Mathematical Models and Applications

Multivariate Analysis

Pedro Cristiano Silva

Instituto Superior de Agronomia

2022-23

Outline

- **LINEAR ALGEBRA**
- **PRINCIPAL COMPONENT ANALYSIS**
- **CLUSTER ANALYSIS**
- LINEAR DISCRIMINANT ANALYSIS

Notations

- Non bold letters (upper or lower case) represent scalar quantities: x, y, A,...
- Lower case bold letters represent vectors $\mathbf{x}$, $\mathbf{y}$, $\vec{\mathbf{x}}$, $\vec{\mathbf{y}}$,...
- Upper case bold letters represent matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{X}$, $\mathbf{Y}$,...

LINEAR ALGEBRA

Eigenvalues and eigenvectors

Definition

$\mathbf{A}_{p\times p} = [a_{ij}]$ *a square matrix of order* p. *A vector* $\mathbf{v} \in \mathbb{R}^p$, $\mathbf{v} \neq \vec{0}$, *is called an* **eigenvector** *of* $\mathbf{A}$ *if there is* $\lambda \in \mathbb{R}$ *such that* $\mathbf{Av} = \lambda\mathbf{v}$. λ *is called the corresponding* **eigenvalue**.

Example

$$\mathbf{A} = \begin{bmatrix} 3 & 0 & 2 \\ 0 & -1 & 1 \\ 2 & 0 & 0 \end{bmatrix}, \qquad \mathbf{v} = \begin{bmatrix} 10 \\ 1 \\ 5 \end{bmatrix}$$

We have

$$\mathbf{Av} = \begin{bmatrix} 3 & 0 & 2 \\ 0 & -1 & 1 \\ 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} 10 \\ 1 \\ 5 \end{bmatrix} = \begin{bmatrix} 40 \\ 4 \\ 20 \end{bmatrix} = 4 \begin{bmatrix} 10 \\ 1 \\ 5 \end{bmatrix} = 4\mathbf{v}$$

Hence $\mathbf{v}$ *is an eigenvector of* $\mathbf{A}$ *associated to the eigenvalue* $\lambda = 4$.

Eigenvalues and eigenvectors (cont.)

- The spectrum of $\mathbf{A}$, denoted $\sigma(\mathbf{A})$, is the collection of the p eigenvalues of $\mathbf{A}$ (including repetitions), i.e., the collection of p roots (real and complex) of its **characteristical polynomial**, $p_{\mathbf{A}}(x) = \det(\mathbf{A} - x\mathbf{I}_p)$ (which has degree p)
- The eigenspace associated with an eigenvalue λ, denoted $E(\lambda)$, is the linear space spanned by the eigenvectors associated with λ
- The trace of $\mathbf{A}$, denoted $\mathrm{tr}(\mathbf{A})$, is the sum of all diagonal elements of $\mathbf{A}$ and equals the sum of all eigenvalues of $\mathbf{A}$ (including repetitions):

$$\mathrm{tr}(\mathbf{A}) = a_{11} + a_{22} + \cdots + a_{pp} = \sum_{\lambda \in \sigma(\mathbf{A})} \lambda$$

- The determinant of $\mathbf{A}$ (not defined here) equals the product of all eigenvalues of $\mathbf{A}$ (including repetitions):

$$\det \mathbf{A} = \prod_{\lambda \in \sigma(\mathbf{A})} \lambda$$

$\mathbf{A}$ is invertible $\Leftrightarrow \det(\mathbf{A}) \neq 0 \Leftrightarrow 0$ is not an eigenvalue of $\mathbf{A}$

Example revisited

Returning to the example of slide 6 we have the the following:

- $\sigma(\mathbf{A}) : \mathbf{-1}, \mathbf{-1}, \mathbf{4}$
- $\mathrm{tr}(\mathbf{A}) = 3 + (-1) + 0 = 2$ corresponds to the sum of its diagonal elements which is also equal to sum of its eigenvalues (counting with repetitions): $(-1) + (-1) + 4 = 2$
- $\det(\mathbf{A}) = (-1) \times (-1) \times 4 = 4 \neq 0$ which is equal to the product of its eigenvalues (counting with repetitions)
- $E(-1) = \langle (1, 1, 0) \rangle$ has dim=1
- $E(4) = \langle (0, 1, 5) \rangle$ has dim=1

Since $\dim E(-1) + \dim E(4) = 2 < 3 = p$, $\mathbf{A}$ is not **diagonalizable**, i.e., there isn't an invertible matrix $\mathbf{P}$ and a diagonal matrix $\mathbf{\Lambda}$ such that $\mathbf{A} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^{-1}$

Exercise

Verify that $(1, 1, 0)$ *is an eigenvector of* $\mathbf{A}$ *associated to the eigenvalue* $\lambda = -1$

Eigenvalues and eigenvectors - R

R

```
A=matrix(c(3,0,2,0,-1,1,2,0,0),ncol=3,byrow=TRUE)
A
EV<-eigen(A) # eigenvalues and eigenvectors of A
det(A) # determinant of A
tr<-sum(diag(A)) # trace of A
tr
```

Orthonormal sets

Definition

Given $\mathbf{v}_1, \ldots, \mathbf{v}_q \in \mathbb{R}^p$ *with* $q \leq p$ *we say that* $\{\mathbf{v}_1, \ldots, \mathbf{v}_q\}$ *is an* **orthonormal set** *if*

$$\|\mathbf{v}_i\| = 1, \forall i \quad \textit{and} \quad \mathbf{v}_i \perp \mathbf{v}_j \quad (i \neq j)$$

If $q = p$, $\{\mathbf{v}_1, \ldots, \mathbf{v}_q\}$ *is called an* **orthonormal basis** *of* $\mathbf{R}^p$

Denoting by $\mathbf{V}_{p\times q} = [\, \mathbf{V}_1 \quad \cdots \quad \mathbf{V}_q \,]$ the matrix whose columns are the q vectors, $\mathbf{v}_1, \ldots, \mathbf{v}_q$, we have the following:

- $\{\mathbf{v}_1, \ldots, \mathbf{v}_q\}$ is an orthonormal set iff $\mathbf{V}^T\mathbf{V} = \mathbf{I}_q$
- $\{\mathbf{v}_1, \ldots, \mathbf{v}_p\}$ is an orthonormal basis iff $\mathbf{V}^T\mathbf{V} = \mathbf{V}\mathbf{V}^T = \mathbf{I}_p$ iff $\mathbf{V}^{-1} = \mathbf{V}^T$

  In this later case we can write for all $\mathbf{u} \in \mathbb{R}$,

$$\mathbf{u} = \underbrace{(\mathbf{u}^T\mathbf{v}_1)\mathbf{v}_1}_{\text{proj}_{\mathbf{v}_1}(\mathbf{u})} + \cdots + \underbrace{(\mathbf{u}^T\mathbf{v}_p)\mathbf{v}_p}_{\text{proj}_{\mathbf{v}_p}(\mathbf{u})} \tag{1}$$

Orthonormal basis

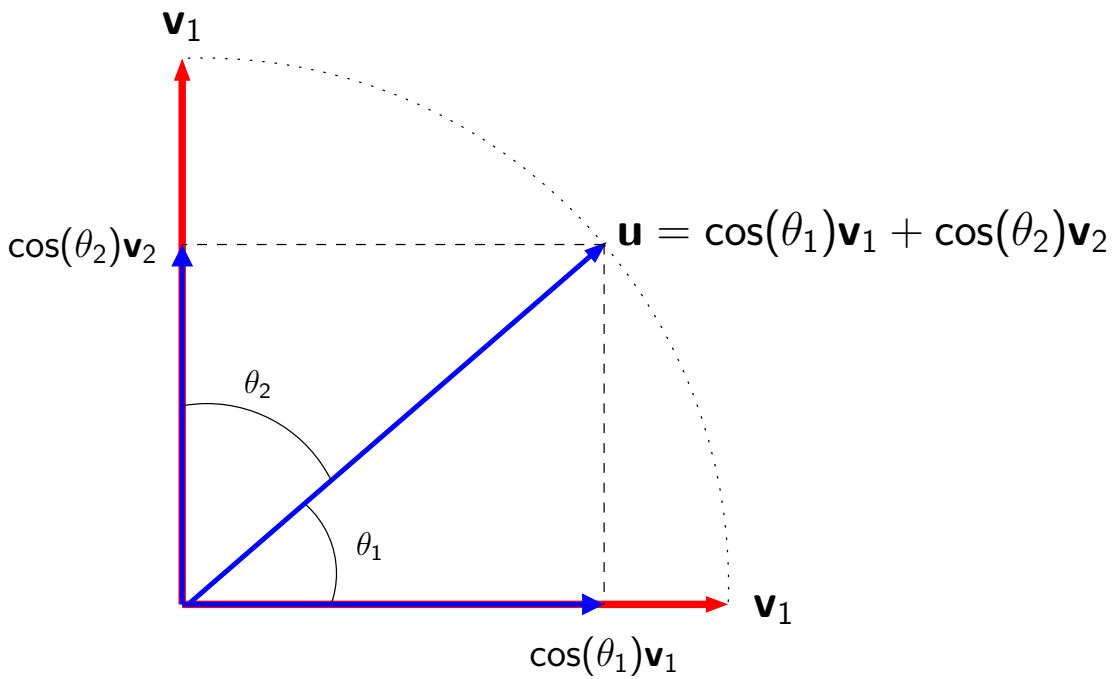
Denote θ_i, $i = 1, \ldots, p$, the angle between $\mathbf{u}$ and $\mathbf{v}_i$

If $\|\mathbf{u}\| = 1$, we have applying (1) of slide 9

$$\boxed{\mathbf{u} = \cos(\theta_1)\mathbf{v}_1 + \cdots + \cos(\theta_p)\mathbf{v}_p}$$

with $\cos^2(\theta_1) + \cdots + \cos^2(\theta_p) = 1$.

The case $p = 2$:



Interlude: matrix multiplications

If $\mathbf{A}_{m\times n} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n \\ | & | & & | \end{bmatrix}$ with $\mathbf{a}_j \in \mathbb{R}^n$, $j = 1, \ldots, n$, and

$$\mathbf{B}_{n\times p} = \begin{bmatrix} -\mathbf{b}_1^T- \\ -\mathbf{b}_2^T- \\ \vdots \\ -\mathbf{b}_n- \end{bmatrix} \text{ then } AB = \sum_{j=1}^n \mathbf{a}_j\mathbf{b}_j^T$$

Example

$$\left[\begin{array}{c|c} 1 & 3 \\ 2 & 4 \end{array}\right] \left[\begin{array}{cc} 1 & 2 \\ \hline 3 & -1 \end{array}\right] = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \begin{bmatrix} 1 & 2 \end{bmatrix} + \begin{bmatrix} 3 \\ 4 \end{bmatrix} \begin{bmatrix} 3 & -1 \end{bmatrix}$$

$$= \begin{bmatrix} 10 & -1 \\ 14 & 0 \end{bmatrix}$$

Note that if $\mathbf{b} = (\beta_1, \ldots, \beta_n)$ one gets, $\mathbf{Ab} = \mathbf{A} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix} = \sum_{j=1}^n \beta_j \mathbf{a}_j$

Eigenvalue decomposition of a symmetric matrix

Theorem

Let $\mathbf{A}$ *be a symmetric matrix* ($\mathbf{A}^T = \mathbf{A}$) *of order* p. *Then we can find matrices* $\mathbf{V}_{p\times p}$ *and* $\mathbf{\Delta}_{p\times p}$, *such that*

$$\boxed{\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T} \tag{2}$$

where:

- $\mathbf{V} = [\,\mathbf{v}_1\ \mathbf{v}_2\ \cdots\ \mathbf{v}_p\,]$ *verify* $\mathbf{V}^T\mathbf{V} = \mathbf{V}\mathbf{V}^T = \mathbf{I}_p$: *matrix of (unit and pairwise orthogonal) eigenvectors of* $\mathbf{A}$
- $\mathbf{\Lambda} = diag(\lambda_1, \lambda_2, \ldots, \lambda_p)$ *with* $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p$: *diagonal matrix containing the corresponding eigenvalues of* $\mathbf{A}$ ($\mathbf{A}\mathbf{v}_i = \lambda_i\mathbf{v}_i$)

Using the decomposition of a matrix product in terms of sums of columns and rows products described in slide 11, we can rewrite (2) as,

$$\boxed{\mathbf{A} = \lambda_1\mathbf{v}_1\mathbf{v}_1^T + \lambda_2\mathbf{v}_2\mathbf{v}_2^T + \cdots + \lambda_p\mathbf{v}_p\mathbf{v}_p^T,}$$

which is called the **spectral decomposition** of $\mathbf{A}$

Singular value decomposition of an arbitrary matrix

Theorem

Let $\mathbf{A}$ *be matrix of type* $N \times p$ *and rank* r*. Then we can find matrices* $\mathbf{U}_{N\times r}$, $\mathbf{\Delta}_{r\times r}$ *and* $\mathbf{V}_{p\times r}$, *such that*

$$\boxed{\mathbf{A} = \mathbf{U}\mathbf{\Delta}\mathbf{V}^T} \tag{3}$$

where:

- $\mathbf{U} = [\,\mathbf{u}_1 \ \cdots \ \mathbf{u}_r\,]$ *verify* $\mathbf{U}^T\mathbf{U} = \mathbf{I}_r$*: matrix of (unit and pairwise orthogonal) left singular vectors of* $\mathbf{A}$
- $\mathbf{V} = [\,\mathbf{v}_1 \ \cdots \ \mathbf{v}_r\,]$ *verify* $\mathbf{V}^T\mathbf{V} = \mathbf{I}_r$*: matrix of (unit and pairwise orthogonal) right singular vectors of* $\mathbf{A}$
- $\mathbf{\Delta} = diag(\delta_1, \ldots, \delta_r)$ *with* $\delta_1 \geq \cdots \geq \delta_r > 0$*: diagonal matrix of the nonzero singular values of* $\mathbf{A}$ *(*$\mathbf{A}\mathbf{v}_i = \delta_i\mathbf{u}_i$ *and* $\mathbf{A}^T\mathbf{u}_i = \delta_i\mathbf{v}_i$*)*

Using the results of slide 11 we can rewrite (3) as,

$$\boxed{\mathbf{A} = \delta_1\mathbf{u}_1\mathbf{v}_1^T + \delta_2\mathbf{u}_2\mathbf{v}_2^T + \cdots + \delta_r\mathbf{u}_r\mathbf{v}_r^T,}$$

which is called the **singular value decomposition** of $\mathbf{A}$

Summary statistics - univariate case

Given $\mathbf{x} = (x_1, \ldots, x_N), \mathbf{y} = (y_1, \ldots, y_N) \in \mathbb{R}^N$ vectors of N observed measurements of two variables/features, we define:

- (sample) mean of $\mathbf{x}$:

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i$$

- (sample) variance of $\mathbf{x}$:

$$s_x^2 = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \bar{x})^2$$

- (sample) covariance between $\mathbf{x}$ and $\mathbf{y}$:

$$s_{xy}^2 = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \bar{x})(y_i - \bar{y}) = \frac{1}{N-1} (\mathbf{x}^*)^T \mathbf{y}^*,$$

  where $\mathbf{x}^* = (x_1 - \bar{x}, \ldots, x_N - \bar{x})$ and $\mathbf{y}^* = (y_1 - \bar{y}, \ldots, y_N - \bar{y})$ are the corresponding centered vectors

- (sample) linear correlation coefficient between $\mathbf{x}$ and $\mathbf{y}$:

$$r_{xy}^2 = \frac{s_{xy}^2}{s_x s_y}$$

Variable's cloud and individual's cloud

$\mathbf{X}_{N\times p} = [\, x_{ij} \,]$ a data matrix with $x_{ij} \in \mathbb{R}$

- Each column of $\mathbf{X}$ represents an observed variable, i.e., the measurements of some variable/feature across N individuals:

  $$\mathbf{X}_{N\times p} = [\mathbf{x}_1 \ \cdots \ \mathbf{x}_p] \quad \text{with} \quad \mathbf{x}_j = (x_{1j}, \ldots, x_{Nj}) \in \mathbb{R}^N, \quad j = 1, \ldots, p$$

  We obtain in this way a cloud of p points in $\mathbb{R}^N$ - variable's cloud

- Each row of $\mathbf{X}$ represents the observations of p variables/features of a single individual:

  $$\mathbf{X}^T_{p\times N} = [\mathbf{x}^1 \ \cdots \ \mathbf{x}^N] \quad \text{with} \quad \mathbf{x}^i = (x_{i1}, \ldots, x_{ip}) \in \mathbb{R}^p, \quad i = 1, \ldots, N$$

  We obtain in this way a cloud of N points in $\mathbb{R}^p$ - individuals's cloud

Summary statistics - multivariate case

- The (sample) mean of $\mathbf{X}$, i.e., the cloud's center of gravity, is

$$\mathbf{x}^G = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}^i \in \mathbb{R}^p,$$

that is, $\mathbf{x}^G = (\bar{x}_1, \ldots, \bar{x}_p)$ with $\bar{x}_j = \frac{1}{N}\sum_{i=1}^{N} x_{ij}$

- The (sample) covariance matrix of $\mathbf{X}$ is

$$\mathbf{S} = [s_{jk}^2] = \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{x}^i - \mathbf{x}^G)(\mathbf{x}^i - \mathbf{x}^G)^T,$$

where the (sample) covariance between variables j and k is equal to

$$s_{jk}^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k)$$

- The total variation of $\mathbf{X}$ is

$$\text{tr}(\mathbf{S}) = s_{11}^2 + \cdots + s_{kk}^2 = \frac{1}{N-1}\sum_{j=1}^{p}\sum_{i=1}^{N}(x_{ij} - \bar{x}_j)^2 = \frac{1}{N-1}\sum_{i=1}^{N}\|\mathbf{x}^i - \mathbf{x}^G\|^2$$

Centered data matrix and covariance

For each $j = 1, \ldots, p$, the centered vector of the N observations of variable j is

$$\mathbf{x}_j^* = (x_{1j} - \bar{x}_j, \ldots, x_{Nj} - \bar{x}_j) \in \mathbb{R}^N,$$

The (sample) covariance s_{jk}^2 can then be written, using the centered variables $\mathbf{x}_j^*$ and $\mathbf{x}_k^*$, as a simple inner product (in $\mathbb{R}^N$) divided by $N - 1$,

$$s_{jk}^2 = cov(\mathbf{x}_j, \mathbf{x}_k) = \frac{1}{N-1}(\mathbf{x}_j^*)^T \mathbf{x}_k^* \tag{4}$$

Likewise, defining the centered data matrix as

$$\mathbf{X}^* = [\, \mathbf{x}_1^* \ \cdots \ \mathbf{x}_p^* \,],$$

i.e.,

$$(\mathbf{X}^*)^T = [\, (\mathbf{x}^1 - \mathbf{x}^G) \ \cdots \ (\mathbf{x}^N - \mathbf{x}^G) \,],$$

the covariance matrix $\mathbf{S} = [s_{jk}^2]$ of $\mathbf{X}$ can be written as

$$\boxed{\mathbf{S} = \tfrac{1}{N-1}(\mathbf{X}^*)^T \mathbf{X}^*}$$

Standardized data matrix and correlation

For each $j = 1, \ldots, p$, the vector of the N observations of standardized variable j is

$$\mathbf{z}_j = \left(\frac{x_{1j} - \bar{x}_j}{s_j}, \ldots, \frac{x_{Nj} - \bar{x}_j}{s_j} \right) = \left(\frac{x^*_{1j}}{s_j}, \ldots, \frac{x^*_{Nj}}{s_j} \right) \in \mathbb{R}^N$$

and we obtain the corresponding standardized data matrix,

$$\mathbf{Z} = [\, \mathbf{z}_1 \ \cdots \ \mathbf{z}_p \,]$$

- The (sample) linear correlation coefficient between variables j and k is

$$r_{jk} = \frac{s^2_{jk}}{s_j s_k} = \frac{1}{N-1} \sum_{i=1}^{N} \left(\frac{x_{ij} - \bar{x}_j}{s_j} \right) \left(\frac{x_{ik} - \bar{x}_k}{s_k} \right) = \frac{1}{N-1} \mathbf{z}_j^T \mathbf{z}_k$$

- Hence the (sample) correlation matrix $\mathbf{R} = [r_{ij}]$ of $\mathbf{X}$ equals the covariance matrix of the standardized data matrix, i.e.,

$$\boxed{\mathbf{R} = \tfrac{1}{N-1} \mathbf{Z}^T \mathbf{Z}}$$

- The total variation of $\mathbf{Z}$ is

$$\operatorname{tr}(\mathbf{R}) = r_{11} + \cdots + r_{pp} = p$$

Principal component analysis - statistical motivation

- Principal component analysis (PCA) is a statistical multvariate method that aims to reduce the dimensionality of a dataset $\mathbf{X}$ while preserving its information, i.e., the variability between individuals, as much as possible
- This goal is achieved by defining a set of uncorrelated variables, called **principal components**, that are linear combinations of the original (or standardized) variables, in such a way that the first few principal components explains a large proportion of the total variability of the data set
- The dimension reduction is (particularly) effective when the original variables are (highly) correlated
- PCA is probably the most widely used multivariate statistical method

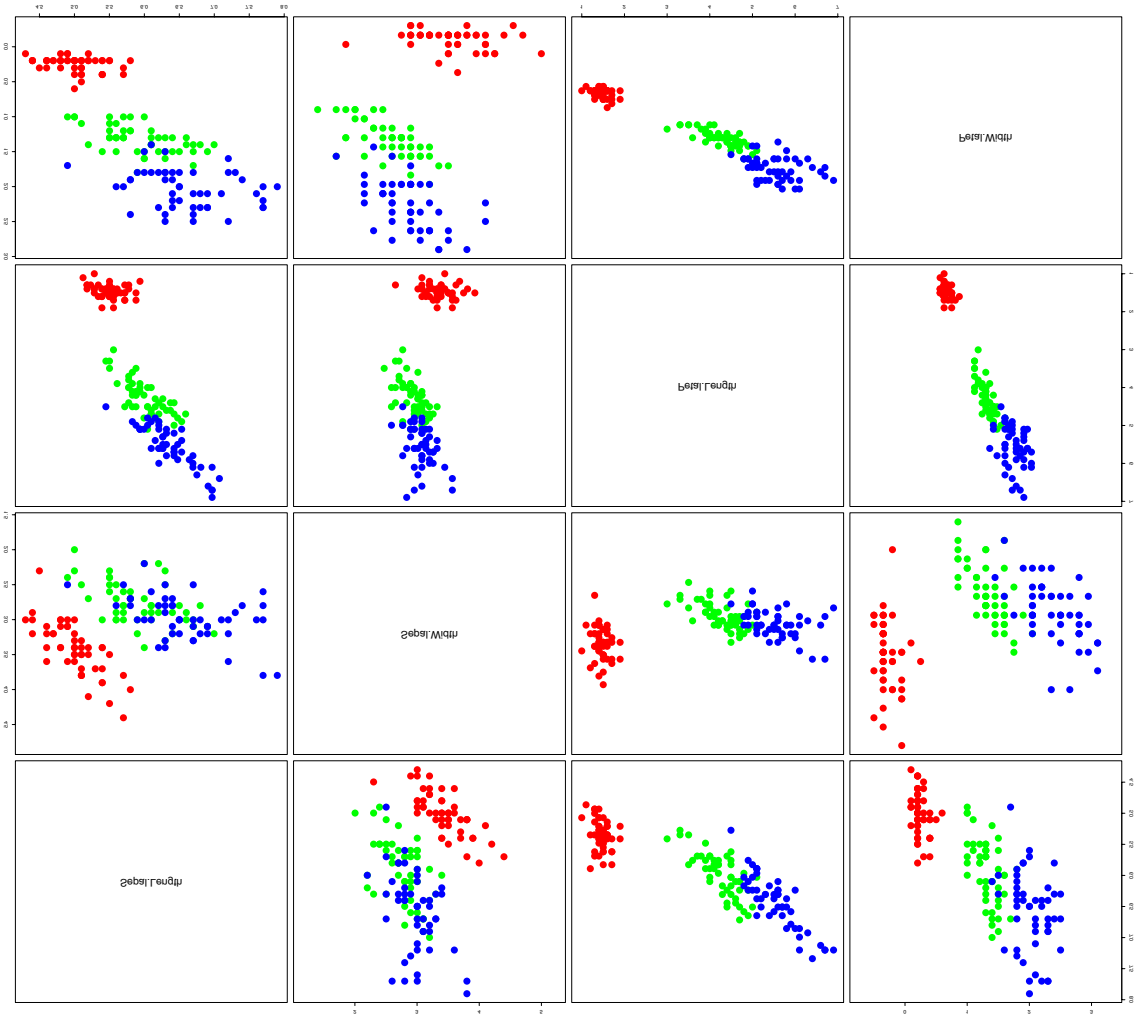
Example: iris flower data set

- The well known iris flower data set consists of 4 measurements, sepal and petal lengths and widths, *SL*,*SW*,*PL*,*PW* (in cm), containing 50 iris flowers of each one of the following three species, setosa, versicolor and virginica

- Hence the iris flower dataset defines a cloud of 150 points in $\mathbb{R}^4$. We can try to have a geometrical grasp of the shape of this 4-dimensional cloud by projecting it on a two dimensional space (plane), using all possible combinations of two variables

Example: iris flower data set

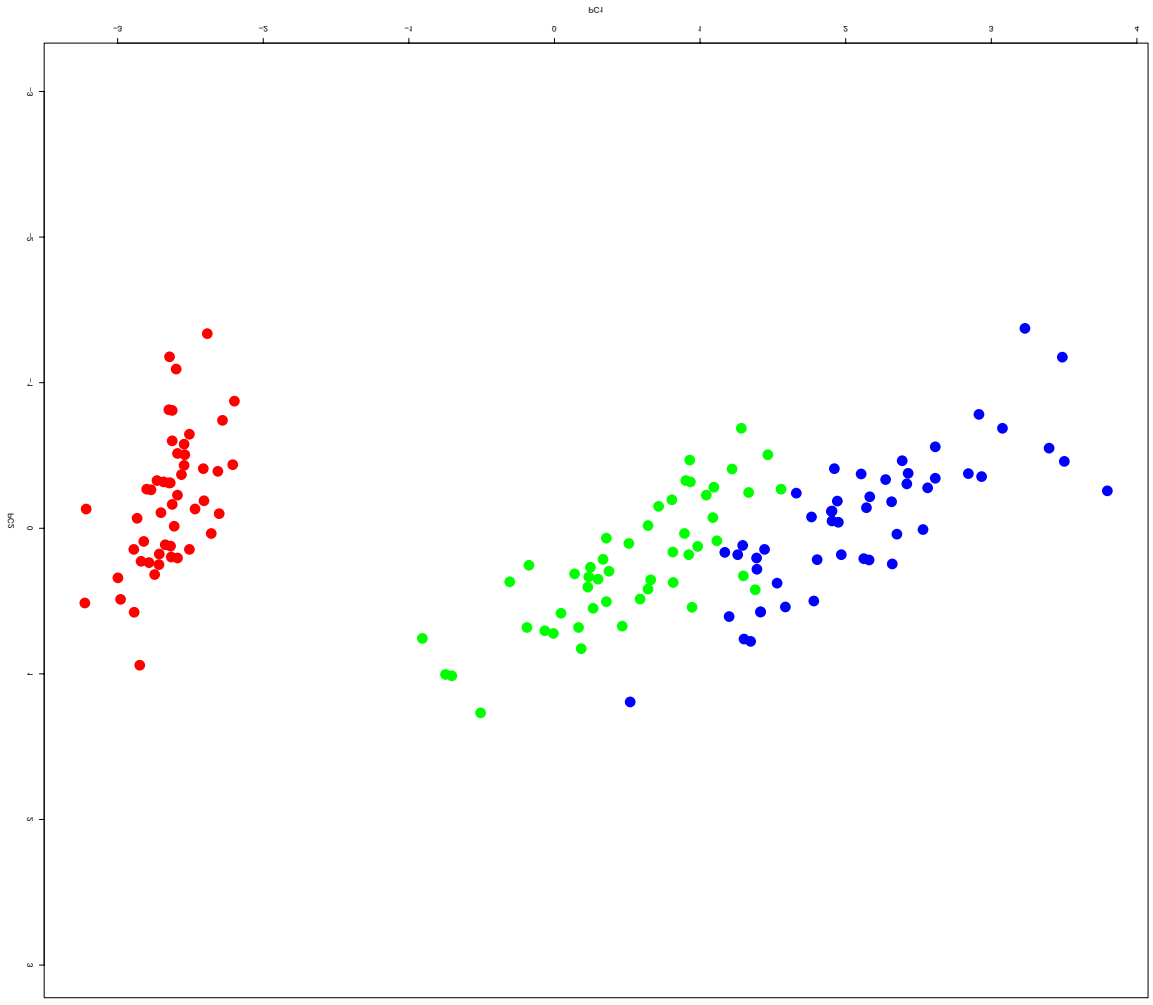
```
pairs(iris[-5],asp=TRUE,pch=16,col=c(rep("red",50),
      rep("green",50),rep("blue",50)))
```



