varimax")) # Other methods also possible.
plot(socf1)
```

It gives us more output, but the factor loadings (`ML`), the communalities (`h2`) and the uniqunesses (`u2`) are the same to rounding error.

It also gives us several variants of a test of the hypothesis that one factor is sufficient, and shows that it is not.

The plot is the load against variable index:

```
(socf2 <- fa(sococon, 2, fm="ml", rotate="varimax")) # Other methods also possible.
plot(socf2)
```

We now have two factors. Note that `ML2` corresponds to `Factor1` in `factanal()` and `ML1` to `Factor2`.

The plot gives us the loading on Factor 1 against those on Factor 2, and highlights which variables "belong" to which factor.

We can also fit and plot 3 factors but notice how we are running out of degrees of freedom:

```
(socf3 <- fa(sococon, 3, fm="ml", rotate="varimax")) # Works but complains.
plot(socf3)
```

Package `psych`'s `fa.parallel()` can also be used to quickly estimate a good number of factors using an analogue of the Kaiser's rule:

```
fa.parallel(sococon, fm="ml")
```

Notice that we do not specify the rotation method: all possible rotations have the same goodness of fit.

Extracting results

You can use the `unclass()` function to extract factor analysis information from the fit results:

```
# Contents of a factanal object:
unclass(socf2stats)
```

```
# Contents of an fa object:
unclass(socf2)
```

Challenge: Factor analysis

If you choose to complete this task in your own RStudio, upload the following file:

Factor_Analysis_Examples.challenge.Rmd

Click on the 'Factor_Analysis_Examples.challenge.Rmd' in the 'Files' section to begin. Enter your response to the tasks in the 'Enter your code here' section.

This activity and the solution will be discussed at the Collaborate session this week. In the meantime, share and discuss your results in the 'Tutorials' discussion forum.

The solution will also be available here on Friday of this week by clicking on the 'Solution' tab in the top right corner.

The contents of the RMD file are also displayed below:

Packages

```
library(here)
library(dplyr)
library(readr)
# stats is loaded by default
library(psych)
library(GGally)
```

Challenges

Recall the dataset `pizza.csv`, containing nutritional data from a variety of pizza brands:

`brand` : Pizza brand (class label)

`id` : Sample analysed

`prot` : Amount of protein per 100 grams in the sample

`fat` : Amount of fat per 100 grams in the sample

`ash` : Amount of ash per 100 grams in the sample

`sodium` : Amount of sodium per 100 grams in the sample

`carb` : Amount of carbohydrates per 100 grams in the sample

`cal` : Amount of calories per 100 grams in the sample

We will, again, focus on brand A:

```
pizzaA <- read_csv(here("datasets","pizza.csv")) %>% filter(brand=="A") %>% select(-brand)
ggpairs(pizzaA)
```

Task 1: Perform the factor analysis on these data; what is the adequate number of factors?

Task 2: What do these factors appear to represent?

Topic 4: Overview of structural equation models

Structural equation modelling

Factor analysis (FA) is only one example of a new approach to data analysis which is **not based on individual observations**. We were not able to use the regression approach since the input factors were **latent** (not observable). There were too many unknowns. We went to analyse the covariance matrix Σ (and its estimator $\boldsymbol{S}$) which involved the actual parameters of interest—σ_i^2 and Λ. That is, we switched **from the level of individual observations** to analyse covariance matrices instead. There are a **series** of methods which are based on analysis of **covariances** rather than individual cases. Instead of minimising functions of observed and predicted **individual values**, we minimise the differences between **sample covariances and covariances predicted by the model**.

The fundamental hypothesis in these analyses is

$$H_0 : \Sigma = \Sigma(\boldsymbol{\theta}) \quad \text{against} \quad H_1 : \Sigma \neq \Sigma(\boldsymbol{\theta}).$$

Here Σ has $p(p+1)/2$ unknown elements (estimated by $\boldsymbol{S}$) **but these are assumed to be reproducible by just** $k = \dim(\boldsymbol{\theta}) < p(p+1)/2$ **parameters**. Note that more generally we could consider fitting **means and covariances, or means and covariances and higher moments** to a given structure. **Regression analysis with random inputs, simultaneous equations systems, confirmatory factor analysis, canonical correlations, (M)ANOVA** can be considered special cases.

Structural equation modelling is an important statistical tool in economics and behavioural sciences. Structural equations express relationships among several variables that can be either directly observed variables (manifest variables) or unobserved hypothetical variables (latent variables). In **structural models**, as opposed to **functional models**, all variables are taken to be **random** rather than having fixed levels. In addition, for maximum likelihood estimation and generalised least squares estimation (see next slide), the random variables are assumed to have an approximately multivariate normal distribution. Hence you are advised to remove outliers and consider transformations to normality before fitting.

General form of the model

$$\boldsymbol{\eta} = B\boldsymbol{\eta} + \Gamma\boldsymbol{\xi} + \boldsymbol{\zeta}. \tag{4.5}$$

Here,

$\boldsymbol{\eta} \in \mathbb{R}^m$ vector of output latent variables;
$\boldsymbol{\xi} \in \mathbb{R}^{n'}$ vector of input latent variables;
$B \in \mathcal{M}_{m,m}$, $\Gamma \in \mathcal{M}_{m,n'}$ coefficient matrices;

Note: $(I - B)$ is assumed to be nonsingular.

$\boldsymbol{\zeta} \in \mathbb{R}^m$ disturbance vector with $E\boldsymbol{\zeta} = 0$.
To this **modelling equation** (4.5) we attach two **measurement equations**:

$$\boldsymbol{Y} = \Lambda_{\boldsymbol{Y}}\boldsymbol{\eta} + \boldsymbol{\epsilon}; \tag{4.6}$$

$$\boldsymbol{X} = \Lambda_{\boldsymbol{X}}\boldsymbol{\xi} + \boldsymbol{\delta}; \tag{4.7}$$

$$\boldsymbol{Y} \in \mathbb{R}^p,\ \boldsymbol{X} \in \mathbb{R}^q;\ \Lambda_{\boldsymbol{Y}} \in m_{p\times m}, \Lambda_{\boldsymbol{X}} \in m_{q\times n'}$$

with $\boldsymbol{\epsilon} \in \mathbb{R}^p,\ \boldsymbol{\delta} \in \mathbb{R}^q$ zero-mean measurement errors. These errors are assumed to be uncorrelated with $\boldsymbol{\xi}$ and $\boldsymbol{\zeta}$ and with each other.

Generative model for X and Y

The above quite general model (4.5)–(4.6)–(4.7) is called **Keesling–Wiley–Jöreskog** model. Its interpretation is that the input and output latent variables $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$ are connected by a system of linear equations (the structural model (4.5) with coefficient matrices B and Γ and an error vector $\boldsymbol{\zeta}$. The random vectors $\boldsymbol{Y}$ and $\boldsymbol{X}$ represent the observable vectors (measurements).

The implied covariance matrix for this model can be obtained. Let

$$\mathrm{Var}(\boldsymbol{\xi}) = \Phi; \mathrm{Var}(\boldsymbol{\zeta}) = \Psi; \mathrm{Var}(\boldsymbol{\epsilon}) = \boldsymbol{\theta}_{\epsilon}; \mathrm{Var}(\boldsymbol{\delta}) = \boldsymbol{\theta}_{\boldsymbol{\delta}}.$$

Then,

$$\begin{aligned}\Sigma = \Sigma(\boldsymbol{\theta}) &= \begin{pmatrix} \sum_{\boldsymbol{YY}}(\boldsymbol{\theta}) & \Sigma_{\boldsymbol{YX}}(\boldsymbol{\theta}) \\ \Sigma_{\boldsymbol{XY}}(\boldsymbol{\theta}) & \Sigma_{\boldsymbol{XX}}(\boldsymbol{\theta}) \end{pmatrix} \\ &= \begin{pmatrix} \Lambda_{\boldsymbol{Y}}(I-B)^{-1}\left(\Gamma\Phi\Gamma^{\top}+\Psi\right)\left[(I-B)^{-1}\right]^{\top}\Lambda_{\boldsymbol{Y}}^{\top} + \boldsymbol{\theta}_{\boldsymbol{\epsilon}} & \Lambda_{\boldsymbol{Y}}(I-B)^{-1}\Gamma\Phi_{\boldsymbol{X}}^{\top} \\ \Lambda_{\boldsymbol{X}}\Phi\Gamma^{\top}\left[(I-B)^{-1}\right]^{\top}\Lambda_{\boldsymbol{Y}}^{\top} & \Lambda_{\boldsymbol{X}}\Phi\Lambda_{\boldsymbol{X}}^{\top} + \boldsymbol{\theta}_{\delta} \end{pmatrix}. \end{aligned} \tag{4.8}$$

Estimation

Under the normality assumption, we can use the MLE. Since the "data" is the estimated covariance matrix $\boldsymbol{S} = \frac{1}{n-1}\sum_{i=1}^{n}\left\{ \begin{pmatrix} \boldsymbol{Y}_i - \hat{\boldsymbol{Y}} \\ \boldsymbol{X}_i - \hat{\boldsymbol{X}} \end{pmatrix} \begin{pmatrix} \boldsymbol{Y}_i - \hat{\boldsymbol{Y}} \\ \boldsymbol{X}_i - \hat{\boldsymbol{X}} \end{pmatrix}^{\top} \right\}$, and since it is known that $(n-1)\boldsymbol{S} \sim W_{p+q}(n-1, \Sigma)$, we can utilise the form of the Wishart density to derive that

$$\log L(\boldsymbol{S}, \Sigma(\boldsymbol{\theta})) = \text{constant} - \frac{n-1}{2}\{\log|\Sigma(\boldsymbol{\theta})| + \text{tr}[\boldsymbol{S}\Sigma^{-1}(\boldsymbol{\theta})]\}$$

This is the function that has to be maximised. Hence, to find MLE, we minimise

$$F_{\text{ML}}(\boldsymbol{\theta}) = \log|\Sigma(\boldsymbol{\theta})| + \text{tr}[\boldsymbol{S}\Sigma^{-1}(\boldsymbol{\theta})] - \log|\boldsymbol{S}| - (p+q). \tag{4.9}$$

The function (4.9) has the advantage that F_{ML} would be zero for the "saturated model" (with $\hat{\Sigma} = \boldsymbol{S}$). I.e., a perfect fit is indicated by zero (and any non-perfect fit gives rise to > 0 value of F_{ML}).

Model evaluation

Under normality, model adequacy is mostly tested by an asymptotic χ^2-test.

Under $H_0 : \Sigma = \Sigma(\boldsymbol{\theta})$ versus $H_1 : \Sigma \neq \Sigma(\boldsymbol{\theta})$, the statistic to be used is $T = (n-1)F_{\mathrm{ML}}(\hat{\boldsymbol{\theta}}_{\mathrm{ML}})$ and under H_0, its asymptotic distribution is χ^2 with $\mathrm{df} = \frac{(p+q)(p+q+1)}{2} - \dim(\boldsymbol{\theta})$.

Reason:

$$\begin{aligned}\log L_0 =& \log L(\boldsymbol{S}, \hat{\Sigma}_{\mathrm{MLE}}) = \log L(\boldsymbol{S}, \Sigma(\hat{\theta}_{\mathrm{ML}})) \\ =& -\frac{n-1}{2}\left\{\log|\hat{\Sigma}_{\mathrm{MLE}}| + \mathrm{tr}[\boldsymbol{S}\hat{\Sigma}_{\mathrm{MLE}}^{-1}]\right\} + \mathrm{constant};\end{aligned}$$

$$\log L_1 = \log L(\boldsymbol{S}, \boldsymbol{S}) = -\frac{n-1}{2}\left\{\log|\boldsymbol{S}| + (p+q)\right\} + \mathrm{constant}.$$

Then,

$$\begin{aligned}-2\log\frac{L_0}{L_1} =& (n-1)\left\{\log|\hat{\Sigma}_{\mathrm{MLE}}| + \mathrm{tr}(\boldsymbol{S}\hat{\Sigma}_{\mathrm{MLE}}^{-1}) - \log|\boldsymbol{S}| - (p+q)\right\} \\ =& (n-1)F_{\mathrm{ML}}(\hat{\boldsymbol{\theta}}_{\mathrm{ML}}).\end{aligned}$$

Some particular SEM

From the general model (4.5)–(4.6)–(4.7), we can obtain the following particular models:

A) $\Lambda_{\boldsymbol{Y}} = I_m$, $\Lambda_{\boldsymbol{X}} = I_{n'}$; $p = m$; $q = n'$; $\boldsymbol{\theta_\epsilon} = 0$; $\boldsymbol{\theta_\delta} = 0 \implies \boldsymbol{Y} = B\boldsymbol{Y} + \Gamma\boldsymbol{X} + \boldsymbol{\zeta}$ (the classical econometric model).

B) $\Lambda_{\boldsymbol{Y}} = I_p$, $\Lambda_{\boldsymbol{X}} = I_q \implies$ The measurement error model:

- $\boldsymbol{\eta} = B\boldsymbol{\eta} + \Gamma\boldsymbol{\xi} + \boldsymbol{\zeta}$
- $\boldsymbol{Y} = \boldsymbol{\eta} + \boldsymbol{\epsilon}$
- $\boldsymbol{X} = \boldsymbol{\xi} + \boldsymbol{\delta}$

C) Factor Analysis Models: Just take the measurement part $\boldsymbol{X} = \Lambda_{\boldsymbol{X}}\boldsymbol{\xi} + \boldsymbol{\delta}$.

Relationship between exploratory and confirmatory FA

In EFA the number of latent variables is not determined in advance; further, the measurement errors are assumed to be uncorrelated. In CFA a model is constructed to a great extent **in advance**, the number of latent variables $\boldsymbol{\xi}$ is set by the analyst, whether a latent variable influences an observed variable is specified, some direct effects of latent on observed values are fixed to zero or some other constant (e.g., one), measurement errors $\boldsymbol{\delta}$ may correlate, the covariance of latent variables can be either estimated or set to any value. In practice, distinction between EFA and CFA is more blurred. For instance, researchers using traditional EFA procedures may restrict their analysis to a group of indicators that they believe are influenced by one factor. Or, researchers with poorly fitting models in CFA often modify their model in an exploratory way with the goal of improving fit.

Software available for fitting structural equation models

R

There are two packages for SEM in R: `lavaan` and `sem`. `sem` is an older package, whereas `lavaan` aims to provide an extensible framework for SEMs and their extensions:

- can mimic commercial packages (including those below)
- provides convenience functions for specifying simple special cases (such as CFA) but also a more flexible interface for advanced users
- mean structures and multiple groups
- different estimators and standard errors (including robust)
- handling of missing data
- linear and nonlinear equality and inequality constraints
- categorical data support
- multilevel SEMs
- package `blavaan` for Bayesian estimation

Demonstration: Structural equation

This demonstration can be completed using the provided RStudio environment or your own RStudio.

To complete this task select the 'SEM_Example.demo.Rmd' in the 'Files' section of RStudio. Follow the demonstration contained within the RMD file.

If you choose to complete the example in your own RStudio, upload the following file:

SEM_Example.demo.Rmd

The contents of the RMD file are also displayed below:

Packages

```
library(lavaan)
library(lavaanPlot)
```

Anomie

The following example is adapted with modifications from Lavaan Tutorial.

In 1977, Wheaton, Muthen, Alwin, and Summers used an structural equation model to model a longitudinal study that measured the sense of anomie ("a condition in which society provides little moral guidance to individuals") and a sense of powerlessness in respondents across two waves: 1967 and 1971.

Data

Measured (observed) variables on each respondent included:

`Anomie67` : A measurement on the sense of anomie scale in 1967.

`Anomie71` : A measurement on the sense of anomie scale in 1971.

`Powerlessness67` : A measurement on the sense of powerlessness scale in 1967.

`Powerlessness71` : A measurement on the sense of powerlessness scale in 1971.

`Education` : A measurement of educational attainment.

`SEI` : SocioEconomic Index: a measurement of economic well-being.

Package `lavaan` provides a convenience function, `getCov`, to load a variance-covariance matrix

from a character string containing its lower triangle and diagonal:

```
lower <- '
 11.834
  6.947   9.364
  6.819   5.091  12.532
  4.783   5.028   7.495   9.986
 -3.839  -3.889  -3.841  -3.625  9.610
-21.899 -18.831 -21.748 -18.775 35.522 450.288 '

wheaton.cov <-
    getCov(lower, names = c("Anomie67", "Powerless67",
                            "Anomie71", "Powerless71",
                            "Education", "SEI"))
wheaton.cov
```

Model specification

The SEM is specified as an R string with syntax similar to R formulas but with special operations: `=~` for defining a latent variable, `~` for defining a regression (possibly among latent variables), and `~~` for defining correlation. The lines of model below can be read as follows (using the notation of the lecture notes).

- Output latent variable $\eta_1 =$ `Alien67` (sense of Alienation in 1967) is measured by output observed variables $Y_1 =$ `Anomie67` and $Y_2 =$ `Powerless67`, with coefficients specified to be 1.0 and 0.833. Note that this does not mean that they exactly equal, since there are also the noise terms ϵ_1 and ϵ_2.
- Similarly, output latent variable $\eta_2 =$ `Alien71` (sense of Alienation in 1971) is measured by output observed variables $Y_3 =$ `Anomie71` and $Y_4 =$ `Powerless71`, with coefficients again specified to be 1.0 and 0.833.
- Input latent variable `SES` (SocioEconomic Status) is measured by input observed variables `Education` and `SEI`. That is, $\xi_1 =$ `SES`, $X_1 =$ `Education`, $X_2 =$ `SEI`, and we also "lock" the coefficient between `Education` and `SES`, $\Lambda_{X,1,1} = 1$. We lock this coefficient to identify the variable, since one can get the same model by scaling the variable and inverse-scaling the coefficients. The coefficient $\Lambda_{X,2,1}$ is free. (Note that we could have given it a name or "locked" it with another coefficient by putting `<NAME>*` in front of `SES`.)
- Alienation in 1967 is a linear function of socioeconomic status (and error). (That is, $\Gamma_{1,1}$ is a free parameter.)
- Alienation in 1971 is a linear function of alienation in 1967 and socioeconomic status (and error). (That is, $\Gamma_{2,1}$ and $B_{1,2}$ are free parameters, and other elements of B are fixed at 0.)
- The residual variance of anomie in 1967 is the same as that in 1971. (That is, ϵ_1 and ϵ_3 have the same variance, which we call `theta1`.)
- Similarly, the residual variance of powerlessness in 1967 is the same as that in 1971. (That is,

ϵ_2 and ϵ_4 have the same variance, which we call `theta2`.)

- Furthermore, there is a residual correlation between a person's anomie measurements in 1967 and 1971 (respectively, ϵ_1 and ϵ_3). We call it `theta3`.
- `theta3` is also the residual correlation between a person's powerlessness measurements in 1967 and 1971 (respectively, ϵ_2 and ϵ_4).

