Unsupervised learning

1: cluster analysis

Antonio D'Ambrosio

Outline I

Introduction I

Unsupervised learning is often performed as part of an exploratory data analysis.

It can be hard to assess the results obtained from unsupervised learning methods: we have not a "supervisor". In unsupervised learning, there is no way to check our work because we don't know the true answer.

There are several models and methods dealing with unsupervised learning: factor analysis, principal component analysis, correspondence analysis, multidimensional scaling, cluster analysis....

Introduction II

We will focus our attention on Cluster analysis.

Clustering methods

Clustering refers to a very broad set of techniques for finding subgroups, or clusters, in a data set.

The aim is seeking to partition statstical units into distinct groups so that the observations within each group are quite *similar* to each other, while observations in different groups are quite *different* from each other.

There exist a great number of clustering methods. We will focus on hierarchical cluster analysis, K-means, K-medoids, *fuzzy* C-means, Probablistic Distance clustering, model-based clustering.

Distance Measures

Distance measures I

Let $\mathbf{X}$ be a data matrix with n rows and p columns. A distance (metric) measure between the i-th and j-th unit is a quantity that has the following properties:

- identity: $d(x_i, x_j) = 0 \implies i = j$;
- non negativity: $d(x_i, x_j) \geq 0$;
- simmetry: $d(x_i, x_j) = d(x_j, x_i)$;
- triangular inequality: $d(x_i, x_j) \leq d(x_i, x_h) + d(x_j, x_h)\ \forall i, j, h$.

Distance measures II

Class of Minowski distances:

$$d(x_i, x_j) = \left[\sum_{s=1}^{p} |x_{is} - x_{js}|^k \right]^{\frac{1}{k}}.$$

If $k = 1$, the Minkowsky metric is the *city-block* distance. If $k = 2$, then the Minkowsky metric is the Euclidean distance.

Canberra distance:

$$d(x_i, x_j) = \sum_{s=1}^{p} \frac{|x_{is} - x_{js}|}{(x_{is} + x_{js})}$$

Distance measures III

Mahalanobis distance:

$$d(x_i, x_j) = (x_i - x_j)S^{-1}(x_i - x_j)^T,$$

where S is the covariance matrix.

Distance measures IV

A dissimilarity measure has the following properties:

- identity: $d(x_i, x_j) = 0 \implies i = j$;
- non negativity: $d(x_i, x_j) \geq 0$;
- simmetry: $d(x_i, x_j) = d(x_j, x_i)$;

The squared Euclidean distance is a *dissimilarity* index, not a distance (metric) measure.

To do clustering, it is necessary at least the use of a dissimilarity measure. Better if the measure is a metric (distance).

Hierarchical Clustering

Hierarchical cluster analysis I

Let $\mathbf{X}$ be a data matrix with n rows and p columns. Choose a distance measure of all the $\binom{n}{2}$ pairwise dissimilarities. At the beginning each observation is a cluster.
for $i = n, n-1, \ldots, 2$

- Examine all pairwise inter-cluster dissimilarities among the i clusters and identify the pair of clusters that are least dissimilar (that is, most similar), then fuse these two clusters;
- Compute the new pairwise inter-cluster dissimilarities among the $i-1$ remaining clusters.

Hierarchical cluster analysis II

1. How fuse the clusters?
2. How visualize the solution?

Fusing clusters I

In order to fuse the clusters, we need to define (and choose) the *linkage method*. In the following, we assume that r and s are two clusters, x_{ri} (x_{sj}) is the i-th (j-th) object in cluster r (s), n_r (n_s) is the sample size within cluster r (s).

- *Single linkage*, or nearest neighbor (smallest distance between objects in the two clusters):
  $d(r,s) = min(d(x_{ri}, x_{sj}))$, $i \in (1,\ldots,n_r)$, $j \in (1,\ldots,n_s)$.
  It can be used with any dissimilarity measure.
- *Complete linkage*, or furthest neighbor (largest distance between objects in the two clusters):
  $d(r,s) = max(d(x_{ri}, x_{sj}))$, $i \in (1,\ldots,n_r)$, $j \in (1,\ldots,n_s)$.
  It can be used with any dissimilarity measure.

Fusing clusters II

- *Average linkage* (or **UPGMA** -unweighted pair group method with arithmetic mean - average distance between all pairs of objects in any two clusters):
  $d(r,s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} d(x_{ri}, x_{sj})$.
  It can be used with any dissimilarity measure.
- *Weighted average linkage* (or **WPGMA** - Weighted Pair Group Method with Arithmetic Mean -, or **McQuitty**, recursive definition for the distance between two clusters):
  Suppose r was created by combining clusters q and t: then
  $d(r,s) = \frac{(d(q,s)+d(t,s))}{2}$.
  It can be used with any dissimilarity measure.