Best 2-dimensional representation using PCA

- Another approach is to define new synthetic uncorrelated variables that are linear combinations of the original iris flowers measurements, the so-called principal components (PC), in such a way each PC explains, as much as possible, of the total dataset variability
- The projection of the cloud of iris flowers on the plane associated with the first two PCs, called principal factorial plane (PFP), explains 98.1% and thus provides an excellent 2-dimensional portray of the original cloud of iris flowers
- This is actually the best representation among all 2-dimensional representations of the iris flower dataset, in the sense that it is the 2-dimensional representation that retains the largest amount of the dataset variability

Best two-dimensional representation of the iris flowers



Eigenvalues of the covariance matrix

Let $\mathbf{X}_{N\times p}$ be a data matrix and $\mathbf{S} = \frac{1}{N-1}(\mathbf{X}^*)^T\mathbf{X}^*$ the corresponding covariance matrix. Then:

- $\mathbf{S}$ is symmetric ($\mathbf{S}^T = \mathbf{S}$)
- $\mathbf{x}^T\mathbf{S}\mathbf{x}$ is a **semi-definite positive** quadratic form, that is,

$$\mathbf{x}^T\mathbf{S}\mathbf{x} \geq 0, \quad \forall \mathbf{x} \in \mathbb{R}^p$$

- the eigenvalues $\lambda_1, \ldots, \lambda_p$ of $\mathbf{S}$ are nonnegative real numbers and we may assume that

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$$

- If moreover all variables are globally non-correlated then all eigenvalues of $\mathbf{S}$ are strictly positive real numbers. In this case $\mathbf{S}$ is invertible, $\mathbf{x}^T\mathbf{S}\mathbf{x}$ is a definite positive quadratic form, which amounts to say that

$$\mathbf{x}^T\mathbf{S}\mathbf{x} > 0, \quad \forall \mathbf{x} \in \mathbb{R}^p, \, \mathbf{x} \neq \vec{\mathbf{0}}$$

Linear combinations

Let $\mathbf{X}_{N\times p} = [\,\mathbf{x}_1 \;\cdots\; \mathbf{x}_p\,]$ be a data matrix of p observed variables

A **linear combination** of the p observed variables $\mathbf{x}_1, \ldots, \mathbf{x}_p$ is a new variable of the form

$$\mathbf{y} = \alpha_1\mathbf{x}_1 + \cdots + \alpha_p\mathbf{x}_p = [\,\mathbf{x}_1 \;\cdots\; \mathbf{x}_p\,] \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_p \end{bmatrix} = \mathbf{X}\mathbf{a},$$

where

$$\mathbf{a} = (\alpha_1, \ldots, \alpha_p) = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_p \end{bmatrix} \in \mathbb{R}^p,$$

is the vector of coefficients (**loadings**) (see slide 11)

Covariance between linear combinations

Given $\mathbf{a}, \mathbf{b} \in \mathbb{R}^p$, the (sample) covariance between the linear combinations $\mathbf{Xa}$ and $\mathbf{Xb}$ is

$$cov(\mathbf{Xa}, \mathbf{Xb}) = \mathbf{a}^T \mathbf{S} \mathbf{b} \tag{5}$$

Actually, using (4) of slide 17 we have,

$$\begin{aligned} cov(\mathbf{Xa}, \mathbf{Xb}) &= \frac{1}{N-1}[(\mathbf{Xa})^*]^T (\mathbf{Xb})^* \overset{exercise}{=} \frac{1}{N-1}(\mathbf{X}^*\mathbf{a})^T \mathbf{X}^*\mathbf{b} \\ &= \frac{1}{N-1}\mathbf{a}^T(\mathbf{X}^*)^T\mathbf{X}^*\mathbf{b} = \mathbf{a}^T \frac{1}{N-1}(\mathbf{X}^*)^T\mathbf{X}^*\mathbf{b} \\ &= \mathbf{a}^T \mathbf{S} \mathbf{b} \end{aligned}$$

In particular, $var(\mathbf{Xa}) = \mathbf{a}^T \mathbf{S} \mathbf{a}$

Exercise

Prove that centering a linear combination of variables $\mathbf{x}_1, \ldots, \mathbf{x}_p$ is equivalent to the linear combination of the centered variables $\mathbf{x}_1^*, \ldots, \mathbf{x}_p^*$ with the same coefficients, that is,

$$(\mathbf{Xa})^* = (\alpha_1 \mathbf{x}_1 + \cdots + \alpha_p \mathbf{x}_p)^* = \alpha_1 \mathbf{x}_1^* + \cdots + \alpha_p \mathbf{x}_p^* = \mathbf{X}^*\mathbf{a},$$

where $\mathbf{X} = [\, \mathbf{x}_1 \ \cdots \ \mathbf{x}_p \,]$, $\mathbf{X}^* = [\, \mathbf{x}_1^* \ \cdots \ \mathbf{x}_p^* \,]$ *and* $\mathbf{a} = (\alpha_1, \ldots, \alpha_p) \in \mathbb{R}^p$

First principal component - formulation

To define the first principal component we seek a linear combination of the p observed variables $\mathbf{x}_1, \ldots, \mathbf{x}_p$ that maximizes the variance, that is, we want to solve the following problem:

$$\text{determine } \mathbf{a} \in \mathbb{R}^p \text{ such that } \; var(\mathbf{Xa}) = \mathbf{a}^T\mathbf{Sa} \; \text{ is maximum}$$

Without further restrictions on vector $\mathbf{a}$ the problem is ill-posed since if we multiply the vector of coefficients $\mathbf{a}$ by a scalar λ we obtain

$$var(\mathbf{X}(\lambda\mathbf{a})) = \lambda\mathbf{a}^T\mathbf{S}\lambda\mathbf{a} = \lambda^2\mathbf{a}^T\mathbf{Sa} = \lambda^2 var(\mathbf{X}(\lambda\mathbf{a})),$$

which shows that the variance of a linear combination can be arbitrarily large. To overcome this issue we reformulate the problem as follows:

$$\boxed{\text{determine } \mathbf{a} \in \mathbb{R}^p \text{ with } \|\mathbf{a}\| = 1 \; : \; var(\mathbf{Xa}) = \mathbf{a}^T\mathbf{Sa} \text{ is maximum}} \quad (6)$$

The previous problem can be equivalently formulated as the problem of maximizing the so-called **Rayleigh-Ritz ratio** *(cf. slides Prof. Cadima)*

$$\boxed{\textit{determine } \mathbf{a} \in \mathbb{R}^p \setminus \{\vec{0}\} \; : \; \frac{\mathbf{a}^T\mathbf{Sa}}{\mathbf{a}^T\mathbf{a}} \textit{ is maximum}} \quad (7)$$

First principal component (cont.)

The covariance matrix $\mathbf{S}$ admits a spectral decomposition (see slide 12) of the form,

$$\mathbf{S} = \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T + \lambda_2 \mathbf{v}_2 \mathbf{v}_2^T + \cdots + \lambda_p \mathbf{v}_p \mathbf{v}_p^T \tag{8}$$

where $\mathbf{v}_1, \ldots, \mathbf{v}_p \in \mathbb{R}^p$ are unit and pairwise orthogonal eigenvectors of $\mathbf{S}$ associated to (sorted) real eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$

By the results of slide 10, we have for all $\mathbf{a} \in \mathbf{R}^p$, $\|\mathbf{a}\| = 1$,

$$\mathbf{a} = \cos(\theta_1)\mathbf{v}_1 + \cdots + \cos(\theta_p)\mathbf{v}_p, \tag{9}$$

with

$$\cos^2(\theta_1) + \cdots + \cos^2(\theta_p) = 1, \tag{10}$$

where θ_i denotes the angle between the vectors $\mathbf{a}$ and $\mathbf{v}_i$, , $i = 1, \ldots, p$

First principal component (cont.)

Applying (8), (9) and (10) from the previous slide, along with relations $\lambda_1 \geq \cdots \geq \lambda_p \geq 0$, $\|\mathbf{v}_i\|^2 = \mathbf{v}^T\mathbf{v} = 1$ for all i and $\mathbf{v}_i^T\mathbf{v}_j = 0$, $i \neq j$, we obtain by straightforward computations (since all products envolving v_i and v_j, $j \neq i$, cancel out),

$$\begin{aligned} \mathbf{a}^T\mathbf{S}\mathbf{a} &= \lambda_1 \cos^2\theta_1 + \cdots + \lambda_p \cos^2\theta_p \\ &\leq \lambda_1 \cos^2\theta_1 + \cdots + \lambda_1 \cos^2\theta_p \\ &= \lambda_1(\cos^2\theta_1 + \cdots + \cos^2\theta_p) = \lambda_1 \end{aligned}$$

Thus $var(\mathbf{X}\mathbf{a}) = \mathbf{a}^T\mathbf{S}\mathbf{a} \leq \lambda_1$ (the largest eigenvalue of $\mathbf{S}$). Taking $\mathbf{a} = \mathbf{v}_1$, we get

$$\mathbf{a}^T\mathbf{S}\mathbf{a} = \mathbf{v}_1^T\mathbf{S}\mathbf{v}_1 = \lambda_1 \underbrace{\cos^2\theta_1}_{1} + \lambda_2 \underbrace{\cos^2\theta_2}_{0} + \cdots + \lambda_1 \underbrace{\cos^2\theta_p}_{0} = \lambda_1$$

The maximum variance of a linear combination $\mathbf{X}\mathbf{a}$ of the p observed variables $\mathbf{x}_1, \ldots, \mathbf{x}_p$, with unit vector of coefficients $\mathbf{a}$, is attained along the direction of a unit eigenvector $\mathbf{v}_1$ of $\mathbf{S}$ associated with the largest eigenvalue λ_1. Hence the first principal component is

$$PC_1 : \quad \mathbf{y}_1 = \mathbf{X}\mathbf{v}_1 \textit{ with maximum variance } \lambda_1$$

The larger the value of λ_1, the more the cloud of points is elongated along the PC_1

Second principal component

To define the second principal component PC_2, we seek a linear combination of the p original observed variables, that maximizes the variance and is uncorrelated with PC_1:

$$\text{determine } \mathbf{a} \in \mathbb{R}^p \text{ with } \begin{cases} \|\mathbf{a}\| = 1 \\ \mathbf{a} \perp \mathbf{v_1} \end{cases} : \; var(\mathbf{Xa}) = \mathbf{a}^T\mathbf{Sa} \text{ is maximum}$$

Since $\mathbf{a} \perp \mathbf{v}_1 \Leftrightarrow \cos\theta_1 = 0$, we seek $\mathbf{a} = \cos(\theta_2)\mathbf{v}_2 + \cdots + \cos(\theta_p)\mathbf{v}_p$, with $\cos^2(\theta_2) + \cdots + \cos^2(\theta_p) = 1$ and we obtain similarly,

$$\begin{aligned} \mathbf{a}^T\mathbf{Sa} &= \lambda_2 \cos^2\theta_2 + \cdots + \lambda_p \cos^2\theta_p \\ &\leq \lambda_2(\cos^2\theta_2 + \cdots + \cos^2\theta_p) = \lambda_2 \end{aligned}$$

Taking $\mathbf{a} = \mathbf{v}_2$ (a unit eigenvector of $\mathbf{S}$ associated with the second largest eigenvalue λ_2 and orthogonal to $\mathbf{v}_1$), one gets

$$\mathbf{a}^T\mathbf{Sa} = \lambda_2$$

The second PC is thus defined by a unit eigenvector $\mathbf{v}_2$ of $\mathbf{S}$, associated with the second largest eigenvalue λ_2 and orthogonal to the vector $\mathbf{v}_1$

$$PC_2 : \quad \mathbf{y}_2 = \mathbf{Xv}_2 \quad \textit{with maximum variance equal to } \lambda_2$$

Principal components

In general, to define the j-th principal component PC_j, $j = 2, \ldots, p$, we seek a linear combination of the p original observed variables, that maximizes the variance and is uncorrelated with $\mathrm{PC}_1, \ldots, \mathrm{PC}_{j-1}$:

$$\text{determine } \mathbf{a} \in \mathbb{R}^p \text{ with } \left\{ \begin{array}{rcl} \|\mathbf{a}\| & = & 1 \\ \mathbf{a} & \perp & \mathbf{v_1} \\ & \vdots & \\ \mathbf{a} & \perp & \mathbf{v_{j-1}} \end{array} \right| \; var(\mathbf{Xa}) = \mathbf{a}^T\mathbf{Sa} \text{ is maximum} \qquad (11)$$

We construct in this way a collection of p principal components

$$\mathbf{y}_1 = \mathbf{X}\mathbf{v}_1, \quad \mathbf{y}_2 = \mathbf{X}\mathbf{v}_2, \quad \ldots, \quad \mathbf{y}_p = \mathbf{X}\mathbf{v}_p$$

with maximum variances,

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p \geq 0,$$

where $\mathbf{v}_1, \ldots, \mathbf{v}_p$ are unit and pairwise orthogonal eigenvectors of $\mathbf{S}$, respectively associated to $\lambda_1, \ldots, \lambda_p$, i.e., for all $j, k = 1, \ldots, p$, $k \neq j$ we have

$$\|\mathbf{v}_j\| = 1, \qquad \mathbf{v}_j \perp \mathbf{v}_k, \qquad \mathbf{S}\mathbf{v}_j = \lambda_j \mathbf{v}_j$$

Vector of loadings

The vector $\mathbf{v}_j$ defining the j-th principal component $\mathbf{y}_j = \mathbf{X}\mathbf{v}_j$, contains the coefficients, also called loadings, of the j-th principal component w.r.t. the original observed variables $\mathbf{x}_1, \ldots, \mathbf{x}_p$. In other words, writing the vector of loadings as $\mathbf{v}_j = (\alpha_1, \ldots, \alpha_p)$ we obtain,

$$\mathbf{y}_j = \alpha_1 \mathbf{x}_1 + \cdots + \alpha_p \mathbf{x}_p$$

- *If the p eigenvalues of the covariance matrix $\mathbf{S}$ are pairwise distinct, i.e., $\lambda_1 > \cdots > \lambda_p \geq 0$, the vector of loadings defining each PC is unique up to sign: if $\mathbf{y}_j = \mathbf{X}\mathbf{v}_j$ is a solution of (11) of slide 31, then $\mathbf{y}'_j = \mathbf{X}(-\mathbf{v}_j)$ is also a solution of (11) - this is the most common situation*
- *If there are repeated eigenvalues of $\mathbf{S}$ the PCs associated with repeated eigenvalues are not uniquely determined. Actually, the vectors of loadings defining these PCs can arise from any orthonormal base of the eigenspace associated with the repeated eigenvalue and therefore can be defined in infinitely many distinct ways*

Scores matrix

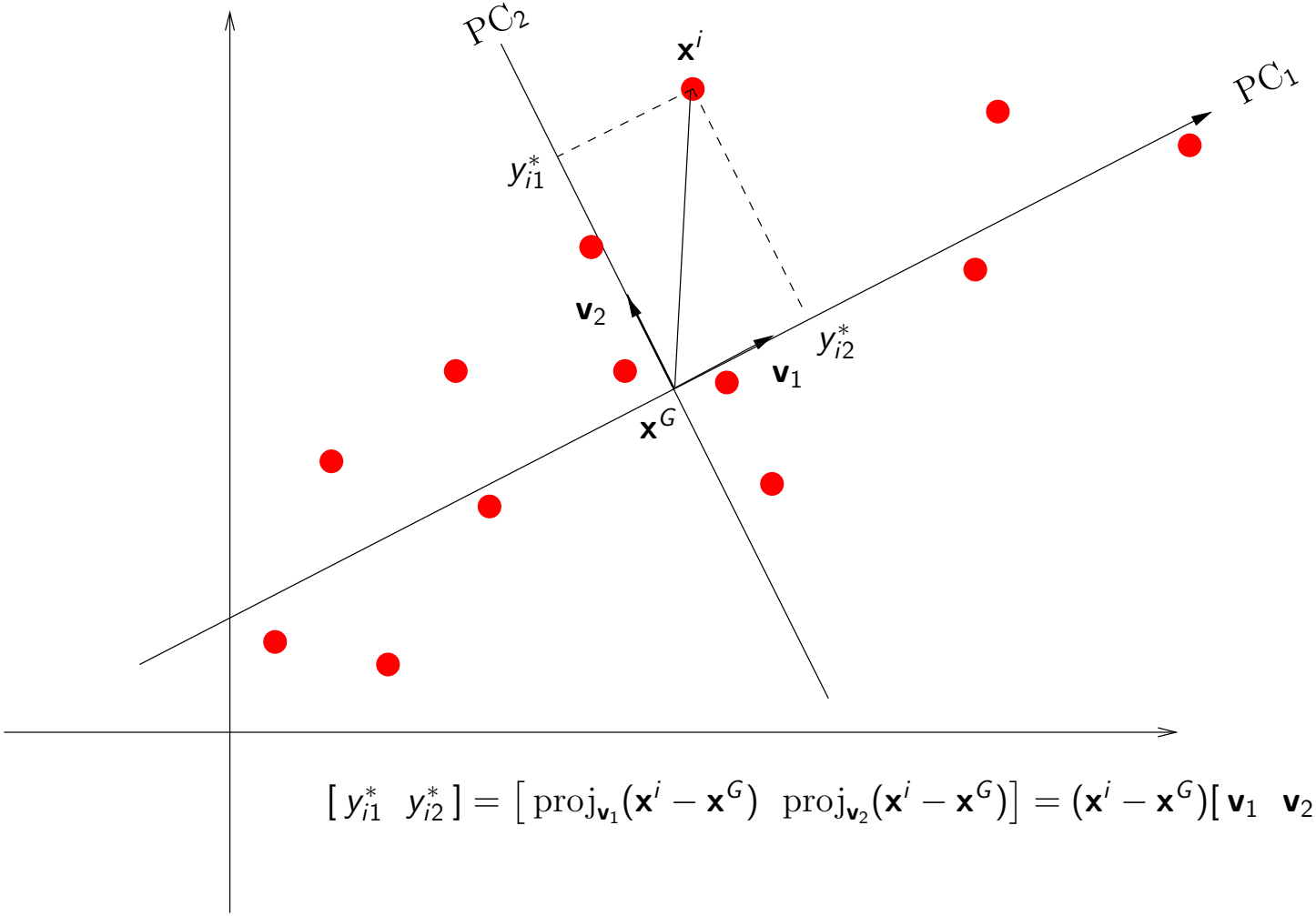
Recall that,

- $\mathbf{X}_{N\times p} = [x_{ij}]$ is the original data matrix of the p observed variables across N individuals
- $\mathbf{X}^T = [\,\mathbf{x}^1 \ \cdots \ \mathbf{x}^N\,]$, with $\mathbf{x}^i = (x_{i1}, \ldots, x_{ip})$ the i-th row of $\mathbf{X}$, i.e., the coordinates of i-individual in the cloud of N points of $\mathbf{R}^p$
- $\mathbf{x}^G = (\bar{x}_1, \ldots, \bar{x}_p)$ is the center of gravity (also called barycenter) of the cloud of individuals
- $\mathbf{X}^* = [x^*_{ij}]$ is the centered data matrix, where $x^*_{ij} = x_{ij} - \bar{x}_j$
- $\mathbf{x}^i - \mathbf{x}^G = (x^*_{i1}, \ldots, x^*_{ip})$ the i-th row of $\mathbf{X}^*$, i.e., the vector of the coordinates of individual i in the centered cloud of N points
- $\mathbf{V} = [\,\mathbf{v}_1 \ \cdots \mathbf{v}_p\,]$ is matrix of loadings

The matrix $\mathbf{Y}^* = [y^*_{ij}] = \mathbf{X}^*\mathbf{V}$ is called scores matrix: the rows of $\mathbf{Y}^*$ correspond to the vectors of coordinates, also called (factor) scores, of the N individuals w.r.t the new coordinate axes defined by the vectors of loadings of the PCs

The column j of $\mathbf{Y}^*$, $\mathbf{y}^*_j$, contains the values of the (centered) cloud of N individuals w.r.t the new sinthetic variable $\mathbf{y}_j$ that defined the PC_j

Scores of individual i when $p = 2$



$$[\, y^*_{i1} \;\; y^*_{i2} \,] = \left[\, \mathrm{proj}_{\mathbf{v}_1}(\mathbf{x}^i - \mathbf{x}^G) \;\; \mathrm{proj}_{\mathbf{v}_2}(\mathbf{x}^i - \mathbf{x}^G) \right] = (\mathbf{x}^i - \mathbf{x}^G)[\, \mathbf{v}_1 \;\; \mathbf{v}_2 \,]$$

Covariance of the scores matrix

- The covariance matrix of the (centred) scores matrix $\mathbf{Y}^*$, is

$$\begin{aligned} cov(\mathbf{Y}^*) &= cov(\mathbf{X}^*\mathbf{V}) = \frac{1}{N-1}(\mathbf{X}^*\mathbf{V})^T(\mathbf{X}^*\mathbf{V}) \\ &= \mathbf{V}^T\frac{1}{N-1}(\mathbf{X}^*)^T\mathbf{X}^*\mathbf{V} = \mathbf{V}^T\mathbf{S}\mathbf{V} = \mathbf{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_p) \end{aligned}$$

- The total variation of $\mathbf{Y}^*$, i.e., the dataset total variability is

$$\sum_{j=1}^{p} \text{var}(\mathbf{y}_j^*) = \sum_{j=1}^{p} \lambda_j = \text{tr}(\mathbf{\Lambda}) = \text{tr}(\mathbf{S}) = \sum_{j=1}^{p} \text{var}(\mathbf{x}_j)$$

- The quality of the reduction obtained by keeping the first k PCs ($1 \leq k \leq p$) is assessed by the proportion of variability explained by the first k PCs:

$$\frac{\lambda_1 + \cdots + \lambda_k}{\lambda_1 + \cdots + \lambda_p}$$

Covariance and correlation

- The covariance between the observed variable $\mathbf{x}_j$ and the PC $\mathbf{y}_k$ is

$$\begin{aligned} cov(\mathbf{x}_j, \mathbf{y}_k) &= \frac{1}{N-1}[(\mathbf{X}\mathbf{e}_j)^*]^T(\mathbf{X}\mathbf{v}_k)^* = \frac{1}{N-1}(\mathbf{X}^*\mathbf{e}_j)^T(\mathbf{X}^*\mathbf{v}_k) \\ &= \mathbf{e}_j^T \frac{1}{N-1}(\mathbf{X}^*)^T\mathbf{X}^*\mathbf{v}_k = \mathbf{e}_j^T\mathbf{S}\mathbf{v}_k = \mathbf{e}_j^T\lambda_k\mathbf{v}_k \\ &= \lambda_k\mathbf{e}_j^T\mathbf{v}_k = \lambda_k\mathbf{v}_{jk} \end{aligned}$$

  where $\mathbf{v}_{jk} = \mathbf{e}_j^T\mathbf{v}_k$ is j-th component of $\mathbf{v}_k$, i.e., (j,k)-entry of the loadings matrix $\mathbf{V}$

- The correlation between $\mathbf{x}_j$ and $\mathbf{y}_k$ is

$$cor(\mathbf{x}_j, \mathbf{y}_k) = \frac{cov(\mathbf{x}_j, \mathbf{y}_k)}{\sqrt{\mathbf{x}_j}\sqrt{\mathbf{y}_k}} = \frac{\lambda_k\mathbf{v}_{jk}}{s_j\sqrt{\lambda_k}} = \frac{\sqrt{\lambda_k}\mathbf{v}_{jk}}{s_j}$$

- The contribution of individual i to the construction of PC_k is the part of the variance of PC_k that is explained by i:

$$ctr_{i,k} = \frac{(y_{i,k}^*)^2}{\sum_{j=1}^N (y_{j,k}^*)^2}$$

Example: iris flower dataset revisited

R

```
X=iris[-5] # non standardized

head(X)

iris.acp<-prcomp(X) # performs PCA on the covariance
matrix

summary(iris.acp)

iris.acp$sdev # std accounted by the PCs

sum(iris.acp$sdev[1]^2) # total variance

iris.acp$rot # matrix of loadings

iris.acp$x # matrix of scores

plot(iris.acp$x[,1:2],asp=TRUE,pch=16,col=c(rep("red",50),
rep("green",50),rep("blue",50))) #
```

Importance of the PC components

The R command `summary(iris.acp)` gives, for each j, the standard deviation $\sqrt{\lambda_j}$ associated with PC_j, the proportion of the total variance explained by PC_j, $\frac{\lambda_j}{\sum_k \lambda_k}$, and the cumulative variance explained by the first j PCs:

	PC1	PC2	PC3	PC4
Standard deviation	2.0563	0.49262	0.2797	0.15439
Proportion of Variance	0.9246	0.05307	0.0171	0.00521
Cumulative Proportion	0.9246	0.97769	0.9948	1.00000

Thus we have that:

- The cloud of points projected on the line associated with the first PC explains about 92% of the dataset's total variability
- The cloud of points projected on the plane associated with the first two PCs (principal factorial plane - PFP) explains about 98% of the total variability of the dataset,
- and so on...

More on the R command prcomp

`iris.acp$sdev` give, for each j, the standard deviation $\sqrt{\lambda_j}$ associated with PC_j: 2.0562689 0.4926162 0.2796596 0.1543862

`sum(iris.acp$sdev[1]^2)` gives the dataset total variance: 4.572957

`iris.acp$rotation` returns the matrix of loadings, where column j contains the coefficients of the PC_j, $\mathbf{y}_j$, written as linear combination of the original observed variables $\mathbf{x}_1, \ldots, \mathbf{x}_4$:

	PC1	PC2	PC3	PC4
Sepal.Length	0.3614	-0.6566	0.5820	0.3155
Sepal.Width	-0.0845	-0.7302	-0.5979	-0.3197
Petal.Length	0.8567	0.1734	-0.0762	-0.4798
Petal.Width	0.3583	0.0755	-0.5458	0.7537

The first PC (for instance), is a linear combination of the observed measurements as:

$$\begin{aligned} \mathbf{y}_1 &= 0.3614\,\text{Sepal.Length} - 0.0845\,\text{Sepal.Width} + 0.8567\,\text{Petal.Length} + 0.3583\,\text{Petal.Wid} \\ &\approx 0.3614\,\text{Sepal.Length} + 0.8567\,\text{Petal.Length} + 0.3583\,\text{Petal.Width} \end{aligned}$$

which represents a kind of overall measurement of the iris flowers that explains a large amount ($\geq 90\%$) of the total variability of the iris dataset

More on the R command prcomp - loadings

The columns of the loading matrix are unit eigenvectors of **S** and pairwise orthogonal

R

```
V<-iris.acp$rotation
S <- cov(S)
round(t(V)%*% V,10) # gives the identity matrix
v1 <- V[,1]
lambda1 <- iris.acp$sdev[1]^2
S %*% v1
lambda1%*% v1
```

More on the R command `prcomp` - scores

`iris.acp$x` returns the matrix of factor scores, where each row i contains coordinates of the individual i w.r.t. the PCs, i.e., w.r.t. the new synthetic variables $\mathbf{y}_1, \ldots, \mathbf{y}_4$:

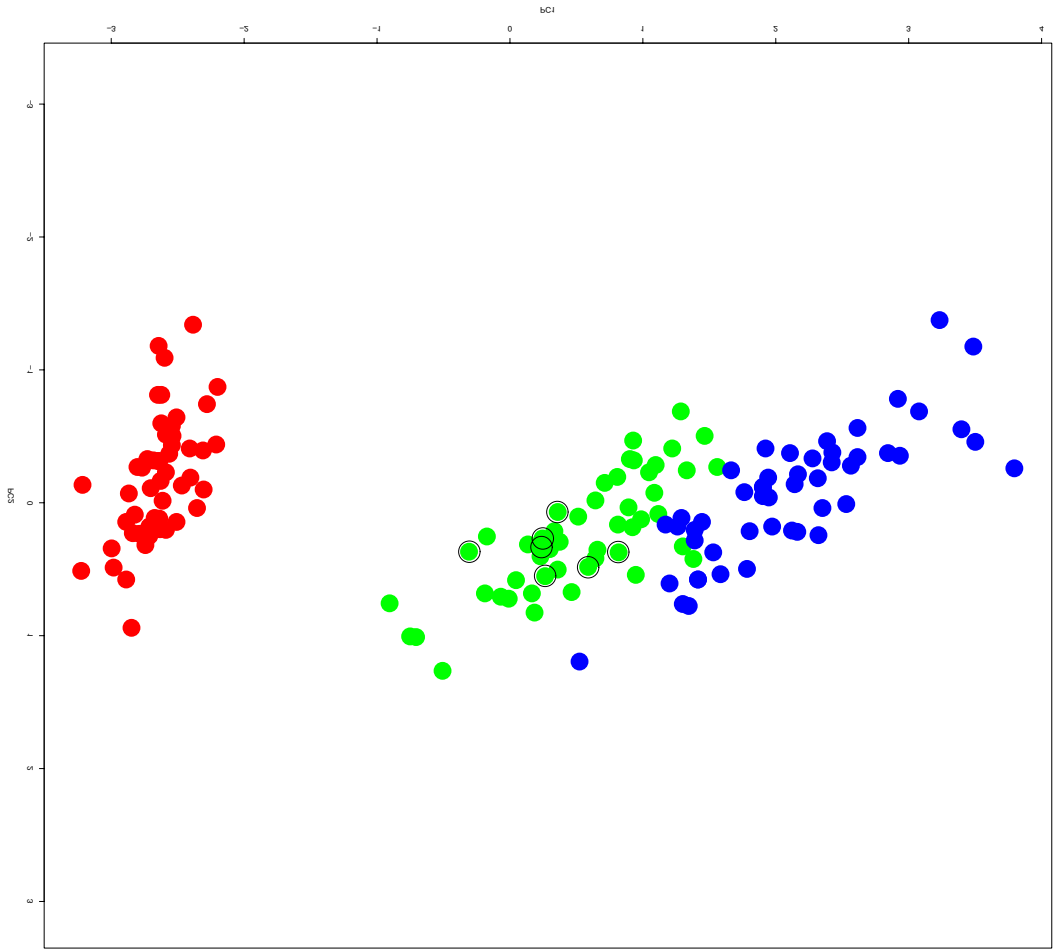
	PC1	PC2	PC3	PC4
	-2.68413	-0.31940	0.02791	0.00226
	-2.71414	0.17700	0.21046	0.09903
	-2.88899	0.14495	-0.01790	0.01997
	-2.74534	0.31830	-0.03156	-0.07558
	⋮	⋮	⋮	⋮

More on the R command prcomp - scores

R

```
N <- 150 X.G <- colMeans(X) # iris's cloud center of gravity
Xc <- scale(X,scale=FALSE) # centred iris data matrix
Yc <- Xc %*% V # scores matrix
head(Yc) ; head(iris.acp$x) # should be equal!
sum(iris.acp$x[,1]^2)/(N-1) ; iris.acp$sdev[1]^2
# contributions of each individual to the 1st PC
Yc[,1]*Yc[,1]/sum(Yc[,1]*Yc[,1])
vspace.5ex
# individuals with contribution above the average
Yc[,1]*Yc[,1]/sum(Yc[,1]*Yc[,1]>1/150
# quality of the representation of individual i in each PC
Yc[1,]*Yc[1,]/sum(Yc[1,]*Yc[1,])
# quality of the representation of individual i in the PFP
(Yc[1,1]*Yc[1,1]+Yc[1,2]*Yc[1,2])/sum(Yc[1,]*Yc[1,])
cos2<-matrix(0,ncol=4,nrow=150)
for (i in 1:150) { cos2[i,]<-Yc[i,]*Yc[i,]/sum(Yc[i,]*Yc[i,]) }
sort(rowSums(cos2[,1:2]))
order(rowSums(cos2[,1:2]))
plot(iris.acp$x[,1:2],pch=16,col=c(rep("red",50),
rep("green",50),rep("blue",50)),asp=TRUE)
points(iris.acp$x[rowSums(cos2[,1:2])<.7,1:2],pch=1)
```

Representation of the iris flower dataset in the PFP



Drawbacks of the PCA on the covariance matrix

- The first PCs tend to be dominated by the variable(s) with the largest variance(s)
- The PCs are invariant under orthogonal transformations of the variables (e.g. rotations), but not under differentiated change of scalars in each variable. As a consequence the PCA is highly dependent on the units of measurements - this is a major drawback
- Another important drawback when there are distinct units of measurements is how to a interpret a PC when that is a linear combination of variables expressed in totally different units of measurements, say, for instance temperature and weight?