```
wheaton.model <- '
  # latent variables
    Alien67 =~ 1.0*Anomie67 + 0.833*Powerless67 # 1.
    Alien71 =~ 1.0*Anomie71 + 0.833*Powerless71 # 2.
    SES     =~ 1.0*Education + SEI # 3.
  # regressions
    Alien67 ~ SES # 4.
    Alien71 ~ Alien67 + SES # 5.
  # consrained variances
    Anomie67 ~~ theta1*Anomie67 # 6.
    Anomie71 ~~ theta1*Anomie71 # 6.
    Powerless67 ~~ theta2*Powerless67 # 7.
    Powerless71 ~~ theta2*Powerless71 # 7.
  # correlated residuals
    Anomie67 ~~ theta3*Anomie71 # 8.
    Powerless67 ~~ theta3*Powerless71 # 9.
'
```

Estimation

We can now fit the model. Before looking at the results, let's visualise it and confirm that the structure it postulates makes sense:

```
fit <- sem(wheaton.model,
           sample.cov = wheaton.cov,
           sample.nobs = 932)
lavaanPlot(model=fit, covs=TRUE)
```

Note: As of 1 September 2020, there appears to be a bug in `lavaanPlot` that prevents it from plotting correlations between observed variables even if `covs=TRUE`.

Interpretation

Finally, let's take a look at the results with fitted coefficients:

```
summary(fit, standardized = TRUE)
lavaanPlot(model=fit, coefs=TRUE, covs=TRUE)
```

Firstly, note the model's lack-of-fit test. With $\chi^2 = 13.5$ with $\mathrm{df} = 9$, $p\text{-value} = 0.141$: there is not

sufficient evidence to believe that the data came from a different model from the one fit.

Some conclusions assuming causation follow.

Latent variables:

- Higher SES, constructed to positively affect educational attainment, turns out to positively affect the SEI.

Regressions:

- Higher SES strongly reduces alienation in both years.
- Higher alienation in 1967 results in higher alienation in 1971.

Covariances:

- After accounting for the above effects and other effects in the model (e.g., the fixed relationships between Alienation and Anomie and Powerlessness), an individual's residual anomie in 1967 is strongly positively correlated with their anomie in 1971 and analogously with powerlessness.

Topic 1: Concepts of classification

Welcome to Week 5

Dr Pavel Krivitsky gives you a brief overview of topics and concepts we'll be covering in this week.

Transcript

Weekly learning outcomes

- Define, calculate for a classifier, and interpret the basic measures of classification, including confusion matrices, true/false positive/negatives, expected cost of misclassification, and related concepts.
- Define various optimal classification rules.
- Use linear and quadratic discriminant analysis to classify observations.
- Use hypothesis testing to determine whether the assumptions of linear discriminant analysis are satisfied.
- Fit, tune, and assess a support vector machine to a specified dataset and use it to predict new observations.
- Explain the assumptions underlying the above inferential procedures and check them.

Topics we will cover are:

- Topic 1: Concepts of classification
- Topic 2: Linear discriminant analysis
- Topic 3: Quadratic discriminant analysis
- Topic 4: Support vector machines concepts and overview of estimation
- Topic 5: Tuning support vector machines

Optional readings

An alternative presentation of the concepts for this week can be found in:

Johnson, R. A., & Wichern, D. (2008). *Applied Multivariate Statistical Analysis* (6th ed.). Pearson Prentice Hall.

- 11.1–11.6.

All readings are available from the course Leganto reading list. Please keep in mind that you will need to be logged into Moodle to access the Leganto reading list.

Questions about this week's topics?

This week's topics were prepared by Dr P. Krivitsky. If you have any questions or comments, please post them under Discussion or email directly: p.krivitsky@unsw.edu.au

Discrimination and classification

Introduction: Separation and classification for two populations

Discriminant analysis and classification are widely used in multivariate techniques. The goal is either *separating sets of objects* (in discriminant analysis terminology) or *allocating new objects to given groups* (in classification theory terminology).

Basically, discriminant analysis is more exploratory in nature than classification. However, the difference is not significant especially because very often a function that separates may sometimes serve as an allocator, and, conversely, a rule of allocation may suggest a discriminatory procedure. In practice, the goals in the two procedures often overlap.

We will consider the case of two populations (classes of objects) first. Typical examples include: an anthropologist wants to classify a skull as a male or female; a patient needs to be classified as needing surgery or not needing surgery etc.

Denote the two classes by π_1 and π_2. The separation is to be performed on the basis of measurements of p associated random variables that form a vector $\boldsymbol{X} \in \mathbb{R}^p$. The observed values of $\boldsymbol{X}$ belong to different distributions when taken from π_1 and π_2 and we shall denote the densities of these two distributions by $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$, respectively.

Allocation or classification is possible due to the fact that one has a *learning sample* at hand, i.e., there are some measurement vectors that are known to have been generated from each of the two populations. These measurements have been generated in earlier similar experiments. The goal is to partition the sample space into 2 mutually exclusive regions, say R_1 and R_2, such that if a *new* observation falls in R_1, it is allocated to π_1 and if it falls in R_2, it is allocated to π_2.

Classification errors

There is always a chance of an erroneous classification (misclassification). Our goal will be to develop such classification methods that in a suitably defined sense minimise the chances of misclassification.

It should be noted that one of the two classes may have a greater likelihood of occurrence because one of the two populations might be much larger than the other. For example, there tend to be a lot more financially sound companies than bankrupt companies. These *prior probabilities* of occurrence should also be taken into account when constructing an optimal classification rule if we want to perform optimally.

In a more detailed study of optimal classification rules, cost is also important. If classifying a π_1 object to the class π_2 represents a much more serious error than classifying a π_2 object to the class π_1 then these cost differences should also be taken into account when designing the optimal rule.

The **conditional** probabilities for misclassification are defined naturally as:

$$\Pr(2|1) = \Pr(\boldsymbol{X} \in R_2|\pi_1) = \int_{R_2} f_1(\boldsymbol{x})d\boldsymbol{x} \tag{5.1}$$

$$\Pr(1|2) = \Pr(\boldsymbol{X} \in R_1|\pi_2) = \int_{R_1} f_2(\boldsymbol{x})d\boldsymbol{x} \tag{5.2}$$

Summarising

We turn briefly to the question of how to summarise a classifier's performance. Each object has a true class membership and the one predicted by the classifier, and for a given dataset for which true memberships are known, we may summarise the counts of the four resulting possibilities in a contingency table called a *confusion matrix*, i.e.,

A confusion matrix can be produced when there are more than two classes as well.

In the special case where there are two classes that can be meaningfully labelled as Negative/Positive, False/True, No/Yes, Null/Alternative, or similar, it is common to use the following terminology for them:

One can then define various performance metrics such as

sensitivity (a.k.a. recall, true positive rate (TPR)): $\Pr(\text{Predicted positive}|\text{Actual positive}) = \frac{\text{TP}}{\text{TP}+\text{FN}}$

specificity (a.k.a. selectivity, true negative rate (TNR)):

$\Pr(\text{Predicted negative}|\text{Actual negative}) = \frac{\text{TN}}{\text{TN}+\text{FP}}$

false positive rate (a.k.a. (FPR), fall-out): $\Pr(\text{Predicted positive}|\text{Actual negative}) = \frac{\text{FP}}{\text{TN}+\text{FP}} = 1 - \text{TNR}$

precision (a.k.a. positive predictive value): $\Pr(\text{Actual positive}|\text{Predicted positive}) = \frac{\text{TP}}{\text{TP}+\text{FP}}$

negative predictive value: $\Pr(\text{Actual negative}|\text{Predicted negative}) = \frac{\text{TN}}{\text{TN}+\text{FN}}$

F1 score: $\frac{2\text{TP}}{2\text{TP}+\text{FP}+\text{FN}}$

Many classifiers return a continuous score that needs to be thresholded to produce a binary decision (e.g., predict "Yes" if the score exceeds some constant k and "No" otherwise), it is a common practice to plot a *receiver operating characteristic* (ROC) curve by varying the threshold and then plotting the TPR (on the vertical axis) against FPR (on the horizontal axis) that result. Both of which decrease as k increases. A perfect classifier would have a threshold for which the curve achieves the $(0, 1)$ point, whereas classifier close to the $y = x$ line is no better than chance.

Optimal classification rules

Rules that minimise the expected cost of misclassification (ECM)

Lemma 5.1. *Denote by p_i the **prior** probability of $\pi_i, i = 1, 2, p_1 + p_2 = 1$. Then the **overall** probabilities of incorrectly classifying objects will be:* $\Pr(\text{misclassified as } \pi_1) = \Pr(1|2)p_2$ *and* $\Pr(\text{misclassified as } \pi_2) = \Pr(2|1)p_1$. *Further, let $c(i|j), i \neq j, i, j = 1, 2$ be the misclassification costs. Then the **expected cost of misclassification** is*

$$ECM = c(2|1)\Pr(2|1)p_1 + c(1|2)\Pr(1|2)p_2 \tag{5.3}$$

The regions R_1 and R_2 that minimise ECM are given by

$$R_1 = \left\{ x : \frac{f_1(\boldsymbol{x})}{f_2(\boldsymbol{x})} \geq \frac{c(1|2)}{c(2|1)} \frac{p_2}{p_1} \right\} \tag{5.4}$$

and

$$R_2 = \left\{ x : \frac{f_1(\boldsymbol{x})}{f_2(\boldsymbol{x})} < \frac{c(1|2)}{c(2|1)} \frac{p_2}{p_1} \right\}. \tag{5.5}$$

Proof: It is easy to see that $\text{ECM} = \int_{R_1} \left[c(1|2)p_2 f_2(\boldsymbol{x}) - c(2|1)p_1 f_1(\boldsymbol{x}) \right] d\boldsymbol{x} + c(2|1)p_1$. Hence, the ECM will be minimised if R_1 includes those values of $\boldsymbol{x}$ for which the integrand $\left[c(1|2)p_2 f_2(\boldsymbol{x}) - c(2|1)p_1 f_1(\boldsymbol{x}) \right] \leq 0$ and excludes all the complementary values.

Note, the significance of the fact that in Lemma 5.1 **only ratios** are involved. Often in practice, one would have a much clearer idea about the cost ratio rather than for the actual costs themselves.

For your own exercise, consider the partial cases of Lemma 5.1 when $p_2 = p_1$, $c(1|2) = c(2|1)$ and when both these equalities hold. Comment on the soundness of the classification regions in these cases.

Rules that minimise the total probability of misclassification (TPM)

If we ignore the cost of misclassification, we can define the total probability of misclassification as

$$\text{TPM} = p_1 \int_{R_2} f_1(\boldsymbol{x}) d\boldsymbol{x} + p_2 \int_{R_1} f_2(\boldsymbol{x}) d\boldsymbol{x}$$

Mathematically, this is a particular case of Lemma 5.1 when the costs of misclassification are equal—so nothing new here.

Bayesian approach

Here, we try to allocate a new observation $\boldsymbol{x}_0$ to the population with the larger posterior probability $\Pr(\pi_i|\boldsymbol{x}_0), i = 1, 2$. According to Bayes's formula we have

$$\Pr(\pi_1|\boldsymbol{x}_0) = \frac{p_1 f_1(\boldsymbol{x}_0)}{p_1 f_1(\boldsymbol{x}_0) + p_2 f_2(\boldsymbol{x}_0)}, \ \Pr(\pi_2|\boldsymbol{x}_0) = \frac{p_2 f_2(\boldsymbol{x}_0)}{p_1 f_1(\boldsymbol{x}_0) + p_2 f_2(\boldsymbol{x}_0)}$$

Mathematically, the strategy of classifying an observation $\boldsymbol{x}_0$ as π_1 if $\Pr(\pi_1|\boldsymbol{x}_0) > \Pr(\pi_2|\boldsymbol{x}_0)$ is again a particular case of Lemma 5.1 when the costs of misclassification are equal. But note that the calculation of the posterior probabilities themselves is in itself a useful and informative operation.

Topic 2: Linear discriminant analysis

Classification with two multivariate normal populations

Until now we did not specify any particular form of the densities $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$. Essential simplification occurs under normality assumption and we are going over to a more detailed discussion of this particular case now. Two different cases will be considered- of equal and of non-equal covariance matrices.

Case of equal covariance matrices $\Sigma_1 = \Sigma_2 = \Sigma$

Now we assume that the two populations π_1 and π_2 are $N_p(\boldsymbol{\mu}_1, \Sigma)$ and $N_p(\boldsymbol{\mu}_2, \Sigma)$, respectively. Then, (5.4) becomes

$$R_1 = \left\{ \boldsymbol{x} : \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_1)^\top \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1) + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_2)^\top \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_2) \right] \geq \frac{c(1|2)}{c(2|1)} \times \frac{p_2}{p_1} \right\}.$$

Similarly, from (5.5) we get

$$R_2 = \left\{ \boldsymbol{x} : \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_1)^\top \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1) + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_2)^\top \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_2) \right] < \frac{c(1|2)}{c(2|1)} \times \frac{p_2}{p_1} \right\},$$

and we arrive at the following result

Theorem 5.1. *Under the above assumptions, the allocation rule that minimises the ECM is given by:*

1. *Allocate $\boldsymbol{x}_0$ to π_1 if*

$$(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top \Sigma^{-1}\boldsymbol{x}_0 - \frac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top \Sigma^{-1}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2) \geq \log\left[\frac{c(1|2)}{c(2|1)} \times \frac{p_2}{p_1} \right].$$

2. *Otherwise, allocate $\boldsymbol{x}_0$ to π_2.*

Note, also that it is unrealistic to assume in most situations that the parameters $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$, and Σ are known. They will need to be estimated by the data instead. Assume, n_1 and n_2 observations are available from the first and from the second population, respectively. If $\bar{\boldsymbol{x}}_1$ and $\bar{\boldsymbol{x}}_2$ are the sample mean vectors and $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ the corresponding sample covariance matrices, then under the assumption of $\Sigma_1 = \Sigma_2 = \Sigma$ we can derive the pooled covariance matrix estimator $\boldsymbol{S}_{\text{pooled}} = \frac{(n_1-1)\boldsymbol{S}_1 + (n_2-1)\boldsymbol{S}_2}{n_1+n_2-2}$ (This is an unbiased estimator of Σ (!)).

Hence the *Sample classification rule* becomes:

1. Allocate $\boldsymbol{x}_0$ to π_1 if

$$(\bar{\boldsymbol{x}}_1 - \bar{\boldsymbol{x}}_2)^\top \boldsymbol{S}_{\text{pooled}}^{-1} \boldsymbol{x}_0 - \frac{1}{2}(\bar{\boldsymbol{x}}_1 - \bar{\boldsymbol{x}}_2)^\top \boldsymbol{S}_{\text{pooled}}^{-1}(\bar{\boldsymbol{x}}_1 + \bar{\boldsymbol{x}}_2) \geq \log\left[\frac{c(1|2)}{c(2|1)} \times \frac{p_2}{p_1}\right] \quad (5.6)$$

2. Otherwise, allocate $\boldsymbol{x}_0$ to π_2.

This empirical classification rule is called **an allocation rule based on Fisher's discriminant function**. The function

$$(\bar{\boldsymbol{x}}_1 - \bar{\boldsymbol{x}}_2)^\top \boldsymbol{S}_{\text{pooled}}^{-1} \boldsymbol{x}_0 - \frac{1}{2}(\bar{\boldsymbol{x}}_1 - \bar{\boldsymbol{x}}_2)^\top \boldsymbol{S}_{\text{pooled}}^{-1}(\bar{\boldsymbol{x}}_1 + \bar{\boldsymbol{x}}_2)$$

itself (which is linear in the vector observation $\boldsymbol{x}_0$) is called **Fisher's linear discriminant function.**

Of course, the latter rule is only an *estimate* of the optimal rule since the parameters in the latter have been replaced by estimated quantities. But we are expecting this rule to perform well when n_1 and n_2 are large. It is to be pointed out that the allocation rule in (5.6) is **linear** in the new observation $\boldsymbol{x}_0$. The simplicity of its form is a consequence of the multivariate normality assumption.

Optimum error rate and Mahalanobis distance

We defined the TPM quantity in general terms for any classification in (5.3). When the regions R_1 and R_2 are selected in an optimal way, one obtains the minimal value of TPM which is called **optimum error rate (OER)** and is being used to characterise the difficulty of the classification problem at hand. Hereby we shall illustrate the calculation of the OER for the simple case of two normal populations with $\Sigma_1 = \Sigma_2 = \Sigma$ and prior probabilities $p_1 = p_2 = \frac{1}{2}$. In this case

$$\mathrm{TPM} = \frac{1}{2}\int_{R_2} f_1(\boldsymbol{x})d\boldsymbol{x} + \frac{1}{2}\int_{R_1} f_2(\boldsymbol{x})d\boldsymbol{x},$$

and OER is obtained by choosing

$$R_1 = \left\{\boldsymbol{x} : (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top \Sigma^{-1}\boldsymbol{x} - \frac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top \Sigma^{-1}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2) \geq 0\right\}$$

and

$$R_2 = \left\{\boldsymbol{x} : (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top \Sigma^{-1}\boldsymbol{x} - \frac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top \Sigma^{-1}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2) < 0\right\}.$$

If we introduce the random variable $Y = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top \Sigma^{-1}\boldsymbol{X} = \boldsymbol{l}^\top \boldsymbol{X}$ then $Y|i \sim N_1(\mu_{iY}, \Delta^2), i = 1, 2$ for the two populations π_1 and π_2 where $\mu_{iY} = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top \Sigma^{-1}\boldsymbol{\mu}_i, i = 1, 2$. The quantity $\Delta = \sqrt{(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top \Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)}$ is the **Mahalanobis distance** between the two normal populations and it has an important role in many applications of Multivariate Analysis. Now

$$\Pr(2|1) = \Pr\left(Y < \frac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top \Sigma^{-1}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)\right) = \Pr\left(\frac{Y - \mu_{1Y}}{\Delta} < -\frac{\Delta}{2}\right) = \Phi\left(-\frac{\Delta}{2}\right),$$

$\Phi(\cdot)$ denoting the cumulative distribution function of the standard normal. Along the same lines we can get (**do it (!)**) : $\Pr(1|2) = \Phi\left(-\frac{\Delta}{2}\right)$ to that finally $\mathrm{OER} = \mathrm{minim}\ \mathrm{TPM} = \Phi\left(-\frac{\Delta}{2}\right)$.

In practice, Δ is replaced by its estimated value $\hat{\Delta} = \sqrt{\bar{\boldsymbol{x}}_1 - \bar{\boldsymbol{x}}_2)^\top \boldsymbol{S}_{\mathrm{pooled}}^{-1}(\bar{\boldsymbol{x}}_1 - \bar{\boldsymbol{x}}_2)}$.

Classification with more than 2 normal populations

Formal generalisation of the theory for the case of $g > 2$ groups $\pi_1, \pi_2, \ldots, \pi_g$ is straightforward but optimal error rate analysis is difficult when $g > 2$. It is easy to see that the ECM classification rule with **equal** misclassification costs (compare to (5.4) and (5.5)) becomes now:

1. Allocate $\boldsymbol{x}_0$ to π_k if $p_k f_k > p_i f_i$ for all $i \neq k$.

Equivalently, one can check if $\log p_k f_k > \log p_i f_i$ for all $i \neq k$.

When applying this classification rule to g normal populations $f_i(\boldsymbol{x}) \sim N_p(\boldsymbol{\mu}_i, \Sigma_i), i = 1, 2, \ldots, g$ it becomes:

1. Allocate $\boldsymbol{x}_0$ to π_k if

$$\begin{aligned} \log p_k f_k(\boldsymbol{x}_0) &= \log p_k - \frac{p}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(\boldsymbol{x}_0 - \boldsymbol{\mu}_k)^\top \Sigma_k^{-1}(\boldsymbol{x}_0 - \boldsymbol{\mu}_k) \\ &= \max_i \log p_i f_i(\boldsymbol{x}_0). \end{aligned}$$

Ignoring the constant $\frac{p}{2}\log(2\pi)$ we get the **quadratic discriminant score for the ith population:**

$$d_i^Q(\boldsymbol{x}) = -\frac{1}{2}\log|\Sigma_i| - \frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_i)^\top \Sigma_i^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_i) + \log p_i \tag{5.7}$$

and the rule advocates to allocate $\boldsymbol{x}$ to the population with a largest quadratic discriminant score. It is obvious how one would estimate from the data the unknown quantities involved in (5.7) in order to obtain the *estimated* minimum total probability of misclassification rule. (You should formulate the precise statement (!)).

In the case we are justified to assume that **all covariance matrices** for the g populations are equal, a simplification is possible (like in the case $g = 2$). Looking only at the terms that vary with $i = 1, 2, \ldots, g$ in (5.7) we can define the **linear discriminant score:** $d_i(\boldsymbol{x}) = \boldsymbol{\mu}_i^\top \Sigma^{-1}\boldsymbol{x} - \frac{1}{2}\boldsymbol{\mu}_i^\top \Sigma^{-1}\boldsymbol{\mu}_i + \log p_i$. Correspondingly, a **sample version** of the linear discriminant score is obtained by substituting the arithmetic means $\bar{\boldsymbol{x}}_i$ instead of $\boldsymbol{\mu}_i$ and $\boldsymbol{S}\text{pooled} = \frac{n_1 - 1}{n_1 + n_2 + \ldots n_g - g}\boldsymbol{S}_1 + \cdots + \frac{n_g - 1}{n_1 + n_2 + \cdots + n_g - g}\boldsymbol{S}_g$ instead of Σ thus arriving at

$$\hat{d}_i(\boldsymbol{x}) = \bar{\boldsymbol{x}}_i^\top \boldsymbol{S}_{\text{pooled}}^{-1}\boldsymbol{x} - \frac{1}{2}\bar{\boldsymbol{x}}_i^\top \boldsymbol{S}_{\text{pooled}}^{-1}\bar{\boldsymbol{x}}_i + \log p_i$$

Therefore the **Estimated Minimum TPM Rule for Equal Covariance Normal Populations** is the following:

1. Allocate $\boldsymbol{x}$ to π_k if $\hat{d}_k(\boldsymbol{x})$ is the largest of the g values $\hat{d}_i(\boldsymbol{x}), i = 1, 2, \ldots, g$.

In this form, the classification rule has been implemented in many computer packages.

In software

R: `MASS:lda`, `MASS:qda`

Check your understanding

Complete the below exercises to check your understanding of concepts presented so far.

Three bivariate normal populations, labelled $i = 1, 2, 3$ have same covariance matrix given by $\Sigma = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$ and means $\boldsymbol{\mu}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \boldsymbol{\mu}_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \boldsymbol{\mu}_3 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ respectively.

a) Suggest a classification rule for an observation $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ that corresponds to one of the three populations. You may assume equal priors for the three populations and equal misclassification costs.

b) Classify the following observations to one of the three distributions:

$$\begin{pmatrix} 0.2 \\ 0.6 \end{pmatrix}, \begin{pmatrix} 2 \\ 0.8 \end{pmatrix}, \begin{pmatrix} 0.75 \\ 1 \end{pmatrix}$$

c) Show that in $\mathbb{R}^2$, the 3 classification regions are bounded by straight lines and draw a graph of these three regions.

Topic 3: Quadratic discriminant analysis

Case of different covariance matrices

Case of different covariance matrices ($\Sigma_1 \neq \Sigma_2$)

Theorem 5.2. *Now we assume that the two populations π_1 and π_2 are $N_p(\boldsymbol{\mu}_1, \Sigma_1)$ and $N_p(\boldsymbol{\mu}_2, \Sigma_2)$, respectively. Repeating the same steps as in Theorem 5.1 we get*

$$R_1 = \left\{ \boldsymbol{x} : -\frac{1}{2}\boldsymbol{x}^\top(\Sigma_1^{-1} - \Sigma_2^{-1})\boldsymbol{x} + (\boldsymbol{\mu}_1^\top\Sigma_1^{-1} - \boldsymbol{\mu}_2^\top\Sigma_2^{-1})\boldsymbol{x} - k \geq \log\left[\frac{c(1|2)}{c(2|1)} \times \frac{p_2}{p_1}\right] \right\}$$

$$R_2 = \left\{ \boldsymbol{x} : -\frac{1}{2}\boldsymbol{x}^\top(\Sigma_1^{-1} - \Sigma_2^{-1})\boldsymbol{x} + (\boldsymbol{\mu}_1^\top\Sigma_1^{-1} - \boldsymbol{\mu}_2^\top\Sigma_2^{-1})\boldsymbol{x} - k < \log\left[\frac{c(1|2)}{c(2|1)} \times \frac{p_2}{p_1}\right] \right\}$$

where $k = \frac{1}{2}\log(\frac{|\Sigma_1|}{|\Sigma_2|}) + \frac{1}{2}(\boldsymbol{\mu}_1^\top\Sigma_1^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2^\top\Sigma_2^{-1}\boldsymbol{\mu}_2)$ and we see that the classification regions are ***quadratic*** *functions of the new observation in this case. One obtains the following rule:*

1. *Allocate $\boldsymbol{x}_0$ to π_1 if*
   $\frac{1}{2}\boldsymbol{x}_0^\top(\boldsymbol{S}_1^{-1} - \boldsymbol{S}_2^{-1})\boldsymbol{x}_0 + (\bar{\boldsymbol{x}}_1^\top\boldsymbol{S}_1^{-1} - \bar{\boldsymbol{x}}_2^\top\boldsymbol{S}_2^{-1})\boldsymbol{x}_0 - \hat{k} \geq \log\left[\frac{c(1|2)}{c(2|1)} \times \frac{p_2}{p_1}\right]$
   where $\hat{k}$ is the empirical analog of k.
2. *Allocate $\boldsymbol{x}_0$ to π_2 otherwise.*

When $\Sigma_1 = \Sigma_2$, the quadratic term disappears and we can easily see that the classification regions from Theorem 5.1 are obtained. Of course, the case considered in Theorem 5.2 is more general but we should be cautious when applying it in practice. It turns out that in more than two dimensions, classification rules based on quadratic functions do not always perform nicely and can lead to strange results. This is especially true when the data are not quite normal and when the differences in the covariance matrices are significant. The rule is very sensitive (non-robust) towards departures from normality. Therefore, it is advisable to try to first transform the data to more nearly normal by using some classical normality transformations. Also, tests discussed in Topic 1 of Week 4 can be used to check if equal variance assumption is valid.

Demonstration: Discriminant analysis

This demonstration can be completed using the provided RStudio or your own RStudio.

To complete this task select the 'Discrim_Examples.demo.Rmd' in the 'Files' section of RStudio. Follow the demonstration contained within the RMD file.

If you choose to complete the example in your own RStudio, upload the following file:

Discrim_Examples.demo.Rmd

The contents of the RMD file are also displayed below:

Packages

