

# UNSUPERVISED (and semi-supervised) LEARNING

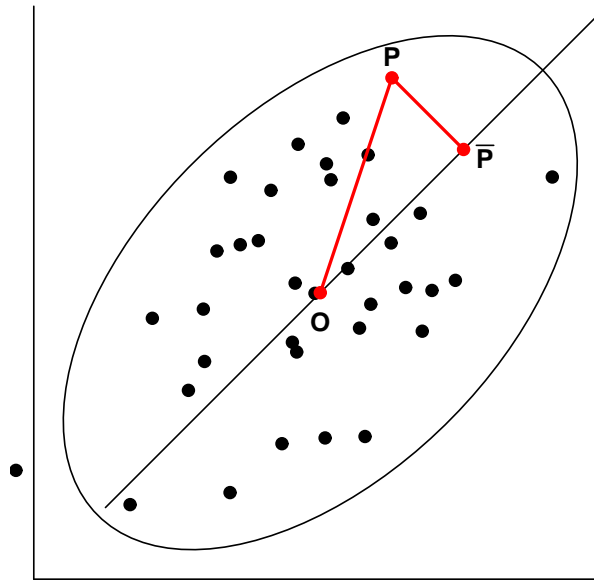
## 20. Finding Structure in Data: Principal Component Analysis, and More

Given  $n$   $p$ -dimensional data points i.e. cloud of  $n$  points in  $p$ -dimensional space.

**Aim:** provide the 'best'  $r$ -dimensional representation of this cloud, where  $r < p$ .

Principal component analysis (PCA) is one of the realizations of this aim with certain adopted meaning of 'best'.

Consider  $p = 2$  and a two-dimensional cloud of points. Position coordinate center  $O$  at the centroid of this points ( $\equiv \mathbf{x}_i := \mathbf{x}_i - \bar{\mathbf{x}}$ ).



We look for one-dimensional representation of this cloud such that the displacement of points is relatively small.

Observe that Pythagoras' theorem implies ( $\bar{P}_i$ : projection of  $P_i$ )

$$(OP_i)^2 = (O\bar{P}_i)^2 + (P_i\bar{P}_i)^2.$$

and thus

$$\sum_{i=1}^n (OP_i)^2 = \sum_{i=1}^n (O\bar{P}_i)^2 + \sum_{i=1}^n (P_i\bar{P}_i)^2.$$

As the lefthand side does not depend on the direction of the line, we see that

minimization of  $\sum_{i=1}^n (P_i\bar{P}_i)^2 \equiv$  maximization of  $\sum_{i=1}^n (O\bar{P}_i)^2$  or equivalently, maximization of

$$\frac{1}{n-1} \sum_{i=1}^n (O\bar{P}_i)^2$$

i.e. **the variance** of the projections on the considered line.

**This justifies the following strategy of PCA (for general  $p$ ):**

**(i) For  $x_1, \dots, x_n \in R^p$  find direction  $a_1$  such that  $\|a_1\| = 1$  and the variance of points projected onto this direction,  $a_1^T x_1, \dots, a_1^T x_n$ , is the largest.**

**(ii) Find direction  $a_2$  such  $\|a_2\| = 1$  and  $a_2$  is perpendicular to  $a_1$  such that the the variance of points projected onto this direction,  $a_2^T x_1, \dots, a_2^T x_n$ , is the largest among perpendicular directions.**

**(iii) continue to choose  $a_1, \dots, a_r$ .**

Let  $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})^T$ ,  $i = 1, \dots, n$  and  $\mathbf{a}_i = (a_{i1}, \dots, a_{ip})^T$ .

$y_i = \mathbf{a}_i^T \mathbf{x}$  is called  $i^{\text{th}}$  principal component

$\mathbf{a}_i$  -vector of loadings (direction) of the  $i^{\text{th}}$  principal component.

The  $r$  principal component values for  $s^{\text{th}}$  sample point are thus given by

$$\begin{aligned}y_{s1} &= a_{11}x_{s1} + a_{12}x_{s2} + \dots + a_{1p}x_{sp} \\y_{s2} &= a_{21}x_{s1} + a_{22}x_{s2} + \dots + a_{2p}x_{sp} \\&\dots \\y_{sr} &= a_{r1}x_{s1} + a_{r2}x_{s2} + \dots + a_{rp}x_{sp}\end{aligned}$$

$y_{s1}, y_{s2}, \dots, y_{sr}$  -  $r$  principal component scores for the  $s^{\text{th}}$  individual.