When the variables have different units of measurements or very different variances it is advisable or even mandatory to standardize (i.e., to center and reduce the variables to unit variance) prior to perform the PCA. This amounts to compute the eigenvectors of the correlation matrix of $\mathbf{X}$

PCA on the correlation matrix

Let $\mathbf{X}_{N\times p} = [\mathbf{x}_{ij}]$ be the usual data matrix and $\mathbf{Z}_{N\times p} = [\mathbf{z}_{ij}]$, be the corresponding data matrix of the standardized variables $z_{ij} = \frac{x_{ij}-\bar{x}_j}{s_j}$

- The covariance matrix of the standardized data $\mathbf{Z}$ is

$$\mathbf{R} = cov(\mathbf{Z}) = \frac{1}{N-1}\mathbf{Z}^T\mathbf{Z},$$

  which corresponds to the correlation matrix of $\mathbf{X}$

- The PCs are now given by $\mathbf{Y}_j = \mathbf{Z}\mathbf{v}_j$ where $\mathbf{v_1}, \ldots, \mathbf{v_p}$ are unit and pairwise orthogonal eigenvectors of $\mathbf{R}$ associated with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_p$
- The total variance is now the number of variables:

$$p = \sum_{i=1}^{p} var(\mathbf{z}_j) = \lambda_1 + \cdots + \lambda_p$$

- The correlation coefficient between $\mathbf{z}_j$ and $\mathbf{y}_k$ reduces to

$$cor(\mathbf{z}_j, \mathbf{y}_k) = \sqrt{\lambda_k}\mathbf{v}_{kj}$$

Interpretation of the results in space of variables

Each standardized variable $\mathbf{z}_j$ and each PC $\mathbf{y}_k$, can be represented as vectors in $\mathbf{R}^N$. This allows to reinterpret geometrically some of the previous statistics:

- The variables $\mathbf{z}_j$, $j = 1, \ldots, p$, lie in a hypersphere of radius $\sqrt{N-1}$:

$$\|\mathbf{z}_j\|^2 = \mathbf{z}_j^T \mathbf{z}_j = (N-1)var(\mathbf{z}_j) = N-1$$

  More generally, the length of centered variable is also proportional to its standard deviation (exercise)

- The length of each PC is proportional to its standard deviation:

$$\begin{aligned} \|\mathbf{y}_k\|^2 &= \mathbf{y}_k^T \mathbf{y}_k = (\mathbf{Z}\mathbf{v}_k)^T (\mathbf{Z}\mathbf{v}_k) \\ &= \mathbf{v}_k^T \mathbf{Z}^T \mathbf{Z} \mathbf{v}_k = (N-1)\mathbf{v}_k^T \mathbf{R} \mathbf{v}_k \\ &= (N-1)\lambda_k = (N-1)var(\mathbf{y}_k) \end{aligned}$$

- The correlation coefficient between $\mathbf{z}_j$ and $\mathbf{y}_k$ is the cosine of the angle θ_{jk} between the variables $\mathbf{z}_j$ and $\mathbf{y}_k$:

$$\begin{aligned} cor(\mathbf{z}_j, \mathbf{y}_k) &= \frac{cov(\mathbf{z}_j, \mathbf{y}_k)}{\sqrt{var(\mathbf{z}_j)}\sqrt{var(\mathbf{y}_k)}} = \frac{\frac{cov(\mathbf{z}_j, \mathbf{y}_k)}{N-1}}{\sqrt{var(\mathbf{z}_j)}\sqrt{var(\mathbf{y}_k)}} \\ &= \frac{1}{N-1} \frac{\mathbf{z}_j^T \mathbf{y}_k}{\frac{\|\mathbf{z}_k\|}{\sqrt{N-1}} \frac{\sqrt{\lambda_k}}{\sqrt{N-1}}} = \frac{\mathbf{z}_j^T \mathbf{y}_k}{\|\mathbf{z}_j\| \, \|\mathbf{y}_k\|} = \cos(\theta_{jk}) \end{aligned}$$

- The correlation coefficient between $\mathbf{z}_j$ and $\mathbf{z}_k$ is the cosine of the angle between the vectors representing these variables (exercise)

How many PCs ?

No exact answer can be given. Some empirical rules are listed below:

- To define a cutoff %: to consider a given cumulative percentage of the total variation (usually between 70% and 90%) and to choose the smallest number m of PC such that the % of explained variance by the first m PCs exceeds the chosen %.
- Scree plot: to look for a elbow point in the scree plot of the variance
- Kaiser's rule (for PCA on correlation matrix): to retain the PCs with variance greater than the average value 1: the PCs with variance inferior to 1 contain less information than the original variables and are not worthing to retain. (for the PCA on the covariance matrix, the cutoff value 1 should be replaced by the average of the PCs variances)
- Jolliffe's variant of Kaiser's rule (for PCA on correlation matrix): is a more conservative rule that proposes a cutoff value of 0.7
- Broken-stick model: a unit stick is randomly broken into p segments. The expected length of the k-th largest segment is $\ell_k^* = \frac{1}{p}\sum_{j=k}^{p}\frac{1}{j}$. This rule retains the PCs while the variance of each PC_k keeps above the length ℓ_k

PCA on the correlation matrix - summary

- All variables have the same variance and therefore their importance is equalized
- The cloud of individuals tend to have a more rounded shape
- The PCA tend to reflect existing correlation patterns among variables
- The first PC tends to be dominated by groups of variables that highly correlated
- The PCs can be interpreted since they are linear combinations of dimensionless variables
- The number of PCs that are necessary to explain a given proportion of the total dataset variability is usually higher compared to the PCA on the covariance matrix

A more geometrical approach to PCA using SVD

Applying the SVD to the centered data matrix $\mathbf{X}^*$ we obtain

$$\boxed{\mathbf{X}^* = \mathbf{U}\mathbf{\Delta}\mathbf{V}^T = \sum_{j=1}^{r} \delta_j \mathbf{u}_j \mathbf{v}_j^T}$$

where

- $\mathbf{\Delta}_{r\times r} = \text{diag}(\delta_1, \ldots, \delta_r)$ is the diagonal matrix containing the (positive) singular values of $\mathbf{Z}$ with $\delta_1 \geq \delta_2 \geq \cdots \geq \delta_r > 0$
- $\mathbf{U}_{N\times r} = [\,\mathbf{u}_1 \ \cdots \ \mathbf{u}_r]$, with $\mathbf{u}_1, \ldots, \mathbf{u}_r \in \mathbb{R}^N$, is the matrix of left singular vectors of $\mathbf{Z}$
- $\mathbf{V}_{p\times r} = [\,\mathbf{v}_1 \ \cdots \ \mathbf{v}_r]$, with $\mathbf{v}_1, \ldots, \mathbf{v}_r \in \mathbb{R}^p$, is the matrix of right singular vectors of $\mathbf{Z}$
- $\mathbf{U}^T\mathbf{U} = \mathbf{V}^T\mathbf{V} = \mathbf{I}_r$, that is, the left and right singular vectors, are unit and pairwise orthogonal vectors vectors

For each $k = 1, \ldots, p$ we have a rank k linear approximation of $\mathbf{X}^*$,

$$\mathbf{X}(k) = \sum_{j=1}^{k} \delta_j \mathbf{u}_j \mathbf{v}_j^T = \mathbf{U}(k)\mathbf{\Delta}(k)\mathbf{V}(k)^T$$

Here $\mathbf{U}(k)$ is the submatrix of $\mathbf{U}$ containing the first k columns, etc...

Best rank k-linear approximation

For instance, we have the following rank one and rank two linear approximations,

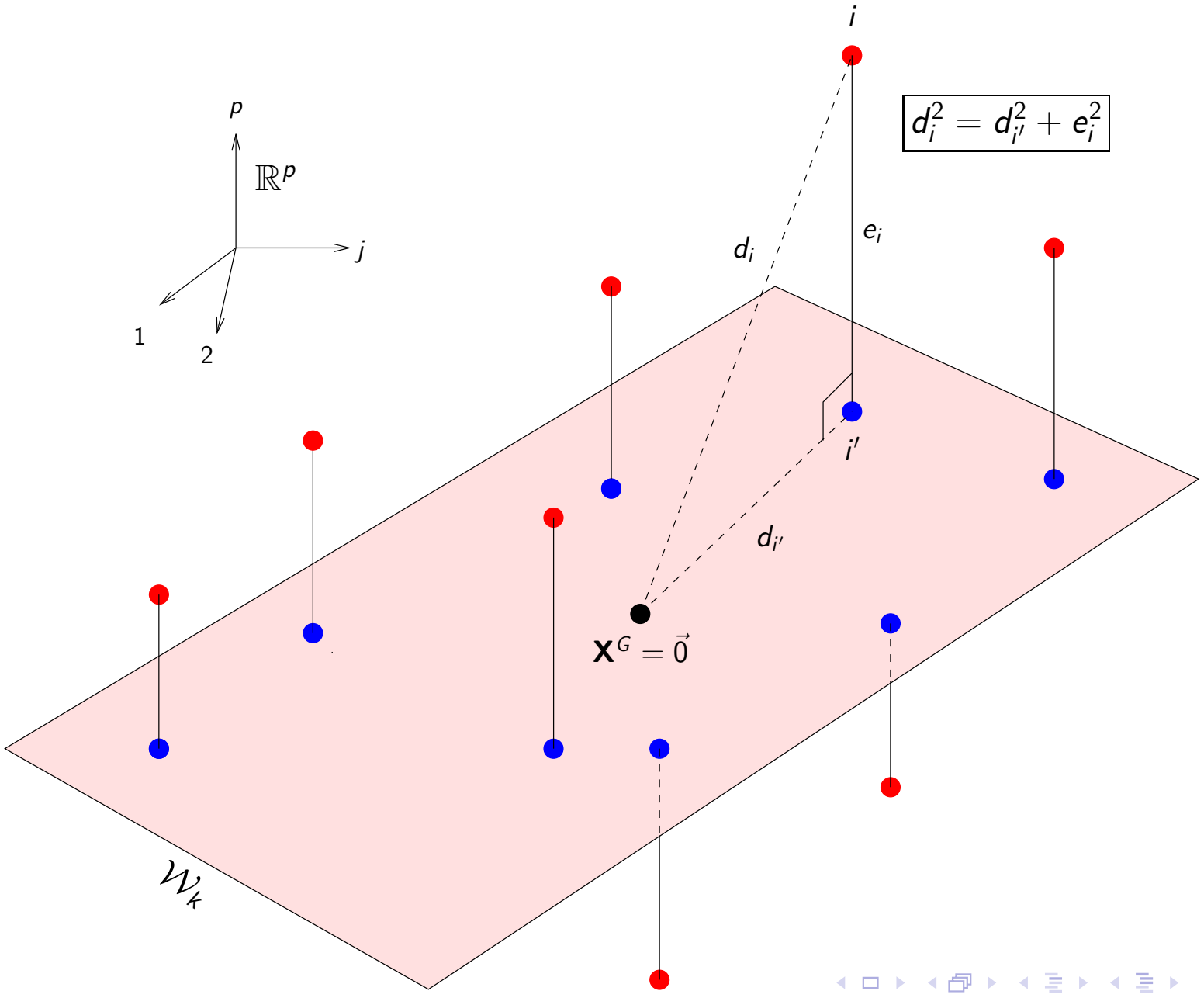
$$\begin{aligned}\mathbf{X}(1) &= \delta_1 \mathbf{u}_1 \mathbf{v}_1^T = \mathbf{U}(1)\mathbf{\Delta}(1)\mathbf{V}(1)^T \\ \mathbf{X}(2) &= \delta_1 \mathbf{u}_1 \mathbf{v}_1^T + \delta_2 \mathbf{u}_1 \mathbf{v}_2^T = \mathbf{U}(2)\mathbf{\Delta}(2)\mathbf{V}(2)^T\end{aligned}$$

All rows of $\mathbf{X}(k)$ are linear combinations of $\mathbf{v}_1^T, \ldots, \mathbf{v}_k^T$. Moreover:

- For each k, the cloud of N points defined by the rows of $\mathbf{X}(k)$ lie in a k-dimension linear subspace $\mathcal{W}(k)$ of $\mathbb{R}^p$ (generated by the vectors $\mathbf{v}_1 \ldots, \mathbf{v}_k$), that is close to the cloud of centered points defined by the rows of $\mathbf{X}^*$
- Denoting by i the point defined by row i of $\mathbf{X}^*$ and by i' the corresponding k-approximated point in $\mathcal{W}(k)$, defined by row i of $\mathbf{X}(k)$, we have that $i - i'$ is a linear combination of $\mathbf{v}_j$, $j > k$, and thus orthogonal to the linear space $\mathcal{W}(k)$
- Denoting by d_i the distance between i and the origin (center of gravity), by $d_{i'}$ the distance between i' and the origin and setting $e_i = d(i, i')$, we have a decomposition

$$d_i^2 = d_{i'}^2 + e_i^2 \tag{12}$$

Best fitting k-dimensional linear space



Best k-dimensional fitting

- The cloud of points $\mathbf{X}(k)$ gives the best rank k approximation of $\mathbf{X}^*$, corresponding to the best fitting k-dimensional linear space in terms of least square distances, between the centered cloud of points defined by $\mathbf{X}^*$ and the cloud of the projected points in the k-dimensional space, $\mathbf{X}(k)$. In other words it minimizes the sum of square distances $\sum_i e_i^2$ (Eckart-Young's Theorem)
- Using the decomposition (12) of the slide 49 we obtain,

$$\underbrace{var(\mathbf{X}^*)}_{\text{total var.}} = \frac{1}{N-1}\sum_i d_i^2 = \frac{1}{N-1}\sum_{i'} d_{i'}^2 + \frac{1}{N-1}\sum_i e_i^2$$

$$= \underbrace{var(\mathbf{X}(k))}_{\text{explain. var.}} + \underbrace{\frac{1}{N-1}\sum_i e_i^2}_{\text{unexplain. var.}}$$

Therefore the optimal solution in the sense of the least square distances, minimizes the variance that is left unexplained, i.e., maximizes the variance of the cloud of N points projected in a k-dimensional space (explained variance) - main goal of PCA!

Equivalence between the EVD and SVD approaches

We shall assume all singular values positive (otherwise we have to work with a slight different version of the SVD decomposition):

$$(\mathbf{X}^*)^T\mathbf{X}^* = (\mathbf{U}\mathbf{\Delta}\mathbf{V}^T)^T(\mathbf{U}\mathbf{\Delta}\mathbf{V}^T) = \mathbf{V}\mathbf{\Delta}^T\mathbf{U}^T\mathbf{U}\mathbf{\Delta}\mathbf{V}^T = \mathbf{V}\mathbf{\Delta}^2\mathbf{V}^T,$$

which is equivalent to say that,

$$\mathbf{S} = \mathbf{V}\left(\frac{1}{\sqrt{N-1}}\mathbf{\Delta}\right)^2\mathbf{V}^T \tag{13}$$

Hence the PC loadings, i.e., the eigenvectors of $\mathbf{S}$, are the right singular vectors of $\mathbf{X}^*$ and the corresponding PC standard deviations, the singular values of $\mathbf{X}^*$ divided by $\sqrt{N-1}$. The PC factor scores are given by

$$\mathbf{Y}^* = \mathbf{X}^*\mathbf{V} = \mathbf{U}\mathbf{\Delta}\mathbf{V}^T\mathbf{V} = \mathbf{U}\mathbf{\Delta},$$

and the left singular vectors verify

$$\mathbf{U} = \mathbf{X}^*\mathbf{V}\mathbf{\Delta}^{-1} = \mathbf{Y}^*\mathbf{\Delta}^{-1},$$

where $\mathbf{Y}^*\mathbf{\Delta}^{-1}$ is a matrix of normalized scores (more precisely, with constant standard deviations $\frac{1}{\sqrt{N-1}}$)

One can consider, alternatively, the SVD of $\frac{1}{\sqrt{N-1}}(\mathbf{X}^*)^T\mathbf{X}^*$. In this case the PCs variances λ_j are the squared singular values δ_j^2 of $\frac{1}{\sqrt{N-1}}(\mathbf{X}^*)^T\mathbf{X}^*$ (see the slides of Prof. Cadima)

comparing PCA via EVD and via SVD in R

R

```
# EVD APPROACH TO PCA
X<-iris[-5] # can be replaced by your own dataset or standardized
X.pca <- prcomp(X) # computes the PCA of X
loadings <- X.pca$rotation # eigenvectors of S=cov(X) (loadings)

sdev <- X.pca$sdev
# standard deviations of the PCs (square roots of the eigenvalues of S)

scores <- X.pca$x # scores (coordinates of the individuals w.r.t PCs)

# SVD APPROACH TO PCA
Xc <- scale(X,scale=FALSE) # Xc = centered X
X.svd<-svd(Xc) # computes the SVD of Xc
left.sing <- X.svd$u # left singular vectors of Xc
singvalues <- X.svd$d # singular values of Xc
right.sing <- X.svd$v # right singular vectors of Xc

# EQUIVALENCE BETWEEN EVD AND SVD APPROACHES

sdev; singvalues/sqrt(N-1)
# eigenvalues of S = square of sing values of Xc (divided by N-1)

head(loadings); head(right.sing) # loadings = right sing vectors

head(scores) ; head(left.sing%*%diag(singvalues))
# scores = normalized left sing vectors
```

A very useful decomposition...

Any matrix $\mathbf{C}_{N\times p}$ of rank r can be decomposed as a

$$\mathbf{C} = \mathbf{A}\,\mathbf{B}^T = \sum_{i=1}^{r} \mathbf{a}_i \mathbf{b}_i^T,$$

where $\mathbf{A} = [\,\mathbf{a}_1 \;\cdots\; \mathbf{a}_r\,]$ and $\mathbf{B} = [\,\mathbf{b}_1 \;\cdots\; \mathbf{b}_r\,]$, with $\mathbf{a}_i \in \mathbf{R}^N$ and $\mathbf{b}_i \in \mathbf{R}^p$

In particular, any matrix $\mathbf{C}$ of rank one, i.e., with proportional rows and proportional columns, can be decomposed as:

$$\mathbf{C} = \mathbf{a}\,\mathbf{b}^T = \begin{bmatrix} a_1 \\ \vdots \\ a_N \end{bmatrix} \begin{bmatrix} b_1 & \cdots & b_p \end{bmatrix}, \quad \text{with}$$

$$\mathbf{a} = \begin{bmatrix} a_1 \\ \vdots \\ a_N \end{bmatrix} = (a_1, \ldots, a_N) \in \mathbb{R}^N, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_p \end{bmatrix} = (b_1, \ldots, b_N) \in \mathbb{R}^p$$

The decomposition is not unique. For instance,

$$\mathbf{C} = \begin{bmatrix} 2 & 4 & 6 \\ 4 & 8 & 12 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \begin{bmatrix} 2 & 4 & 6 \end{bmatrix}$$

In the general case the decomposition can be obtained using the SVD...

Biplots

The biplots provide simultaneous representations of the individuals and variables of a data matrix in a low dimension space (usually of dimension two or three), using the SVD applied to the centered data matrix in order to obtain a decomposition of the type described in the previous slide

Let $\mathbf{X}^*$ be the matrix obtained by centering the p observed variables of a data matrix $\mathbf{X}_{N\times p}$ (i.e., column centering the matrix). We will assume $\mathbf{X}^*$ has rank p. Applying the SVD we can write,

$$\mathbf{X}^* = \mathbf{U}\mathbf{\Delta}\mathbf{V}^T \tag{14}$$

where,

- $\mathbf{U}_{N\times p}$ verifies $\mathbf{U}^T\mathbf{U} = \mathbf{I}_p$ is the matrix of left singular vectors of $\mathbf{X}^*$
- $\mathbf{V}_{p\times p}$ verifies $\mathbf{V}^T\mathbf{V} = \mathbf{I}_p$ is the matrix of right singular vectors of $\mathbf{X}^*$, i.e., the matrix of loadings of $\mathbf{X}$
- $\mathbf{\Delta}_{p\times p} = \text{diag}(\delta_1, \ldots, \delta_p)$ is a diagonal matrix containing the singular values of $\mathbf{X}^*$

Biplots (cont.)

Using the decomposition (14) of the previous slide we can decompose $\mathbf{X}^* = \mathbf{G}\mathbf{H}^T$ in many different ways. We will refer here two of them:

- $\mathbf{G} = \mathbf{U}\boldsymbol{\Delta}$ and $\mathbf{H} = \mathbf{V}$ - focuses on distances between individuals
- $\mathbf{G} = \mathbf{U}$ and $\mathbf{H} = \mathbf{V}\boldsymbol{\Delta}$ - focuses on covariances/correlations between variables

In the first case, $\mathbf{G} = \mathbf{U}\boldsymbol{\Delta}$ contains the left singular vectors scaled by the respective singular values which gives the factor scores (coordinates) of the individuals. Actually, the right singular vectors of $\mathbf{X}^*$ are eigenvectors of the covariance matrix $\mathbf{S}$, i.e, vectors of loadings of $\mathbf{X}$ and therefore the scores matrix is given

$$\mathbf{Y}^* = \mathbf{X}^*\mathbf{V} = \mathbf{U}\boldsymbol{\Delta}\mathbf{V}^T\mathbf{V} = \mathbf{U}\boldsymbol{\Delta}$$

The matrix $\mathbf{H} = \mathbf{V}$,corresponds to the matrix of right singular vectors, i.e. to the matrix of the vectors of loadings

Biplots (cont.)