```
library(here)
library(readr)
library(MASS) # lda(), qda()
library(dplyr) # Also, we want select() from here and not MASS, so we load it second.
library(GGally)
library(heplots) # boxM
```

Iris Example

The famous iris dataset contains four measurements on 150 flowers from three subspecies of iris: setosa, versicolor, and virginica. We will use it to illustrate linear and quadratic discriminant analysis.

Data

This dataset is included with R. Let's load and visualise it.

```
data(iris)
iris
ggpairs(iris, aes(colour=Species, alpha = 0.3))
```

Linear discriminant analysis

Suppose that we are prepared to assume that the three species have equal variances and covariances among the variables. Then, we can fit the linear discriminant analysis with the `lda()` function:

```
(iris.lda <- lda(Species~., data=iris))
```

Note that the output format used by `lda` is somewhat different from the one we had derived in lecture. This is because it is optimised for rapidly classifying large numbers of observations.

To illustrate the structure of classification, let's fit LDA based only on two variables: sepal length and sepal width. These are not the "best" variables for this, and this is deliberate.

```
# Fit the LDA with only two predictors.
(iris.lda2 <- lda(Species~Sepal.Length+Sepal.Width, data=iris))

# Generate a grid of values covering the data.
SLs <- seq(min(iris$Sepal.Length), max(iris$Sepal.Length), length.out=200)
SWs <- seq(min(iris$Sepal.Width), max(iris$Sepal.Width), length.out=200)
xy <- expand.grid(Sepal.Length=SLs, Sepal.Width=SWs)

# Make the predictions:
z <- factor(predict(iris.lda2, newdata=xy)$class) # See ? predict.lda() for structure.

plot(xy[,1],xy[,2], col=z, xlab="Sepal Length",ylab="Sepal Width", pch=".")
points(iris$Sepal.Length,iris$Sepal.Width, col=factor(iris$Species), pch=15) # pch=15
-> Filled squares
```

We see that the region boundaries are linear.

But, of course, we should check that the variances are, in fact, equal. We learned how to do that last week:

```
select(iris, -Species) %>% boxM(iris$Species)
```

The test confirms that they are different, though we could have also just looked at the plots. This does not mean that we cannot use LDA: it may still do a perfectly good job separating the groups. None the less, let's now consider QDA.

Quadratic discriminant analysis

Let's begin by taking a look at the classification regions:

```
# Fit the QDA with only two predictors.
(iris.qda2 <- qda(Species~Sepal.Length+Sepal.Width, data=iris))

# The grid is the same as before. Make the predictions:
z <- factor(predict(iris.qda2, newdata=xy)$class) # See ? predict.qda() for structure.

plot(xy[,1],xy[,2], col=z, xlab="Sepal Length",ylab="Sepal Width", pch=".")
points(iris$Sepal.Length,iris$Sepal.Width, col=factor(iris$Species), pch=15) # pch=15
-> Filled squares
```

The boundaries are now curved. Let's zoom out:

```
# Generate a grid of values covering the data.
```

```
SLs <- seq(2, 10, length.out=200)
SWs <- seq(0, 6, length.out=200)
xy <- expand.grid(Sepal.Length=SLs, Sepal.Width=SWs)

# Make the predictions:
z <- factor(predict(iris.qda2, newdata=xy)$class) # See ? predict.lda() for structure.

plot(xy[,1],xy[,2], col=z, xlab="Sepal Length",ylab="Sepal Width", pch=".")
points(iris$Sepal.Length,iris$Sepal.Width, col=factor(iris$Species), pch=15) # pch=15
-> Filled squares
```

As you can see, the regions---particularly outside of the range of the data---can have weird shapes.

Next, let's fit using all the predictors:

```
(iris.qda <- qda(Species~., data=iris))
```

Comparing classifiers

Confusion matrices

Lastly, let's do some quick comparisons of these two classifiers. The simplest one is the confusion matrix: we obtain it by cross-tabulating the observed classes against the predicted. This can be done easily with the `table` function:

```
# LDA (all 4)
table(truth=iris$Species, prediction=predict(iris.lda)$class)
# QDA (all 4)
table(truth=iris$Species, prediction=predict(iris.qda)$class)
# LDA (Sepal only)
table(truth=iris$Species, prediction=predict(iris.lda2)$class)
# QDA (Sepal only)
table(truth=iris$Species, prediction=predict(iris.qda2)$class)
```

As you can see, the two behave about equally well.

Cross-validation

This naive form of evaluation has the disadvantage that the same data are used to train the classifier as to test it. To get around this, we can use *cross-validation*: each observation in the dataset gets a turn being the predicted value based on the rest of the dataset. We can enable it using `CV=TRUE` option:

```
# LDA (all 4)
table(truth=iris$Species, prediction=lda(Species~.,data=iris,CV=TRUE)$class)
# QDA (all 4)
table(truth=iris$Species, prediction=qda(Species~.,data=iris,CV=TRUE)$class)
```

```
# LDA (Sepal only)
table(truth=iris$Species,
prediction=lda(Species~Sepal.Length+Sepal.Width,data=iris,CV=TRUE)$class)
# QDA (Sepal only)
table(truth=iris$Species,
prediction=qda(Species~Sepal.Length+Sepal.Width,data=iris,CV=TRUE)$class)
```

We see that LDA is slightly better than QDA in both cases. This is probably because the separating boundaries are close to linear.

Challenge: Discriminant analysis

If you choose to complete this task in your own RStudio, upload the following file:

Discrim_Examples.challenge.Rmd

Click on the 'Discrim_Examples.challenge.Rmd' in the 'Files' section to begin. Enter your response to the tasks in the 'Enter your code here' section.

This activity and the solution will be discussed at the Collaborate session this week. In the meantime, share and discuss your results in the 'Tutorials' discussion forum.

The solution will also be available here on Friday of this week by clicking on the 'Solution' tab in the top right corner.

The contents of the RMD file are also displayed below:

Packages

```
library(here)
library(readr)
library(MASS) # lda(), qda()
library(dplyr) # Also, we want select() from here and not MASS, so we load it second.
library(GGally)
library(heplots) # boxM
```

Challenges

Recall the dataset `pizza.csv`, containing nutritional data from a variety of pizza brands:

`brand` : Pizza brand (class label)

`id` : Sample analysed

`prot` : Amount of protein per 100 grams in the sample

`fat` : Amount of fat per 100 grams in the sample

`ash` : Amount of ash per 100 grams in the sample

`sodium` : Amount of sodium per 100 grams in the sample

`carb` : Amount of carbohydrates per 100 grams in the sample

`cal` : Amount of calories per 100 grams in the sample