Note: as principal directions are orthogonal, projections of a data set on different directions are **uncorrelated**.

How to find principal directions ?

Simple algebraic solution exists:

Consider the empirical covariance matrix  $S$  corresponding to the cloud of points. Find **eigenvalues**  $\lambda_i$  of  $S$  and order them:  $\lambda_1 \geq \lambda_2 \cdots \geq \lambda_p \geq 0$ . Principal directions are given by **eigenvectors**  $\mathbf{a}_1, \dots, \mathbf{a}_p$  (of unit length) corresponding to the ordered eigenvalues.

Another point of view: PCA yields transformation of data matrix  $X = (x_{ij})$  with consecutive observations being rows such that

$$Y_{n \times p} = X_{n \times p} A_{p \times p},$$

where columns of  $A$  are eigenvectors of covariance matrix  $S$ .

How to choose the number of principal directions  $r$  ?

**Fact:** The variance of projections of  $\mathbf{x}_1, \dots, \mathbf{x}_n$  on the hyperplane spanned by  $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_r$  is equal to  $\lambda_1 + \lambda_2 + \dots + \lambda_r$ .

First choice of  $r$ :

$$P_k = \frac{\lambda_1 + \lambda_2 + \dots + \lambda_k}{\lambda_1 + \lambda_2 + \dots + \lambda_p}$$

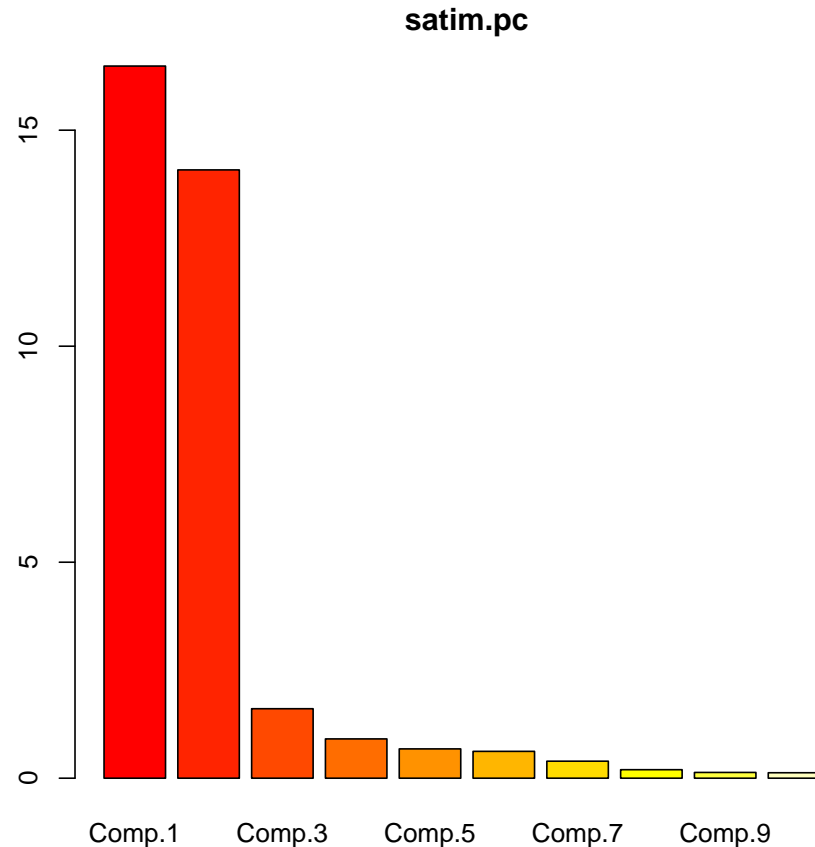
Take as  $r$

$\min r : P_r \geq$  given threshold  $\gamma$  (usually  $\gamma \approx 0.7 - 0.9$ )

**Interpretation :** The principal directions chosen explain at least  $100\gamma\%$  variability of data.

Second choice of  $r$ :

Consider a **scree-plot** i.e. the plot of  $\lambda_i$ 's against index  $i$  and choose as  $r$  minimal index  $i_0$  such that for  $i > i_0$  the plot levels off.