Consider now the second case, $\mathbf{G}_{N\times p} = \mathbf{U}$ and $\mathbf{H}_{p\times p} = \mathbf{V}\mathbf{\Delta}$ and denote

$$\mathbf{G}^T = [\,\mathbf{g}^1 \;\cdots\; \mathbf{g}^N\,],$$

where $\mathbf{g}^j \in \mathbb{R}^p$ is the j-th row of $\mathbf{G}$. Similarly denote

$$\mathbf{H}^T = [\,\mathbf{h}^1 \;\cdots\; \mathbf{h}^p\,],$$

where $\mathbf{h}^k \in \mathbb{R}^p$ is the k-th row of $\mathbf{H}$ The rows of G and H are called, respectively, **markers of individuals and variables**. We have,

$$\begin{aligned}(N-1)\,\mathbf{S} &= (\mathbf{X}^*)^T\mathbf{X}^* = (\mathbf{G}\mathbf{H}^T)^T\mathbf{G}\mathbf{H}^T \\ &= \mathbf{H}\mathbf{G}^T\mathbf{G}\mathbf{H}^T = \mathbf{H}\mathbf{U}^T\mathbf{U}\mathbf{H}^T = \mathbf{H}\mathbf{H}^T\end{aligned}$$

Hence

$$(\mathbf{h}^j)^T\mathbf{h}^k = (N-1)s_{jk}^2,$$

that is, the inner product between the markers $\mathbf{h}^j$ and $\mathbf{h}^k$ is proportional to the covariance between the observed variables $\mathbf{x}_j$ and $\mathbf{x}_k$. In particular, the length of each variable marker is proportional to the standard deviation of the corresponding variable and we get, denoting θ_{jk} the angle between the variable markers $\mathbf{h}^j$ and $\mathbf{h}^k$,

$$\cos(\theta_{jk}) = r_{jk}$$

Euclidean and Mahalanobis distances

The usual squared (euclidean) distance between the individuals $\mathbf{x}^i, \mathbf{x}^\ell \in \mathbf{R}^p$ is

$$d_{i\ell}^2 = \|\mathbf{x}^i - \mathbf{x}^\ell\|^2 = (\mathbf{x}^i - \mathbf{x}^\ell)^T(\mathbf{x}^i - \mathbf{x}^\ell)$$

The (squared) Mahalanobis distance accounts for the dataset variability and generalizes the euclidean distance. Assuming the covariance matrix $\mathbf{S}$ invertible, the Mahalanobis distance between the individuals $\mathbf{x}^i, \mathbf{x}^\ell \in \mathbf{R}^p$ is defined as

$$\delta_{i\ell}^2 = (\mathbf{x}^i - \mathbf{x}^\ell)^T\mathbf{S}^{-1}(\mathbf{x}^i - \mathbf{x}^\ell)$$

The Mahalanobis distance between the individuals $\mathbf{x}^i = \mathbf{H}\mathbf{g}^i$ and $\mathbf{x}^\ell = \mathbf{H}\mathbf{g}^\ell$ is proportional to the (squared) euclidean distance between the corresponding markers $\mathbf{g}^i$ and $\mathbf{g}^\ell$. Actually, from relation (13) of slide 53, we obtain

$$(N-1)\,\mathbf{V}\boldsymbol{\Delta}^{-2}\mathbf{V}^T = (N-1)\,((\mathbf{X}^*)^T\mathbf{X}^*))^{-1} = \mathbf{S}^{-1}$$

and therefore

$$\begin{aligned}
(N-1)(\mathbf{g}^i - \mathbf{g}^\ell)^T(\mathbf{g}^i - \mathbf{g}^\ell) &= (N-1)(\mathbf{g}^i - \mathbf{g}^\ell)^T\boldsymbol{\Delta}\boldsymbol{\Delta}^{-2}\boldsymbol{\Delta}(\mathbf{g}^i - \mathbf{g}^\ell) \\
&= (N-1)(\mathbf{g}^i - \mathbf{g}^\ell)^T\boldsymbol{\Delta}(\mathbf{V}^T\mathbf{V})\boldsymbol{\Delta}^{-2}(\mathbf{V}^T\mathbf{V})\boldsymbol{\Delta}(\mathbf{g}^i - \mathbf{g}^\ell) \\
&= (\mathbf{g}^i - \mathbf{g}^\ell)^T(\mathbf{V}\boldsymbol{\Delta})^T\mathbf{S}^{-1}(\mathbf{V}\boldsymbol{\Delta})(\mathbf{g}^i - \mathbf{g}^\ell) \\
&= (\mathbf{g}^i - \mathbf{g}^\ell)^T\mathbf{H}^T\mathbf{S}^{-1}\mathbf{H}(\mathbf{g}^i - \mathbf{g}^\ell) \\
&= (\mathbf{H}(\mathbf{g}^i - \mathbf{g}^\ell))^T\mathbf{S}^{-1}\mathbf{H}(\mathbf{g}^i - \mathbf{g}^\ell) \\
&= (\mathbf{x}^i - \mathbf{x}^\ell)^T\mathbf{S}^{-1}(\mathbf{x}^i - \mathbf{x}^\ell) = \delta_{i\ell}^2, \quad \text{(UFF!)}
\end{aligned}$$

"Exact" interpretation of a biplot

Summarizing, we have the following "exact interpretations":

- The cosine of the angle between two variable markers is the correlation coefficient between these variables
- The length of a variable marker is proportional to the standard deviation of the variable
- The euclidean distance between individual markers is proportional to the Mahalanobis distance between the corresponding individuals
- The coordinate of the orthogonal projection of an individual marker $\mathbf{g}^i$ onto the line defined by a variable marker $\mathbf{h}^j$ equals value of the individual on that variable divided by the standard deviation of the variable

The last property follows directly from relation $\mathbf{X}^* = \mathbf{G}\mathbf{H}^T$, which implies that $x^*_{ij} = (\mathbf{g}^i)^T\mathbf{h}^j$ and therefore,

$$\text{proj}_{\mathbf{h}^j}(\mathbf{g}^i) = \frac{(\mathbf{g}^i)^T\mathbf{h}^j}{\|\mathbf{h}^j\|^2}\mathbf{h}^j = \frac{x^*_{ij}}{\|\mathbf{h}^j\|^2}\mathbf{h}^j$$

Note that $\|\text{proj}_{\mathbf{h}^j}(\mathbf{g}^i)\| = \frac{|x^*_{ij}|}{\|\mathbf{h}^j\|}$

"Approximated interpretations" of a biplot

Let $\mathbf{G}^T(m) = \mathbf{U}(m)^T$ and $\mathbf{H}^T(m) = \mathbf{\Delta}(m)\mathbf{V}(m)^T$, $1 \leq m \leq p$ be the submatrices containing the first m rows of $\mathbf{G}^T$ and $\mathbf{H}^T$, resp. Denote

$$(\mathbf{G}(m))^T = [\, \mathbf{g}_m^1 \ \cdots \ \mathbf{g}_m^N \,], \qquad (\mathbf{H}(m))^T = [\, \mathbf{h}_m^1 \ \cdots \ \mathbf{h}_m^p \,]$$

The rows of $G(m)$ and $H(m)$ give approximations to the markers of the individuals and variables. We have:

- The cosines of the angles between variable markers are approximately equal to the correlation coefficients between these variables
- The length of a variable marker is approximately proportional to the standard deviation of the variable
- The (euclidean) distances between individual markers are approximately proportional to the Mahalanobis distance between these individuals
- The coordinate of the orthogonal projection of an individual marker $\mathbf{g}^i$ onto the line defined by a variable marker $\mathbf{h}^j$ is approximately equal to the value of the individual on that variable divided by the standard deviation of the variable

The higher the proportion of the explained variance by the first m PCs, the better the approximations in the previous points

Displaying a biplot using in R software

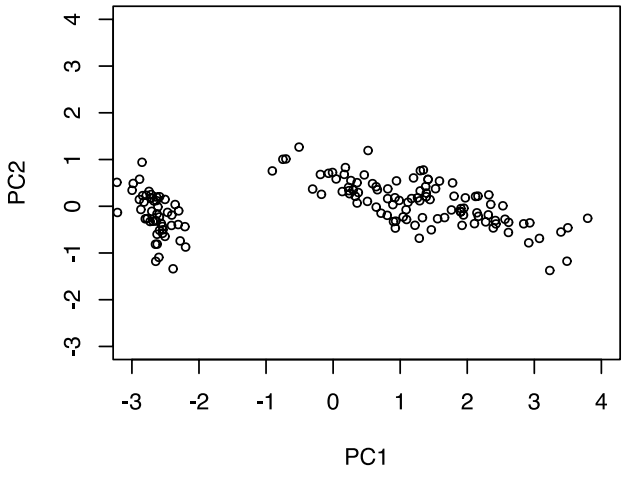
We can display the biplot of the iris flowers data set in two distinct ways, with the `biplot` function:

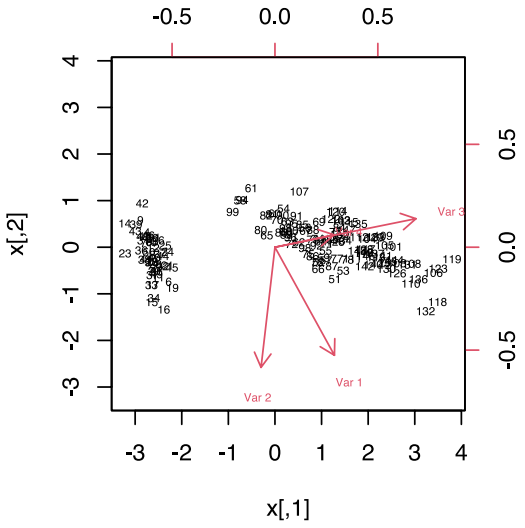
R

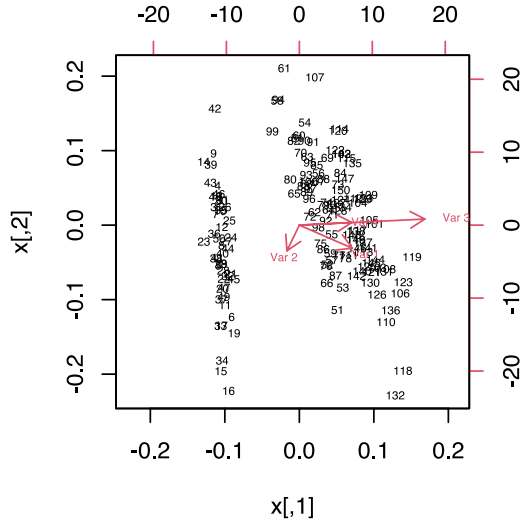
```
Xc <- scale(iris[-5],scale=FALSE) #centred iris flower dataset
iris.svd <- svd(Xc) # compute the svd UDV^T of the centred iris dataset
U <- iris.svd$u
V <- iris.svd$v
Delta <- diag(iris.svd$d) # creates a diagonal matrix with diagonal with
the singular values
par(mfrow=c(2,2)) # 4 simultaneous windows
plot(iris.pca$x[,1:2],asp=TRUE,pch=16) # plot
biplot(U % * % Delta, V, asp=TRUE,cex=.5) # G=U Delta; H=V
biplot(U, V% * %Delta, asp=TRUE,cex=.5) # G=U; H=V Delta
biplot(iris.acp, asp=TRUE,cex=.5) # computes the second species
```

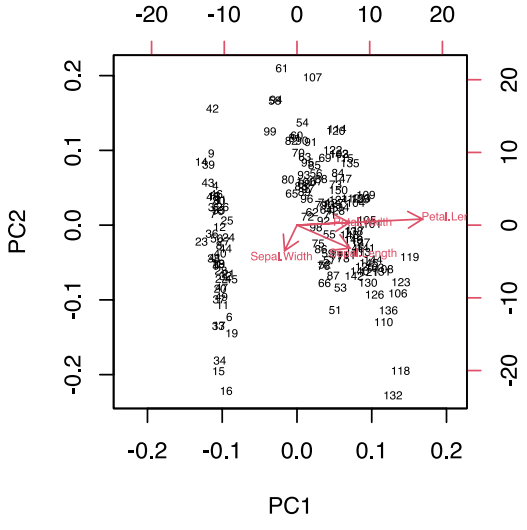
Iris flower biplots

The output obtained by the script of the previous sliede









Some notes on generalized euclidean distances

If $\mathbf{S}$ is a symmetric positive definite (hence invertible) matrix of order p, we define the (squared) **generalized euclidean distance** between the vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^p$ as

$$d^2_{\mathbf{S}}(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T \mathbf{S}^{-1} (\mathbf{x} - \mathbf{y})$$

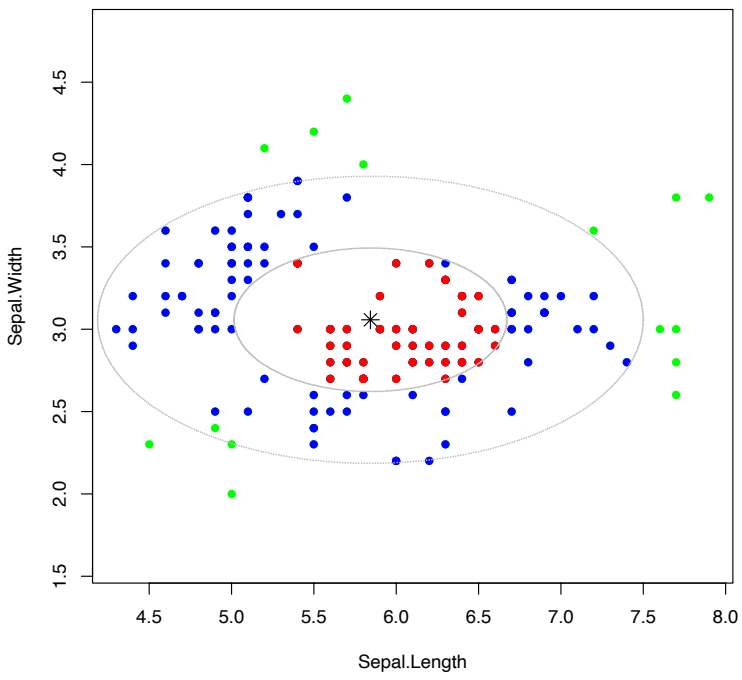
- If $\mathbf{S} = \mathbf{I}_p$, $d^2_{\mathbf{S}}(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T (\mathbf{x} - \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^2$ is the usual (squared) Euclidean distance between $\mathbf{x}$ and $\mathbf{y}$
- If $\mathbf{S} = cov(\mathbf{X})$, $d^2_{\mathbf{S}}(\mathbf{x}, \mathbf{y})$ is the (squared) Mahalanobis distance between $\mathbf{x}$ and $\mathbf{y}$
- When the variables are uncorrelated, the covariance matrix $\mathbf{S}$ is a diagonal matrix containing the variances of the p variables $d^2_{\mathbf{S}}(\mathbf{x}, \mathbf{y})$ equals (squared) euclidean distance between the corresponding standardized variables
- The Mahalanobis distance of between an individual and the cloud's center of gravity is 'smaller' along the directions of $\mathbf{X}$ of greater variability and generalizes to the multivariate case the idea of how many standard deviations each observed vector $\mathbf{x}$ is far away from the mean. This can be useful, for instance, to detect outliers. . .

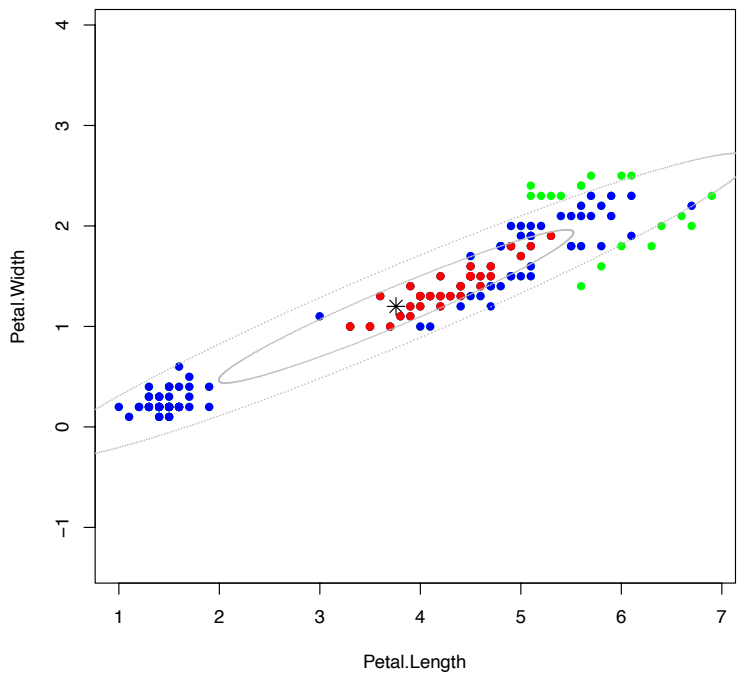
Mahalanobis distances for the iris flower data set

The variance-covariance matrices of the sepal and the petal widths are, respectively:

$$\begin{bmatrix} 0.6856935 & -0.0424340 \\ -0.0424340 & 0.1899794 \end{bmatrix}, \quad \begin{bmatrix} 3.116278 & 1.2956094 \\ 1.295609 & 0.5810063 \end{bmatrix}$$

The iris flowers at Mahalanobis distances from the mean less than or equal to 1 are displayed in red and the iris flowers at mahalanobis distances greater than 1 and smaller than or equal to 2 displayed in blue color





Contribution and square cosine

- Recall that the contribution of individual i to a PC_k is the part of the variance of PC_k that is due to individual i:

$$ctr_{i,k} = \frac{(y^*_{i,k})^2}{\sum_{j=1}^{N}(y^*_{j,k})^2}$$

Individuals with contributions above the average are usually more important to interpret the PC

- A related notion is the square cosine of a PC k with an individual i, which gives the contribution of the PC to the squared distance of the individual to the origin:

$$\cos^2_{i,k} = \frac{(y^*_{i,k})^2}{\sum_{j=1}^{p}(y^*_{i,j})^2}$$

Square cosines can be added together to assess the quality of representation of an individual i by its projection on the space defined by several PCs. For instance, the quality of representation of individual i in the PFP is given by,

$$\cos^2_{i,1} + \cos^2_{i,2} = \frac{(y^*_{i,1})^2 + (y^*_{i,2})^2}{\sum_{j=1}^{p}(y^*_{i,j})^2}$$

Only well represented individuals should be interpreted!

PCA interpretation - summary

- Proportion of the variance explained by a PC
- Correlation between a variable and a PC
- Contribution of an individual to a PC
- Squared cosine of a PC with an individual
- Biplot

CLUSTER ANALYSIS

Definition of clustering

Given a collection of N objects, $X = \{x_1, \ldots, x_N\}$, one seeks a partition of X into K nonempty disjoint sets (the *clusters*),

$$X = \mathcal{C}_1 \cup \cdots \cup \mathcal{C}_K$$

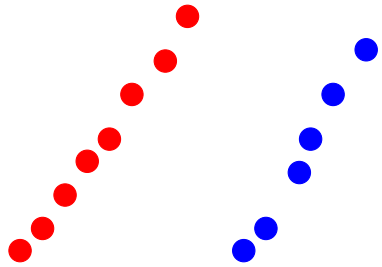
such that, *given the notion of resemblance considered*, it

- maximizes the internal homogeneity or cluster cohesion, or equivalently, it minimizes the intra-cluster variability - objects belonging to the same cluster should share the similar features
- it maximizes the external heterogeneity or cluster separation, i.e., it maximizes the inter-cluster separability - objects belonging to distinct clusters should be very dissimilar and have clear distinguished features

Examples

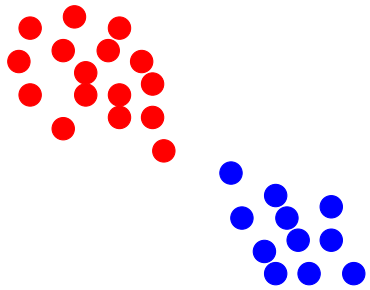
strong internal cohesion

strong separation



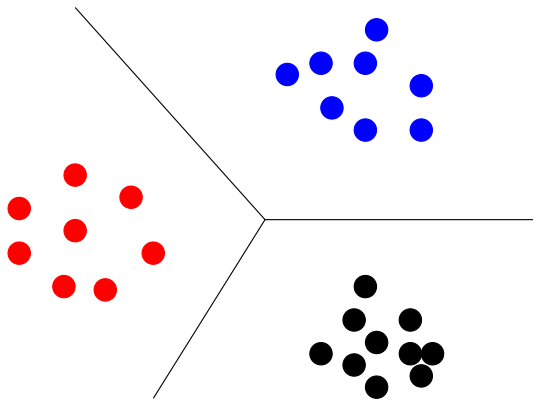
weak internal cohesion

strong separation

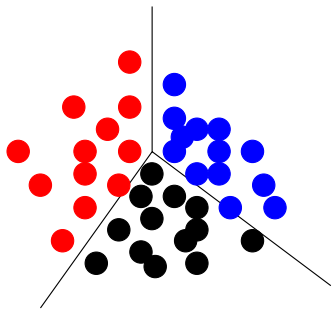


strong internal cohesion

weak separation



clear clustering structure



artificial clustering structure

Clustering

- Clustering always imposes some kind structure on the data, even when no special structure or discontinuities are present!

  For instance, many clustering techniques tend to form globular clusters, e.g., with elliptical or spherical shapes

- How to choose the best partition ?

Huge solution space...

The number of distinct partitions of N elements into K clusters $(1 \leq K \leq N)$ equals

$$\xi(N,K) = \frac{1}{K!}\sum_{j=1}^{K}\binom{K}{j}(-1)^{K-j}j^N,$$

which is a huge number, known as Stirling number of second kind, even for relatively small values of N and K, making impossible to to find the best partition by exhaustive search.

For instance, the number of partitions of a set with 25 elements into 8 clusters equals

$$\xi(25,8) = 69022372111836858$$

For N large and K fixed, $\xi(N,K) \approx \frac{K^N}{K!}$

In the previous example, one gets $\xi(25,8) \approx \frac{8^{25}}{8!} = 9.369775e{+}17$

Common steps in a cluster analysis

- Variables/features selection
  - Which variables (continuous, categorical, ordinal, binary, ...), encode as much as possible the information concerning the task, avoiding redundancy (i.e., highly correlated variables) ?
  - Standardize/normalize the variables to balance their importance ?
- Clustering model
  - Which combination of a clustering method with a distance/dissimilarity is more appropriate?
- Cluster validation
  - Internal: How many groups and how to assess the quality of the clusters ?
  - External: How the clustering results compare with the outcomes obtained using different clustering models or how they compare with known information ?
- Interpretation of the results
  - Are the outcomes interpretable in the context of the problem ?
  - Which variables/features (active/supplementary) are more important to characterize the clusters ?

Cluster model

A cluster model is build upon two concepts:

- the notion of distance/dissimilarity between individuals and clusters should be adequate to the type of variables involved and to the type of results sought
- the clustering method should take into account the type of structure/shape of the clusters sought (rounded shape/arbitrary shape/...) and characteristics of the method itself (sensitivity to outliers/noise/ldots), computational issues (scalability for large datasets), etc...

When several cluster models are appropriate one should compare the outputs of such models to seek for common patterns that emerge from these clustering models - robust solutions

Example of numerical dataset - iris flower dataset

The well known iris flower dataset contains the sepal and petal lengths and widths (in cm) of 150 iris flowers

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	5.1	3.5	1.4	0.2
2	4.9	3.0	1.4	0.2
3	4.7	3.2	1.3	0.2
4	4.6	3.1	1.5	0.2
5	5.0	3.6	1.4	0.2
6	5.4	3.9	1.7	0.4
7	4.6	3.4	1.4	0.3
8	5.0	3.4	1.5	0.2
9	4.4	2.9	1.4	0.2
10	4.9	3.1	1.5	0.1

- How to measure the distance between each pair of iris flowers ?
- Standardize (z-score normalization) or normalize (min-max scaling) the variables in order that the differences between all variables contribute equally ?

Example - a freshwater fish dataset in West Africa

In the biogeography it is common to use biological markers (e.g., river fish species) to distinguish between sites (e.g., river basins)

	annectens	ansorgi	bichir	endlicheri
GAMBIE	1	0	1	0
GEBA	0	1	1	1
CRUBAL	0	1	0	0
KONKOURE	0	0	0	0
KOLENTE	0	0	0	0
LSCARC	0	0	0	0
ROKEL	0	0	0	0

- Which type of variable/feature is more appropriate to encode this type data ?
- How to assess the similarity between river basins given the distribution of fish species ?
- How to assess the similarity between fish species given their distribution by the sites ?

Example categorical dataset

The following two-way contingency table encodes the country of residence and language spoken by 1000 inhabitants in 5 countries

	English	French	Spanish	German	Italian	Total
Canada	688	280	10	11	11	1000
USA	730	31	190	8	41	1000
England	798	74	38	31	59	1000
Italy	17	13	11	15	944	1000
Switzer.	15	222	20	648	95	1000
Total	2248	620	269	713	1150	5000

({\bf source}: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3718710/)

- How to assess the similarity between countries given the languages spoken in these countries ?
- How to assess the similarity between the spoken languages given their distribution by the countries ?

Properties of a dissimilarity measure / distance

In order to tackle the previous questions we first need to establish which properties a dissimilarity/distance notion should have.

A dissimilarity measure on a set X is a real function

$$d : X \times X \to \mathbb{R},$$

such that, for all $x, y \in X$, we have

- $d(x, y) \geq 0$
- $d(x, y) = 0$ if and only if $x = y$
- $d(x, y) = d(y, x)$

We call d a distance if moreover d verifies the *triangle inequality*

- $d(x, z) \leq d(x, y) + d(y, z)$ for all $x, y, z \in X$,

Three important distances

Consider $\mathbf{x} = (x_1, \ldots, x_N)$ and $\mathbf{y} = (y_1, \ldots, y_N)$ of $\mathbf{R}^n$

- The usual euclidean distance:

$$d(x, y) = \sqrt{\sum_{i=1}^{N} |x_i - y_i|^2}$$

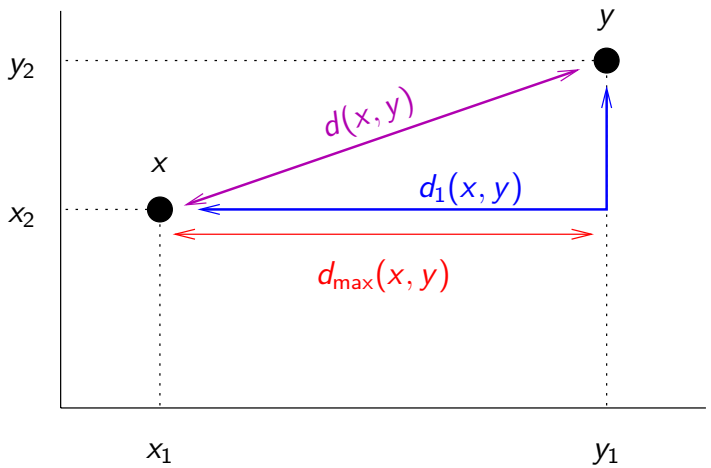
- The Manhattan distance (also called city block or taxicab distance):

$$d_1(x, y) = \sum_i |x_i - y_i|.$$

- The so-called maximum distance (also called Chebyshev distance):

$$d_{\max}(x, y) = \max_i |x_i - y_i|$$

Relation among the 3 distances



For all $x, y \in \mathbb{R}^N$ we have $d_1(x, y) \geq d(x, y) \geq d_{\max}(x, y)$

- For the taxi-cab and euclidean distances all differences $|x_i - y_i|$, $i = 1, \ldots, N$, have approximately the same relative weight in the computation of the overall distance
- For the maximum distance only the variable(s) i yielding the largest difference $|x_i - y_i|$ accounts for the overall distance

The Canberra distance

If $\mathbf{x}, \mathbf{y}$ are N-dimensional vectors with positive components, one can define the so-called **Canberra** distance

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{N} \frac{|x_i - y_i|}{x_i + y_i}$$

- This distance is a weighted version of the Manhattan distance that is sensitive to differences between values x_i and y_i of small amplitudes.
- It is invariant under differentiated changes of scale in each variable but not under variables centering. Only the relative proportion between the differences of the coordinates and their sum are importante.

When to standardize the data ?

- Usually, the euclidean distance between original numerical variables is employed if all variables are expressed in the same units and similar scales of measurement. Otherwise, it is usually better to standardize the data to give the same weight to all variables.

- It could also be interesting to explore if other types of dissimilarities (for instance, the Canberra or Mahalanobis distance), could be more appropriate...

Dissimilarity measures for binary data

Consider binary vectors $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{y} = (y_1, \ldots, y_n)$ and define

a: nr components where both variables take value 1 (positive agreement)

b: nr of components where $\mathbf{x}$ take value 1 and $\mathbf{y}$ value 0 (disagreement)

c: nr of components where $\mathbf{x}$ take value 0 and $\mathbf{y}$ value 1 (disagreement)

d: nr components where both variables take value 0 (negative agreement)

- Simple matching (counts double-zeroes, is suitable if 0-1 represent equally valued attributes like male-female):

$$S(\mathbf{x}, \mathbf{y}) = \frac{a+d}{a+b+c+d} \implies D(\mathbf{x}, \mathbf{x}) = 1 - S(\mathbf{x}, \mathbf{x}) = \frac{b+c}{a+b+c+d}$$

- Jaccard coefficient (does not count double zeroes. Suitable if 0-1 represent unequal valued attributes, like species presences-absences):

$$J(\mathbf{x},) = \frac{a}{a+b+c} \implies D(\mathbf{x}, \mathbf{y}) = 1 - J(\mathbf{x}, \mathbf{y}) = \frac{b+c}{a+b+c}$$

Example

Assume that we have two binary variables **x** and **y** representing presences (1) and absences (0) of two species at 16 spots:

$$\mathbf{x} = (0,1,1,1,0,0,0,0,0,0,0,0,0,1,0,0), \qquad \mathbf{y} = (0,1,0,0,1,0,0,0,0,1,0,0,0,0,0,1)$$

We want to determine how similar are the two species with regard to their distribution in the 16 spots. Computing the positive and negative agreements/disagreements, we get $a = 1$, $b = 3$, $c = 3$ and $d = 9$ ($a + b + c + d = 16$). Therefore we have.

- Simple matching: $\frac{a+d}{a+b+c+d} = 10/16$
- Jaccard coefficient: $\frac{a}{a+b+c} = 1/7$

The asymmetrical character of Jaccard's coefficient seems to the be a more suitable similarity to create homogeneous groups of species with respect to their distribution in the spots

R

```
# The R function dist with the method ‘‘binary’’ computes the
dissimilarity as d(x,y) = 1 − S(x,y), where S is the Jaccard coefficient

d = dist(cbind(x,y),method=‘‘binary’’,diag=FALSE,upper=FALSE,p=2)

Several other dissimilarity measures well suited for binary data in the
framework of ecology and community composition data are available via
the function dist.ldc from the ADESPATIAL package
```

χ^2-distance for nominal data

- Let $\mathbf{X} = [x_{ij}]$ be a contingency table, where x_{ij} is the observed frequency in category A_i of a nominal variable A and category B_j of a nominal variable B (assuming nonzero row and column sums). Let I and J be the number of categories of A and B and $N = \sum_{i,j} x_{ij}$ the total number of observations.
- Dividing each row i by the corresponding row total, $x_{i\cdot} = \sum_j x_{ij}$, we obtain the so-called ith row-profile, $\left(\frac{x_{i1}}{x_{i\cdot}}, \ldots, \frac{x_{iJ}}{x_{i\cdot}}\right)$, which corresponds to the conditional distribution of variable B assuming category a_i of A.
- The set of the I row-profiles defines a cloud of I points in $\mathbb{R}^J$ and the centroid of this cloud, $\frac{1}{I}\sum_i \left(\frac{x_{i1}}{x_{i\cdot}}, \ldots, \frac{x_{iJ}}{x_{i\cdot}}\right) \in \mathbb{R}^J$, is called is the mean row-profile.
- If variables A and B are independent, i.e., $x_{ij} = \frac{x_{i\cdot}x_{\cdot j}}{N}\ \forall i,j$, ith row-profile verifies

$$\left(\frac{x_{i1}}{x_{i\cdot}}, \ldots, \frac{x_{iJ}}{x_{i\cdot}}\right) = \left(\frac{x_{\cdot 1}}{N}, \ldots, \frac{x_{\cdot J}}{N}\right) = (f_{\cdot 1}, \ldots, f_{\cdot J}),$$

  where $f_{\cdot j} = \sum_i f_{ij}$ are the column marginals of the relative frequencies $f_{ij} = \frac{x_{ij}}{N}$. In particular, all row-profiles are equal to the mean row-profile. If A and B are not independent, the row-profiles spread away from the mean row-profile.
- The squared χ^2-distance between the ith and ℓth row-profiles is defined as,

$$d^2_{\chi^2}(i,\ell) = \sum_{j=1}^{J} \frac{1}{f_{\cdot j}}\left(\frac{x_{ij}}{x_{i\cdot}} - \frac{x_{\ell j}}{x_{\ell\cdot}}\right)^2 = \sum_{j=1}^{J} \frac{1}{f_{\cdot j}}\left(\frac{f_{ij}}{f_{i\cdot}} - \frac{f_{\ell j}}{f_{\ell\cdot}}\right)^2$$

  (the weights in the inverse proportion of the column marginal frequencies $f_{\cdot j}$ increase the importance of the small differences between rare categories).