```
pizza <- read_csv(here("datasets","pizza.csv"))
ggpairs(pizza, mapping=aes(col=brand, alpha=0.3))
```

Task 1: Compare LDA and QDA for classifying the pizza brand based on measurements using in-sample predictions (i.e., not cross-validated). Which looks better?

Hint: the error rate is `1-sum(diag(CM))` where `CM` is the confusion matrix.

Task 2: Now compare LDA and QDA using cross-validation. How does your answer change?

Task 3: Use your selected classifier to classify a pizza that has the following properties. Obtain predicted probabilities.

- protein content of 12
- fat content of 15
- ash content of 1.5
- sodium content of 0.6
- carbohydrate content of 25
- caloric content of 3

Hint: See `?predict.lda`.

Topic 4: Support vector machines concepts and overview of estimation

Support Vector Machines

Introduction and motivation

As seen in earlier topics of this Week (Topic 2, Topic 3), when classifying into one of two p-dimensional multivariate normal populations, the scores are either linear (when the same covariance matrices are used) or quadratic (when the covariance matrices are different). Even optimality for such simple classifiers could be shown due to the multivariate normality assumption.

However, when the two populations are **not** multivariate normal, the situation is more difficult, the bounds between the populations may be more blurry and significantly more non-linear classification techniques may be necessary to achieve a good classification. Support vector machines (SVM) are an example of such non-linear statistical classification techniques.

They usually achieve superior results in comparison to more traditional non-linear parametric classification techniques such as *logit analysis* or non-parametric techniques such as *neural networks*. Mathematically, when using SVM, we try to formulate the classification as an empirical risk minimisation problem and to solve the problem under additional restrictions on the allowed (nonlinear) classifier functions.

Expected versus Empirical Risk minimisation

Let Y be an "indicator" with values $+1$ and -1 that indicate if certain p dimensional observation belongs to one of two groups of interest. We want to find a "best" classifier in a class $\mathcal{F}$ of functions f. Each classifier function $f(\boldsymbol{x})$ is meant to deliver a value of $+1$ or -1 for a given observation vector $\boldsymbol{x}$. To this end, we consider the *expected risk*

$$R(f) = \int \frac{1}{2}|f(\boldsymbol{x}) - y| dP(\boldsymbol{x}, y)$$

Since the joint distribution $P(\boldsymbol{x}, y)$ is unknown in practice, we consider the *empirical risk* over a training set $(\boldsymbol{x}_i, y_i), i = 1, 2, \ldots, n$ of observations instead:

$$\hat{R}(f) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2}|f(\boldsymbol{x}_i) - y_i|$$

The loss in the risk's definition is the "zero-one loss" given by

$$L(\boldsymbol{x}, y) = \frac{1}{2}|f(\boldsymbol{x}) - y|$$

and, thanks to the chosen labels ± 1 for Y obviously has the values 0 (if classification is correct) and 1 (if classification is wrong).

Minimising the empirical (instead of the unknown expected) risk means to find $f_n = \arg\min_{f \in \mathcal{F}} \hat{R}(f)$ as an approximation to $f_{\text{opt}} = \arg\min_{f \in \mathcal{F}} R(f)$. (Here and elsewhere, $\arg\min_a h(a)$ is that a which minimises the value of $h(a)$.) Generally speaking the two solutions f_n and f_{opt} do not coincide and without further assumptions may be quite different. However, thanks to some ground breaking work by V. Vapnik there are theoretical results which, loosely speaking, state that if $\mathcal{F}$ is not too large and $n \to \infty$, there is an upper bound on their difference with probability $(1 - \eta)$:

$$R(f) \leq \hat{R}(f) + \phi\left(\frac{h}{n}, \frac{\log \eta}{n}\right)$$

The above inequality can be interpreted as stating that the test error is bounded from above by the sum of the training error and the complexity of the set of models under consideration. We can then try to minimise the bound from above and hope that in that way we keep under control to a

minimum the (unknown) test error.

The function ϕ above is monotone increasing in h (at least for large enough sample sizes n). Here h denotes the Vapnik–Chervonenkis **(VC) dimension** (i.e., a measure of the complexity of the class $\mathcal{F}$).

For a linear classification rule $f(\boldsymbol{x}) = \text{sign}(\boldsymbol{w}^\top \boldsymbol{x} + b)$ with a p dimensional predictor $\boldsymbol{x}$ it is known that

$$\phi\left(\frac{h}{n}, \frac{\log \eta}{n}\right) = \sqrt{\frac{h(\log(\frac{2n}{h}) + 1) - \log(\frac{\eta}{4})}{n}}.$$

and that the VC dimension is $h = p + 1$. You can now **directly check** that

$$\frac{\partial}{\partial h}\left[\frac{h(\log(\frac{2n}{h}) + 1) - \log(\frac{\eta}{4})}{n}\right] = \frac{1}{n}\log\left(\frac{2n}{h}\right) > 0$$

as long as $h < 2n$ which confirms the monotone increasing property stated above.

In general, the VC dimension of a given set of functions is equal to the maximal number of points that can be separated in *all possible ways* by that set of functions.

At first glance, the "more rich" the function class $\mathcal{F}$ the better the classification rule would be. Indeed you can construct a classifier that has zero classification error on the training set. However, this classifier will be too specialised for the given training set with no ability to generalise for other sets. Hence such a classifier would be undesirable. "More rich" is tantamount to require bigger complexity of $\mathcal{F}$ or equivalently higher value of h (and therefore of ϕ). The term $\phi(\frac{h}{n}, \frac{\log \eta}{n})$ can be considered penalty for the excessive complexity of the classifier function. You can see directly that the derivative $\frac{\partial \phi(\frac{h}{n}, \frac{\log(\eta)}{n})}{\partial h} \geq 0$ if and only if $2n \geq h$. For large enough n this means that the function ϕ is increasing with the complexity of the model. Hence the sum of the two terms: $\hat{R}(f)$ (precision) and $\phi(\frac{h}{n}, \frac{\log \eta}{n})$ (complexity) represents the compromise between precision in the risk estimation and the complexity of the classifier. Therefore minimising this sum is the sensible thing to do in order to perform "optimally".

Basic idea of SVMs

A *linear classifier* is one that given feature vector $\boldsymbol{x}_{\text{new}}$ and weights $\boldsymbol{w}$, classifies y_{new} based on the value of $\boldsymbol{w}^\top \boldsymbol{x}_{\text{new}}$; for example,

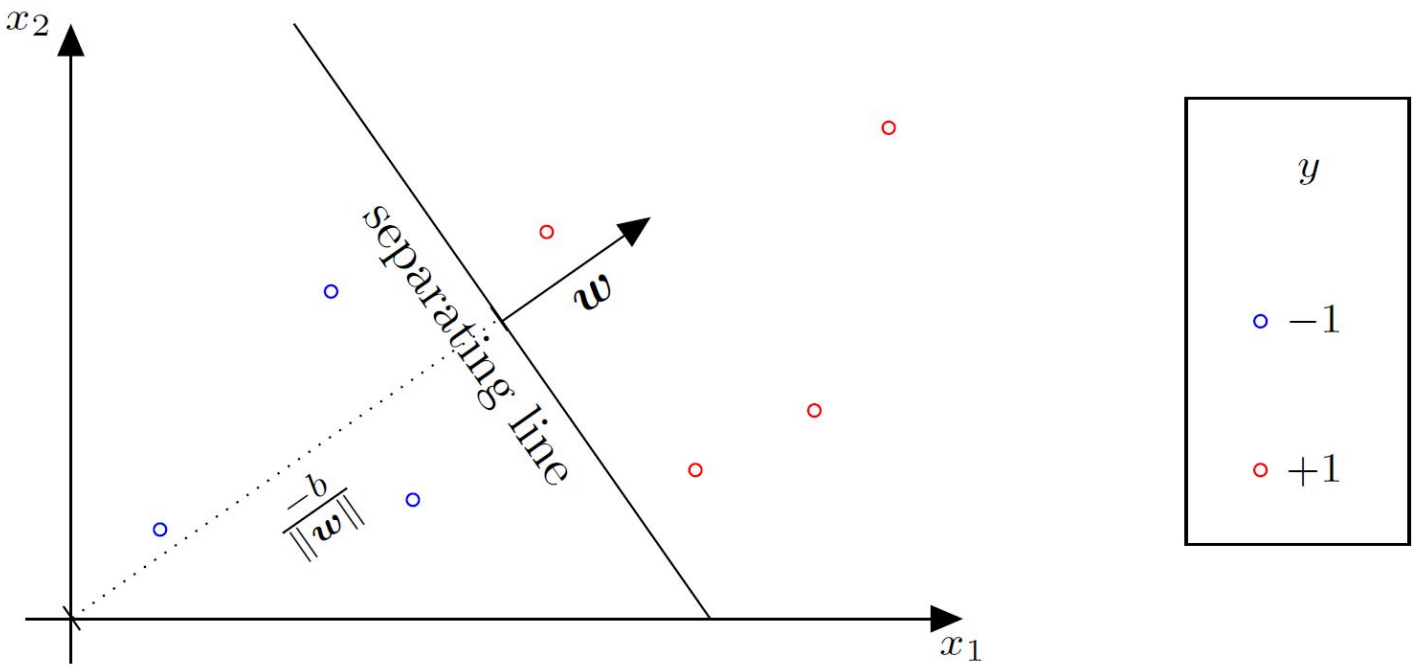
$$\hat{y}_{\text{new}} = \begin{cases} +1 & \text{if } \boldsymbol{w}^\top \boldsymbol{x}_{\text{new}} + b > 0 \\ -1 & \text{if } \boldsymbol{w}^\top \boldsymbol{x}_{\text{new}} + b < 0 \end{cases}$$

for a threshold $-b$. Here, we see that every element of $\boldsymbol{x}$, x_i, gets a weight w_i:

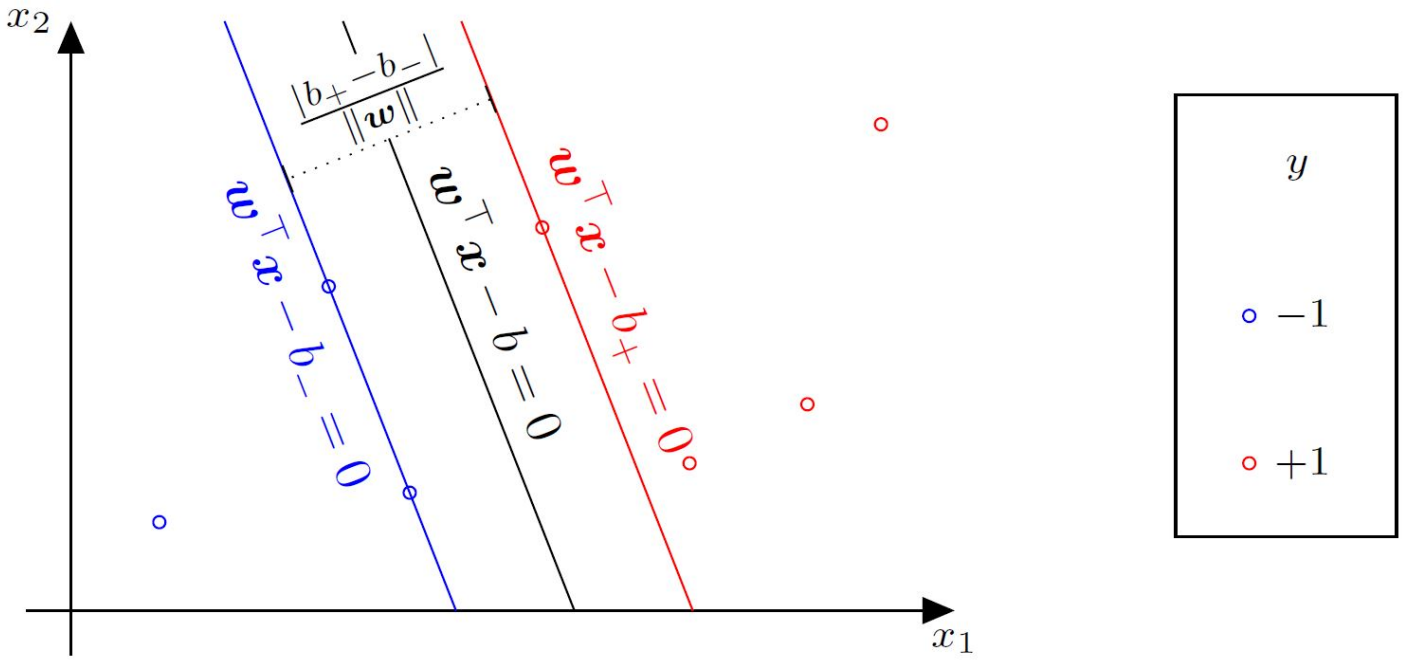
The **Sign** of w_i determines whether increasing x_i pushes the prediction toward $y_i = -1$ or $y_i = +1$.

The **Magnitude** of w_i determines how strongly.

The regions of $\boldsymbol{x}$ for which the model predicts $+1$ as opposed to -1 are defined by $\boldsymbol{w}^\top \boldsymbol{x} + b = 0$. Points $\boldsymbol{x}$ that satisfy that equation exactly form a line (if $d = 2$), a plane (if $d = 3$), or a hyperplane (if $d \geq 3$). We call the data *linearly separable* if a hyperplane that separates them exists. Let us focus on this linearly separable case (and consider the nonseparable case later.) The following diagram illustrates one such line:



Now, usually, there are infinitely many different hyperplanes which could be used to separate a linearly separable dataset. We therefore have to define the "best" one. The "best" choice can be regarded as the middle of the widest empty strip (or higher dimensional analogue) between the two classes, one that maximises the *margin* $\frac{|b_+ - b_-|}{\|\boldsymbol{w}\|}$ in the following illustration:



The scale of $\boldsymbol{w}$ and b is arbitrary: for arbitrary $\alpha \neq 0$, any $\boldsymbol{x}$ that satisfies $\boldsymbol{w}^\top \boldsymbol{x} + b = 0$ also satisfies $(\alpha \boldsymbol{w})^\top \boldsymbol{x} + (\alpha b) = \alpha(\boldsymbol{w}^\top \boldsymbol{x} + b) = 0$, so $(\boldsymbol{w}, b)$ and $(\alpha \boldsymbol{w}, \alpha b)$ define the same plane. We fix $|b_+ - b| = |b_- - b| = 1$, and only vary $\boldsymbol{w}$: our "outer" hyperplanes become

$$\boldsymbol{w}^\top \boldsymbol{x} + (b + 1) = 0$$
$$\boldsymbol{w}^\top \boldsymbol{x} - (b + 1) = 0$$

Then, the margin of $\frac{|b_+ - b_-|}{\|\boldsymbol{w}\|} = \frac{2}{\|\boldsymbol{w}\|}$ is maximised by minimising $\|\boldsymbol{w}\|$. Therefore, a *Linear Support Vector Machine* minimises $\|\boldsymbol{w}\|^2$ subject to separating -1s and $+1$s.

Estimation

Linear SVM: Separable Case

We write the boundaries of the empty region as

$$\begin{aligned} \boldsymbol{w}^\top \boldsymbol{x} + (b+1) = 0 &\implies \boldsymbol{w}^\top \boldsymbol{x} + b = +1 \\ \boldsymbol{w}^\top \boldsymbol{x} - (b+1) = 0 &\implies \boldsymbol{w}^\top \boldsymbol{x} + b = -1 \end{aligned}$$

and observe that

$$\hat{y}_i = \begin{cases} +1 & \text{if } \boldsymbol{w}^\top \boldsymbol{x}_i + b > 0 \\ -1 & \text{if } \boldsymbol{w}^\top \boldsymbol{x}_i + b < 0 \end{cases} = \text{sign}(\boldsymbol{w}^\top \boldsymbol{x}_i + b).$$

This means that if $\boldsymbol{w}^\top \boldsymbol{x} + b = 0$ separates -1s and $+1$s (i.e., $y_i = \hat{y}_i$ for all $i = 1, \ldots, n$),

$$y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1.$$

Therefore, a linear SVM learning task for can be expressed as a constrained optimisation problem:

$$\arg\min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{w}\|^2 \text{ subject to } y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1,\ i = 1, \ldots, n.$$

(Here and elsewhere, $\arg\min_a h(a)$ is that a which minimises the value of $h(a)$.)

This problem is solved using the Lagrange Multiplier technique. We omit the details for brevity, but it is valuable to observe the following intermediate result: we can show that the solution to the minimisation problem will be at

$$\begin{aligned} y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) - 1 \geq 0,\ i = 1, \ldots, n \\ \alpha_i \left(y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) - 1\right) = 0,\ i = 1, \ldots, n \end{aligned}$$

for some $\alpha_i \geq 0,\ i = 1, \ldots, n$. Notice that the second equation implies that *either* $\alpha_i = 0$ *or* $y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) = 1$ (or both). But that means that if $\alpha_i \neq 0$, the training instance lies on a corresponding hyperplane and is known as a *support vector*.

It turns out that we can further reexpress this problem as maximising

$$\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^\top \boldsymbol{x}_j,$$

subject to

$$\alpha_i \geq 0,\ i = 1, \ldots, n$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0.$$

This is a *quadratic programming* problem, for which many software tools are available.

Linear SVM: Nonseparable Case

Of course, in real-world problems, it is not possible to find hyperplanes which perfectly separate the target classes. The *soft margin* approach considers a trade-off between margin width and number of training misclassifications. *Slack* variables $\xi_i \geq 0$ are included in the constraints: we insist that

$$y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 - \xi_i. \tag{5.8}$$

The optimisation then becomes

$$\arg\min_{\boldsymbol{w}} \left(\frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n} \xi_i \right) \text{ subject to } y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 - \xi_i,\ i = 1, \ldots, n,$$

or a tuning constant C. Small C means a lot of slack, whereas a large C means little slack. In particular, if we set $C = \infty$, we require separation to be perfect, a *hard margin.*

Now, taking (5.8) and solving for ξ_i gives $\xi_i \geq 1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b)$. We want to make ξ_i as small as possible, so we can get $\xi_i = 1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b)$. After some further algebraic manipulation, we can eliminate these slack variables and end up with the following optimisation problem:

$$\arg\max_{\boldsymbol{\alpha}} \left(\sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n} \alpha_j \alpha_k y_j y_k (\boldsymbol{x}_j^\top \boldsymbol{x}_k) \right)$$
$$\text{subject to } \sum_{i=1}^{n} \alpha_i y_i = 0 \text{ and } 0 \leq \alpha_i \leq C,\ i = 1, \ldots, n.$$

Prediction

We can also express the prediction in two ways:

$$\text{Primal: } \hat{y}(x) = \text{sign}\left(\boldsymbol{w}^\top \boldsymbol{x} + b\right) \tag{5.9}$$

$$\text{Dual: } \hat{y}(x) = \text{sign}\left\{ \sum_{j=1}^{n} \alpha_j y_j \left(\boldsymbol{x}_j^\top \boldsymbol{x}\right) + b \right\} \tag{5.10}$$

Primal ($\boldsymbol{w}$) form requires d parameters, while *dual* ($\boldsymbol{\alpha}$) form requires n parameters. This means that for high-dimensional problems — those with $d \gg n$, a huge number of predictors, the dual

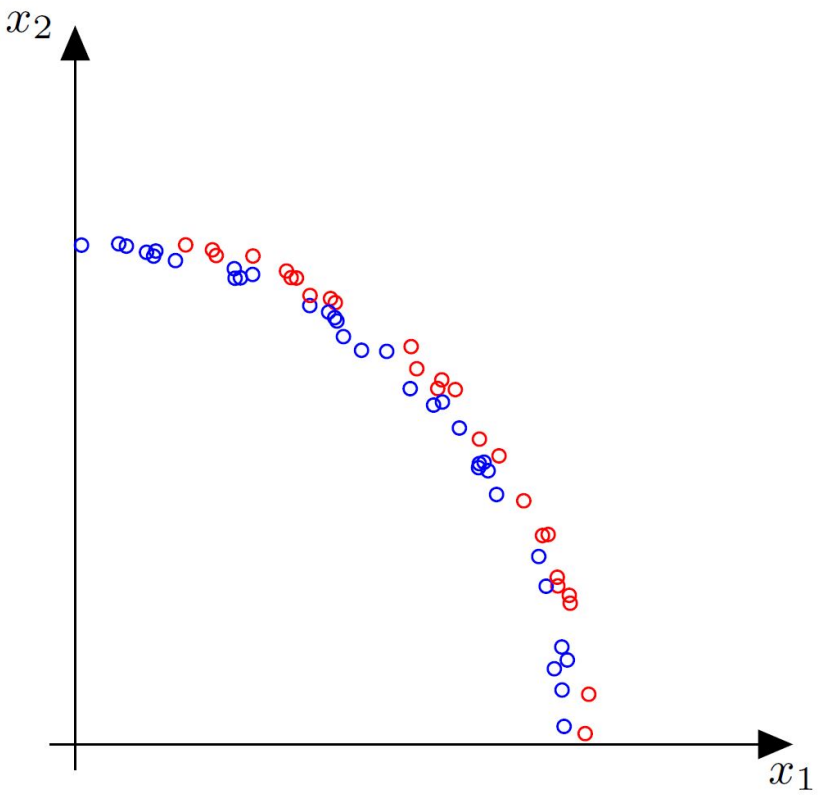
representation can be more efficient.

But it gets better! Notice that only the $\boldsymbol{x}_i$s closest to the separating hyperplane — those with $\alpha_j > 0$ — matter in determining $\hat{y}(\boldsymbol{x})$, so most of them will have no effect. Thus, computationally, effective "n" will actually be much smaller than the sample size, so the above condition can be met far more often than one might expect. Again, those $\boldsymbol{x}_i$s that "support" the hyperplane are called *support vectors*.

In addition, notice that the dual form only depends on $\left(\boldsymbol{x}_j^\top \boldsymbol{x}_k\right)$s. This opens the door to nonlinear SVMs.

Nonlinear SVMs

Consider:



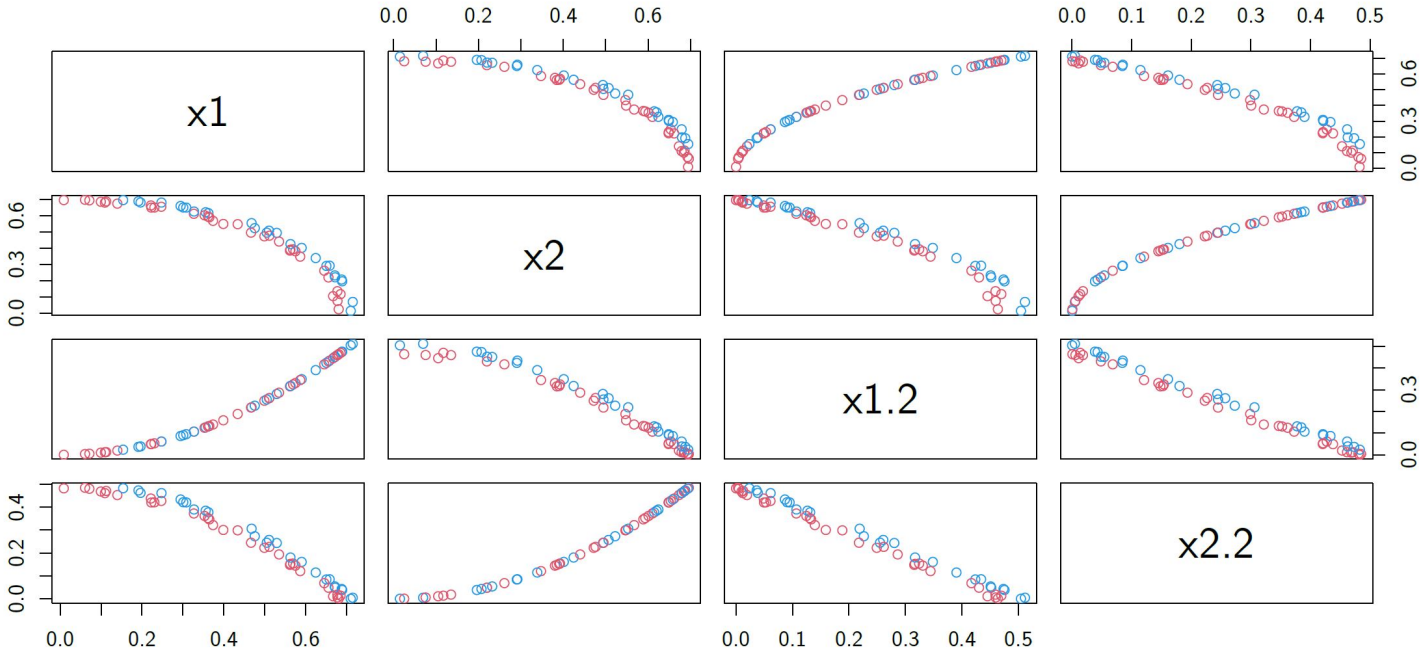
The true classification for these points is

$$y = \begin{cases} +1 & x_1^2 + x_2^2 < 0.75^2 \\ -1 & x_1^2 + x_2^2 > 0.75^2 \end{cases},$$

but one can hardly draw a line separating them.

What we can do is transform $\boldsymbol{x}$ so that a linear decision boundary can separate them. In this case, suppose we augmented our $\boldsymbol{x}$ with squared terms:

$$(x_1, x_2) \to (x_1, x_2, x_1^2, x_2^2) :$$



Now, a linear separator exists!

Now, recall that the dual form (5.10) depends only on dot products $\boldsymbol{x}_i^\top \boldsymbol{x}_j$. However, we can specify other *kernels* $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. For example, a "kernel" function of the form $k(\boldsymbol{u}, \boldsymbol{v}) = (\boldsymbol{u}^\top \boldsymbol{v} + 1)^2$ can be regarded as a dot product

$$u_1^2 v_1^2 + u_2 v_2^2 + 2u_1 v_1 + 2u_2 v_2 + 1$$

$$= (u_1^2, u_2^2, \sqrt{2}u_1, \sqrt{2}u_2, 1)^\top (v_1^2, v_2^2, \sqrt{2}v_1, \sqrt{2}v_2, 1).$$

which reconstructs the above augmentation. In general, kernel functions can be expressed in terms of high dimensional dot products. Computing dot products via kernel functions is computationally cheaper" than using transformed attributes directly.

A common type of kernel is a *radial basis function*: a function of distance from the origin, or from another fixed point $\boldsymbol{v}$. Usually, the distance is *Euclidean, i.e.*

$$\|\boldsymbol{u} - \boldsymbol{v}\| = \sqrt{(u_1 - v_1)^2 + \cdots + (u_n - v_n)^2}.$$

A common radial basis function is *Gaussian*:

$$\phi(\boldsymbol{u}, \boldsymbol{v}) = \exp\left(-\gamma \|\boldsymbol{u} - \boldsymbol{v}\|^2\right).$$

We can use $\phi(\cdot, \cdot)$ as our SVM kernel.

Multiple classes

Finally, we briefly consider the problem when there are more than two classes. Suppose that there are $K > 2$ categories.

Recall that $\boldsymbol{w}^\top \boldsymbol{x}_i$ gives us a "score" that we normally compare to b. However, we do not have to do so. Instead, for each $k = 1, \ldots, K$, we can fit a separate SVM (i.e., $\boldsymbol{w}_k$ and b_k) for whether an observation is in k vs. not. We can then predict $\hat{y}_{\text{new}}$ by evaluating $\boldsymbol{w}_k^\top \boldsymbol{x}_{\text{new}} + b_k$ for each k and taking highest biggest one. This is called the *One-against-rest* approach.

A computationally more expensive approach that tends to perform better is the *One-against-one*: an SVM is fit for every distinct pair $k_1, k_2 = 1, \ldots, K$, fit an SVM for k_1 vs. k_2, and predict the "winner" of all the rounds (if any). This requires fitting $K(K-1)/2$ binary classifiers, but to smaller datasets.

Topic 5: Tuning support vector machines

SVM specification and tuning

Categorical data can be handled by introducing binary *dummy* variables to indicate each possible value.

When fitting an SVM, the user must specify some control parameters, these include cost constant C for slack variables, the type of kernel function, and its parameters. Unlike the more probabilistic forms of classification, it is difficult to predict the out-of-sample classification error for SVMs, so cross-validation is used.

The following kernel functions available via the `R e1071` package:

linear: $\boldsymbol{u}^\top \boldsymbol{v}$

polybomial: $(\gamma \boldsymbol{u}^\top \boldsymbol{v} + c_0)^p$

radial basis: $\exp(-\gamma ||\boldsymbol{u} - \boldsymbol{v}||^2)$

sigmoid: $\tanh(-\gamma \boldsymbol{u}^\top \boldsymbol{v} + c_0)$

for constants γ, p, and c_0.

Conclusion

We conclude this week with a brief discussion of the advantages and disadvantages of SVMs. SVM training can be formulated as a convex optimisation problem, with efficient algorithms for finding the global minimum, and the final result involves support vectors rather than the whole training set. This is both a computational benefit, but also one to robustness: outliers have less effect than for other methods.

On the other hand, they are much more difficult to interpret than modelbased classification techniques like the linear discriminant analysis. Furthermore, SVMs do not actually provide class probability estimates. These can be estimated by cross-validation, however.

To conclude the week, complete the demonstration and associated challenge task in the following slides.

Demonstration: Support vector machines

This demonstration can be completed using the provided RStudio or your own RStudio.

To complete this task select the 'Cancor_Examples.demo.Rmd' in the 'Files' section of RStudio. Follow the demonstration contained within the RMD file.

If you choose to complete the example in your own RStudio, upload the following file:

SVM_Examples.demo.Rmd

The contents of the RMD file are also displayed below:

Packages