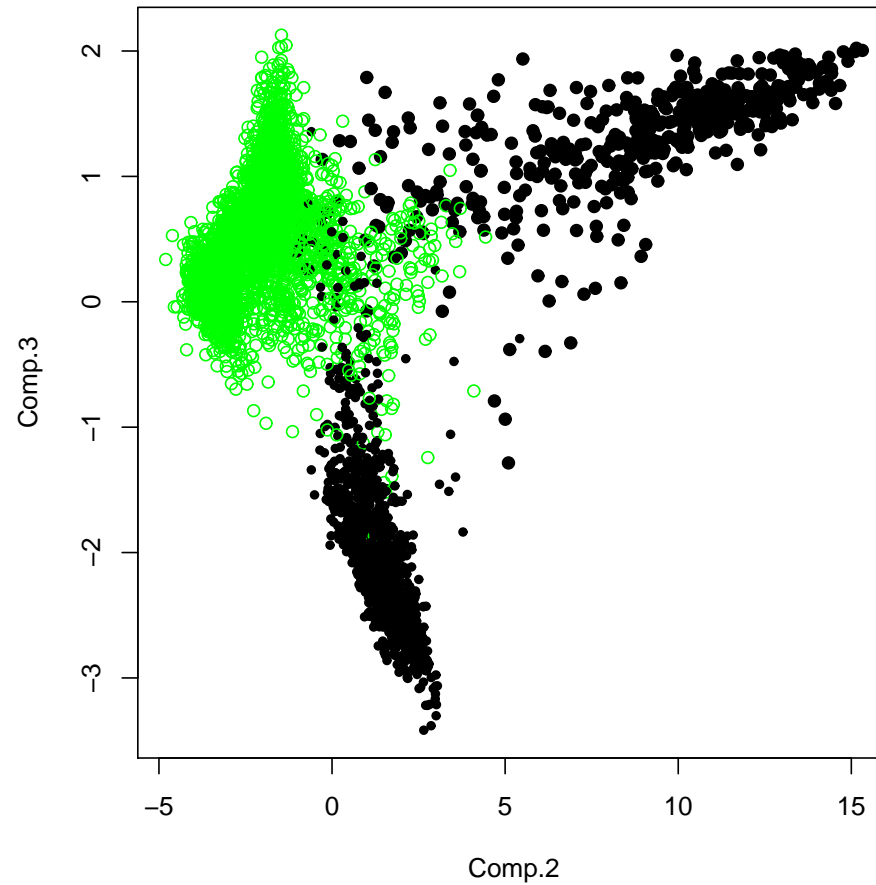
This does not always work: it can happen that we have many components with comparable and small variabilities which jointly have non-negligible impact on variability of data.

## Applications

A plot of the first two or three component scores frequently yields a new insight into the structure of the data. Some ideas:

- useful to get some idea about possible clusters, outliers etc.
- PCA is frequently used as a feature extraction method. We work with the first few principal components instead of original variables. This is used e.g. in PCA regression when response is regressed on the first  $r$  principal components of predictors. This should be used with caution: principal components do not use the response. It is possible that a lesser principal component is actually very important in predicting the response.

In classification problems with large number of attributes it frequently pays off to perform LDA or QDA on several first principal components of  $\mathbf{x}$ . Scatterplot of the 2nd and 3rd principal components for satellite image data:



Nonlinear Principal Components ...

Projection Pursuit Density Estimation ...

Multidimensional Scaling ...

Factor Analysis, ICA ...