Example

Consider again the two-way contingency table containing the distribution by **country of residence** of the **primary language spoken** of 5000 inhabitants (see slide 13)

	English	French	Spanish	German	Italian	Total
Canada	688	280	10	11	11	1000
USA	730	31	190	8	41	1000
England	798	74	38	31	59	1000
Italy	17	13	11	15	944	1000
Switz.	15	222	20	648	95	1000
Total	2248	620	269	713	1150	5000

(source: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3718710/)

χ^2-distance between the row-profiles

- The corresponding 5 row-profiles and mean row-profile are given below

	English	French	Spanish	German	Italian	Totals	
Canada	0.688	0.280	0.010	0.011	0.011	1.000	
USA	0.730	0.031	0.190	0.008	0.041	1.000	
England	0.798	0.074	0.038	0.031	0.059	1.000	
Italy	0.017	0.013	0.011	0.015	0.944	1.000	
Switz.	0.015	0.222	0.020	0.648	0.095	1.000	
mean	0.4496	0.124	0.0538	0.1426	0.230	1.000	(verify)
f.j	0.4496	0.124	0.0538	0.1426	0.230	1.000	(verify)

- The 5 row-profiles define a a cloud of $I = 5$ points in $\mathbb{R}^J$, with $J = 5$ (number of columns) with centroid given by the mean row-profile
- The squared χ^2-distance between the row profiles of *Canada* and *Switzerland* is

$$d^2_{\chi^2}(1,5) = \frac{(0.688-0.015)^2}{0.4496} + \frac{(0.280-0.222)^2}{0.124} + \frac{(0.010-0.020)^2}{0.0538} + \frac{(0.011-0.648)^2}{0.1426} + \frac{(0.011-0.095)^2}{0.230} = 3.912575$$

- We define similarly the set of 5 column-profiles, which can be regarded as a cloud of $J = 5$ points in $\mathbb{R}^I$, with $I = 5$ and the corresponding pairwise squared χ^2-distances (left as an exercise).
- The correspondence analysis (CA) allows to study and visualize the relationships of a contingency table when the number of categories is high.

The corresponding R code

The R function dist.ldc from the package ADESPATIAL computes the χ^2-distance matrix between every pair of row-profiles

R

```
library(adespatial)
tab<-matrix(c( 688, 280, 10 , 11 , 11, 730, 31, 190, 8 , 41, 798, 74,
38, 31, 59, 17, 13, 11, 15, 944, 15, 222, 20, 648, 95),
nrow=5, byrow = TRUE)
colnames(tab)<-c("English", "French", "Spanish", "German", "Italian")
rownames(tab)<-c("Canada","USA","England","Italy","Switz.")
tab
d.chisqr<-dist.ldc(tab,method="chisquare")
d.chisqr
```

We obtain the following distance matrix (d_{χ^2}) between row-profiles

Countries	Canada	USA	England	Italy
USA	1.0536310			
England	0.6297091	0.6780536		
Italy	2.3154271	2.2966246	2.1925680	
Switzerland	1.9780231	2.2030640	2.0546442	2.5094977

For instance, $d^2_{\chi^2}(r_1, r_5) = (1.9780231)^2 = 3.912575$, as computed in the previous slide

Dissimilarity measures for variables

- An usual similarity notion between two variables x and y is Pearson's correlation coefficient

$$r = \frac{s_{xy}^2}{s_x \, s_y}$$

  This similarity can be transformed into a dissimilarity using the transformation $d = \sqrt{1 - r^2}$, which take values in the interval $[0, 1]$

- Highly linearly correlated variables (positively or negatively) will have $d \approx 0$ while for uncorrelated variables $d \approx 1$
- Alternatively, we can define $d = (1 - r)/2$. In this case the strength of the linear relationship and the direction are both accounted
- We can use the above dissimilarity measures to cluster variables. Each cluster will consist of a set of variables highly correlated. This can be useful to detect redundancies and can give an idea of the number of principal dimensions of data

Clustering methods

- Distance-based models rely only on pairwise dissimilarities between individuals
- Density-based clustering seeks for high density regions of points (clusters) separated by low density of points (noise)
- Model-based clustering assumes that the data in each cluster is drawn from some probabilistic distribution (the standard model is a finite mixture of multivariate gaussians) and assign a degree of membership (probability) to each element to belong to a cluster. Can be considered as generalizations of some distance-based clustering methods
- Constrained-clustering methods, are clustering methods that also account for other type of information, like spatial relationships between observations (for instance, contiguity relationships between cells in a map)
- ...

Two important types of clustering

- Hierarchical clustering - produces a *nested* structure of partitions and do not requires that the number of clusters is known *a priori*:
  - *Hierarchical agglomerative (or ascending) clustering algorithm* (HAC) - starts from the partition consisting of N clusters with one individual per cluster (*singletons*) and proceeds until a unique group is obtained.
  - *Divisive clustering algorithm* - proceeds in the opposite way and are usually more computacional demanding, being more seldom used (not considered in this course)
- Partitional clustering - produces *flat* (non-nested) partition and requires that the number of clusters is known *a priori*. Usually seeks to maximize some criterion like the **intra-cluster homogeneity** or the **inter-cluster heterogeneity**.

Hierarchical ascending clustering algorithm

Algorithm

Input: *the proximity matrix containing the pairwise dissimilarities between N individuals $x_1, \ldots, x_N$*

- *Starts with N clusters containing a single object each (singletons);*
- *Merges the least dissimilar pair of clusters into a new cluster, according to the given definition of distance between clusters, and updates the proximity matrix (reducing its order by one);*
- *Repeats step 2 $N - 1$ steps, until only the cluster containing all individuals is obtained.*

Output: *the sequence (of length $N - 1$) of the clusters aggregated during the clustering algorithm along with pairwise distances between these merged clusters*

Once two individuals are grouped together they cannot be separate at a posterior stage.

Dissimilarity between clusters

The dissimilarity $d_{i,j} = D(\mathcal{C}_i, \mathcal{C}_j)$, between clusters C_i and C_j with n_i and n_j elements, respectively, depends on the aggregation method:

- Single-linkage or nearest-neighbor:

$$d_{i,j} = \min_{x \in \mathcal{C}_i, y \in \mathcal{C}_j} d(x, y)$$

- Complete-linkage or furthest-neighbor:

$$d_{i,j} = \max_{x \in \mathcal{C}_i, y \in \mathcal{C}_j} d(x, y)$$

- Average:

$$d_{i,j} = \frac{1}{n_i n_j} \sum_{x \in C_i} \sum_{y \in C_j} d(x, y)$$

- Centroid
- Median
- Ward or minimum-variance clustering
- ...

Updating formula for HAC

- For all aggregation methods that we are going to consider, the dissimilarity between two merged clusters, say $\mathcal{C}_i \cup \mathcal{C}_j$, and each one of the remaining clusters $\mathcal{C}_k$,

$$d_{ij,k} = D(\mathcal{C}_i \cup \mathcal{C}_j, \mathcal{C}_k),$$

  can be determined in terms of the pairwise dissimilarities,

$$d_{i,j} = D(\mathcal{C}_i, \mathcal{C}_j), \quad d_{i,k} = D(\mathcal{C}_i, \mathcal{C}_k), \quad d_{j,k} = D(\mathcal{C}_j, \mathcal{C}_k)$$

- In other words, the proximity matrix containing the pairwise distances between the clusters at a given step $\ell + 1$ can be determined in terms of the proximity matrix containing the pairwise distances between the clusters at the previous step ℓ, via a convenient updating formula

- Therefore and unlike many other statistical methods like PCA, the HAC algorithm does not require the knowledge of the original data matrix $\mathbf{X}$, but only the knowledge of the proximity matrix containing the pairwise distances between the elements of $\mathbf{X}$.

Example of updating formulas

- Single-linkage or nearest-neighbor:

$$d_{ij,k} = \min\{d_{i,k}, d_{j,k}\}$$

- Complete-linkage or furthest-neighbor

$$d_{ij,k} = \max\{d_{i,k}, d_{j,k}\}$$

- Average

$$d_{ij,k} = \frac{n_i d_{i,k} + n_j d_{j,k}}{n_i + nj}$$

- see Lance-Williams table
- see Lance-Williams table
- see Lance-Williams table

Dendrogram

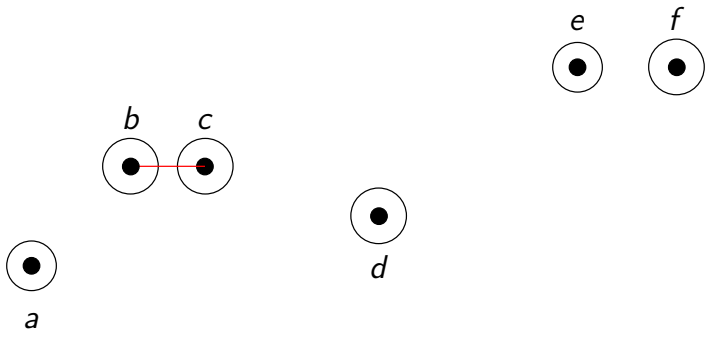
The sequence of length $N - 1$ of the merged clusters and the corresponding fusion costs (i.e., the distance between the merged clusters) can be graphically represented by a special tree graph called dendrogram

- Dendrograms are tree-like diagrams made of branches that join terminal nodes (*leaves*)
- The branches represent clusters and the heights at which the branches are connected represent fusion costs. The leaves represent the objects
- The lifetime of a branch is the difference of fusion costs between the step in which it appears and the step in which it is aggregated

Example: step -1 (initial step)

As an example we are going to apply the single-linkage clustering algorithm to a set of 6 points

$X = \{a, b, c, d, e, f\}$



PROXIMITY MATRIX

	a	b	c	d	e
b	0.7				
c	1.0	0.3			
d	1.8	.1.3	0.9		
e	2.9	2.4	1.9	1.3	
f	3.4	2.8	2.4	1.7	.5

At the initial step all clusters are singletons

Next step merges the clusters $\{b\}$ and $\{c\}$

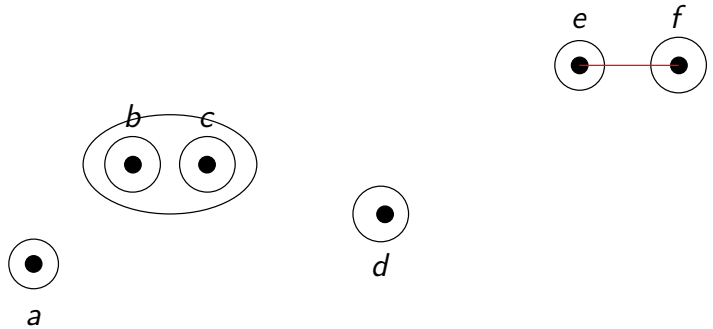
with fusion cost 0.3 (the least dissimilar pair) and in each dashed box

the minimum value is chosen, reducing the proximity matrix order by one,

and defining the dissimilarities between each one of the singletons and the new formed cluster $\{b, c\}$

Step -2

$X = \{a, b, c, d, e, f\}$

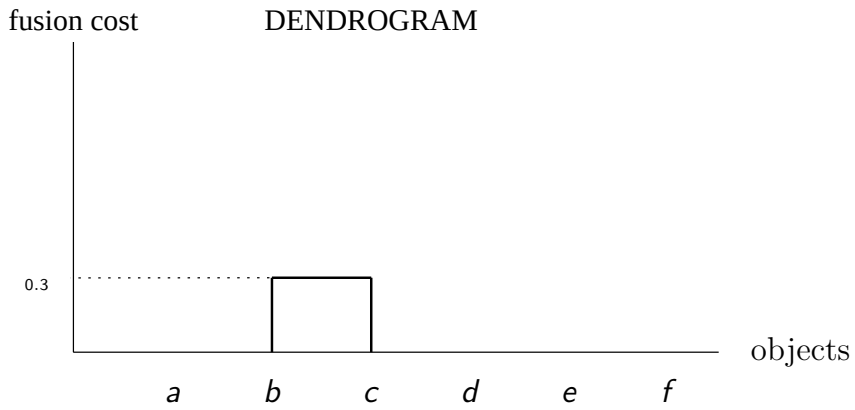


PROXIMITY MATRIX

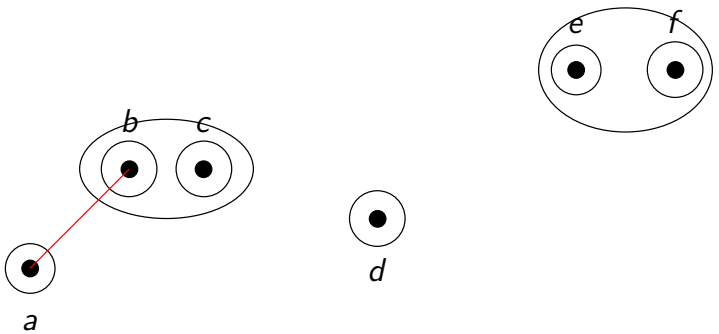
	a	$\{b, c\}$	d	e
$\{b, c\}$	0.7			
d	1.8	0.9		
e	2.9	1.9	1.3	
f	3.4	2.4	1.7	0.5

Next step merges the singletons $\{e\}$ and $\{f\}$

with fusion cost 0.5



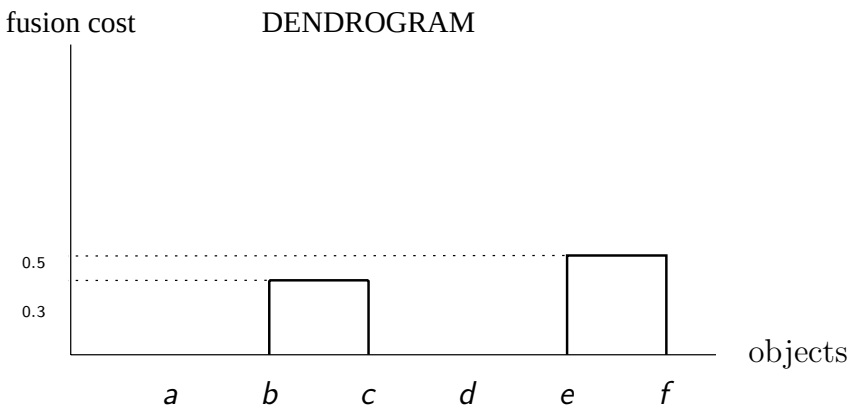
Step - 3



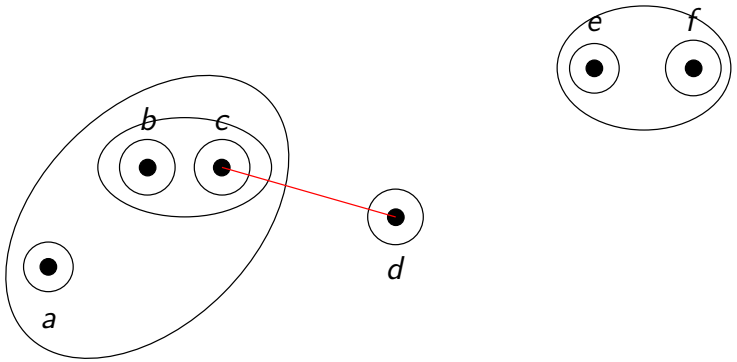
PROXIMITY MATRIX

	a	$\{b,c\}$	d
$\{b,c\}$	0.7		
d	1.8	0.9	
$\{e,f\}$	2.9	1.9	1.3

Next step merges the pair of clusters $\{a\}$ and $\{b,c\}$ with fusion cost 0.7



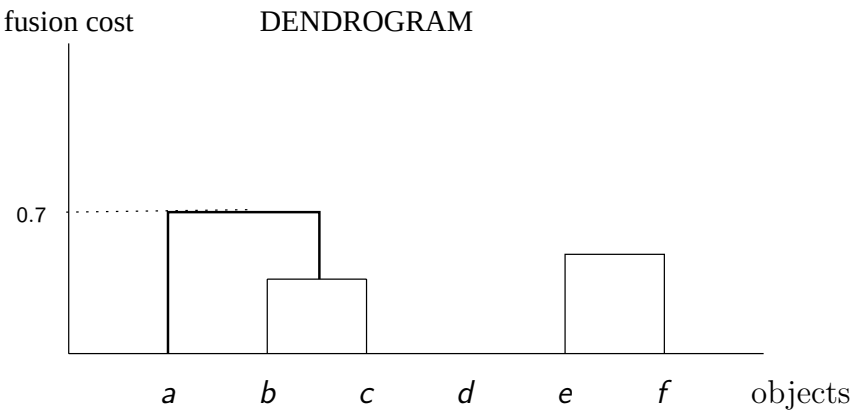
Step - 4



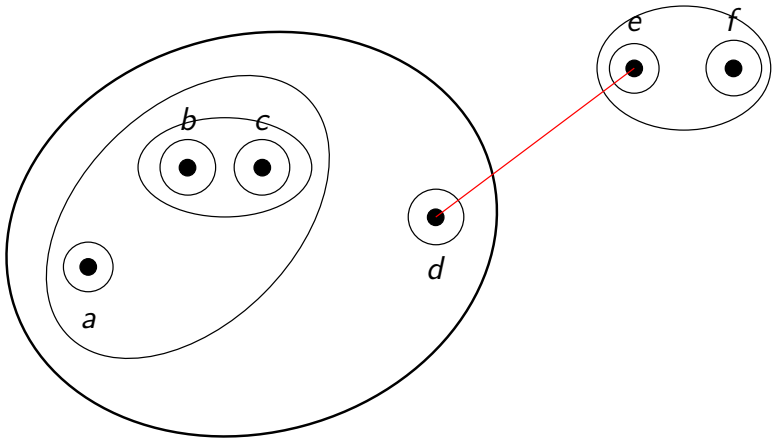
PROXIMITY MATRIX

	$\{a, b, c\}$	d
d	0.9	
$\{e, f\}$	1.9	1.3

Next step merges the clusters $\{a, b, c\}$ and $\{d\}$ with fusion cost 0.91



Step - 5

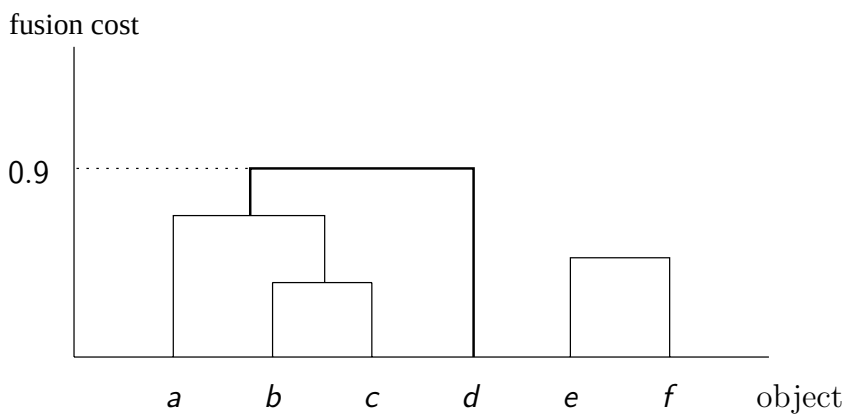


Next step is the final one and merges the clusters $\{a, b, c, d\}$ and $\{e, f\}$ with fusion cost 1.3

PROXIMITY MATRIX

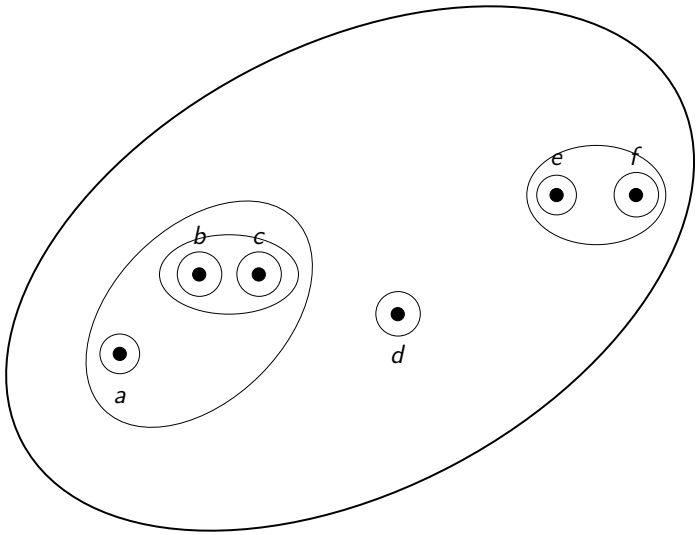
	$\{a, b, c, d\}$
$\{e, f\}$	1.3

DENDROGRAM



step - 6 (final step)

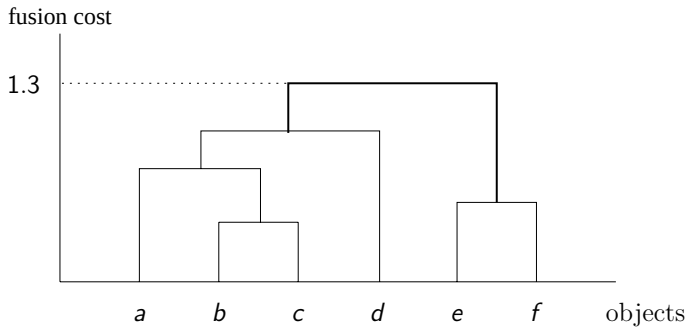
The final structure of nested clusters and the dendrogram encoding the clustering procedure are the following



PROXIMITY MATRIX

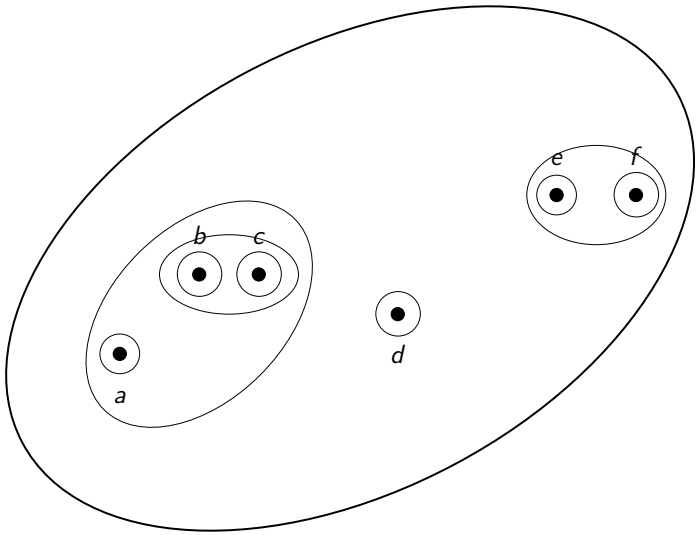
EMPTY

DENDROGRAM



step - 6 (final step)

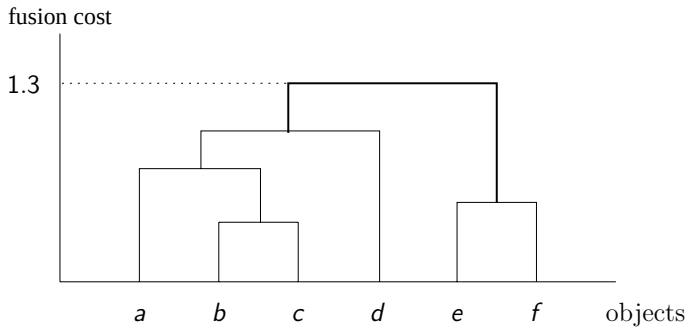
The final structure of nested clusters and the dendrogram encoding the clustering procedure are the following



PROXIMITY MATRIX

EMPTY

DENDROGRAM



The R function hclust

It performs hierarchical agglomerative clustering using several aggregation criterion methods and it admits an arbitrary dissimilarity matrix as input

input: a *dissimilarity matrix d* and the clustering *method* among the options, "ward", "single", "complete" (default), "average", "mcquitty", "median" or "centroid".

value: the function returns an object of the class *hclust*, which consists of a list including, among others, the following elements:
merge - a $(n-1) \times 2$ matrix indicating the clusters being merged
heigth - the list of fusion costs

R (hclust function)

`hc<-hclust(d, method=‘‘complete’’, members=NULL)`
`plot(hc)` *or* `plot(hc, hang=-1)` *to plot the dendrogram with all leaves at the same height*

Example

R (single-linkage example with output)

```
X<-matrix(c(0,0,0.5,0.5,0.85,0.5,1.75,0.25,2.75,1,3.25,1),
nrow=6,byrow=TRUE) # the set of 6 points {a, b, c, d, e, f} in two variables
     [,1] [,2]
[1,] 0.00 0.00 point "a"
[2,] 0.50 0.50 point "b"
[3,] 0.85 0.50 point "c"
[4,] 1.75 0.25 point "d"
[5,] 2.75 1.00 point "e"
[6,] 3.25 1.00 point "f"
d<-dist(X) # by default uses the euclidean distance
SL<-hclust(d, method="single")
SL$height
[1] 0.375 0.5 0.707 0.91 1.25
SL$merge
[,1] [,2]
[1,] -2 -3 (merges singletons {b} and {c})
[2,] -5 -6 (merges singletons {e} and {f})
[3,] -1 1 (merges singleton {a} with cluster {b, c})
[4,] -4 3 (merges singleton {d} with cluster {a, b, c})
[5,] 2 4 (merges cluster {e, f} with cluster {a, b, c, d})
# The number with minus sign refers to a singleton ID,
# otherwise refers to the step number where the cluster was aggregated

plot(SL, hang=-1) # plot the dendrogram
```

Where to cut the dendrogram?