```
library(here)
library(GGally)
library(e1071) # svm()
library(pROC) # ROC curves
library(dplyr)
library(readr)
```

Iris Example

Data

Recall the Iris dataset:

```
data(iris)
iris
ggpairs(iris, aes(colour=Species, alpha = 0.3))
```

We will use it to explore SVMs and tuning.

Two-class SVM

We will begin by focusing on two classes (virginica and versicolor) and two variables (Sepal Length and Width):

```
iris2 <- iris %>% filter(Species!="setosa") %>%
select(Species,Sepal.Length,Sepal.Width) %>% mutate(Species=factor(Species))
plot(iris2[,2:3],col=as.numeric(iris$Species)+2)
legend("topleft", levels(iris$Species)[-1], fill=3:4, ncol=1)
```

Note that `svm()` when classifying requires the response variable to be of type `factor` rather than `character`.

Let's try a linear SVM with default settings and map its classification regions:

```
svm.linear <- svm(Species~.,data=iris2,kernel="linear")
plot(svm.linear, data=iris2)
```

The separation boundary is linear. (Here, the support vectors are denoted using "X" and point colour identifies the class.)

Now, let's try an RBF SVM with $\gamma = 1/d$ (the default):

```
svm.radial <- svm(Species~.,data=iris2,kernel="radial")
plot(svm.radial, data=iris2)
```

Observe that the classification boundary is now curved.

Let's try making the RBF decay faster by increasing γ: $\gamma = 1$:

```
svm.radial1 <- svm(Species~., data=iris2,kernel="radial",gamma=1)
plot(svm.radial1, data=iris2)
```

$\gamma = 10$:

```
svm.radial10 <- svm(Species~., data=iris2,kernel="radial",gamma=10)
plot(svm.radial10, data=iris2)
```

$\gamma = 100$:

```
svm.radial100 <- svm(Species~., data=iris2,kernel="radial",gamma=100)
plot(svm.radial100, data=iris2)
```

Eventually, each point becomes an island.

Something similar happens when we make the slack penalty too high: $\gamma = 1/d$, $C = 100$:

```
svm.radialC100 <- svm(Species~.,data=iris2,kernel="radial",cost=100)
plot(svm.radialC100, data=iris2)
```

Tuning the SVM

Lower γ / Lower C:

- more like linear SVM
- few support vectors

- contiguous predictions
- higher misclassification rate in-sample

Higher γ / Higher C:

- prediction regions can have more complex shapes
    - "islands" in the extreme case
- many support vectors
- higher misclassificaton rate out-of-sample (usually)
    - i.e., overfitting

Here, we have in-sample confusion matrices for: $\gamma = 1/d$ and $\gamma = 100$:

```
table(iris2$Species,fitted(svm.radial))
table(iris2$Species,fitted(svm.radial100))
```

It would seem that the higher slack penalty leads to better classification. But does it?

We can use cross-validation to get a better idea of out-of-sample error rate. Here, we use 10-fold cross-validation, in which the dataset is randomly split into 10 equal-sized parts, and each part takes a turn being predicted from the rest of the dataset.

$\gamma = 1/d$:

```
summary(svm.radial <- svm(Species~.,data=iris2,kernel="radial", cross=10))
```

$\gamma = 100$:

```
summary(svm.radial100 <- svm(Species~.,data=iris2,kernel="radial", gamma=100,
cross=10))
```

And so, in fact, it's much better not to set γ to be too high.

We can automate the tuning over a grid:

```
summary(tuned.svm <- tune.svm(Species~.,data=iris2,kernel="radial", gamma = 10^(-1:1),
cost = 10^(-1:1)))
tuned.svm$best.model
```

Here, the combination with the smallest error wins. It's stored in the `$best.model` element.

Multiclass SVM

Multiclass SVMs work as well:

```
svm3.radial <- svm(Species~Sepal.Length+Sepal.Width,data=iris,cross=10)
```

```
summary(svm3.radial)
plot(svm3.radial, data=iris, Sepal.Length~Sepal.Width)
```

Prediction

Now, suppose that we have measured a new flower, with Sepal Width of 3.4 and Sepal Length of 6.0. What species is it likely to be?

```
predict(svm3.radial, newdata=data.frame(Sepal.Width=3.4, Sepal.Length=6.0),
decision.values=TRUE)
```

ROC curves

Since linear classifier predictors are continuous ($\sum_{j=1}^{n} \alpha_j y_j (x_j' x)$ for SVM), we can use some threshold other than $-b$. This can be useful, in particular, if the prior probability or the cost of misclassifying matters. E.g.,

```
hist(y.hat.radial <- c(attr(predict(svm.radial, newdata=iris2,
decision.values=TRUE),"decision.values")))
```

We can use the `pROC` package:

```
roc.radial <- roc(iris2$Species=="versicolor", y.hat.radial)
y.hat.linear <- c(attr(predict(svm.linear, newdata=iris2,
decision.values=TRUE),"decision.values"))
roc.linear <- roc(iris2$Species=="versicolor", y.hat.linear)
plot(roc.radial, col=2)
plot(roc.linear, col=3, add=TRUE)
legend("bottomright",c("RBF SVM", "Linear SVM"),lty=1,col=2:3)
```

This is mainly useful when there are two classes.

Challenge: Support vector machines

If you choose to complete this task in your own RStudio, upload the following file:

SVM_Examples.challenge.Rmd

Click on the 'SVM_Examples.challenge.Rmd' in the 'Files' section to begin. Enter your response to the tasks in the 'Enter your code here' section.

This activity and the solution will be discussed at the Collaborate session this week. In the meantime, share and discuss your results in the 'Tutorials' discussion forum.

The solution will also be available here on Friday of this week by clicking on the 'Solution' tab in the top right corner.

The contents of the RMD file are also displayed below:

Packages

```
library(here)
library(GGally)
library(e1071) # svm()
library(pROC) # ROC curves
library(dplyr)
library(readr)
```

Recall the dataset `pizza.csv`, containing nutritional data from a variety of pizza brands:

`brand` : Pizza brand (class label)

`id` : Sample analysed

`prot` : Amount of protein per 100 grams in the sample

`fat` : Amount of fat per 100 grams in the sample

`ash` : Amount of ash per 100 grams in the sample

`sodium` : Amount of sodium per 100 grams in the sample

`carb` : Amount of carbohydrates per 100 grams in the sample

`cal` : Amount of calories per 100 grams in the sample

```
pizza <- read_csv(here("datasets","pizza.csv")) %>% mutate(brand=factor(brand))
```