## 21. Finding groups in data - cluster analysis

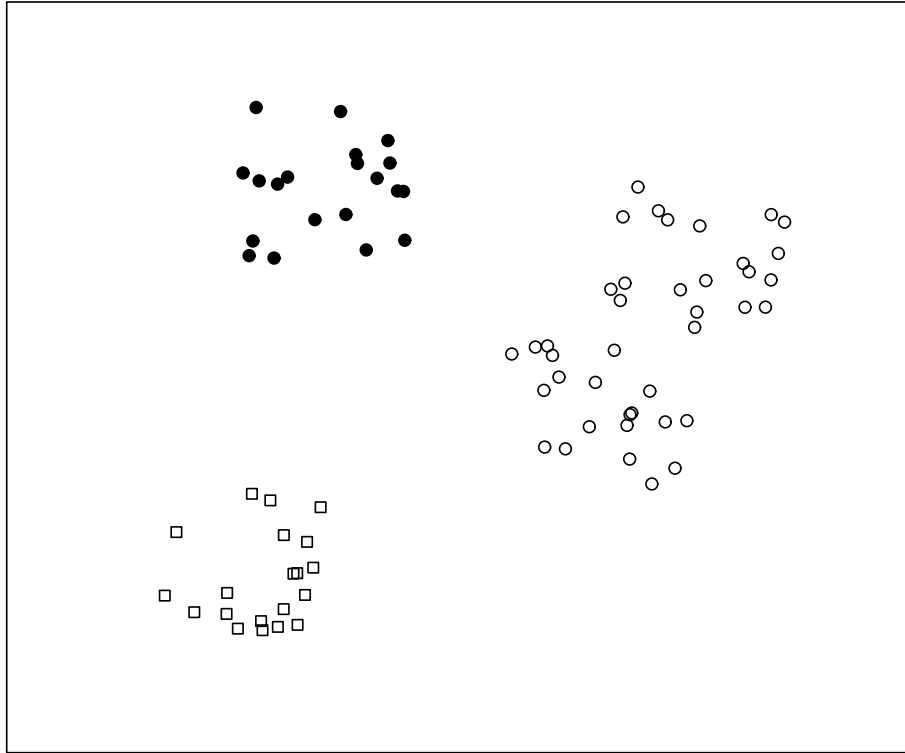
**Given  $n$   $p$ -dimensional data points, i.e. we are given a cloud of  $n$  points in  $p$ -dimensional space.**

**Aim: Divide data points into groups (clusters) such that dissimilarity between points belonging to the same group is on the average smaller than dissimilarity between points belonging to different groups.**

**An important problem: how to measure dissimilarity between objects?**

**Let us first focus on the case when objects are given as points in  $R^p$  and dissimilarity is related to Euclidean distance.**

**Assume that we want to divide the data into  $K$  groups with  $K$  given (for the time being).**



$K = 3$  or  $K = 4$  in this case ?

**Define**

$C(i) = k$  when  $x_i$  belongs to  $k^{th}$  cluster

and  $d(x_i, x_j)$  - square of Euclidean distance between  $x_i$  and  $x_j$

Let

$$T = \frac{1}{2} \sum_{i=1}^n \sum_{i'=1}^n d_{ii'} \quad (1)$$

and note that

$$T = W + B, \quad (2)$$

where

$$W = \frac{1}{2} \sum_{k=1}^K \sum_{C(i)=k} \sum_{C(i')=k} d(i, i') \quad (3)$$

and

$$B = \frac{1}{2} \sum_{k=1}^K \sum_{C(i)=k} \sum_{C(i') \neq k} d(i, i'). \quad (4)$$

**It is known that there are**

$$\frac{1}{K!} \sum_{k=1}^K (-1)^{K-k} \binom{K}{k} k^n$$

**different groupings of  $n$  observations into  $K$  groups, which is a forbiddingly large number even for modest  $n$  and  $K$  ( $K \ll n$ )!**

**It is easy to show that**

$$W = \sum_{k=1}^K \sum_{C(i)=k} d(\mathbf{x}_i, \mathbf{m}_k) n_k, \quad (5)$$

**where  $\mathbf{m}_k$ ,  $k = 1, \dots, K$ , is the mean of the observations in the  $k$ -th cluster,**

$$\mathbf{m}_k = \frac{1}{n_k} \sum_{C(i)=k} \mathbf{x}_i, \quad (6)$$

**with  $n_k$  being the size of the  $k$ -th cluster.**

**Write**

$$\tilde{W} = \sum_{k=1}^K \sum_{C(i)=k} d(\mathbf{x}_i, \mathbf{m}_k) = \sum_{i=1}^n d(\mathbf{x}_i, \mathbf{m}_{C(i)}) \quad (7)$$

## K-means criterion

Find a partition into  $K$  groups such that

$$\tilde{W} = \sum_{i=1}^n d(\mathbf{x}_i, \mathbf{m}_{C(i)})$$

is minimal (i.e. we want to minimize the within-groups sum of squares). This is a combinatorial optimization problem but optimization by direct enumeration is usually not feasible.