A cut in a dendrogram at a given height τ produces the (flat) partition into the clusters whose fusion cost is smaller than τ

Usually one seeks cuts in the dendrogram such that:

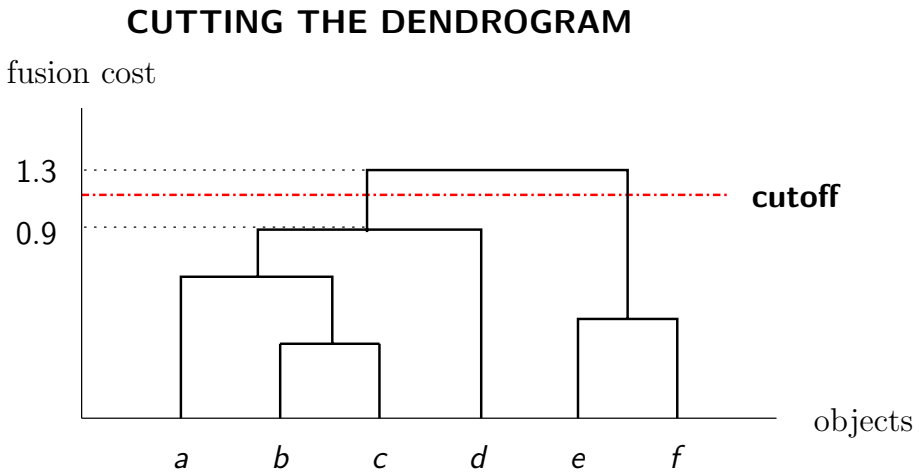
- split high height branches (high lifetimes) to get high inter-cluster heterogeneity
- as close as possible to the leaves to get high intra-class homogeneity

Some caution has to be applied regarding the decision where to cut the dendrogram (and what is the "best" number of clusters). With some methods (for instance, the Ward method), the dendrogram lifetimes tend to increase when the larger clusters are merged, due to the way the fusion costs are defined

Several internal validity indices can be used to estimate the optimal number of clusters

Example

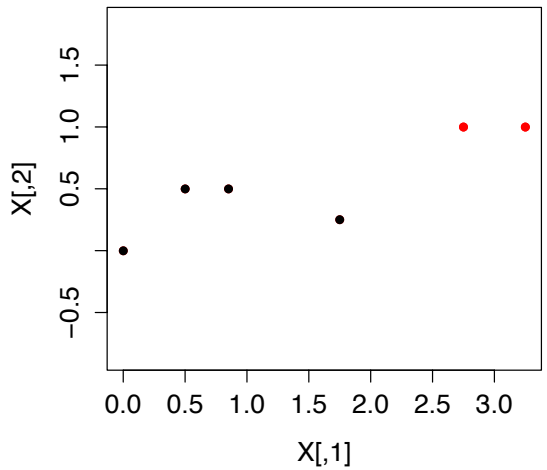
For instance, to obtain a partition into 2 clusters we have to cut the dendrogram at some height in the interval $]0.9, 1.3[$, yielding the clusters $\mathcal{C} = \{a, b, c, d\}$ and $\mathcal{C}' = \{e, f\}$



- The cluster $\{e, f\}$ is relatively well separate from the cluster $\{a, b, c, d\}$ since the fusion cost (1.3) between these groups is relatively high
- But cluster $\{a, b, c, d\}$ is not very homogeneous since the fusion cost (0.9) of aggregating all of its elements is also relatively high

Cutting the dendrogram in R

The resulting partition into two clusters $\{a, b, c, d\}$ and $\{e, f\}$ (depicted using distinct colors)



R (cutree function)

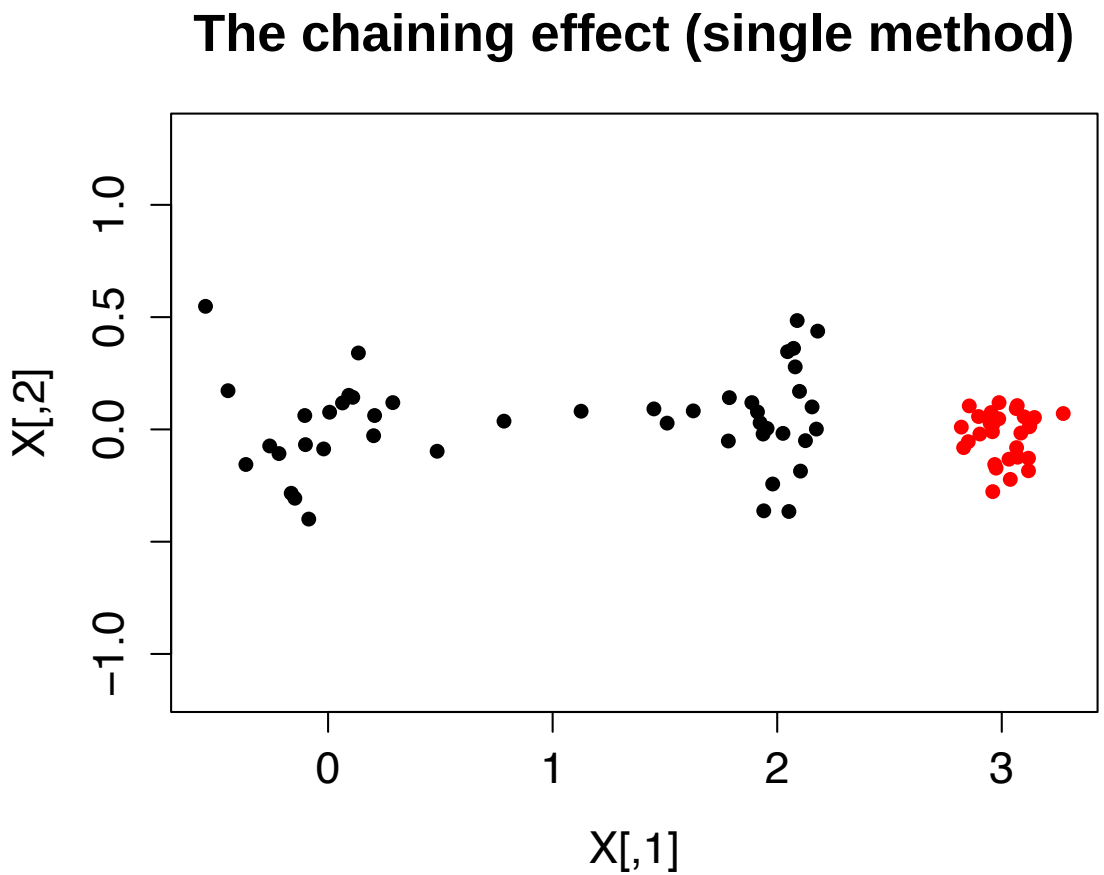
```
SL<-hclust(X,method="single")
part<-cutree(SL,2) # 2 clusters
# # or
part<-cutree(SL,h=1.1) # h is the height
part
plot(X,type="p",cex=0.8,pch=16, col=part,asp=TRUE)
```

Chaining effect

- In single-linkage if two clusters are merged at a fusion cost τ, every pair of objects, one in each cluster, have pairwise distance greater than or equal to τ.
- As the clusters growth it becomes more and more easier to incorporate new elements in the cluster since the distances between these elements and the cluster is the distance to the nearest point in the cluster
- As a consequence, the singletons tend to aggregate to the larger clusters, often producing elongated clusters (chain effect) and/or very unbalanced partitions

Chaining effect

The chaining effect is usually produced by the existence of intermediate points between clusters, giving rise to elongated clusters connecting distant points



Single-linkage emphasizes clusters separation

The nearest neighbor distance can be used to measure of separability between clusters. More precisely, we can measure the separability of a partition $X = C_1 \cup \cdots \cup C_k$ as the distance between the closest pair of clusters for the nearest neighbor criterion, i.e., as

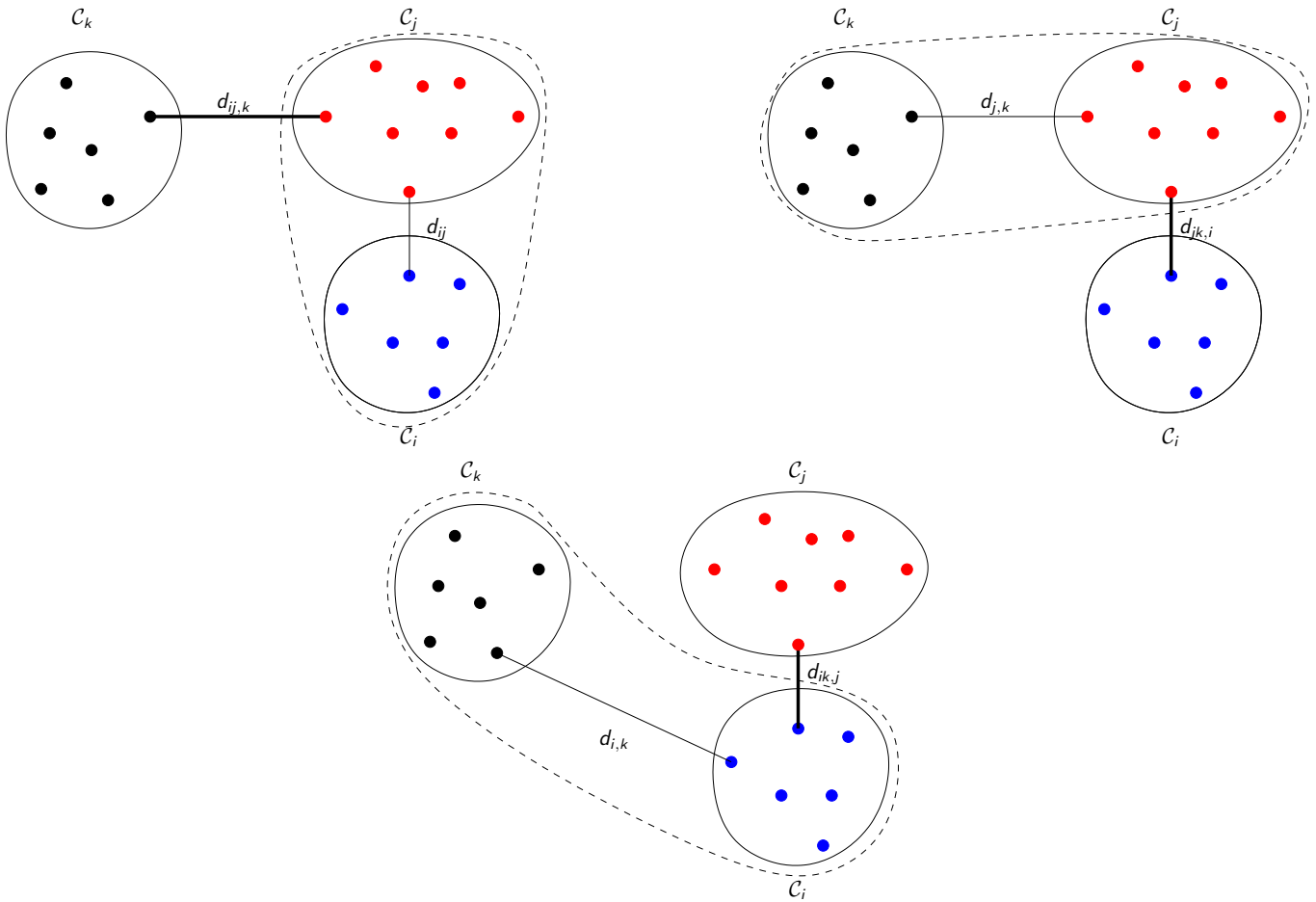
$$\min_{i \neq j} D(C_i, C_j) = \min_{i \neq j} \left(\min_{x \in C_i,\, y \in C_j} d(x, y) \right).$$

In each step the single-linkage algorithm merges the pair of closest clusters, which amounts to say that it merges the pair of clusters that maximizes the separability of the resulting partition.

Therefore we have the following.

The single-linkage clustering algorithm tends to produce well separate partitions but not necessarily homogeneous!

Single-linkage and clusters separation



The aggregation of the pair of closest clusters (top row, on the left) yield the better separated 2-partition among the 3 possible 2-partitions:

$$\{C_{ij}, C_k\}, \qquad \{C_{jk}, C_i\}, \qquad \{C_{ik}, C_j\}$$

Single-linkage clustering - summary

Pros

- Can detect arbitrary cluster shapes
- Can be applied to large datasets since it is computationally efficient, i.e., there are polynomial-time clustering algorithms
- Emphasizes clusters separation, i.e., tends to form well separated clusters
- It is invariant under monotonic transformations of the proximity matrix since it only depends on the rank orders of the pairwise distances between the points of the dataset
- Insensitive to ties in the proximity matrix

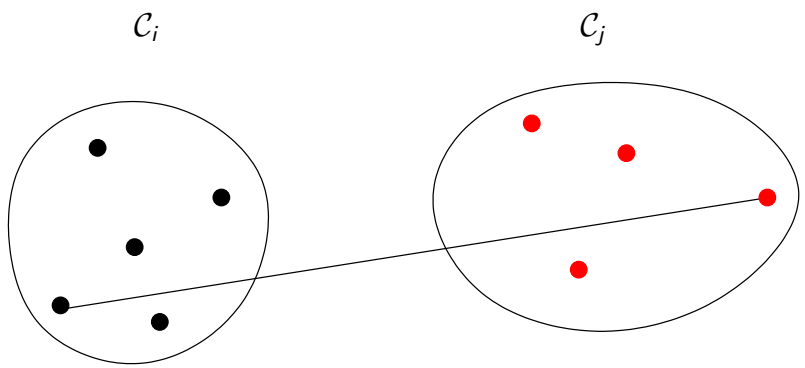
Cons

- Suffers from the chaining effect - often produces elongated clusters with very distinct sizes
- Sensitive to observation errors and noise
- The decision of aggregate two clusters relies only on a pair of elements, one in each cluster

Complete-linkage

The complete-linkage or furthest neighbor is the opposite of nearest-neighbor clustering algorithm The fusion cost between two clusters $\mathcal{C}_i$ and $\mathcal{C}_j$ in this method is defined as the distance between the furthest pair of points, one in each cluster, that is,

$$d_{i,j} = D(\mathcal{C}_i, \mathcal{C}_j) = \max_{x \in \mathcal{C}_i, y \in \mathcal{C}_j} d(x, y)$$



Updating formula for the complete-linkage:

$$\boxed{d_{ij,k} = \max\{d_{i,k}, d_{j,k}\}}$$

Complete-linkage method

- In complete-linkage two clusters are merged at a height τ only if all elements of one cluster are at a distance inferior than or equal to τ with respect to the elements of the other cluster.
- As the cluster growths it becomes more and more difficult to incorporate new elements in a cluster. Therefore the aggregations tend to occur between clusters with few elements.
- The complete method tend to be sensitive to the presence of outliers.

Exercise

- Perform a clustering analysis with the complete-linkage method on the set of points of the real line $X = \{0.2, 3, 4.2, 5, 5.9\}$ and represent the respective dendrogram.

- Cut the dendrogram in order to obtain two clusters. What you conclude?

Complete-linkage emphasizes clusters homogeneity

The **diameter** of a set C is the largest dissimilarity between pairs of elements of C, i.e.,

$$\operatorname{diam}(C) = \max_{x,y \in C} d(x,y)$$

We can measure the cohesion of a partition $X = C_1 \cup \ldots \cup C_k$, as the partition diameter, i.e., as the largest value among the diameters of $C_1, \ldots, C_k$:

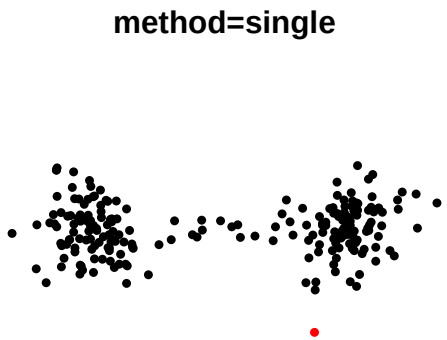
$$\max_i \operatorname{diam}(C_i) = \max_i \left(\max_{x,y \in C_i} d(x,y) \right).$$

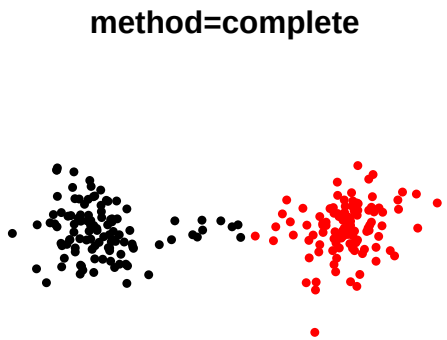
In each step the complete-linkage (also called diameter clustering) method, seeks to aggregate the clusters that produce the smallest increase in the partition diameter, i.e., such that the resulting partition has the smallest possible diameter. Hence we have

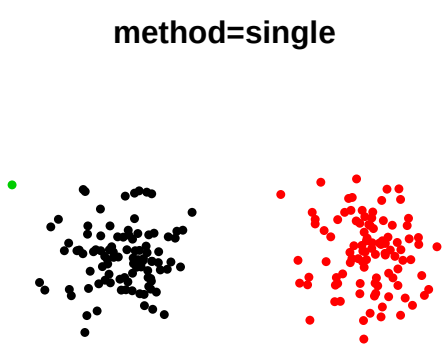
The complete-linkage clustering algorithm tends to produce compact clusters (but not necessarily well separated!)

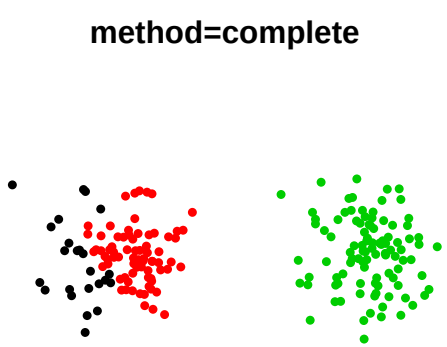
Noise and outliers: single vs complete aggregation methods

The following examples illustrates that the single clustering method is more sensitive to noise than complete, whereas the opposite occurs with outliers (the partitions on the top row have two clusters each and partitions on bottom row 3 clusters)









Complete-linkage clustering - summary

Pros

- Emphasizes cluster compactness - tend to form tight spherical clusters with small diameters, i.e., homogenous clusters
- It is invariant under monotonic transformations of the proximity matrix - only the ranks of the pairwise dissimilarities are important.

Cons

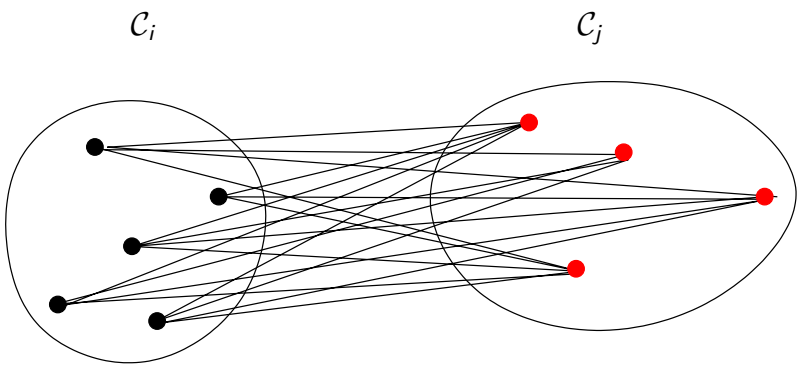
- Sensitive to outliers
- Cannot detect arbitrary cluster shapes
- The decision of aggregate two cluster only relies on a pair of individuals, one in each cluster

Average clustering method

In-between the single-linkage and the complete-linkage clustering methods, we have the average method, also known as *unweighted pair group method average* (UPGMA) The merging cost between two clusters $\mathcal{C}_i$ and $\mathcal{C}_j$ is defined as the arithmetic mean of the distances between every point of $\mathcal{C}_i$ and eevery point of $\mathcal{C}_j$, i.e., equals

$$d_{i,j} = \frac{\sum_{x \in \mathcal{C}_i} \sum_{y \in \mathcal{C}_j} d(x,y)}{n_i \, n_j},$$

where $n_i = |C_i|$ and $n_j = |C_j|$.



The updating formula is given by (left as an exercise),

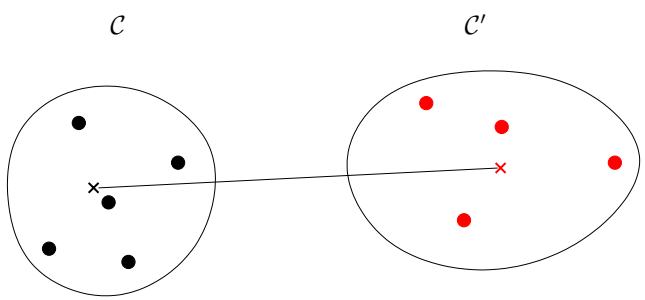
$$\boxed{d_{ij,k} = \frac{n_i d_{i,k} + n_j d_{j,k}}{n_i + n_j}}$$

This method often outperforms single-linkage and complete linkage but it is not invariant under monotonic transformations of the proximity matrix.

Centroid clustering model

This method, also known as UPGMC (unweighted pair group method centroid) implements the very natural idea that the clusters are represented by their centroids and thus define distance $d_{i,j}$ between two clusters $\mathcal{C}_i$ and $\mathcal{C}_j$ as the distance between the respective centroids m_i and m_j:

$$d_{i,j} = \left\| \frac{1}{|\mathcal{C}_i|} \sum_{x_i \in \mathcal{C}_i} x_i - \frac{1}{|\mathcal{C}_j|} \sum_{x_j \in \mathcal{C}_j} x_j \right\| = \|m_i - m_j\|$$



The centroid of the group obtained by merging the clusters C_i and C_j is given by

$$m_{ij} = \frac{n_i m_i + n_j m_j}{n_i + n_j}$$

The updating formula is more complicated in this case. We shall resort to a general procedure to define the updating formula for the centroid method.

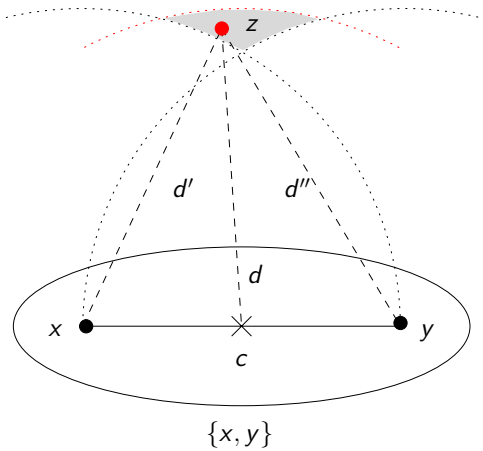
Exercise

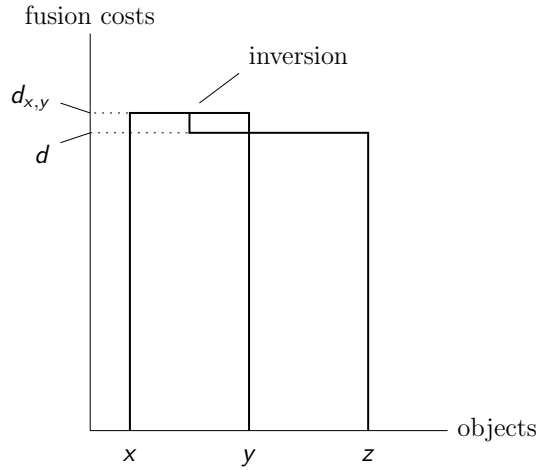
- Perform a clustering analysis using the centroid method on the set of 3 points of $\mathbb{R}^3$, $X = \{(0,0),(8,0),(4,7.5)\}$ and represent the respective dendrogram
- What happened ?

Centroid clustering model - inversions

In the centroid method the merging cost can be non-monotonic, giving rise crossovers (also called inversions) in the dendrogram

All circles have radii equal to the distance between x and y, $d_{x,y}$.





Since z (red point) lie in the grey area, ouside the black circles, $d_{x,y} < d', d''$. Hence x and y are the first pair of objects to be merged. Since z lie inside the red circle centred at the centroid c of x and y,

$$D(\{x, y\}, z) = d_{c,z} < d_{x,y} = D(\{x\}, \{y\})$$

Lance-Williams general updating formula

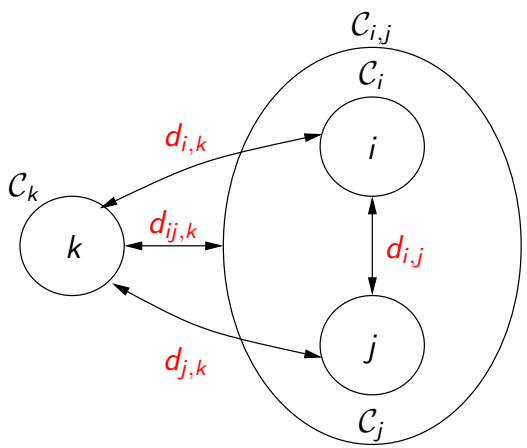
Given clusters $\mathcal{C}_i$, $\mathcal{C}_j$, $\mathcal{C}_k$ and $\mathcal{C}_{ij} = \mathcal{C}_i \cup \mathcal{C}_j$ we will define updating formulas for a family of clustering methods

$$d_{ij,k} = \alpha_i d_{i,k} + \alpha_j d_{j,k} + \beta d_{i,j} + \gamma |d_{i,k} - d_{j,k}|$$

or

$$d^2_{ij,k} = \alpha_i d^2_{i,k} + \alpha_j d^2_{j,k} + \beta d^2_{i,j} + \gamma |d^2_{i,k} - d^2_{j,k}|$$

depending on the method considered, where α_i, α_j, β and γ are convenient parameters that may depend only on the clusters cardinality $n_i = |C_i|$, $n_j = |C_j|$, $n_k = |C_k|$ and $n_i + n_j = |C_{ij}|$:



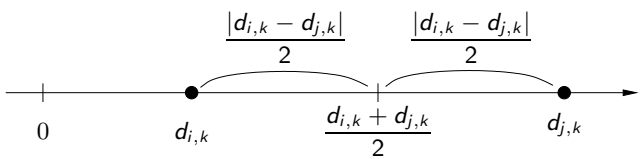
Lance-Williams updating formula - examples

Let us see how to obtain the updating formulas for the single-linkage and complete linkage of slides 39 and 56 (verificar!)

$$\begin{aligned} d_{ij,k} &= \min(d_{i,k}, d_{j,k}) \quad \text{(single-linkage)}, \\ d_{ij,k} &= \max(d_{i,k}, d_{j,k}) \quad \text{(complete-linkage)}, \end{aligned}$$

from the Lance-Williams table. We can assume $d_{i,k} \leq d_{j,k}$. Therefore

$$\begin{aligned} d_{ij,k} &= \min(d_{i,k}, d_{j,k}) = d_{i,k} = \frac{d_{i,k} + d_{j,k}}{2} - \frac{1}{2}|d_{i,k} - d_{j,k}| \\ d_{ij,k} &= \max(d_{i,k}, d_{j,k}) = d_{j,k} = \frac{d_{i,k} + d_{j,k}}{2} + \frac{1}{2}|d_{i,k} - d_{j,k}|, \end{aligned}$$



Hence the Lance-Williams coefficients for the single-linkage and complete-linkage, are:

$$\alpha_i = \alpha_j = \frac{1}{2},\ \gamma = -\frac{1}{2} \text{ and } \beta = 0 \text{ (single-linkage)}$$

$$\alpha_i = \alpha_j = \frac{1}{2},\ \gamma = \frac{1}{2} \text{ and } \beta = 0 \text{ (complete-linkage)}$$

Lance-Williams chart

	α_i	α_j	β	γ	dissimilarity matrix	reversals
single	$\frac{1}{2}$	$\frac{1}{2}$	0	$-\frac{1}{2}$	d_{ij}	NO
complete	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	d_{ij}	NO
average (UPGMA)	$\frac{n_i}{n_i+n_j}$	$\frac{n_j}{n_i+n_j}$	0	0	d_{ij}	NO
McQuitty (WPGMA)	$\frac{1}{2}$	$\frac{1}{2}$	0	0	d_{ij}	NO
centroid (UPGMC)	$\frac{n_i}{n_i+n_j}$	$\frac{n_j}{n_i+n_j}$	$\frac{-n_i n_j}{(n_i+n_j)^2}$	0	d_{ij}^2	can occur
median (WPGMC)	$\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{4}$	0	d_{ij}	can occur
Ward	$\frac{n_i+n_k}{n_i+n_j+n_k}$	$\frac{n_j+n_k}{n_i+n_j+n_k}$	$-\frac{n_k}{n_i+n_j+n_k}$	0	d_{ij}^2	NO

Example: updating formula for the centroid method

Using the previous Lance-Williams table we obtain the following updating formula for the centroid method:

$$\boxed{d^2_{ij,k} = \frac{n_i}{n_i+n_j} d^2_{i,k} + \frac{n_j}{n_j+n_j} d^2_{j,k} - \frac{n_i n_j}{(n_i+n_j)^2} d^2_{i,j}}$$

Note that the distances are squared!

Repeat the clustering performed on the set $\mathbf{X}$ of slide 121 and using the update formula given here

Monotonic condition and inversions

We say that a clustering method satisfies the monotonic condition if whenever two clusters $\mathcal{C}_i$ and $\mathcal{C}_j$ are merged into a cluster $\mathcal{C}_{ij}$ we have

$$d_{ij,k} \geq d_{i,j} \qquad \forall k \neq i, j, ij$$

This implies that the dendrogram cannot have inversions

Proposition

If in the Lance-Williams's formula the parameters α_i, α_j are nonnegative, $\alpha_i + \alpha_j + \beta \geq 1$, and either $\gamma \geq 0$ or $\max\{-\alpha_i, -\alpha_j\} \leq \gamma \leq 0$, the clustering method satisfies the monotonic condition (*)

(*) A stronger condition is given by Batagelj : the Lance-Williams clustering algorithm is monotonic if and only if,

$$\gamma \geq -\min(\alpha_1, \alpha_2), \quad \alpha_1 + \alpha_2 \geq 0, \quad \alpha_1 + \alpha_2 + \beta \geq 1$$

From the Lance-Williams table we deduce immediately that the clustering aggregation methods, *single*, *complete*, *average*, *McQuitty* and *Ward* verify the conditions of the proposition above and therefore satisfy the monotonic condition. In particular, their dendrograms cannot have inversions.

Ward's method

Let $\mathbf{X}$ be a dataset with N individuals, $\mathbf{x}^1, \ldots, \mathbf{x}^N$ in p (observed) variables with mean vector $\mathbf{x}^G = (\bar{x}_1, \ldots, \bar{x}_p)$. Given a partition of $\mathbf{X}$ into K clusters

$$\mathbf{X} = \mathcal{C}_1 \cup \cdots \cup \mathcal{C}_K$$

we define,

- $SSQ_t = \sum_{i=1}^{N} \|\mathbf{x}^i - \mathbf{x}^G\|^2 = \sum_{i=1}^{N} \sum_{j=1}^{p} (x_{ij} - \bar{x}_j)^2$ (total inertia)
- $SSQ_b = \sum_{k=1}^{K} n_k \|\mathbf{m}_k - \mathbf{x}^G\|^2$ (between-clusters inertia)
- $SSQ_w = \sum_{k=1}^{K} \sum_{\mathbf{x} \in \mathcal{C}_k} \|\mathbf{x} - \mathbf{m}_k\|^2$ (total within-clusters inertia),

where $\mathbf{m}_k$ is the centroid of cluster $\mathcal{C}_k$ and n_k the number of its elements

Ward's method

- The between-clusters inertia SSQ_b represents the inertia of the dataset assuming that each cluster $\mathcal{C}_k$ is represented by n_k copies of the cluster centroid $\mathbf{m}_k$.
- The total within-clusters inertia SSQ_w represents the information that is lost by replacing the n_k elements of each cluster $\mathcal{C}_k$ by n_k copies of the cluster centroid.
- By Huygens theorem, $SSQ_t = SSQ_b + SSQ_w$, which is a constant.
- Ward's clustering method, also called minumum variance criterion, tries to minimize the total within-clusters inertia SSQ_w, i.e., the clusters heterogeneity/variability, which, by Huygens theorem, amounts to maximize the between-clusters inertia SSQ_b, i.e., the clusters separation
- Hence Ward's method seeks to simultaneously optimize two criteria: maximize the clusters separation and minimize the clusters variability

Increase in the sum of within-cluster inertia

- At beginning all clusters have a unique element and therefore,

$$SSQ_t = SSQ_b, \qquad SSQ_w = 0$$

- At each step, Ward's method merges the pair of clusters $\mathcal{C}_i$, $\mathcal{C}_j$ yielding the smallest increase in the total within-cluster inertia SSQ_w
- We shall write SSQ_w as

$$SSQ_w = \sum_{k=1}^{K} \mathbf{e}_k^2,$$

where $\mathbf{e}_k^2$ is the inertia of cluster k in, i.e.,

$$\mathbf{e}_k^2 = \sum_{\mathbf{x} \in \mathcal{C}_k} \|\mathbf{x} - \mathbf{m}_k\|^2 = \frac{\sum_{\mathbf{x},\mathbf{y} \in \mathcal{C}_k} \|\mathbf{x} - \mathbf{y}\|^2}{2n_k}$$

(note that the later expression only depends on the pairwise distances between elements of C_k).