```
ggpairs(pizza, mapping=aes(col=brand, alpha=0.3))
```

Task 1: Select and tune an SVM for classifying the pizza brand based on measurements. What is its cross-validated accuracy?

Task 2: Use your selected classifier to classify a pizza that has the following properties.

- protein content of 12
- fat content of 15
- ash content of 1.5
- sodium content of 0.6
- carbohydrate content of 25
- caloric content of 3

Topic 1: Clustering

Welcome to Week 6

Dr Pavel Krivitsky gives you a brief overview of topics and concepts we'll be covering in this week.

Transcript

Weekly learning outcomes

- Explain different types of clustering and their advantages and disadvantages.
- Fit, visualise, and evaluate a K-Means, K-Medioids and hierarchical clustering to data.
- Translate substantive information about clusters into multivariate normal model assumptions.
- Fit, visualise, and evaluate model-based clustering to data.
- Select the optimal number of clusters and cluster model.
- Select a copula appropriate to the data.
- Estimate, visualise, diagnose, and simulate from a copula model.
- Explain the assumptions underlying the above inferential procedures and check them.

Topics we will cover are:

- Topic 1: Clustering
- Topic 2: Copula methods

Optional reading

An alternative presentation of the concepts for this week can be found in:

Johnson, R. A., & Wichern, D. (2008). *Applied Multivariate Statistical Analysis* (6th ed.). Pearson Prentice Hall.

- 12.1 - 12.5

Additional software demonstration of model-based clustering can be found in:

Scrucca, L., Fop, M., Murphy, T. B., & Raftery, A. E. (2016). mclust 5: clustering, classification and density estimation using Gaussian finite mixture models. *The R Journal*, 8(1), 289.

All readings are available from the course Leganto reading list. Please keep in mind that you will need to be logged into Moodle to access the Leganto reading list.

Questions about this week's topics?

This week's topics were prepared by Dr P. Krivitsky. If you have any questions or comments, please post them under Discussion or email directly: p.krivitsky@unsw.edu.au

Cluster analysis

The goal of cluster analysis is to identify groups in data. In contrast to SVMs and discriminant analysis, no preexisting group labels are provided. This makes it an example of *unsupervised learning*.

The input of cluster analysis is therefore an *unlabelled* sample $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^p$, and the output is a *grouping* of observations such that more similar (in some sense) observations are placed in the group. That is, cluster analysis assigns to each $\boldsymbol{x}_i$ a group index $G_i \in 1, \ldots, K$ such that if $G_i = G_j$, $\boldsymbol{x}_i$ *and* $\boldsymbol{x}_j$ are "on average" more similar in some sense than if $G_i \neq G_j$.

Throughout this week, we will use the following additional notation.

$\boldsymbol{G} = (G_1, \ldots, G_n)^\top$: a column vector of cluster memberships.

$S_1, \ldots, S_K$: a *partitioning* of the observations $\{1, \ldots, n\}$ into K non-overlapping sets such that for every $i \in S_k$, $G_i = k$.

$\boldsymbol{S} = (S_1, \ldots, S_K)$: a shorthand for the clustering expressed in terms of sets.

We will consider a taxonomy of approaches to clustering. The "classical" approach is to specify an *algorithm* that assigns observations to clusters. (Often, but not always, an objective function may be defined that is optimised by the algorithm.) Classical approaches can be further subdivided into *hierarchical* clustering, which produces a hierarchy of nested clusterings in a tree which has observations as leaves; and *non-hierarchical*, which merely assigns a label to each point.

The *model-based* approach to clustering is to postulate a *mixture model*—a model consisting of a mixture of probability distributions with different location parameters. The parameters of this model embody information about the clusters (e.g., their means and frequencies), and estimating them enables probabilistic, or *soft* clusterings.

We discuss these approaches in this topic.

“Classical”

Defining a clustering

In order to cluster data—particularly multivariate data—we must first define a *proximity measure*: some function $d(\boldsymbol{x}_1, \boldsymbol{x}_2)$ that determines difference between two observations. (Equivalently we can define a similarity score and negate or invert it.) Here are some common metrics measures:

Euclidean: $\|\boldsymbol{x}_1 - \boldsymbol{x}_2\| = \sqrt{\sum_{k=1}^{p}(x_{1k} - x_{2k})^2}$, the "ordinary" straight-line distance.

Taxicab/Manhattan: $\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_1 = \sum_{k=1}^{p}|x_{1k} - x_{2k}|$, distance if one is only allowed to travel parallel to the axes (like a taxicab on the Manhattan city grid).

Gower: $p^{-1}\sum_{k=1}^{p}\mathbb{I}(x_{1k} \neq x_{2k})$: for binary measures.

A metric should be substantively meaningful and appropriate for the data. It is also common to scale all of the dimensions (say, to have variance of 1 or to be between 0 and 1) before clustering.

Given these distances, we specify the algorithm that minimises within-cluster and maximises between-cluster distances in some sense — that sense often operationalised in an *objective function*.

Example: K-means

Perhaps the best known clustering algorithm is the K-means. It has the advantage of being simple and intuitive. The objective function that it ultimately minimises (over the partitioning $\boldsymbol{S} = (S_1, \ldots, S_K)$) is

$$\sum_{g=1}^{K}\frac{1}{2|S_g|}\sum_{i,j\in S_g}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2,$$

the sum of squared Euclidean distances between every distinct pair of observations within each cluster (appropriately scaled). It can be shown (using a decomposition similar to that of ANOVA) that this is equivalent to minimising

$$\sum_{g=1}^{K}\sum_{i\in S_g}\|\boldsymbol{x}_i - \bar{\boldsymbol{x}}_{S_g}\|^2,\ \bar{\boldsymbol{x}}_{S_g} = \frac{1}{|S_g|}\sum_{i\in S_g}\boldsymbol{x}_i,$$

which is simply the sum of the squared Euclidean distances between each data point and the mean of its cluster.

The following algorithm often does a good job finding such a clustering:

1. Randomly assign a cluster index to each element of $\boldsymbol{G}^{(0)}$.
2. Calculate cluster means (centroids):

$$\bar{\boldsymbol{x}}_{S_g^{(t-1)}} = \frac{1}{|S_g^{(t-1)}|} \sum_{i \in S_g^{(t-1)}} \boldsymbol{x}_i,\ g = 1, \ldots, K.$$

3. Calculate distances of each data point from each mean:

$$d_{ig} = \| \boldsymbol{x}_i - \bar{\boldsymbol{x}}_{S_g^{(t-1)}} \|,\ i = 1, \ldots, n,\ g = 1, \ldots, K.$$

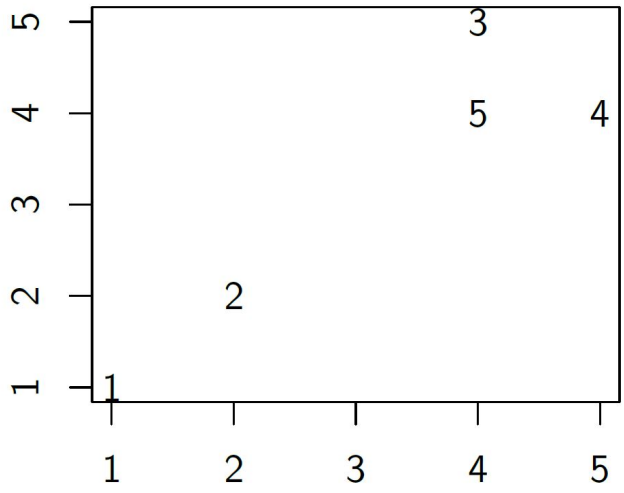
4. Reassign each point to its nearest mean:

$$G_i^{(t)} = \arg\min_g d_{ig}.$$

   (Here and elsewhere, $\arg\min_a h(a)$ is that a which minimises the value of $h(a)$.)
5. Repeat from Step 2 until $\boldsymbol{G}^{(t)} = \boldsymbol{G}^{(t-1)}$.

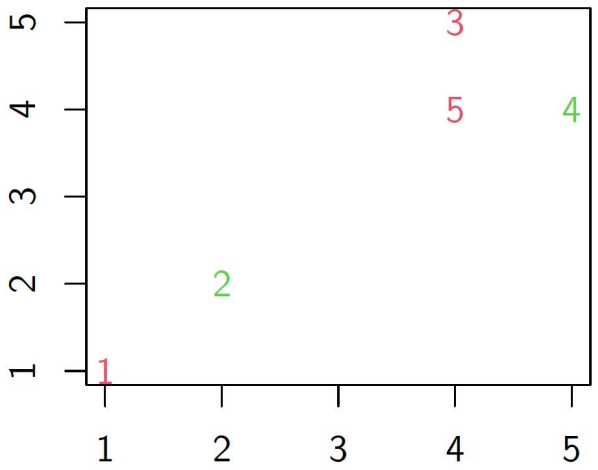
Example 6.1. Consider the following dataset:

Index	V1	V2
1	1	1
2	2	2
3	4	5
4	5	4
5	4	4



We begin by creating an initial clustering at random:

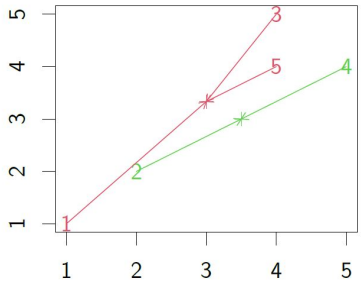
Index	V1	V2	C
1	1	1	1
2	2	2	2
3	4	5	1
4	5	4	2
5	4	4	1



Calculate the centroids:

Index	V1	V2	C
1	1	1	1
2	2	2	2
3	4	5	1
4	5	4	2
5	4	4	1

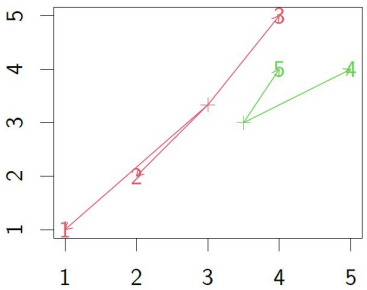
C	V1	V2
1	3.0	3.333333
2	3.5	3.000000



Update the clustering:

Index	V1	V2	C
1	1	1	1
2	2	2	1
3	4	5	1
4	5	4	2
5	4	4	2

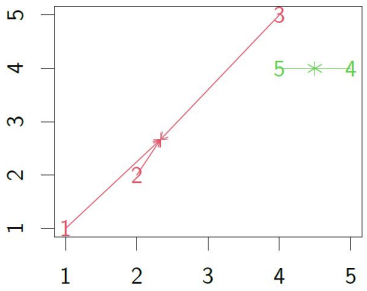
Index	1	2
1	3.073182	3.201562
2	1.666667	1.802776
3	1.943651	2.061553
4	2.108185	1.802776
5	1.201850	1.118034



Calculate the centroids:

Index	V1	V2	C
1	1	1	1
2	2	2	1
3	4	5	1
4	5	4	2
5	4	4	2

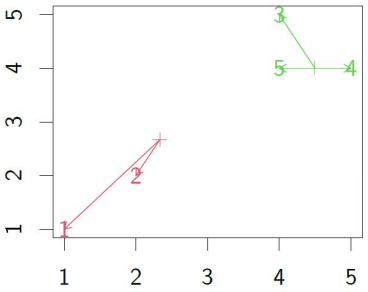
C	V1	V2
1	2.333333	2.666667
2	4.500000	4.000000



Update the clustering:

Index	V1	V2	C
1	1	1	1
2	2	2	1
3	4	5	2
4	5	4	2
5	4	4	2

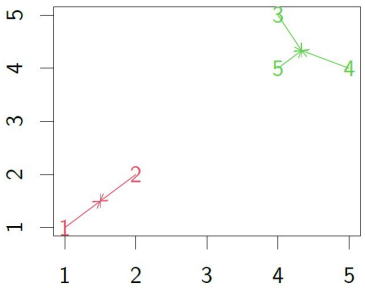
Index	1	2
1	2.134375	4.609772
2	0.745356	3.201562
3	2.867442	1.118034
4	2.981424	0.500000
5	2.134375	0.500000



Calculate the centroids:

Index	V1	V2	C
1	1	1	1
2	2	2	1
3	4	5	2
4	5	4	2
5	4	4	2

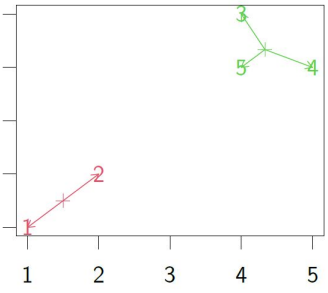
C	V1	V2
1	1.500000	1.500000
2	4.333333	4.333333



Update the clustering:

Index	V1	V2	C
1	1	1	1
2	2	2	1
3	4	5	2
4	5	4	2
5	4	4	2

Index	1	2
1	0.7071068	4.7140452
2	0.7071068	3.2998316
3	4.3011626	0.7453560
4	4.3011626	0.7453560
5	3.5355339	0.4714045



The clustering is unchanged from the previous iteration, so we declare convergence.

Extension: K-medioids

A generalisation of K-means is the K-medioids technique. We define a *medioid* $\tilde{\boldsymbol{x}}_g$ *of cluster* g to be a *specific observation* that has the closest summed distance (however defined) to all other observations in S_g:

$$\tilde{\boldsymbol{x}}_{S_g} = \arg\min_{\boldsymbol{x}_i} \sum_{i \in S_g} d(\boldsymbol{x}_j, \boldsymbol{x}_i).$$

The *Method of* K *– medioids* or *partitioning around medioids (PAM)* minimises the sum of these distances:

$$\arg\min_{\boldsymbol{S}} \sum_{g=1}^{K} \sum_{i \in S_g} d(\boldsymbol{x}_i, \tilde{\boldsymbol{x}}_{S_g}).$$

This method is much more expensive computationally than K-means, but it is also more robust to outliers. It is typically fit as follows:

1. Randomly assign a cluster index to each element of $\boldsymbol{G}^{(0)}$.
2. Calculate cluster medioids:

$$\tilde{\boldsymbol{x}}_{S_g^{(t-1)}} = \arg\min_{\boldsymbol{x}_i} \sum_{j \in S_g^{(t-1)}} d(\boldsymbol{x}_j, \boldsymbol{x}_i),\ g = 1, \ldots, K.$$

3. Calculate distances of each data point from each medioid:

$$d_{ig} = d(\boldsymbol{x}_i, \tilde{\boldsymbol{x}}_{S_g^{(t-1)}}),\ i = 1, \ldots, n,\ g = 1, \ldots, K.$$

4. Reassign each point to its nearest medioid:

$$G_i^{(t)} = \arg\min_g d_{ig}.$$

5. Repeat from Step 2 until $\boldsymbol{G}^{(t)} = \boldsymbol{G}^{(t-1)}$.

Hierarchical clustering

Hierarchical clustering, instead of partitioning the data into K groups, produces a hierarchy of clusterings whose sizes range from 1 (no splits) to as high as n (every observation its own cluster). This clustering is typically visualised in a *dendrogram*, a tree diagram whose branching represents subdivisions of the data into clusters and whose height represents the distances between points or clusters.

The algorithms for producing these clusterings are either *agglomerative*, in that they start with each observation in its own cluster, then combine nearest observations into clusters, nearest clusters into bigger clusters, etc.; or *divisive*, starting with the whole dataset, then splitting it into a small number of clusters, those clusters into smaller clusters, etc..

The former require defining a notion of a *distance between clusters*. The latter require to defining a criterion based on which a cluster is split. Some common examples of distances are provided in the following list:

- Single linkage $d(S_1, S_2) = \min\{d(\boldsymbol{x}_i, \boldsymbol{x}_j) : i \in S_1, j \in S_2\}$
- Complete linkage $d(S_1, S_2) = \max\{d(\boldsymbol{x}_i, \boldsymbol{x}_j) : i \in S_1, j \in S_2\}$
- Average linkage (unweighted) $d(S_1, S_2) = \frac{1}{|S_1||S_2|} \sum_{i \in S_1} \sum_{j \in S_2} d(\boldsymbol{x}_i, \boldsymbol{x}_j)$
- Average linkage (weighted) $d(S_1 \cup S_2, S_3) = \frac{d(S_1, S_3) + d(S_2, S_3)}{2}$
- Centroid $d(S_1, S_2) = \|\bar{\boldsymbol{x}}_{S1} - \bar{\boldsymbol{x}}_{S_2}\|$
- Ward $d\left(S_1, S_2\right) = \sum_{i \in S_1 \cup S_2} \left|\boldsymbol{x}_i - \overline{\boldsymbol{x}}_{S_1 \cup S_2}\right|^2 - \sum_{i \in S_1} \left|\boldsymbol{x}_i - \overline{\boldsymbol{x}}_{S_1}\right|^2 - \sum_{i \in S_2} \left|\boldsymbol{x}_i - \overline{\boldsymbol{x}}_{S_2}\right|^2 = \frac{|S_1||S_2|}{|S_1|+|S_2|} \left|\overline{\boldsymbol{x}}_{S_1} - \overline{\boldsymbol{x}}_{S_2}\right|^2$

A framework that is useful for expressing different between-cluster distances is the *Lance–Williams* framework. Given three clusters, S_1, S_2, and S_3, and suppose that we have some metric for evaluating pairwise distances between them, i.e., $d(S_1, S_2)$, $d(S_1, S_3)$, and $d(S_2, S_3)$. Then, we define the distance resulting from combining S_1 and S_2 in terms of these pairwise distances and coefficients α_1, α_2, β, and γ:

$$d(S_1 \cup S_2, S_3) = \alpha_1 d(S_1, S_3) + \alpha_2 d(S_2, S_3) + \beta d(S_1, S_2) + \gamma |d(S_1, S_3) - d(S_2, S_3)|.$$

This, plus the distance metric between individual points (which applies when the clusters have only one observation in them), allows us to define and efficiently calculate distances between clusters.

For example, the unweighted average linkage can be expressed in this framework as follows:

$$
\begin{aligned}
d\left(S_1 \cup S_2, S_3\right) &= \frac{1}{\left|S_1 \cup S_2\right| \left|S_3\right|} \sum_{i \in S_1 \cup S_2} \sum_{j \in S_3} d\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) \\
&= \frac{1}{\left(\left|S_1\right| + \left|S_2\right|\right) \left|S_3\right|} \left(\sum_{i \in S_1} \sum_{j \in S_3} d\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) + \sum_{i \in S_2} \sum_{j \in S_3} d\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) \right) \\
&= \frac{\left|S_1\right| \left|S_3\right| d\left(S_1, S_3\right) + \left|S_2\right| \left|S_3\right| d\left(S_2, S_3\right)}{\left(\left|S_1\right| + \left|S_2\right|\right) \left|S_3\right|}
\end{aligned}
$$

$$
\implies \alpha_1 = \frac{\left|S_1\right|}{\left|S_1\right| + \left|S_2\right|}, \alpha_2 = \frac{\left|S_2\right|}{\left|S_1\right| + \left|S_2\right|} \beta = \gamma = 0
$$

Ward's method—the most popular hierarchical clustering criterion— similarly, uses the squared Euclidean distances $d(\boldsymbol{x}_i, \boldsymbol{x}_j) = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2$ between points and then

$$
\begin{aligned}
\alpha_1 &= \frac{\left|S_1\right| + \left|S_3\right|}{\left|S_1\right| + \left|S_2\right| + \left|S_3\right|}, \alpha_2 = \frac{\left|S_2\right| + \left|S_3\right|}{\left|S_1\right| + \left|S_2\right| + \left|S_3\right|} \\
\beta &= \frac{-\left|S_3\right|}{\left|S_1\right| + \left|S_2\right| + \left|S_3\right|}, \gamma = 0
\end{aligned}
$$

Ward's method joins the groups that will increase the within-group variance least.

In Software

R:

Hierarchical: `stats::hclust`, `cluster::agnes`

Non-hierarchical: `stats::kmeans`, `cluster::pam`

- Many others

Assessing

We now briefly discuss how a clustering $\boldsymbol{G}$ may be assessed. Ideally, this measurement should be "fair" to the number of clusters K. For example, in K-means clustering, splitting a cluster will *always* reduce the within-cluster variances, and so those cannot be used as a criterion.

A popular method, inspired by K-medioid clustering, is the *silhouettes*. For each $i = 1, \dots, n$, let

$$a(i) = \frac{1}{|S_{G_i}| - 1} \sum_{j \in S_{G_i}} d\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)$$

$$b(i) = \min_{g \neq G_i} \frac{1}{|S_g|} \sum_{j \in S_g} d\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)$$

Observe that $a(i)$ is the distance between i and other observations its own cluster and $b(i)$ is the distance between i and observations in the cluster nearest to i to which i does not belong. In a good clustering each observation will be much closer to its own cluster than to its neighbouring cluster, so $b(i) \gg a(i)$.

Then, *silhouette of* i is a value between -1 and $+1$ calculated as follows:

$$s(i) = \begin{cases} \frac{b(i) - a(i)}{\max(a(i), b(i))} & \text{if } |S_{G_i}| > 1 \\ 0 & \text{otherwise} \end{cases}.$$

That is $s(i)$ evaluates how much closer is i to the rest of its cluster than it is to its nearest cluster, and a higher silhouette indicates a better clustering for point i. Mean silhouette $n^{-1} \sum_{i=1}^{n} s(i)$ then measures the overall quality of clustering.

Model-based clustering

Lastly, we turn to model-based clustering. We will discuss the theoretical underpinnings of this approach—mixture models—and an important special case of Gaussian clustering and its parametrisation. The Expectation–Maximisation algorithm, often used to estimate these models will also be described, as it is useful in a wide variety of circumstances, but it is not examinable.

Mixture Models

A *finite mixture model* is a probability model under which each observation comes from one of several distributions, but we do not observe from which one. (Infinite mixture models exist as well, but they are outside of the scope of this class).

A mixture model is specified as follows. We set K to be the number of distributions (clusters), and a collection of K density functions on the support of $\boldsymbol{x}_i$, $f_g(\boldsymbol{x}_i; \boldsymbol{\theta}_g)$ (for $g = 1, \ldots, K$) each having a parameter vectors $\boldsymbol{\theta}_g$ (e.g., its expectation), which we do not know and must estimate. We also postulate K (unknown) probabilities π_g that an observation (any observation) comes from cluster g. (Standard restrictions apply: $0 \leq \pi_g \leq 1, \sum_{g=1}^K \pi_g$.)

For brevity, we define $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)^\top$, a vector of these probabilities; and $\Psi = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K, \boldsymbol{\pi}\}$, the collection of all model parameters. Then, we assume the following data-generating process: for each $i = 1, \ldots, n$,

1. Sample $G_i \in 1, \ldots, K$ with $\Pr(G_i = g; \boldsymbol{\pi}) = \pi_g$.
2. Sample $\boldsymbol{X}_i | G_i \sim f_{G_i}(\cdot; \boldsymbol{\theta}_{G_i})$.
3. Observe $\boldsymbol{X}_i$, and "forget" G_i.

The pdf of this *mixture density* is

$$f_{\boldsymbol{X}_i}(\boldsymbol{x}_i; \Psi) = \sum_{g=1}^K \pi_g f_g(\boldsymbol{x}_i; \boldsymbol{\theta}_g). \tag{6.1}$$

We wish to estimate the parameters Ψ from the sample of $\boldsymbol{x} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$. This leads to the likelihood

$$L_{\boldsymbol{x}}(\Psi) = \prod_{i=1}^n \sum_{g=1}^K \pi_g f_g(\boldsymbol{x}_i; \boldsymbol{\theta}_g). \tag{6.2}$$

This formulation is convenient for a number of reasons. It is a probability model for the $\boldsymbol{X}_i$s, and therefore we can use it to obtain a *soft clustering* rather than a *hard clustering* that assigns a point to a

single cluster, we can apportion an observation's membership by how likely it to have come from each cluster. An application of Bayes's rule and (6.1) gives

$$\Pr(G_i = g|\boldsymbol{x}_i; \Psi) = \frac{\pi g f_g(\boldsymbol{x}_i; \boldsymbol{\theta}_g)}{\sum_{g'=1}^{K} \pi_{g'} f_{g'}(\boldsymbol{x}_i; \boldsymbol{\theta}_{g'})}.$$

We can also embed it into a *hierarchical model* (a meaning distinct from the hierarchical clustering above), in which either $\boldsymbol{x}_i$s are parameters for some model for the data or for the observation process or $\boldsymbol{\theta}$s are functions of some *hyper–parameters*. Lastly, the fact that we have a well-defined likelihood facilitates model selection.

Multivariate normal clusters

As with other analysis scenarios discussed in this course, the multivariate normal distribution provides a useful formulation for the clusters. Consider the following parametrisation:

$$f_g(\boldsymbol{x}_i; \boldsymbol{\theta}_g) = \frac{1}{(2\pi)^{p/2}|\Sigma(\boldsymbol{\theta}_g)|^{1/2}} \, \mathrm{e}^{-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}(\boldsymbol{\theta}_g))^\top \{\Sigma(\boldsymbol{\theta}_g)\}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}(\boldsymbol{\theta}_g))}.$$

Here, $\boldsymbol{\mu}(\boldsymbol{\theta}_g)$ is the mean vector of cluster g (e.g., first p elements of $\boldsymbol{\theta}_g$), and $\Sigma(\boldsymbol{\theta}_g)$ is the model for the variances. We may also have different clusters "share" elements of $\boldsymbol{\theta}$, and a more general case is

$$f_g(\boldsymbol{x}_i; \boldsymbol{\theta}) = \frac{1}{(2\pi)^{p/2}|\Sigma_g(\boldsymbol{\theta})|^{1/2}} \, \mathrm{e}^{-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_g(\boldsymbol{\theta}))^\top \{\Sigma_g(\boldsymbol{\theta})\}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_g(\boldsymbol{\theta}))}, \tag{6.3}$$

where $\boldsymbol{\mu}_g(\boldsymbol{\theta})$ and $\Sigma_g(\boldsymbol{\theta})$ "extract" the appropriate elements from $\boldsymbol{\theta}$.

One advantage of multivariate normal clusters is in its flexibility in specifying cluster size and shape. Recall the eigendecomposition of the covariance matrix $\Sigma = P\Lambda P^\top$, with P orthogonal and Λ diagonal and nonnegative. Let us further parametrise it as

$$\Sigma = \lambda P A P^\top,$$

with $P \in \mathcal{M}_{p,p}$ orthogonal, $A \in \mathcal{M}_{p,p}$ diagonal and nonnegative *with* $|A| = 1$ (*unimodular*), and scalar $\lambda > 0$. This allows us to interpret the structure of the matrix in simple, substantive terms.

Starting with λ, recall recalling that the determinant of a matrix can be viewed as its volume. Then,

$$|\Sigma| = \lambda^p |P||A||P^\top| = \lambda^p,$$

which makes λ is the "spread", "size", or "volume" of the cluster.

To interpret the diagonal, unimodular matrix A, observe that if $A = I_p$, then

$$\Sigma = \lambda PAP^\top = \lambda PP^\top = \lambda I_p,$$

making the cluster spherical—equal variances on all dimensions. Similarly, if some diagonal elements of A are much larger than others, then the cluster will be an ellipsoid more stretched in one direction than in others.

Lastly, observe that if $P = I_p$, then

$$\Sigma = \lambda PAP^\top = \lambda A,$$

an ellipsoid whose axes are parallel to coordinate axes, implying the elements of $\boldsymbol{X}_i$ within each cluster are uncorrelated with unequal variances. More generally, P controls the rotation of ellipsoid--- the correlation between the dimensions and the orientation of the cluster.

When it comes to estimating K clusters, we can permit the λs, the As, and the Ps to vary between the clusters, be constant between the clusters, or, for A and P, be fixed at the identity. Each combination embodies different assumption about the shape and the relationship between clusters; and, in general, the more we permit to vary, the more parameters we must estimate and the more data we therefore require. Generally,

1. For a mixture of K clusters, we must, invariably, estimate the cluster membership probabilities $\pi_1, \ldots, \pi_K$ ($K - 1$ parameters) and cluster means $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ (Kp parameters).
2. Then, λ can be constrained $\lambda_1 = \lambda_2 = \cdots = \lambda_K$ (1 parameter) or allowed to vary (K parameters).
3. Then, A can be fixed $A_1 = A_2 = \cdots = A_K = I_d$ (0 parameters), constrained $A_1 = A_2 = \cdots = A_K$ ($p - 1$ parameters), or allowed to ($K(p - 1)$ parameters).
4. Lastly, if A is not fixed at the identity matrix, P can either be fixed $P_1 = P_2 = \cdots = P_K = I_d$ (0 parameters), constrained $P_1 = P_2 = \cdots = P_K$ ($\binom{p}{2}$ parameters), or allowed to vary ($K\binom{p}{2}$ parameters)

The different cluster shapes identified by their constraint tripple (λ, A, P) encoding being fixed at identity as I, being constrained to equality between clusters as E, and being allowed to vary freely as V are given in the following figure:

Figure 2 of: Luca Scrucca, Michael Fop, T. Brendan Murphy, and Adrian E. Raftery (2016). `mclust` 5: Clustering, Classification and Density Estimation Using Gaussian Finite Mixture Models. *The R Journal* 8:1, pages 289-317.

Model selection

As mentioned before, model-based clustering requires one to specify both the number of clusters K and the within-cluster models $f_g(\boldsymbol{x}_i; \Psi)$. In the case of multivariate normal clustering, we have a large number of possible specifications for the Σ_gs, and the number of parameters can grow quickly for "XXV" models in particular.

At the same time, because it is likelihood-based, a variety of standard model-selection techniques can be used. For example, BIC is recommended:

$$\mathrm{BIC}\nu = -2 \log L_{\boldsymbol{x}}(\hat{\Psi}) + \nu \log n,$$

where ν the number of parameters estimated. (Here, lower BIC is better, but some authors and software packages use $2 \log L_{\boldsymbol{x}}(\hat{\Psi}) - \nu \log n$, with higher BIC being better.)

Substantive considerations also matter. For example, how many clusters does our research hypothesis predict? Do we expect correlations between dimensions to vary between clusters?

In Software

R: package `mclust` and others.

Expectation–Maximisation Algorithm (optional)

In this final topic, we discuss the typical computational approach for estimating these mixture models. The $\log L(\Psi)$ in (6.2) is computationally tractable, but it does not simplify much, because while the logarithm of a product is a sum of the logarithms, the logarithm of a sum does not, in general, simplify further. Thus, we introduce the *Expectation–Maximisation (EM)* algorithm.

1. Introduce an unobserved (latent) variable G_i, $i = 1, \ldots, n$ giving the cluster membership of i.
2. Suppose that $G_1, \ldots, G_n$ *are* observed; then, this *complete-data likelihood*

$$L_{\boldsymbol{x},G_1,\ldots,G_n}(\Psi) = \prod_{i=1}^{n} \pi_{G_i} f_{G_i}(\boldsymbol{x}_i; \boldsymbol{\theta}_{G_i}) :$$

   we "know" the exact cluster from which each observation came, so we no longer have to sum over the possible clusters. Then, the log-likelihood decomposes into two summations

$$\log L_{\boldsymbol{x},G_1,\ldots,G_n}(\Psi) = \sum_{i=1}^{n} \log \pi G_i + \sum_{i=1}^{n} \log f_{G_i}(\boldsymbol{x}_i; \boldsymbol{\theta} G_i), \tag{6.4}$$

   one that depends only on the π_gs and the other only on the $\boldsymbol{\theta}_g$s.
3. Start with an initial guess $\Psi^{(0)}$.
4. Iterate **E-step** and **M-step** described below to convergence.

E-step

The *Expectation step* consists of starting with a parameter guess $\Psi^{(t-1)}$ and evaluating

$$Q(\Psi|\Psi^{(t-1)}) = \mathrm{E}_{G_1,\ldots,G_n|\boldsymbol{x};\Psi^{(t-1)}}(\log L_{\boldsymbol{x},G_1,\ldots,G_n}(\Psi)) :$$

the expected value of the complete-data log-likelihood. We can evaluate it by calculating (using the Bayes's rule)

$$q_{ig}^{(t-1)} = \Pr(G_i = g|\boldsymbol{x}; \Psi^{(t-1)}) = \frac{\pi_g^{(t-1)} f_g(\boldsymbol{x}_i; \boldsymbol{\theta} g^{(t-1)})}{\sum_{g'=1}^{K} \pi_{g'}^{(t-1)} f_{g'}(\boldsymbol{x}_i; \boldsymbol{\theta}_{g'}^{(t-1)})}$$
$$i = 1, \ldots, n,\ g = 1, \ldots, K,$$

then substituting them in as

$$Q(\Psi|\Psi^{(t-1)}) = \sum_{i=1}^{n}\sum_{g=1}^{K} q_{ig}^{(t-1)} \log \pi_g + \sum_{i=1}^{n}\sum_{g=1}^{K} q_{ig}^{(t-1)} \log f_g(\boldsymbol{x}_i;\boldsymbol{\theta}_g). \tag{6.5}$$

Observe that, like (6.4), (6.5) decomposes into a summation that depends only on the π_gs and a summation that depends only on the $\boldsymbol{\theta}_g$s.

M-step

The *Maximisation step* then consists of maximising the $Q(\Psi|\Psi^{(t-1)})$ with respect to Ψ to obtain the next parameter guess:

$$\Psi^{(t)} = \arg\max_{\Psi} Q(\Psi|\Psi^{(t-1)}), \text{ s.t. } \sum_{g=1}^{K} \pi_g = 1.$$

Conveniently, the form (6.5) separates the π_gs from the $\boldsymbol{\theta}_g$s, and so we can maximise them separately (i.e., if we differentiate with respect to one, the summation involving the other will vanish).

Maximising (6.5) with respect to $\boldsymbol{\theta}_g$s, we take the derivative

$$\frac{\partial Q(\Psi|\Psi^{(t-1)})}{\partial \boldsymbol{\theta}_g} = \sum_{i=1}^{n} q_{ig}^{(t-1)} \frac{\partial \log f_g(\boldsymbol{x}_i;\boldsymbol{\theta}_g)}{\partial \boldsymbol{\theta}_g},$$

and set to 0. This is a *weighted* maximum likelihood estimator.

Maximising (6.5) with respect to to π_gs is also straightforward. We will use Lagrange Multipliers to do so:

$$\text{Lag}(\boldsymbol{\pi}) = \sum_{i=1}^{n}\sum_{g=1}^{K} q_{ig}^{(t-1)} \log \pi_g - \alpha \left(\sum_{g=1}^{K} \pi_g - 1 \right).$$

Differentiating,

$$\text{Lag}'_g(\boldsymbol{\pi}) = \sum_{i=1}^{n} q_{ig}^{(t-1)} \pi_g^{-1} - \alpha.$$

Setting to 0,

$$\pi_g = \sum_{i=1}^{n} q_{ig}^{(t-1)} / \alpha.$$

Summing and solving for α,

$$\sum_{g=1}^{K} \pi_g = \frac{1}{\alpha} \sum_{g=1}^{K} \sum_{i=1}^{n} q_{ig}^{(t-1)} = 1,$$

$$\alpha = \sum_{g=1}^{K} \sum_{i=1}^{n} q_{ig}^{(t-1)}.$$

Therefore,

$$\pi_g^{(t)} = \frac{\sum_{i=1}^{n} q_{ig}^{(t-1)}}{\sum_{g=1}^{K} \sum_{i=1}^{n} q_{ig}^{(t-1)}}.$$

"Sharing" $\boldsymbol{\theta}$s

Lastly, recall that when we select one of the "$\mathbb{E}$" models and (6.3) in the 'Model-based clustering' section, we no longer have a separate $\boldsymbol{\theta}_g$ for every f_g. We may then need to redefine $\boldsymbol{\theta} \in \mathbb{R}^{Kp+1 \text{ or more}}$ to contain parameters for all groups (separate means, distinct variance parameters, etc.), and $f_g(\boldsymbol{x}_i; \boldsymbol{\theta})$ to "extract" those elements of $\boldsymbol{\theta}$ that it needs, with $\Psi = (\boldsymbol{\theta}, \boldsymbol{\pi})$.

Inferentially, $\boldsymbol{\theta}$ replaces $\boldsymbol{\theta}_g$ in all derivations above. In particular,

$$Q(\Psi|\Psi^{(t-1)}) \sum_{i=1}^{n} \sum_{g=1}^{K} q_{ig}^{(t-1)} \log \pi_g + \sum_{i=1}^{n} \sum_{g=1}^{K} q_{ig}^{(t-1)} \log f_g(\boldsymbol{x}_i; \boldsymbol{\theta}),$$

so

$$\frac{\partial Q(\Psi|\Psi^{(t-1)})}{\partial \boldsymbol{\theta}} = \sum_{i=1}^{n} \sum_{g=1}^{K} q_{ig}^{(t-1)} \frac{\partial \log f_g(\boldsymbol{x}_i; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}},$$

which is still a weighted MLE, but now it is joint for all groups, and without simplification.

Demonstration: Cluster analysis

This demonstration can be completed using the provided RStudio or your own RStudio.

To complete this task select the 'Cluster_Examples.demo.Rmd' in the 'Files' section of RStudio. Follow the demonstration contained within the RMD file.

If you choose to complete the example in your own RStudio, upload the following file:

Cluster_Examples.demo.Rmd

The contents of the RMD file are also displayed below:

Packages