## K-means algorithm

1. For a given cluster assignment  $C$  the total cluster variance

$$W = \sum_{k=1}^K \sum_{C(i)=k} d(\mathbf{x}_i, \mathbf{m}_k)$$

is minimized over  $(m_k)$ ,  $k = 1, \dots, K$  yielding the means of currently assigned clusters;

2. Given a current set of means  $\{m_1, m_2, \dots, m_k\}$  for each  $\mathbf{x}_i$ , find the closest current cluster mean and assign  $\mathbf{x}_i$  to this cluster;

3. Steps 1 and 2 are iterated until the assignments do not change.

**Note: The versions of the algorithm differ depending on:**

**(i) the moment the centers are modified:**

**- batch version: center is modified after the whole batch  $x_1, \dots, x_n$  is assigned in 2;**

**-sequential version: center is modified each time a new element is assigned.**

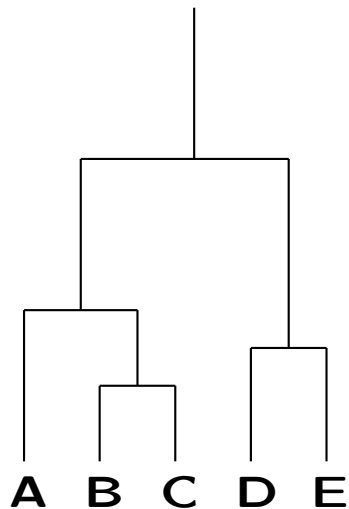
**(ii) initial cluster assignment.**

**Algorithm converges as  $\tilde{W}$  decreases at each step but convergence to local minimum is possible.**

## Hierarchical methods

- agglomerative methods
- divisive methods

An agglomerative method begins with  $n$  subclusters, each containing one data point and at each step merges the two most similar groups to form a new cluster. The algorithm proceeds until forming a single cluster. This is usually visualised in terms of a **dendrogram**.



**The merging process is:**

$$A, B, C, D, E \rightarrow A, \{B, C\}, D, E$$

$$A, \{B, C\}, D, E \rightarrow A, \{B, C\}, \{D, E\}$$

$$A, \{B, C\}, \{D, E\} \rightarrow \{A, B, C\}, \{D, E\}$$

$$\{A, B, C\}, \{D, E\} \rightarrow \{A, B, C, D, E\}$$

A divisive method starts from a single cluster and uses division instead of merging.

Each method requires definition of clusters' dissimilarity which is based on dissimilarities between members of the clusters.

### Clusters' dissimilarity

Consider two clusters  $i$  and  $j$ . Let their dissimilarity be denoted by  $D_{ij}$ .

### Single-linkage dissimilarity

$$D_{ij} = \min d_{kk'},$$

where  $k$  ranges over cluster  $i$  and  $k'$  ranges over cluster  $j$ . Called also **closest nearest neighbor**

### Complete-linkage dissimilarity

$$D_{ij} = \max d_{kk'},$$

where  $k$  ranges over cluster  $i$  and  $k'$  ranges over cluster  $j$ . Called also **furthest nearest neighbor**

## Group-average linkage dissimilarity

$$D_{ij} = \frac{1}{n_i n_j} \sum d_{kk'},$$

where  $k$  ranges over cluster  $i$  and  $k'$  ranges over cluster  $j$  and  $n_i$  is the number of observations in cluster  $i$ .

It is possible to calculate dissimilarity between merged cluster  $i$  and  $j$  and cluster  $k$  using  $D_{ik}$  and  $D_{jk}$ . E.g., in the case of single-linkage algorithm

$$D_{k.i,j} = \min(D_{ki}, D_{kj}).$$

For a fixed number of clusters  $K$  we stop hierarchical algorithm when exactly  $K$  clusters are obtained.