Increase in the sum of within-cluster inertia

- When two clusters $\mathcal{C}_i$ and $\mathcal{C}_j$ are merged into a cluster $\mathcal{C}_{ij}$, the increase in the total within-cluster inertia SSQ_w reduces to the following statistic,

$$\Delta_{ij} SSQ_w = \mathbf{e}_{ij}^2 - \mathbf{e}_i^2 - \mathbf{e}_j^2,$$

  since all other within-group inertias are not affected. After $N-1$ aggregation steps (assuming $|X| = N$) the sum of the successive increases $\Delta_{ij,k}$ is equal to the total inertia SSQ_t.

- It can be proved that

$$\Delta_{ij} SSQ_w = \frac{n_i n_j}{n_i + n_j} \|\mathbf{m}_i - \mathbf{m}_j\|^2,$$

  which represents a weighted distance between the cluster centroids (cf. with centroid method).

- In particular, $\Delta_{ij} SSQ_w$ is always nonnegative (i.e., the SSQ_w is increasing) and only depends on the squared distance between the cluster centroids $\mathbf{m}_i$ and $\mathbf{m}_j$ and on the cluster sizes n_i and n_j.

A better updating formula for Ward's method using LW

- The fusion cost between the clusters $\mathcal{C}_{ij} = \mathcal{C}_i \cup \mathcal{C}_j$ and $\mathcal{C}_k$ is

$$\Delta_{ij,k} SSQ_w = \frac{(n_i + n_j)n_k}{n_i + n_j + n_k} \|m_{ij} - m_k\|^2,$$

  which can be used as an updating formula for Ward's clustering method but has the disadvantage that it requires the knowledge of the original dataset to compute the centroids.

- Using the Lance-Williams table we can derive an alternative updating formula for Ward's method that only requires the (squared) proximity matrix at previou step:

$$\boxed{d^2_{ij,k} = \frac{(n_i + n_k)d^2_{i,k} + (n_j + n_k)d^2_{j,k} - n_k d^2_{i,j}}{n_i + n_j + n_k}}$$

- The above expression actually returns twice the value of $\Delta_{ij,k} SSQ_w$ and corresponds to the square of the dendrogram height computed with R function `hclust` and the ward.D2 method.

Example

Consider the univariate dataset $X = \{a, b, c, d\} = \{1, 2, 4, 8\}$
The pairwise distances and squared pairwise distances between elements of X are given, respectively, by

$$\left[\begin{array}{c|ccc} D & a & b & c \\ \hline b & 1 & & \\ c & 3 & 2 & \\ d & 7 & 6 & 4 \end{array}\right] \quad \text{and} \quad \left[\begin{array}{c|ccc} D^2 & a & b & c \\ \hline b & 1 & & \\ c & 9 & 4 & \\ d & 49 & 36 & 16 \end{array}\right]$$

The minimum of the squared distances is attained for $D^2(a, b)$ so the first pair to be clustered will be $a \cup b$ with squared fusion cost equal to 1

Example (cont.)

$$D^2(a \cup b, c) = \frac{2\,D^2(a,c) + 2\,D^2(b,c) - D^2(a,b)}{3} = \frac{2 \cdot 9 + 2 \cdot 4 - 1}{3} = \frac{25}{3}$$

and

$$D^2(a \cup b, d) = \frac{2\,D^2(a,d) + 2\,D^2(b,d) - D^2(a,b)}{3} = \frac{2 \cdot 49 + 2 \cdot 36 - 1}{3} = \frac{169}{3}$$

$D^2(c,d)$ is not affected. Thus the new squared dissimilarity matrix is

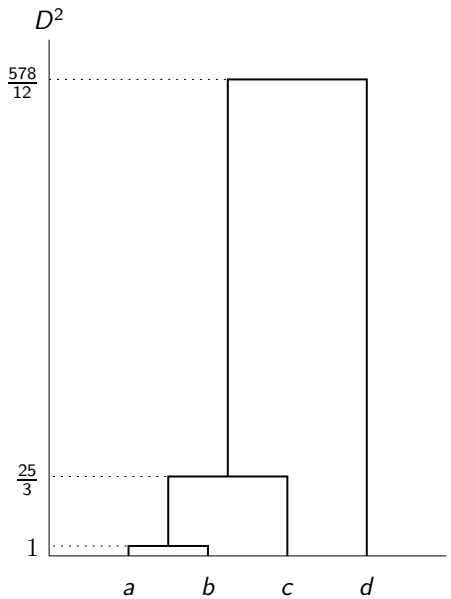
D^2	$a \cup b$	c
c	$\frac{25}{3}$	
d	$\frac{169}{3}$	16

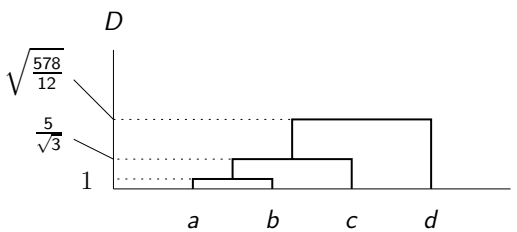
The minimum of the squared distances is attained for $D^2(a \cup b, c)$ so the next pair to be clustered will be $(a \cup b) \cup c$ with squared fusion cost $\frac{25}{3}$

Ward clustering using LW updating formula (concl.)

$$D^2((a \cup b) \cup c, d) = \frac{3\, D^2(a \cup b, d) + 2\, D^2(c, d) - D^2(a \cup b, c)}{4}$$
$$= \frac{3 \cdot \frac{169}{3} + 2 \cdot 16 - \frac{25}{3}}{4} = \frac{578}{12}$$

The dendrogram can be presented either using squared or not squared fusion costs. Its topology however does not change





Computing using the R function *hclust*

The previous dendrogram can also be computed using the R software in the following way:

R (Ward's method)

```
X<-c(1,2,4,8)
N<-length(X)
d<-dist(X) # (euclidean) distance matrix
h.ward<-hclust(d,method="ward.D2")
h.ward$height
sum(h.ward$height**2)/2
SSQt=var(X)*(N-1)
plot(h.ward, hang=-1)
```

Ward's clustering method - summary

Pros

- Tend to form hyperspherical shape clusters, with approximately the same number of elements each (balanced)
- No crossovers
- It is regarded by some authors as a natural hierarchical method to be used with the factorial analysis, such as, PCA, MCA (multiple correspondence analysis), etc, since it seeks to optimize the same variance criterion
- The sum of all dendrogram heights is equal to $2 \times SSQ_t$.

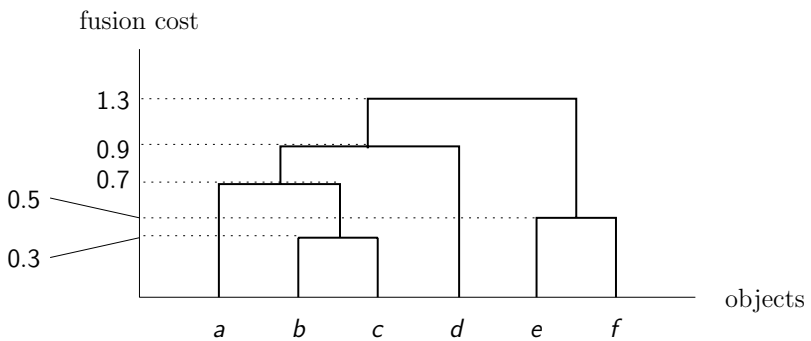
Cons

- Computationally intensive
- Cannot detect arbitrary cluster shapes
- Sensitive to outliers since it uses centroids

Cophenetic distances

The cophenetic distance between two individuals x and y with respect to a given HAC is the merging cost at which x and y become members of the same cluster, during the course of the hierarchical clustering.

Any dendrogram can be represented by its matrix of cophenetic distances up to permutation of the order of the leaves. This matrix can be used to compare distinct classifications



$$\begin{array}{c|ccccc} & a & b & c & d & e \\ \hline b & 0.7 & \cdot & \cdot & \cdot & \cdot \\ c & 0.7 & 0.3 & \cdot & \cdot & \cdot \\ d & 0.9 & 0.9 & 0.9 & \cdot & \cdot \\ e & 1.3 & 1.3 & 1.3 & 1.3 & \cdot \\ f & 1.3 & 1.3 & 1.3 & 1.3 & 0.5 \end{array}$$

Two elements x, y belong to the same cluster of a partition obtained cutting the dendrogram at height τ if and only if their cophenetic distance is less than τ

Distortion measures - Cophenetic Pearson's Coefficient

The **cophenetic Pearson's correlation coefficient** (CPCC) is Pearson's correlation between the original distances (d_{ij}), $i < j$, and the cophenetic distances (c_{ij}), $i < j$, (using half of the proximity matrix), i.e.,

$$CPCC = \frac{\text{cov}(D, C)}{s_D s_C} = \frac{\sum_{i<j}(d_{ij} - \bar{d})(c_{ij} - \bar{c})}{\sqrt{\sum_{i<j}(d_{ij} - \bar{d})^2}\sqrt{\sum_{i<j}(c_{ij} - \bar{c})^2}}$$

- CPCC is considered an internal validation criterion for hierarchical clustering that can be used to evaluate and compare different hierarchical clustering methods, although should be used with caution
- A high value of the CPCC means that the cophenetic distances are a good portray of the original distances
- The cophenetic correlation usually ranges between 0.6 and 0.95.
- Cophenetic correlations between 0.7 and 0.8 are considered reasonable good, between 0.8 and 0.9 good and above 0.9 very good.

Distortion measures - Cophenetic Spearman's Coefficient

Another distortion measure is the **cophenetic Spearman's rank order correlation coefficient** (CSCC), which only depends on the ranks of the variables and corresponds to Pearson's correlation coefficient between the respective ranked variables $rk(C) = (c'_{ij})$ and $rk(D) = (d'_{ij})$ defined by the vectors of original and cophenetic distances,

$$CSCC = \frac{\text{cov}(rk(D), rk(C))}{s_{rk(D)} s_{rk(C)}} = \frac{\sum_{i<j}(d'_{ij} - \bar{d})(c'_{ij} - \bar{c'})}{\sqrt{\sum_{i<j}(d'_{ij} - \bar{d'})^2}\sqrt{\sum_{i<j}(c'_{ij} - \bar{c'})^2}}.$$

- Unlike the Pearson correlation coefficient, Spearman's rank order correlation coefficient can be applied to compare original and cophenetic dissimilarities even if no linear relation between both dissimilarities exists
- A Spearman's rank order correlation close to 1 means that we have a strong correlation between the ranks of original and the ranks of the cophenetic distances, suggesting monotonic relationship between the original distances and the corresponding cophenetic distances

Cophenetic correlations of example of slide 97

The original d_{ij} distances of the example of slide 97 and the corresponding cophenetic distances c_{ij} for the single, complete and avarage methods are

d_{ij}	a	b	c	d	e
b	0.7	·	·	·	·
c	1	0.3	·	·	·
d	1.8	1.3	0.9	·	·
e	2.9	2.4	1.9	1.3	·
f	3.4	2.8	2.4	1.7	5

c_{ij}^s	a	b	c	d	e
b	0.7	·	·	·	·
c	0.7	0.3	·	·	·
d	0.9	0.9	0.9	·	·
e	1.3	1.3	1.3	1.3	·
f	1.3	1.3	1.3	1.3	0.5

Computing the cophenetic Pearson and Spearman correlation coefficients we obtain,

$$CPCC = r(d_{ij}, c_{ij}) = 0.82, \qquad CSCC = r(rk(d_{ij}), rk(c_{ij})) = 0.84$$

HAC - summary

Pros

- The number of clusters does not need to be defined *a priori*
- Many methods rely on a proximity matrix allowing almost any kind of resemblance notion

Cons

- The aggregation of a point in a group at a given step cannot be revised, even if the point is misplaced in that group
- Computationally demanding for large datasets since keeps track of a square matrix of order n (number of individuals): the time and space complexity of most algorithms are not better than $O(n^2 \log(n))$
- Dendrogram difficult to visualize and interpret for large datasets
- Most HAC algorithms are greedy and produce suboptimal solutions

The average and Ward methods are often considered among the best overall HAC methods

Nonhierarchical clustering

To find a single partition into K clusters of a set of N objects in a p dimensional space. Two types of criteria are commonly found:

- Global criterion such as to represent each cluster by a *type-object* (e.g., centroid, medoid) and to assign each object to the nearest *type-object*, optimizing some global criterion of internal homogeneity and/or external heterogeneity, such as, minimizing the within cluster inertia

  Usually requires a prior estimate of the number of clusters

  Examples: k-means and k-medoids (PAM) algorithms

- Local criterion such as to seek for regions of higher density in data. May require to set some parameters

  Example: **DBSCAN**

k-means

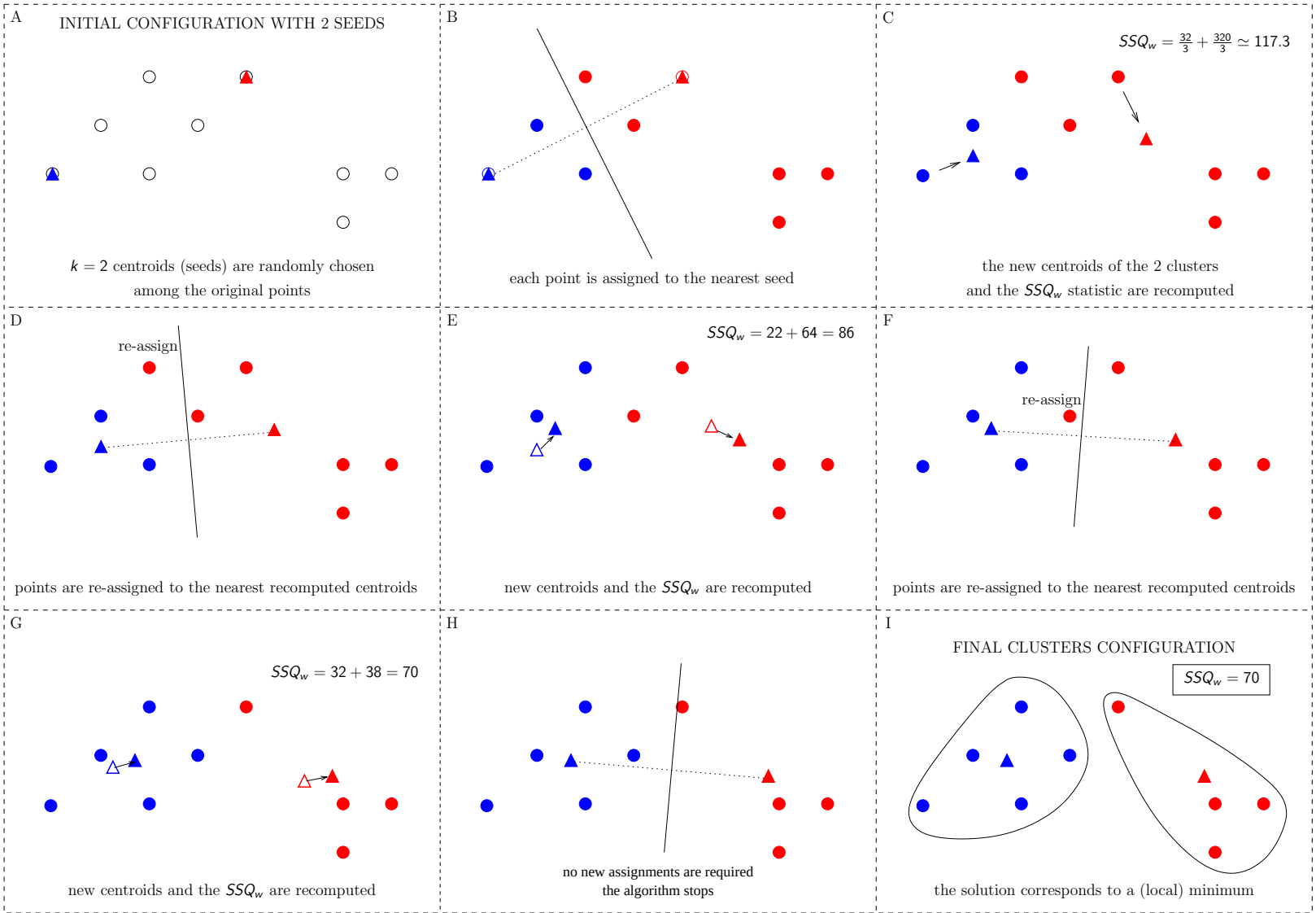
Shares the same global criterion with Ward's method:

To minimize the total within-clusters sum of squares (SSQ_w) of a set of points partitioned into K clusters in a d-dimensional space

Algorithm (Lloyd)

1. *Starts with K randomly chosen initial seeds representing initial candidates to centroids;*
2. *Assigns each object to the nearest centroid*
3. *Recomputes the centroids of the K groups and use them as the new seeds*
4. *Repeat the steps 2 and 3 until no new reassignments occur (in pratice, until the differences between the old seeds and the new recomputed seeds are below a given tolerance threshold)*

k-means algorithm



Convergence of the k-means algorithm

The k-means algorithm consists essentially of a sequence of two steps that are repeatedly iterated:

- Reassignment of the points of X to the closest centroid - this step clearly lowers the statistic $SSQ_w = \sum_{k=1}^{K} \sum_{x \in \mathcal{C}_k} \|x - m_k\|^2$
- Recalculation of the centroids of the K groups to use as the new seeds - this step also lowers the SSQ_w statistic, since it is a well known fact that the minimum of the quadratic function

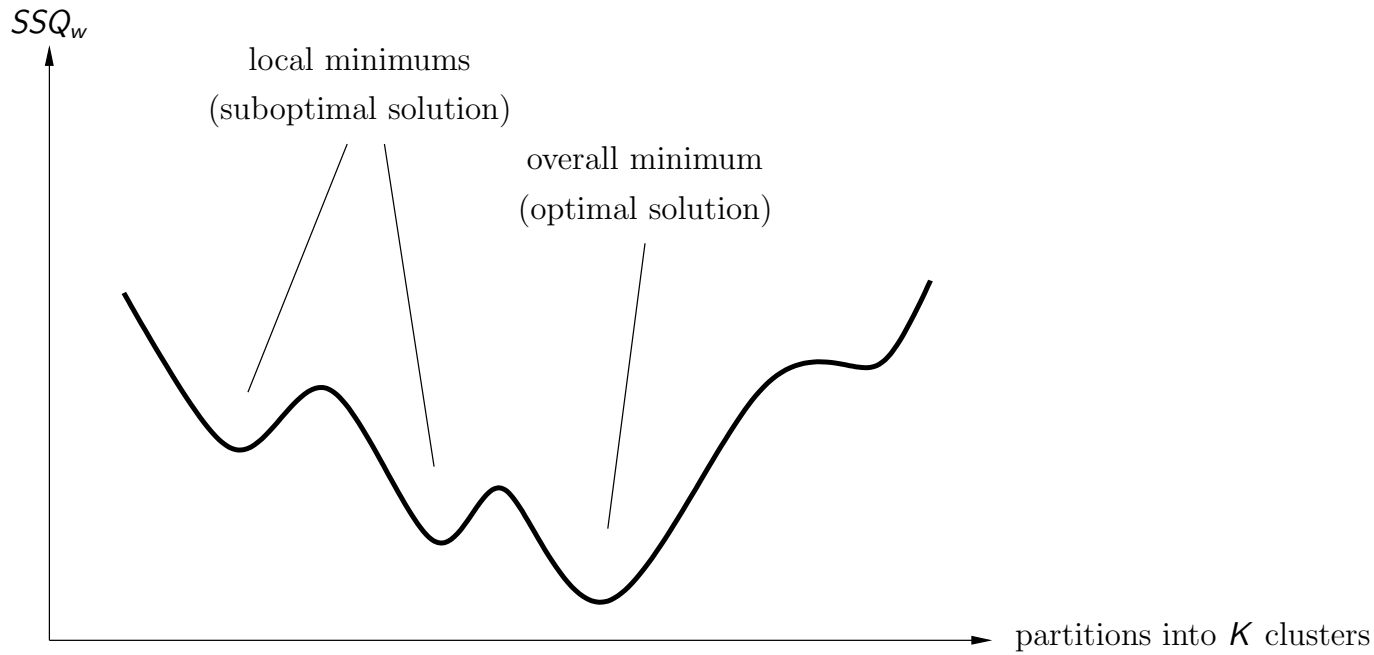
$$f(y) = \sum_{x \in G} \|x - y\|^2,$$

with G a finite subset of $\mathbb{R}^d$, is attained at the centroid of G, i.e., when $y = m_G$

Since there are only finite number of partitions of X into K clusters, the algorithm cannot continue indefinitely strictly lowering the SSQ_w statistic and therefore has to converge to a (possibly local) minimum

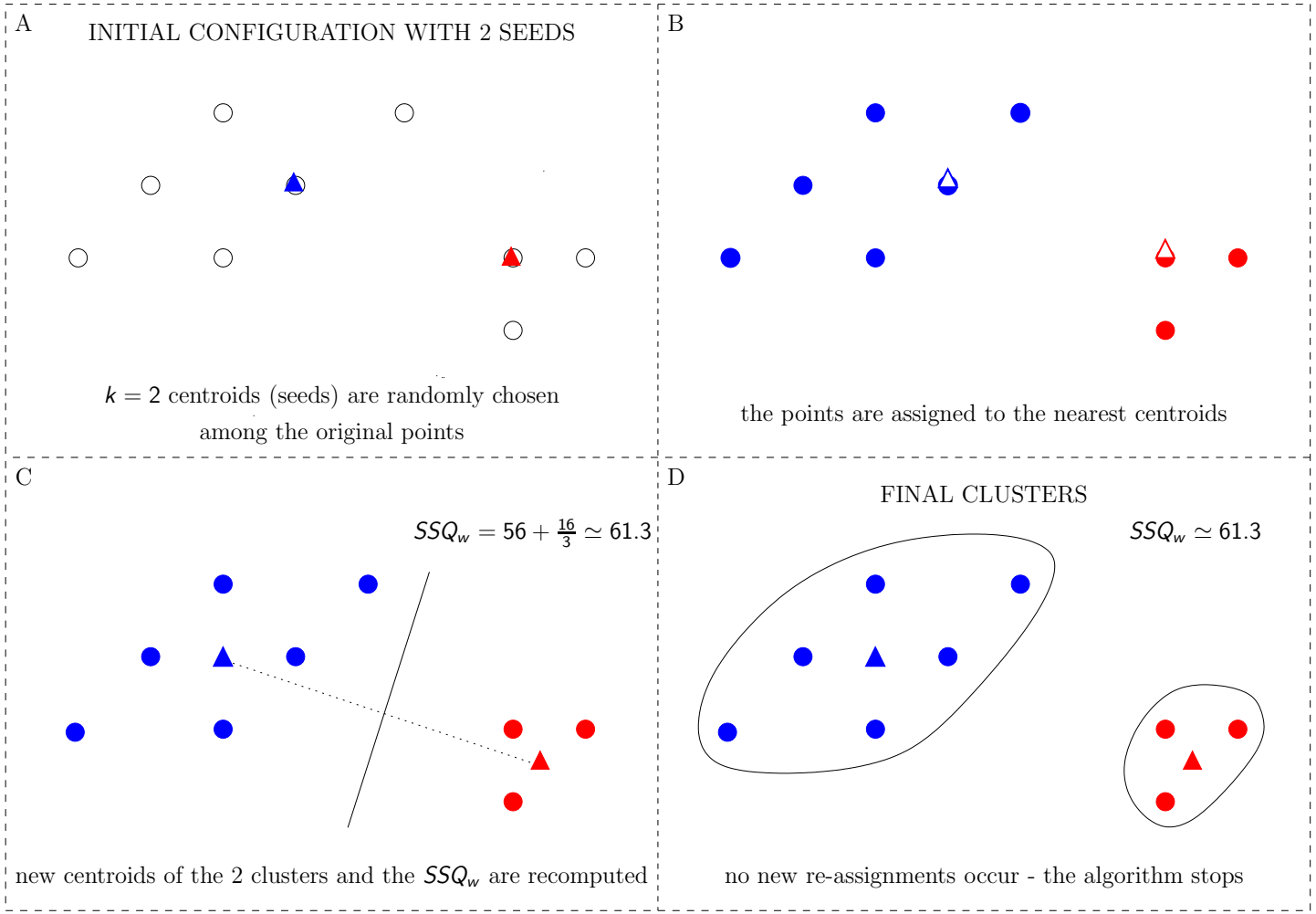
k-means: local minimum problem

The clustering solution can be highly depend on the choice of the initial position of the centroids (seeds) and may converge to a local minimum



Example

The solution found by the k-means algorithm in the previous example is not a global minimum. Actually, with new seeds the algorithm can converge to a solution that improves (i.e., lowers) the SSQ_w statistic



Possible strategies to improve the local minimum?

- To repeat the algorithm several times with randomized sets of K seed points and keep the configuration giving the smallest SSQ_w value of the within-cluster inertia
- To provide an initial configuration of K seed points close to the final solution relying on some real hypothesis
- To provide an initial configuration of seed points issued from some hierarchical aggregation method (e.g., Ward), using for instance, their clusters centroids - this is sometimes called the consolidation of the hierarchical clustering

k-means in the plane and the Voronoi diagram

Given a set of N points in the plane,

$$\{c_1, \ldots, c_K\}$$

the Voronoi diagram is defined as the partition of the plane into K convex regions, called Voronoi cells,

$$R_1, \ldots, R_K$$

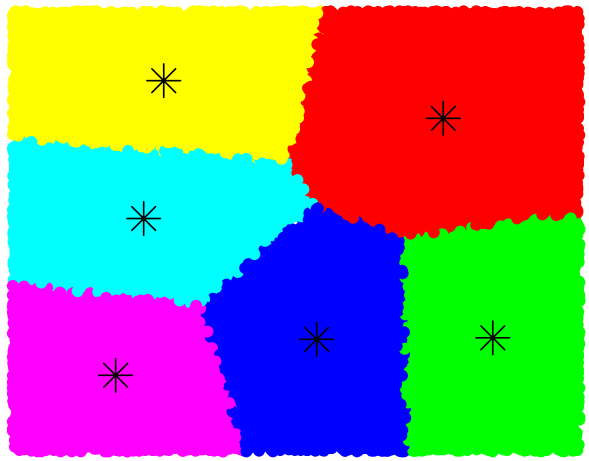
such that each cell R_i consists of the set points of the plane closest to c_i

In each step of the k-means algorithm each cluster corresponds to the set of points of X belonging to one of the Voronoi cells defined be the K centroids $c_1, \ldots, c_K$, which is called Lloyd's algorithm or Voronoi iteration

The above construction can generalized to a set of K points in the N-dimensional space

The Voronoi partition and its centroids

The partition below into 6 clusters was obtained applying the k-means algorithm to a highly dense set of points in the plane with 6 seeds, to give an approximated idea of the Voronoi cells defined by the final centroids



Each cluster arising from a k-means clustering algorithm lies inside the Voronoi cell containing the respective cluster centroid.
In particular, the convex hulls of the clusters don't overlap, i.e., each pair of clusters can be linearly separated.