```
library(here)
library(GGally)
# stats is loaded by default
library(readr)
library(dplyr)
library(candisc)
library(cluster)
library(mclust)
```

Iris example

Data

Suppose that we had mixed up our lab samples and no longer know which specimen comes from which species. Can we recover the species?

```
data(iris)
iris0 <- iris[-5]
```

This plotting function will be helpful for visualising the clusterings produced:

```
irisplot <- function(c){
# Plot the Iris data variables pairwise, with clusters identified by colour and
species by symbol.
    pairs(iris0, col=c+1, pch=as.numeric(as.factor(iris$Species)))
# Alternative approach: find the canonical discriminants for separating newly detected
clusters, and plot those:
    m <- lm(as.matrix(iris0)~factor(c)) # Fit a linear model with clustering as RHS.
    cd <- candisc(m)
```

```
    plot(cd, col=c("red","green","purple"))
}
```

Hierarchical clustering

Built-in (`stats` package)

We begin with hierarchical clustering performed by `hclust()`, using the Ward's method with squared Euclidean distance. The following code produces a complete hierarchy:

```
iris.h <- hclust(dist(iris0), method="ward.D2")
plot(iris.h)
```

We then cut it at 3 clusters

```
(iris.h3 <- cutree(iris.h,3)) # Three clusters
table(species=iris$Species, cluster=iris.h3) # Can it recover the species?
irisplot(iris.h3)
```

We observe 16 misclusterings, mostly virginicas being put into the majority-versicolor cluster.

The canonical discrimination plot identifies all the variables except for sepal width as being relatively redundant with each other for the purposes of this clustering.

`cluster` package

The `cluster` package allows for more flexible clustering, as well as support for silhouette calculation:

```
iris.h <- agnes(iris0, method="ward") # More flexible; see help.
irisplot(cutree(iris.h,3))
# Which cluster number gives the best average silhouette width?
d <- dist(iris0)
plot(2:10,sapply(2:10, function(k) summary(silhouette(cutree(iris.h,k),d))$avg.width))
plot(silhouette(cutree(iris.h,3), d))
```

The clustering is the same, but the silhouette plot does not select 3 clusters: in fact, it prefers 2, perhaps blurring together versicolor with virginica.

Non-hierarchical clustering

k-means

Now, let's try k-means from the `stats` package:

```
iris.3means <- kmeans(iris0,3)
```

```
irisplot(iris.3means$cluster)
table(species=iris$Species, cluster=iris.3means$cluster)
plot(2:10,sapply(2:10, function(k)
summary(silhouette(kmeans(iris0,k)$cluster,d))$avg.width))
plot(silhouette(iris.3means$cluster, d))
```

k-means has the same misclassification rate, again mostly virginicas being placed into majority-versicolor clusters.

k-medioids

k-medioids (a.k.a. "Partitioning Around Medioids"):

```
iris.3meds <- pam(iris0,3)
irisplot(iris.3meds$cluster)
table(species=iris$Species, cluster=iris.3meds$cluster)
plot(2:10,sapply(2:10, function(k)
summary(silhouette(pam(iris0,k)$cluster,d))$avg.width))
plot(silhouette(iris.3meds$cluster, d))
```

Similar results from k-medioids.

Model-based

Lastly, let's try to find the best model-based clustering:

```
# Automatic best model and cluster count selection via BIC:
(iris.bestMBC <- Mclust(iris0))
plot(iris.bestMBC, what=c("BIC", "classification"))
```

The model selected is the VEV model: varying dispersions, common eccentricities, and varying orientations. Two clusters are selected, again blurring virginica with versicolor.

Alternatively, let's try forcing 3 clusters:

```
(iris.3MBC <- Mclust(iris0,G=3))
plot(iris.3MBC, what=c("classification", "uncertainty", "density"))
irisplot(iris.3MBC$classification)
table(species=iris$Species, cluster=iris.3MBC$classification)
```

A VEV model is again selected. Forcing 3 clusters actually produces a significantly lower error rate than others, with only 5 versicolors clustered with the virginicas.

Challenge: Cluster analysis

If you choose to complete this task in your own RStudio, upload the following file:

Cluster_Examples.challenge.Rmd

Click on the 'Cluster_Examples.challenge.Rmd' in the 'Files' section to begin. Enter your response to the tasks in the 'Enter your code here' section.

This activity and the solution will be discussed at the Collaborate session this week. In the meantime, share and discuss your results in the 'Tutorials' discussion forum.

The solution will also be available here on Friday of this week by clicking on the 'Solution' tab in the top right corner.

The contents of the RMD file are also displayed below:

Packages

```
library(here)
library(GGally)
# stats is loaded by default
library(readr)
library(dplyr)
library(candisc)
library(cluster)
library(mclust)
```

Challenges

Recall the dataset `pizza.csv`, containing nutritional data from a variety of pizza brands:

`brand` : Pizza brand (class label)

`id` : Sample analysed

`prot` : Amount of protein per 100 grams in the sample

`fat` : Amount of fat per 100 grams in the sample

`ash` : Amount of ash per 100 grams in the sample

`sodium` : Amount of sodium per 100 grams in the sample

`carb` : Amount of carbohydrates per 100 grams in the sample

`cal` : Amount of calories per 100 grams in the sample

```
pizza <- read_csv(here("datasets","pizza.csv"))
ggpairs(pizza, mapping=aes(col=brand, alpha=0.3))
```

Suppose that we have lost track of which pizza came from which brand, i.e.,

```
pizza0 <- select(pizza, -brand)
```

Task 1: Supposing that we know that there are 10 brands, try using the various clustering methods to group the pizzas by brand. How well can you recover them?

Task 2: Now, suppose that we don't know how many brands there were. How many would you infer from the unlabelled data?

Topic 2: Copula methods

Copulae

Formulation

For the multivariate normal, independence is equivalent to absence of correlation between any two components. In this case the joint cdf is a product of the marginals. When the independence is violated, the relation between the joint multivariate distribution and the marginals is more involved. An interesting concept that can be used to describe this more involved relation is the concept of *copula*. We focus on the two-dimensional case for simplicity. Then the copula is a function $C : [0,1]^2 \to [0,1]$ with the properties:

1. $C(0,u) = C(u,0) = 0$ for all $u \in [0,1]$.
2. $C(u,1) = C(1,u) = u$ for all $u \in [0,1]$.
3. For all pairs $(u_1,u_2),(v_1,v_2) \in [0,1] \times [0,1]$ with $u_1 \leq v_1, u_2 \leq v_2$:

$$C(v_1,v_2) - C(v_1,u_2) - C(u_1,v_2) + C(u_1,u_2) \geq 0.$$

The name is due to the implication that the copula links the multivariate distribution to its marginals. This is explicated in the following theorem:

Theorem 6.1. *Let $F(\cdot,\cdot)$ be a joint cdf with marginal cdf's $F_{X_1}(.)$ and $F_{X_2}(.)$. Then there exists a copula $C(\cdot,\cdot)$ with the property*

$$F(x_1,x_2) = C(F_{X_1}(x_1), F_{X_2}(x_2))$$

for every pair $(x_1,x_2) \in \mathbb{R}^2$. When $F_{X_1}(\cdot)$ and $F_{X_2}(\cdot)$ are continuous the above copula is unique. Vice versa, if $C(\cdot,\cdot)$ is a copula and $F_{X_1}(\cdot), F_{X_2}(\cdot)$ are cdf then the function $F(x_1,x_2) = C(F_{X_1}(x_1), F_{X_2}(x_2))$ is a joint cdf with marginals $F_{X_1}(\cdot)$ and $F_{X_2}(\cdot)$.

Taking derivatives we also get:

$$f(x_1,x_2) = c(F_{X_1}(x_1), F_{X_2}(x_2)) f_{X_1}(x_1) f_{X_2}(x_2)$$

where

$$c(u,v) = \frac{\partial^2}{\partial u \partial v} C(u,v)$$

is the *density* of the copula. This relation clearly shows that the contribution to the joint density of X_1, X_2 comes from two parts: one that comes from the copula and is "responsible" for the dependence ($c(u,v) = \frac{\partial^2}{\partial u \partial v} C(u,v)$) and another one which takes into account marginal information only ($f_{X_1}(x_1) f_{X_2}(x_2)$).

It is also clear that the independence implies that the corresponding copula is $\Pi(u,v) = uv$ (this is called the independence copula).

These concepts are generalised also to p dimensions with $p > 2$.

Common copula types

An interesting example is the *Gaussian copula*. For $p = 2$ it is equal to

$$\begin{aligned} C_\rho(u, v) &= \Phi_\rho(\Phi^{-1}(u), \Phi^{-1}(v)) \\ &= \int_{-\infty}^{\Phi^{-1}(u)} \int_{-\infty}^{\Phi^{-1}(v)} f_\rho(x_1, x_2) dx_2 dx_1. \end{aligned}$$

Here $f_\rho(\cdot, \cdot)$ is the joint bivariate normal density with zero mean, unit variances and a correlation ρ, $\Phi_\rho(\cdot, \cdot)$ is its cdf, and $\Phi^{-1}(\cdot)$ is the inverse of the cdf of the standard normal. (This is "The formula that killed Wall street".) When $\rho = 0$ we see that we get $C_0(u, v) = uv$ (as is to be expected).

Non-Gaussian copulae are much more important in practice and inference methods about copulae are a hot topic in Statistics. The reason for importance of non-Gaussian copulae is that Gaussian copulae do not allow us to model reasonably well the tail dependence, that is, joint *extreme* events have virtually a zero probability. Especially in financial applications, it is very important to be able to model dependence in the tails.

A *Multivariate-t copula* uses the multivariate t distribution instead. This distribution has heavier tails, which allows it to model extreme events better.

The Gumbel–Hougaard copula is much more flexible in modeling dependence in the upper tails. For an arbitrary dimension p is is defined as

$$C_\theta^{\mathrm{GH}}(u_1, u_2, \ldots, u_p) = \exp\left\{ -\left[\sum_{j=1}^{p} (-\log u_j)^\theta \right]^{1/\theta} \right\},$$

where $\theta \in [1, \infty)$ is a parameter that governs the strength of the dependence. You can easily see that the Gumbell–Hougaard copula reduces to the independence copula when $\theta = 1$ and to the Fréchet–Hoeffding upper bound copula $\min(u_1, \ldots, u_p)$ when $\theta \to \infty$.

The Gumbel–Hougaard copula is also an example of the so-called *Archimedean* copulae. The latter are characterised by their *generator* $\phi(\cdot)$: a continuous, strictly decreasing, convex function from $[0, 1]$ to $[0, \infty)$ such that $\phi(1) = 0$. Then the Archimedean copula is defined via the generator as follows:

$$C(u_1, u_2, \ldots, u_p) = \phi^{-1}(\phi(u_1) + \cdots + \phi(u_p)).$$

Here, $\phi^{-1}(t)$ is defined to be 0 if t is not in the image of $\phi(\cdot)$.

Example 6.2. Show that the Gumbell–Hougaard copula is an Archimeden copula with a generator $\phi(t) = (-\log t)^\theta$.

The benefit of using the Archimedean copulae is that they allow for simple description of the p-dim dependence by using a function of one argument only (the generator). However, it is seen immediately that the Archimedean copula is symmetric in its arguments and this limits its applicability for modelling dependencies that are not symmetric in their arguments. The so-called *Liouville* copulae are an extension of the Archiemedean copulae and can be used also to model dependencies that are not symmetric in their arguments.

Computing

R: Packages `copula`, `VineCopula`, and others.

Optional viewing: Gaussian copula

Bionic Turtle. (2009). Gaussian copula. Retrieved from: https://youtu.be/z43_pf5Y6A8

Check your understanding

Complete the below exercises to check your understanding of concepts presented so far.

The (p-dimensional) Clayton copula is defined for a given parameter $\theta > 0$ as

$$C_\theta\left(u_1, u_2, \ldots, u_p\right) = \left[\sum_{i=1}^{p} u_i^{-\theta} - p + 1\right]^{-1/\theta}$$

Show that it is an Archimedean copula and that its generator is $\phi(x) = \theta^{-1}(x^{-\theta} - 1)$.

Demonstration: Copula methods

This demonstration can be completed using the provided RStudio or your own RStudio.

To complete this task select the 'Copula_Examples.demo.Rmd' in the 'Files' section of RStudio. Follow the demonstration contained within the RMD file.

If you choose to complete the example in your own RStudio, upload the following file:

Copula_Examples.demo.Rmd

The contents of the RMD file are also displayed below:

Packages

```
library(here)
library(readr)
library(dplyr)
library(purrr)
library(reshape2)
library(copula)
library(GGally)
library(stats4) # mle()
library(Rsolnp)
```

Microwave Ovens Example

Data

Recall the data about microwave ovens and their radiation when open as opposed to closed. We noted that the distributions were far from normal, and we therefore transformed them by taking the fourth root. Copula models allow us to model them directly.

Load the data:

```
ovens <- read_csv(here("datasets","ovens.csv"))
ggpairs(ovens)
```

Gaussian Copula

Using empirical CDF

We begin by obtaining the empirical quantiles of the observations:

```
head(ovensU <- pobs(ovens)) # Empirical quantiles of observations
```

A Gaussian copula can be fit using the `fitCopula()` function, with `normalCopula(dim=2)` specifying a 2-dimensional Gaussian copula. Method `irho` fits by trying to match the empirical correlation to the one induced by copula.

```
(e_fit <- fitCopula(normalCopula(dim=2), ovensU, method="irho"))
```

```
head(ovensN <- matrix(qnorm(ovensU), ncol=2)) # Map the empirical quantiles onto
normal quantiles
cor(ovensN)
```

Notice that the parameter estimates are the same.

Using Gamma margins

Now, let's suppose that we think that microwave radiation measurements are gamma-distributed. How do we incorporate this assumption? For this, we have `mvdc`, which lets us do precisely that, specifying the copula, the margin families, and their parameters. (Note that we could mix the families, if we wanted to.)

We then use the `fitMvdc()` function, specifying the initial parameter values. This produces a parametric copula fit, including confidence intervals, and a fitted copula model from which we can simulate.

```
normGGc <- mvdc(normalCopula(dim=2), margins=c("gamma","gamma"),
                paramMargins = list(list(shape=1, rate=1),
                                    list(shape=1, rate=1)))
(g_fit <- fitMvdc(as.matrix(ovens), normGGc, start = c(
1,1, # First Gamma
1,1, # Second Gamma
0 # Copula correlation
)))
summary(g_fit)
confint(g_fit)
normGGc_fitted <- g_fit@mvdc

# Simulate another dataset from the fit:
(rovens <- as.data.frame(rMvdc(nrow(ovens), normGGc_fitted)))
names(rovens) <- names(ovens)
ggpairs(as.data.frame(rovens))

# Bigger dataset:
(rovens <- as.data.frame(rMvdc(1000, normGGc_fitted)))
names(rovens) <- names(ovens)
ggpairs(as.data.frame(rovens))
```

Stocks example

Data

Recall that the following stocks were considered:

- IBM (IBM)
- Microsoft (MSFT)
- British Petroleum (BP)
- Coca-Cola (KO)
- Duke Energy (DUK)

We loaded the data, combined them, and sorted them by date. This is all data management, but I suggest looking up help for each of the functions here to see what they do.

```
symbols <- c("IBM", "MSFT", "BP", "KO", "DUK")
path <- here("datasets","stocks")

readstock <- function(symbol, path){
    file.path(path, paste0(symbol,".csv")) %>%
        read_csv() %>% select(Date, `Adj Close`) %>%
        set_names(c("Date", symbol))
}
stocks <- symbols %>% map(readstock, path=path) %>% reduce(left_join, by="Date") %>%
arrange(Date)
stocks %>% melt(id="Date", variable.name="Stock", value.name="Value") %>%
ggplot(aes(x=Date, y=Value, colour=Stock, group=Stock)) + geom_line()
```

While we could work with the daily return on investment,

$$= \frac{\text{Price today}}{\text{Price yesterday}} - 1,$$

it is more convenient to work on log scale:

$$\log \frac{\text{Price today}}{\text{Price yesterday}}.$$

Then,

$$\log \frac{\text{Price today}}{\text{Price 2 days ago}} = \log \frac{\text{Price today}}{\text{Price yesterday}} + \log \frac{\text{Price yesterday}}{\text{Price 2 days ago}},$$

so we can model log-returns over long periods as sums of log-returns over short periods. We therefore calculated log-daily returns:

```
returns <- stocks %>% select(-Date) %>% map(~log(./lag(.))) %>% as_tibble %>% na.omit
```

```
ggpairs(returns)
```

As is often the case for stock returns, they are correlated and there are outliers and long tails, even on the log scale. We therefore converted the observations to quantiles, while mostly preserving the correlations:

```
returnsU <- pobs(returns) # Empirical quantiles of observations
ggpairs(as_tibble(returnsU))
```

Fitting and simulation

We used a *Multivariate t-Copula* with an unstructured (`dispstr="un"`) covariance matrix. Other options would have included exchangeable---which would have assumed that all pairwise correlations are equal. We also specified maximum likelihood to be used. We then simulated from the fitted copula and plotted it.

```
(t_fit <- fitCopula(tCopula(dim=ncol(returnsU), dispstr="un"), returnsU, method="ml"))
# NB: We can use getSigma() to extract the estimated correlation matrix:
getSigma(t_fit@copula)
# We then took our fitted model, and simulated the empirical quantiles.
simreturnsU<- rCopula(100000,t_fit@copula) %>% as_tibble %>%  set_names(symbols)

# This is a function sorely needed in the copula package: inverse of pobs.
qobs <- function(p, x, lower.tail=FALSE, log.p=FALSE){
    p <- cbind(p)
    x <- cbind(x)
    if(ncol(p)!=ncol(x)) stop("Dimension of pseudo observation does not match the
dimension of the original distribution.")
    if(log.p) p <- exp(p)
    if(lower.tail) p <- 1-p
    sapply(seq_len(ncol(p)), function(i) quantile(x[,i], probs=p[,i]))
}

# Map returns back to the original scale:
simreturns <- qobs(simreturnsU, returns) %>% as_tibble %>% set_names(symbols)
pairs(simreturns, pch=".")
```

Goodness-of-fit testing

We can also check whether this t-Copula does a good job representing the correlation among the stocks. The function `gofCopula()` provides several tests for this. Note that it currently has a limitation that it cannot handle non-integer degrees of freedom, so we are going to extract them from the fit and round.

This procedure is very time-consuming, so we'll set the simulation size to only 40.

```
gofCopula(tCopula(dim=ncol(returnsU), dispstr="un",
df=round(t_fit@copula@parameters[11]), df.fixed=TRUE), returns, N=40,
estim.method="ml")
```

Based on the high p-value, we seem to be OK.

Portfolio returns

We can now try different portfolio strategies.

Equally weighted portfolio

```
eq_rets <- rowMeans(exp(simreturns))-1 # I.e., rowwise sums divided by 5.
m.equal <- mean(eq_rets)
sd.equal <- sd(eq_rets)
```

Suppose that we invest \$1, splitting it iequally among the stocks, so \$0.20 into each stock. Then, average daily return will be about $rm.equal$ (with standard deviation $rsd.equal$). We expect to lose money $rmean(eq_rets < 0)$ of the days. And, we expect to lose more than 1% $rmean(eq_rets < -.01)$ of the days.

We can also plot the density of daily returns:

```
qplot(eq_rets, geom="density")+xlab("Daily portfolio return")
```

A better portfolio

Some stocks are correlated, so we might want to try hedging. Can we get the $rm.equal$ return more reliably?

Here, we use constrained optimisation from a package for constrained optimisation. (THIS IS NOT EXAMINABLE!)

```
rets <- function(w) rowSums(sweep(exp(simreturns), 2, w, `*`))-1

eqfun <- function(w) c(mean(rets(w))-m.equal, sum(w)-1)
objfun <- function(w) sd(rets(w))

best_w <- gosolnp(fun=objfun, eqfun=eqfun, eqB=c(0,0), LB=rep(0,5), UB=rep(1,5),
n.sim=100)$pars
names(best_w) <- symbols
best_rets <- rets(best_w)
best_w
```

Notice that only one of IBM and MSFT taken. The average daily return is still $rmean(best_rets)$, but the standard deviation is now about $rsd(best_rets)$, and the probabilities of loss have decreased

accordingly to $rmean(best_rets < 0)$ and $rmean(best_rets < -.01)$, respectively.

Challenge: Copula methods

If you choose to complete this task in your own RStudio, upload the following file:

Copula_Examples.challenge.Rmd

Click on the 'Cancor_Examples.challenge.Rmd' in the 'Files' section to begin. Enter your response to the tasks in the 'Enter your code here' section.

This activity and the solution will be discussed at the Collaborate session this week. In the meantime, share and discuss your results in the 'Tutorials' discussion forum.

The solution will also be available here on Friday of this week by clicking on the 'Solution' tab in the top right corner.

The contents of the RMD file are also displayed below:

Packages

```
library(here)
library(readr)
library(dplyr)
library(purrr)
library(reshape2)
library(copula)
library(GGally)
library(stats4) # mle()
library(Rsolnp)
```

Challenges

Recall the red wines dataset:

- `fixed acidity`
- `volatile acidity`
- `free sulfur dioxide`
- `total sulfur dioxide`
- `density`
- `pH`
- `sulphates`

```
red <- read_csv(here("datasets","winequality-red.extract7.csv"))
```

Previously, we would have transformed these data. Let's try to use copulas instead.

Task 1: Fit and diagnose a Gaussian copula with nonparametric margins to these data.