## Main characteristics of various methods:

- **single linkage:** tends to produce "long" groups with large diameters (effect of chaining)
- **complete-linkage:** *relatively compact clusters relatively far apart*
- **average-linkage:** usually compromise between these two methods

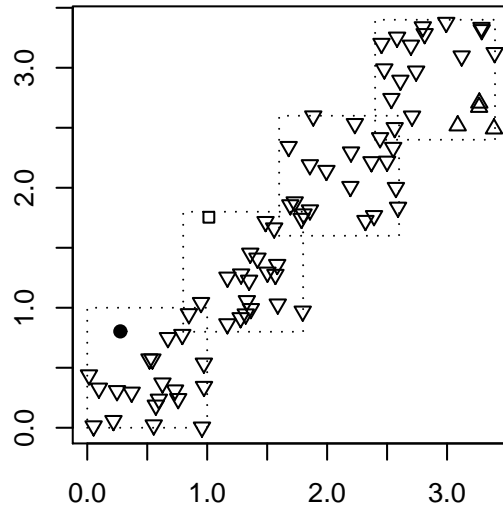
**Choice of the number of clusters: usually difficult, no universal algorithm exists. General heuristics:**

**Calculate**

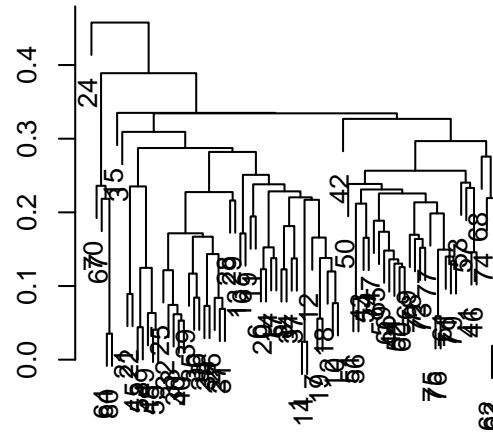
$$W_K = \sum_{k=1}^K \sum_{C(i)=k} d(\mathbf{x}_i, \mathbf{m}_k)$$

**for different  $K$ . As in the case of scree-plot choose as true value of  $K$  the value  $K^*$  for which the plot of  $W_K$  against  $K$  levels off.**

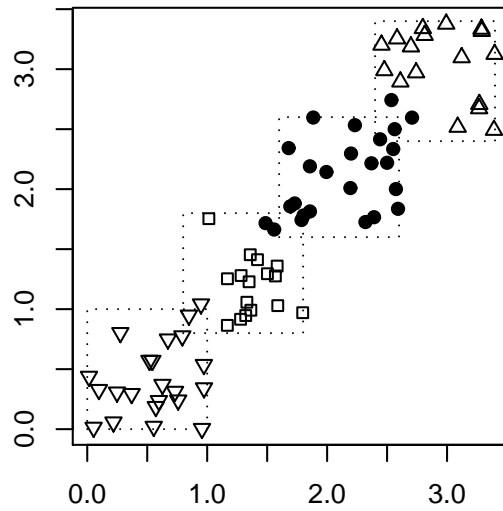
**SINGLE LINKAGE**



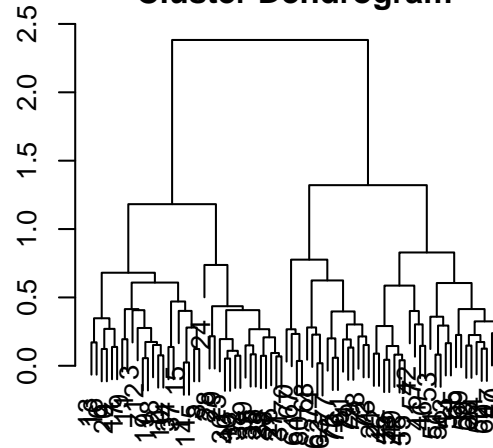
**Cluster Dendrogram**



**AVERAGE LINKAGE**



**Cluster Dendrogram**



$K = 4$  was chosen.

## Dissimilarity measures

Dissimilarity measure does not need to be a metrics (triangle inequality is not always satisfied).