Computing k-means with R

The k-means clustering can be performed using the R function

kmeans(x, centers, iter.max = 10, nstart = 1, ...)

x: numeric matrix of data

centers: the number of clusters or a set of initial (distinct) cluster centres. If a number, a random set of (distinct) rows in x is chosen as the initial centres

nstart: if centers is a number, how many random sets should be chosen (repeat) Returns a list with components:

cluster: A vector of integers (from 1:k) indicating the number of the cluster where each point is assigned

centers: A matrix of cluster centers.

totss: The total sum of squares, i.e., SSQ_t

withinss: Vector of within-cluster sum of squares, one component per cluster

tot.withinss: Total within-cluster sum of squares, i.e., SSQ_w

betweenss:The between-cluster sum of squares, i.e., SSQ_b

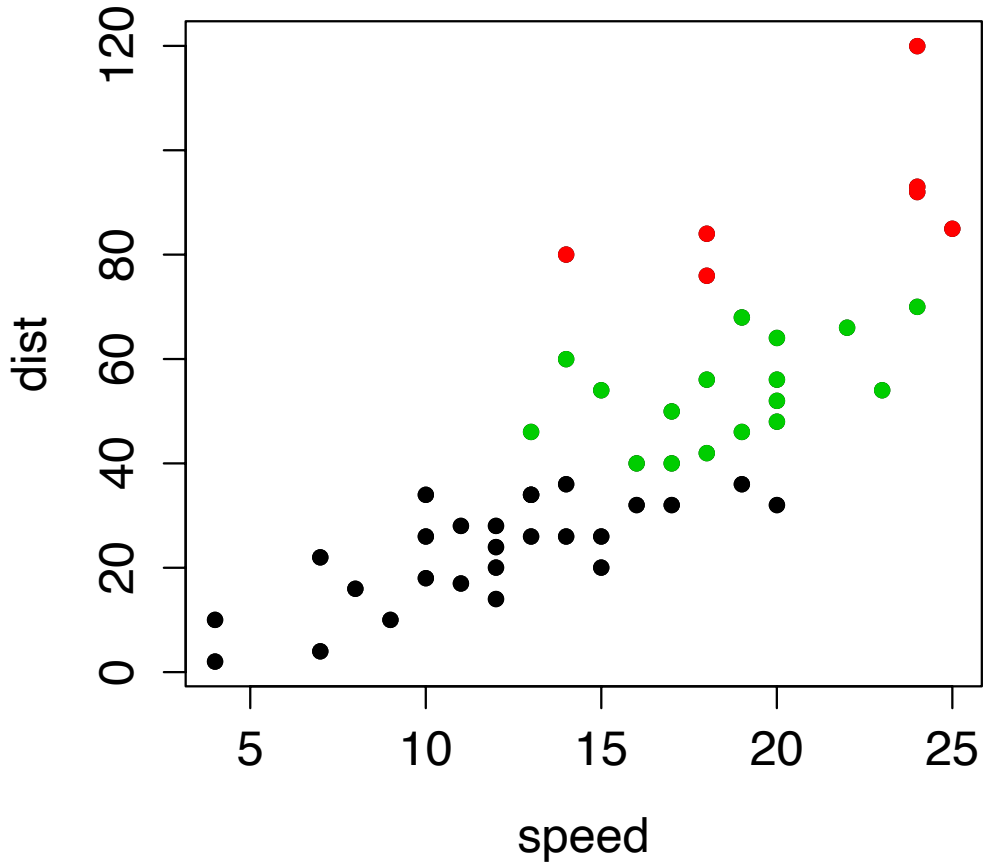
size: The number of points in each cluster

Example

R

```
require(datasets)

data(cars)

?cars

head(cars)

cars.cl<-kmeans(cars, 3, nstart=100)
# 3 centers randomly chosen repeated 100 times

cars.cl

plot(cars,type=‘‘p’’,pch=16,cex=.5)

for(i in 1:50){points(cars[i,1],
cars[i,2],col=cars.cl$cluster[i], pch=16,type=‘‘p’’)}
```

Clustering result



k-means: summary

- The optimizing function SSQ_w is always monotonic decreasing, i.e., the intra-group inertia decreases in each step, converging to some (possibly local) optimum
- The number of iterations required to converge is usually small (≈ 10 iterations are enough)
- Finding an optimal solution is NP-hard. Actually the time complexity is $O(n^{dK+1} \ln d)$, where K denotes the number of clusters, d the dimension and N the number of points)
- It tends to form rounded shaped clusters that can be linearly separated (since each cluster is contained in a Voronoi cell).

  In particular, it cannot detect arbitrarily shaped clusters
- Nearby points can end in distinct classes. Groups can end empty
- Sensitive to noise and outliers
- Requires some geometric notion of centroid. In particular, it cannot be applied to categorical data assumes that the points lie in some euclidean space

The model-based clustering as a generalization of k-means

- The standard model-based clustering is a finite mixture of multivariate Gaussians, i.e., it is assumed that each cluster C_i is generated by a multivariate Gaussian distribution with pdf

$$\phi(x|\mu_i, \Sigma_i)$$

  where μ_i and Σ_i are the mean and covariance matrix of C_i

- One seeks a partition of X into clusters C_i and a mixture of Gaussians with pdf given by a convex combination of the form

$$\phi = \sum_i \eta_i \phi(x|\mu_i, \Sigma_i),$$

  with nonnegative weights η_i, $i = 1, \ldots, K$, such that $\sum_i \eta_i = 1$. To determine the parameters uses the so-called **expectation-maximization** algorithm

- In the model-based clustering the partition can have clusters with different covariance matrices i.e., with distinct ellipsoidal shapes, volumes and orientations, that account with distinct weights to the pdf of the finite mixture

- The k-means clustering can be considered a particular case of the model-based clustering, with all weights η_i equal to $\frac{1}{K}$ and identical isotropic covariance matrices $\Sigma_i = \sigma^2 I$ (I denotes the identity matrix).

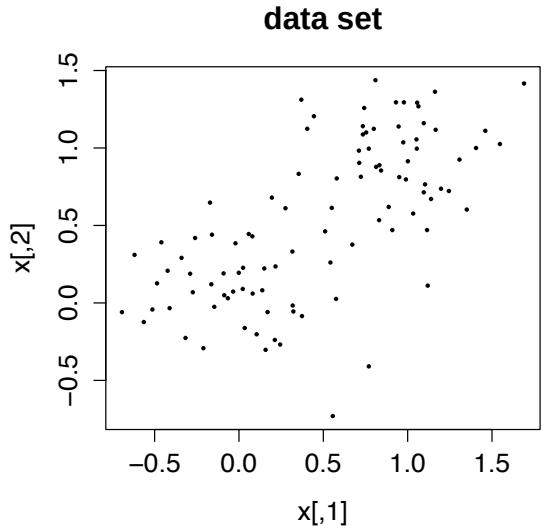
Best number of clusters and internal cluster quality

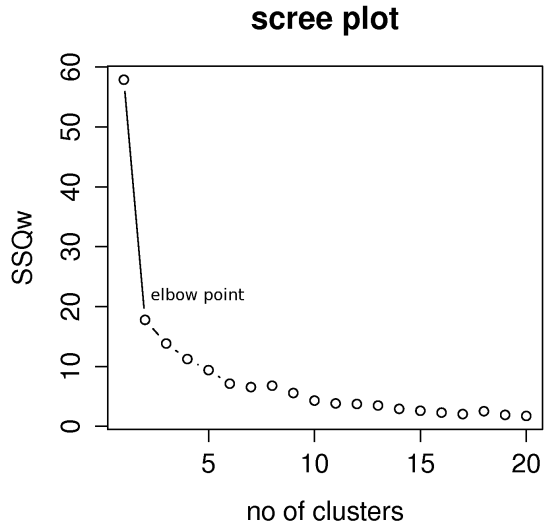
- To estimate the optimal number of clusters we usually look for a good trade-off between a relatively small number of clusters (parsimony principle) and the minimization of the information (variability) loss due to replacing the observations in each cluster by some cluster representative (for instance, the cluster centroid).
- This is one of the most difficult tasks in clustering analysis and no definitive answer can usually be given.
- Several internal cluster validity indices can be used to estimate the optimal number of clusters and/or to assess the cluster quality. Among the most well-known indices we have:
  - SSQ_w.
  - Calinski-Harabasz index.
  - Silhouette coefficient.
  - Davies-Boudin.
  - Duhn index.
  - Several other indices can be computed with the R functions `clustCrit` and `NbClust`.

For a more detailed account on validity indices, See, for instance, O. Arbelaitz et al. *An extensive comparative study of cluster validity indices*, Pattern Recognition 46 (2013) 243–256

Scree plot of SSQ_w statistic

- A simple method to estimate the best number of clusters consists to study the variation of SSQ_w with number of clusters in a scree plot, which essentially amounts, by Huygens's theorem, to study the variation of the percentage of total inertia retained by the clusters, i.e., explained by the partition, $\frac{SSQ_b}{SSQ_t}$
- An elbow point in the scree plot indicating high decrease in the SSQ_w statistic while further increments in the number of clusters will only marginally improves this statistic, could suggest a good estimate for the optimal number of clusters





- Although the statistic SSQ_w depends on the number of clusters, it can be used to compare partitions of a given dataset X with the same number of clusters. Partitions yielding smaller SSQ_w values are preferable for this criterion.

Calinski-Harabaz index

- The Calinski-Harabaz index also known as variance ratio criterion is defined as

$$CH(K) = \frac{SSQ_b/(K-1)}{SSQ_w/(N-K)}$$

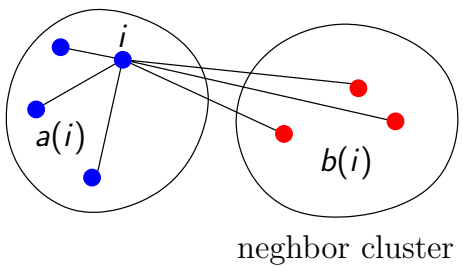
  with the optimal number of clusters being estimated as the number yielding the largest value for $CH(K)$. (Inspired in the F-ratio test of one-way ANOVA)
- Since we have

$$\begin{aligned} CH(K) &= \frac{SSQ_b/(K-1)}{SSQ_w/(N-K)} = \frac{N-K}{K-1} \times \frac{SSQ_b}{SSQ_w} \\ &= \frac{N-1+1-K}{K-1} \times \frac{SSQ_b}{SSQ_w} = \left(\frac{N-1}{K-1} - 1\right) \frac{SSQ_b}{SSQ_w}, \end{aligned}$$

  high values of $CH(K)$ are obtained with well separated and homogeneous clusters, i.e., with large values of SSQ_b and small values of SSQ_w, keeping at the same time, the number of clusters K relatively small, i.e., $\frac{N-1}{K-1}$ relatively large.
- Particularly well adapted when clusters tend to have spherical shapes due to its definition based on the variance
- Several studies suggest Calinski-Harabaz index as being one of the internal cluster validity indices yielding the best results - see, for instance one of the reference papers on internal cluster validation,
  Milligan GW, Cooper MC (1985) An Examination of Procedures for Determining the Number of Clusters in a Data Set. Psychometrika 50:159–179.
- Can be computed using the R function `calinhara` of the package `fpc`

Silhouette coefficient

- For each observation i we compute the average dissimilarity $a(i)$ between i and the remaining points in its cluster
- For each one of the other clusters we compute the average dissimilarity from point i to the points of that cluster and take the minimum $b(i)$ of these average dissimilarities
- The cluster for which the minimum $b(i)$ is attained, i.e., the cluster with lowest average dissimilarity w.r.t to observation i, is called the **neighbor cluster** of i



neghbor cluster

The silhouette coefficient of observation i is defined as

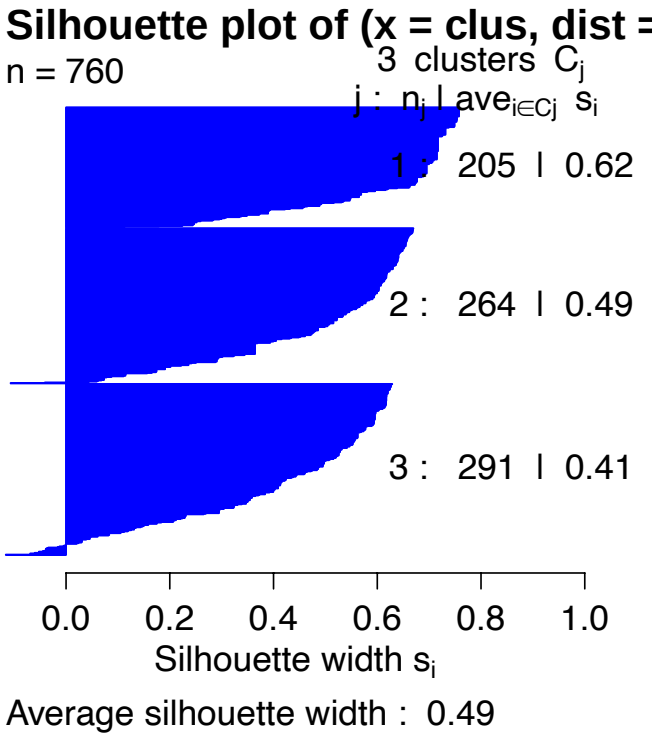
$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

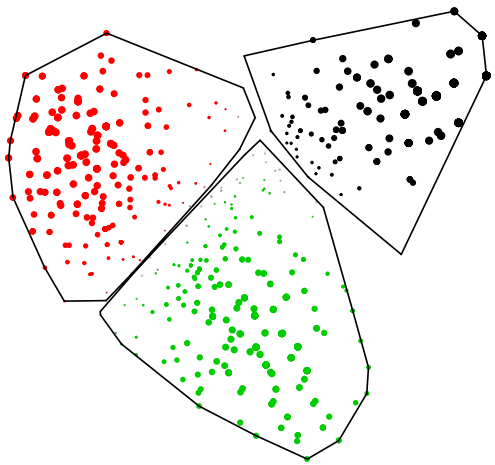
and gives an indication of how well an element is classified in its cluster

Interpretation of silhouette coefficients

- The denominator $\max\{a(i), b(i)\}$ is a normalization term allowing that the index vary in the range $[-1, 1]$
- Small values of $a(i)$ along with large values of $b(i)$ yield a silhouette coefficient close to one
- Likewise, large values of $a(i)$ along with small values of $b(i)$ yield a silhouette coefficient close to minus one
- Observations with silhouette coefficients close to one are very well classified
- Observations with silhouette coefficients close to zero probably lie between clusters
- Observations with negative silhouette coefficients are probably misplaced in their clusters

Silhouette plot





In the figure on the right the dot sizes are proportional to their silhouette coefficients. Larger dots lie in core regions of the clusters whereas smaller dots lie in border regions or between clusters

Average silhouette width - an internal validity criterion

The average silhouette width (ASW) is defined as the average of the silhouette coefficients for all observations

- It assess both cluster cohesion and cluster separation
- It increases with a strong cluster separation (higher $b(i)$ values) and cluster tightness (small values of $a(i)$)

Range of ASW

It is common to consider that

- between 0.71 and 1.0: a strong structure has been found
- between 0.5 and 0.7: a reasonable structure has been found
- between 0.26 and 0.5: the structure is weak and can be artificial
- below 0.25: no substantial structure has been found

The optimal number of clusters can be estimated maximizing the ASW

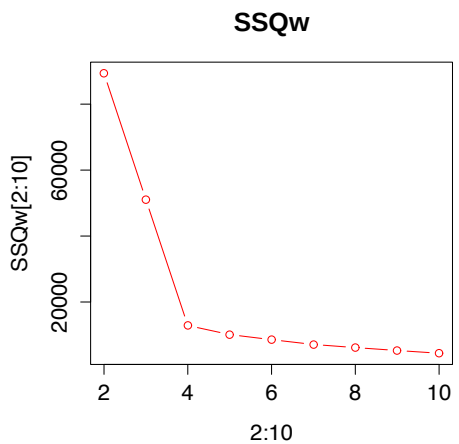
A closely related internal validation criterion is Davies-Bouldin index

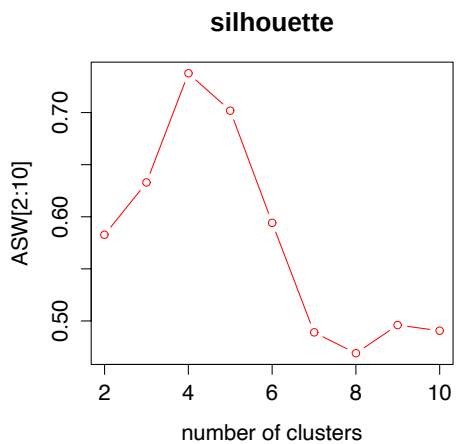
$$DB = \frac{1}{K} \sum_{i=1}^{K} \max_{j \neq i} \frac{S_i + S_j}{m_{ij}}$$

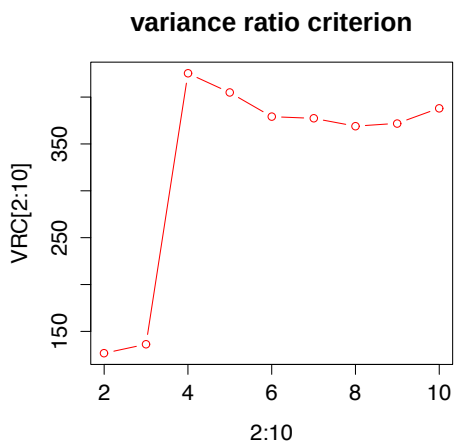
Here S_i denotes some internal cohesion measure of cluster C_i and m_{ij} a separation measure between clusters C_i and C_j, verifying certain properties...
For instance, S_i can be the average distance of the points of C_i to its centroid and m_{ij} the distance between the centroids of C_i and C_j

Number of clusters?

Applying the criteria SSQ_W statistic, ASW and CH to the Ruspini data, a popular dataset in clustering analysis, all criteria agree on 4 clusters

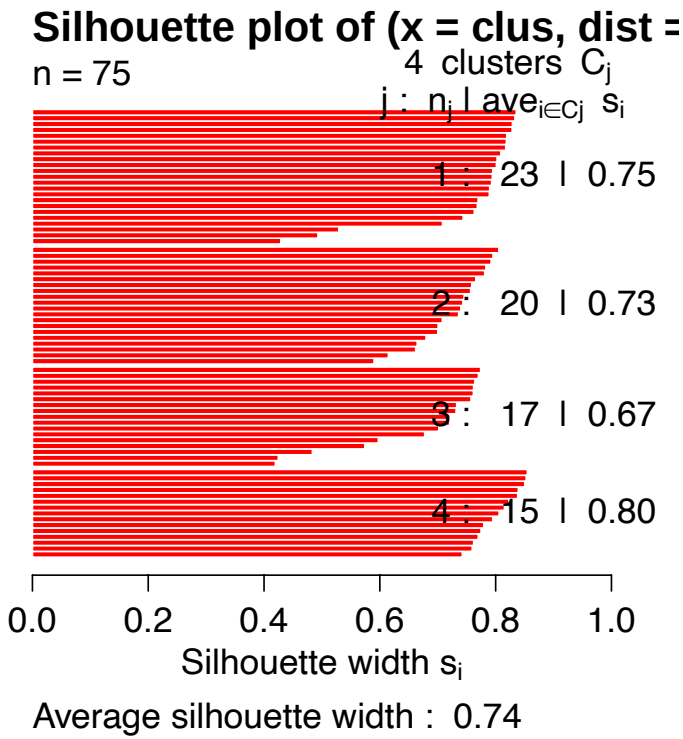




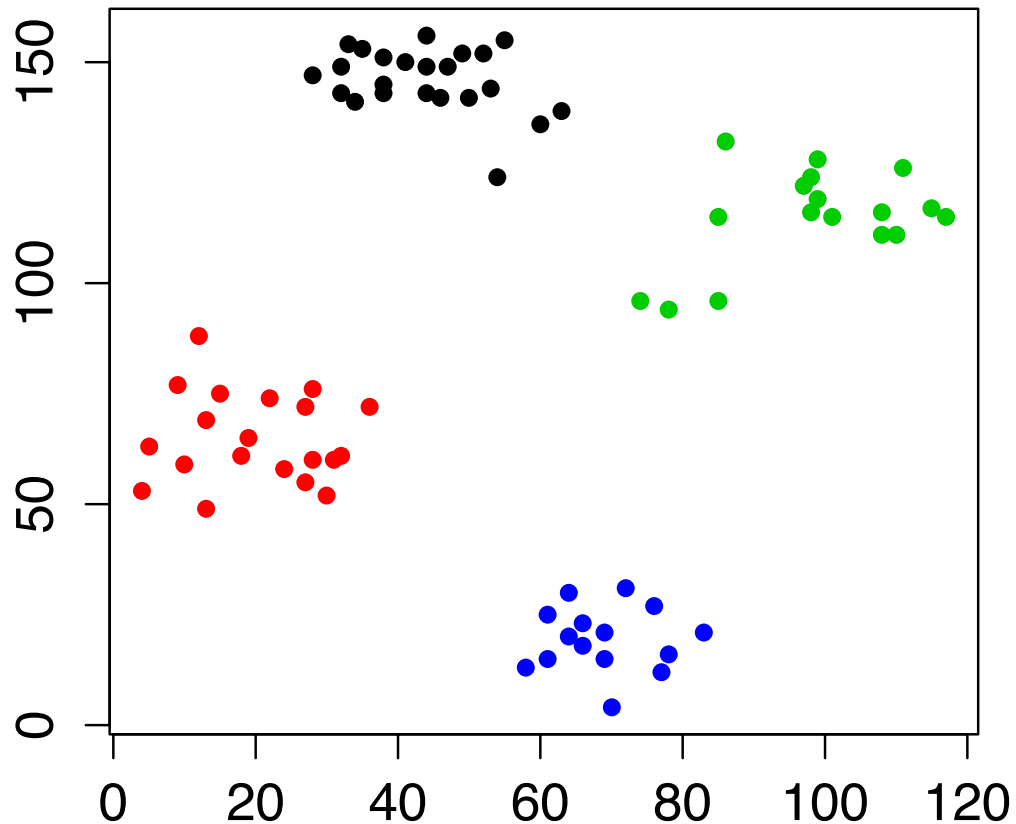


An (internal) cluster validity criterion

The average of the silhouette widths of the previous example is close to .75 suggesting that a strong clustering structure was found in Ruspini data. Since all silhouette coefficients are above .4 no points are misplaced in their clusters



Ruspini plot into 4 clusters using the k-means algorithm



Ruspini plot into 4 clusters using the *k*-means algorithm

R (code)

```
library(cluster)
ch.res<-rep(NA,10)
si.res<-rep(NA,10)
ssqw.res<-rep(NA,10)
plot(ruspini)
for (n in 2:10){
km <- kmeans(ruspini,n,nstart=500)
ch.res[n]<-round(calinhara(ruspini,km$cluster),digits=2)
si.res[n]<-mean(silhouette(km$cluster,dist(ruspini))[,3])
ssqw.res[n]<-km$tot.withinss
# ssqw.res[n]<-km$betweenss/km$tot.withinss
}
par(mfrow=c(2,2))
plot(ssqw.res,type="b",col="black",main="SSQw")
plot(si.res,type="b",col="blue",main="SIL")
plot(ch.res,type="b",col="red",main="CH")
km <- kmeans(ruspini,4,nstart=500)
plot(ruspini, col=km$cluster)
```

External cluster validation

COMPARING PARTITIONS

Motivation

- Several clustering analyses of the same data can be done using distinct meaningful combinations of clustering methods and resemblance notions;
- Clustering analyses having a high degree of agreement may suggest that the common patterns produced by these methods is robust;
- If the clustering structure is known *a priori* and it is important to assess how well the clustering method was able to reproduce this structure;
- It is very difficult (if not impossible or meaningless) to match each cluster of a partition with the correct cluster of the other partition
- The usual way is to compute the number of pairs of individuals that both clustering methods agree to assign in the same/distinct class

Rand index

- Assume that N individuals are classified by two distinct clustering methods. The total number of pairs of individuals is $\binom{N}{2} = \frac{N(N-1)}{2}$. Denote by:

  A: number of pairs classified in the same class in both partitions

  B: number of pairs classified in the same [distinct] class in the first [second] partition

  C: number of pairs classified in the distinct [same] class in the first [second] partition

  D: number of pairs classified in distinct classes in both partitions

- The above quantities can be represented in a contingency table as follows:

	Part. 2		
Part. 1	Classif. in the same group	Classif. in distinct groups	
Classif. in the same group	A	B	A+B
Classif. in distinct groups	C	D	C+D
	A+C	B+D	$\binom{N}{2}$

Rand index

- Rand index (RI) is a simple concordance index used as an external validity index to compare partitions and is defined as,

$$RI = \frac{A + D}{\binom{N}{2}} = \frac{A + D}{A + B + C + D},$$

  where $A+D$ is the number of agreements for both partitions

- It ranges from 0 *(total disagreement)* to 1 *(total agreement)*
- To each partition of a set of N individuals, $x_1, \ldots, x_N$ we associate a binary vector of length $\binom{N}{2}$, where the component corresponding to pair (i, j) is equal 1 if x_i and x_j are assigned in the same class and 0 otherwise
- The Rand index of two partitions is just the simple matching index between the binary vectors associated to these partitions
- Note that the number of groups in each partition can be distinct

Rand index: example

$X = \{a, b, c, d, e, f\}$

Partition 1: $a\ b\ e \mid c \mid d\ f$ Partition 2: $a\ c \mid b\ d \mid e\ f$

	a	b	c	d	e
b	1	·	·	·	·
c	0	0	·	·	·
d	0	0	0	·	·
e	1	1	0	0	·
f	0	0	0	1	0

	a	b	c	d	e
b	0	·	·	·	·
c	1	0	·	·	·
d	0	1	0	·	·
e	0	0	0	0	·
f	0	0	0	0	1

The contingency table between partition 1 and partition 2 is

	1	0	
1	A	B	$A+B$
0	C	D	$C+D$
	$A+C$	$B+D$	$\binom{N}{2}$

$=$

	1	0	
1	**0**	4	4
0	3	**8**	11
	3	12	**15**

Hence

$$RI = \frac{\mathbf{0+8}}{\mathbf{15}} = 0.53333\ldots$$

Computing the Rand index in R

To compute the Rand index of the two partitions in 3 classes,

$$\mathcal{P}_1: \quad a\ b\ e \mid c \mid d\ f \qquad\qquad \mathcal{P}_1: \quad a\ c \mid b\ d \mid e\ f,$$

we encoded these partitions as vectors

$$(1,1,2,3,1,3), \qquad (1,2,1,2,3,3),$$

representing the classes of the elements a, b, c, d, e, f

R (Rand index)

```
# Codigo da funcao do Professor Cadima
rand <- function(class1,class2){
n <- length(class1)
c <- as.dist(outer(class1,class1,"=="))
d <- as.dist(outer(class2,class2,"=="))
rand <- sum(c == d)/(n*(n-1)/2)
return(rand) }
rand(c(1,1,2,3,1,3),c(1,2,1,2,3,3))
# 0.5333333
2 random samples of length 1000 with elements extracted from 1,...,10
p1<-sample(1:10,1000,replace=TRUE)
p2<-sample(1:10,1000,replace=TRUE)
rand(p1,p2)
# 0.8196997
```

Correction for chance: adjusted Rand index

The expected value of Rand index between random partitions is not constant (e.g., 0). To overcome this issue Hubert and Arabie proposed the so-called adjusted Rand index

$$ARI = \frac{RI - E[RI]}{\max(RI) - E[RI]} = \frac{RI - E[RI]}{1 - E[RI]},$$

assuming the Permutation Model as the null model for random clusterings i.e., each partition $\mathcal{P}_i$, $i = 1, 2$, is drawn at random, subject to having a prescribed number of classes K_i and a prescribed number of elements $N_{i,j}$ in each class $j = 1, \ldots, K_i$.

It can be proved that,

$$E[RI] = \frac{2Q_1\, Q_2 - \binom{N}{2}(Q_1 + Q_2) + \binom{N}{2}}{\binom{N}{2}^2},$$

where $Q_i = \sum_{j=1}^{K_i} \binom{N_{ij}}{2}$, $i = 1, 2$, yielding

$$ARI = \frac{\binom{N}{2}(A + D) - U}{\binom{N}{2}^2 - U},$$

where $U = (A + B)(A + C) + (D + B)(D + C)$ and $\binom{N}{2} = \frac{N(N-1)}{2}$.

$ARI \in [-1, 1]$ with $ARI \approx 0$ for independent random partitions, $ARI = 1$ for identical partitions and $ARI < 0$ if the partitions have a low agreement.

More difficult to interpret than the more simple Rand index

Computing the adjusted Rand index

Consider the two partitions in 3 classes of slide 141,

$$\mathcal{P}_1: \quad a\ b\ e \mid c \mid d\ f \qquad\qquad \mathcal{P}_2: \quad a\ c \mid b\ d \mid e\ f,$$

By the results of slide 141, we have $U = (A+B)(A+C)+(D+B)(D+C) = 144$, $\binom{N}{2} = A+B+C+D = \frac{N(N+1)}{2} = 15$ and we therefore we get

$$ARI = \frac{\binom{N}{2}(A+D) - U}{\binom{N}{2} - U} = -0.2962963$$

We can recompute this index using the `adjustedRandIndex` function of the MCLUST package. In order to accomplish that we consider the vectors v_1 e v_2 representing the classes of the elements a, b, c, d, e, f, in the two partitions, $\mathcal{P}_1$ and $\mathcal{P}_2$, that is,

$$v_1 = (1,1,2,3,1,3), \qquad v_2 = (1,2,1,2,3,3)$$

R (Adjusted rand index)

```
require(mclust)
# with the partitions above we get,
adjustedRandIndex(c(1,1,2,3,1,3),c(1,2,1,2,3,3))= -0.2962963

# with the same random samples of slide 142 we get,
adjustedRandIndex(p1,p2)=0.0002526569 ≈ 0 as intended :)
```