For vectors  $\in R^p$ : any metric on  $R^p$  may be considered; for quantitative features on different scales weighted Euclidean distance is appropriate

$$d(\mathbf{x}, \mathbf{y}) = \left( \sum_{i=1}^p w_i^2 (x_i - y_i)^2 \right)^{1/2}$$

where  $w_i$

is either (standard deviation of  $i^{th}$  variable) $^{-1}$  or (range) $^{-1}$ .

**For vectors with binary (0 and 1) coordinates:**  $x = (x_1, x_2, \dots, x_p)$ ,  
 $y = (y_1, y_2, \dots, y_p)$  we define

$$a = \#\{x_i = 1 \ \& \ y_i = 1\}, \quad b = \#\{x_i = 0 \ \& \ y_i = 1\};$$

$$c = \#\{x_i = 1 \ \& \ y_i = 0\}, \quad d = \#\{x_i = 0 \ \& \ y_i = 0\}.$$

### Dissimilarity measures for binary data

- **Normalized Hamming distance:**  $\frac{b+c}{a+b+c+d}$

- **Jacquard:**  $\frac{b+c}{a+b+c}$

(lack of occurrence of a feature does not make objects more similar)

- **Czekanowski:**  $1 - \frac{2a}{2a+b+c}$

**For qualitative vectors having more than two levels**

$$1 - \frac{\# \text{coordinates having the same value}}{\# \text{coordinates}}$$

**Gower coefficients for mixed variables....**

## **22. Hierarchical Clustering via Joint Between-Within Distance**

**Such methods have been of interest to statisticians for decades now. Recently, their valuable version has been proposed by G.J. Székely and M.L. Rizzo. The method can be recommended as applicable in general and useful in particular when standard dendrograms fail to give satisfactory results.**

Let  $A = \{\mathbf{a}_1, \dots, \mathbf{a}_{n_1}\}$  and  $B = \{\mathbf{b}_1, \dots, \mathbf{b}_{n_2}\}$  be two sets in  $\mathbb{R}^p$  and let

$$e(A, B) = \frac{n_1 n_2}{n_1 + n_2} \left( \frac{2}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \|\mathbf{a}_i - \mathbf{b}_j\| \right) \quad (8)$$

$$- \frac{1}{n_1^2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_1} \|\mathbf{a}_i - \mathbf{a}_j\| - \frac{1}{n_2^2} \sum_{i=1}^{n_2} \sum_{j=1}^{n_2} \|\mathbf{b}_i - \mathbf{b}_j\| \quad (9)$$

**Theorem.** Suppose  $\mathbf{X}, \mathbf{X}' \in \mathbb{R}^p$  are i.i.d. random vectors with distribution  $F$ ,  $\mathbf{Y}, \mathbf{Y}' \in \mathbb{R}^p$  are i.i.d. random vectors with distribution  $G$ , independent of  $\mathbf{X}, \mathbf{X}'$ . Suppose  $E\|\mathbf{X}\| < \infty$  and  $E\|\mathbf{Y}\| < \infty$ . Then

$$2E\|\mathbf{X} - \mathbf{Y}\| - E\|\mathbf{X} - \mathbf{X}'\| - E\|\mathbf{Y} - \mathbf{Y}'\| \geq 0,$$

and equality holds if and only if  $F = G$ .

**Corollary.** For all finite nonempty sets  $A, B \in \mathbb{R}^p$ ,  $e(A, B) \geq 0$  and equality holds if and only if  $A = B$ .

**The Authors** have developed a hierarchical algorithm that merges the pair of clusters with minimum  $e$ -distance at each level.

## 23. Clustering on subsets of attributes (by Friedman and Meulman)

Let

$$W(C) = \sum_{k=1}^K \frac{W_k}{n_k^2} \sum_{C(i)=k} \sum_{C(i')=k} d(i, i'). \quad (10)$$

(In particular, by setting  $W_k = n_k^2$  we assign the same weight to all pairs of objects, what amounts to aiming at clusters of possibly similar sizes.)

More generally, let

$$W(C, \{\mathbf{w}_k\}_1^K) = \sum_{k=1}^K \frac{W_k}{n_k^2} \sum_{C(i)=k} \sum_{C(i')=k} \left( \sum_{j=1}^p w_{j,k} d(i, i')_j + \lambda w_{j,k} \log w_{j,k} \right), \quad (11)$$

where  $\lambda \geq 0$ ,  $d(i, i')_j$  is the squared distance on  $j$ -th attribute for objects  $i$  and  $i'$ ,  $\mathbf{w}_k = \{w_{j,k}\}_{j=1}^p$ ,  $k = 1, \dots, K$ ,

$$\{w_{j,k} \geq 0\}_{j=1}^p, \quad \sum_{j=1}^p w_{j,k} = 1, \quad k = 1, \dots, K.$$

Now, (9) is minimized wrt clusters  $C$  and weights  $\{\mathbf{w}_k\}_1^K$ .

## COSA as a preliminary step to hierarchical clustering

Let us replace minimization of

$$W(C, \{\mathbf{w}_k\}_1^K) = \sum_{k=1}^K \frac{W_k}{n_k^2} \sum_{C(i)=k} \sum_{C(i')=k} \left( \sum_{j=1}^p w_{j,k} d(i, i')_j + \lambda w_{j,k} \log w_{j,k} \right), \quad (12)$$

by that of

$$W(\mathbf{W}) = \sum_{i=1}^n \frac{1}{K} \sum_{i' \in KNN(i)} \left( \sum_{j=1}^p w_{j,i} d(i, i')_j + \lambda \sum_{j=1}^p w_{j,i} \log w_{j,i} \right), \quad (13)$$

where  $KNN(i)$  denotes  $K$  nearest neighbors of object  $i$  and  $\mathbf{W}$  is a  $p \times n$  matrix. ( $K$  is chosen experimentally, say,  $K \approx \sqrt{n}$ .) In this way, the following distances are defined

$$D(i, i') = \sum_{j=1}^p w_{j,i} d(i, i')_j.$$

## 24. Semisupervised Learning

One semisupervised learning algorithm (proposed by Davidson and Ravi) for set  $S$  of observations:

1. Construct the transitive closure of the ML ("must link") constraints resulting in  $r$  connected components  $M_1, M_2, \dots, M_r$ .
  2. If two points  $\{x, y\}$  are both a CL ("cannot link") and ML constraint, then output "No Solution" and stop.
  3. Let  $S_1 = S - (\cup_{i=1}^r M_i)$ . Let  $k_{\max} = r + |S_1|$ .
  4. Construct an initial feasible clustering with  $k_{\max}$  clusters consisting of the  $r$  clusters  $M_1, \dots, M_r$  and a singleton cluster for each point in  $S_1$ . Set  $t = k_{\max}$ .
  5. while (there exists a pair of mergeable clusters) do
    - (a) Select a pair of clusters  $C_l$  and  $C_m$  according to the specified distance criterion.
    - (b) Merge  $C_l$  into  $C_m$  and remove  $C_l$ . {The result is  $Dendrogram_{t-1}$ .
    - (c)  $t = t - 1$ .
- endwhile

## 25. In lieu of a coda: some comments and new developments

Solution paths for regression and supervised classification ...

Orthogonal trees ...

Hoefding trees ...

Feature selection for supervised classificatio ...

Clusterability measures ...

...

## References

1. **T. Hastie, R. Tibshirani, J. Friedman** *Elements of Statistical Learning: Data Mining, Inference and Prediction*, **Springer, 2009, Second Edition**
2. **A. Webb** *Statistical Pattern Recognition* **Wiley, 2002, Second Edition**
3. **C.M. Bishop** *Pattern Recognition and Machine Learning*, **Springer, 2006**
4. **J. Koronacki, J. Ćwik** *Statystyczne systemy uczące się*, **Second Edition, Exit, 2009 (1st Edition, WNT, 2005)**
5. **W. Venables, B. Ripley** *Modern Applied Statistics with S-PLUS*, **Wiley, 2003**
6. **B. Ripley** *Formulae for Linear Models*,  
<http://www.stats.ox.ac.uk/~teo/statsmethod/formulae.pdf>
7. **J. Koronacki, J. Mielniczuk** *Statystyka dla studentów kierunków technicznych i przyrodniczych*, **WNT 2001, 2004, 2006, 2009**