Fusing clusters III

- *Centroid linkage* (Euclidean distance between the centroids of the two clusters).
  $d(r,s) = \|\bar{x}_r - \bar{x}_s\|_2$, where $\bar{x}_r = \frac{1}{n_r}\sum_{i=1}^{n_r} x_{ri}$
  Only with Euclidean distance:
- *Median linkage* (Euclidean distance between weighted centroids of the two clusters).
  $d(r,s) = \|\tilde{x}_r - \tilde{x}_s\|_2$, where where $\tilde{x}_r$ and $\tilde{x}_s$ are weighted centroids for the clusters r and s. Suppose cluster r was created by merging clusters q and t, then recursively $\tilde{x}_r = \frac{1}{2}(\tilde{x}_q + \tilde{x}_t)$.
  Only with Euclidean distance

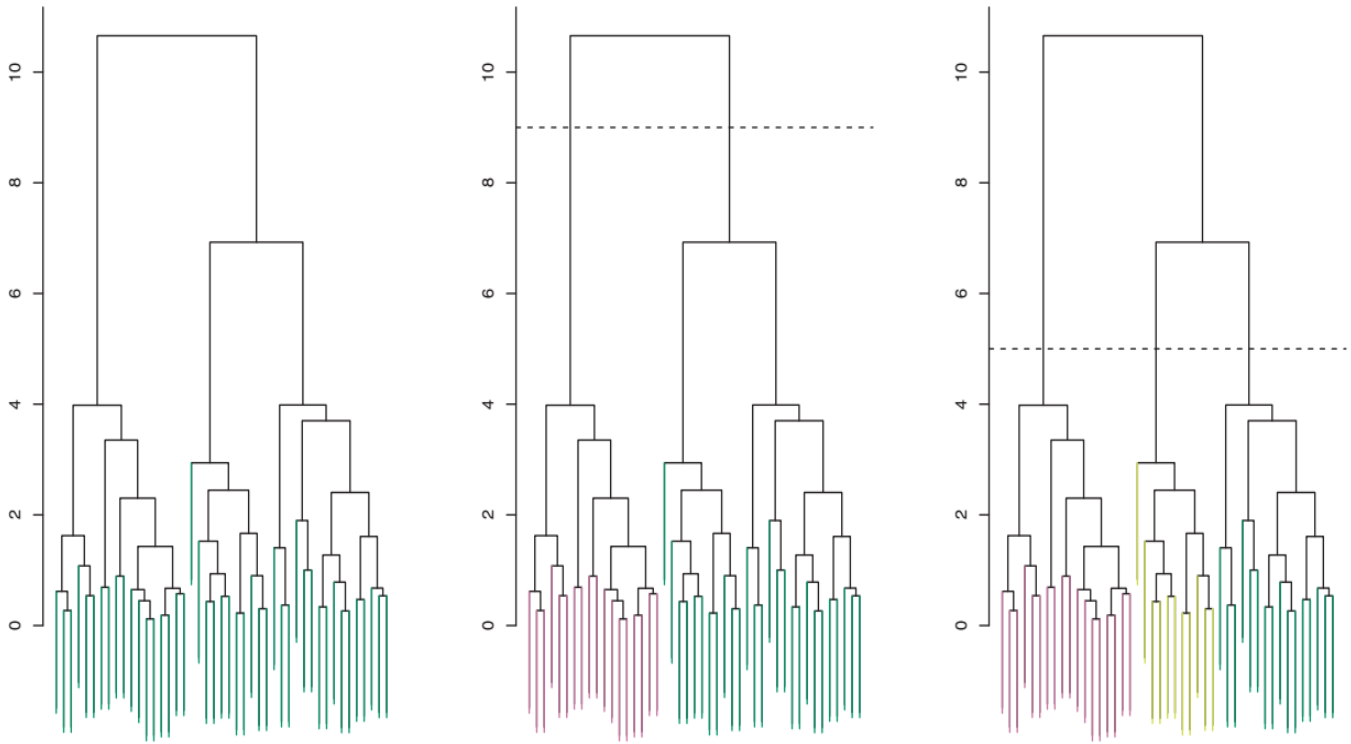
Fusing clusters IV

- *Ward linkage* (incremental sum of squares, the increase in the total within-cluster sum of squares as a result of joining two clusters).
  $d(r,s) = \sqrt{\frac{2n_r n_s}{n_r+n_s}}\|\bar{x}_r - \bar{x}_s\|_2$
  Only with Euclidean distance
- *Minimax* (Bien et al. (2011), Hierarchical Clustering with Prototypes via Minimax Linkage).
  First define radius of a group of points s around x_i as $r(x_i, s) = max_{(j \in s)} d_{ij}$. Then:

$$d(r,s) = min_{i \in s \cup r} r(x_i, s \cup r).$$

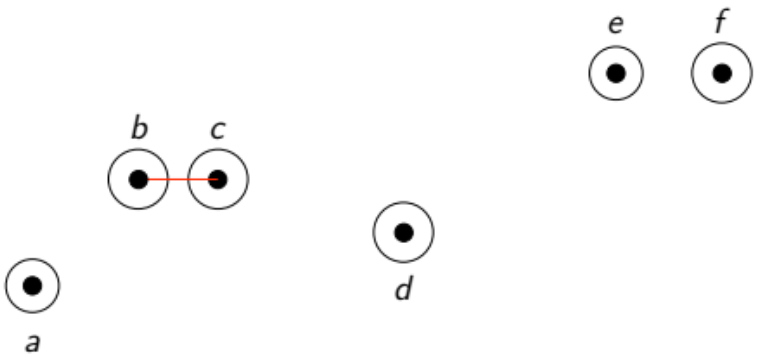
Visualize the solution

To visualize the solution, a graphical tool called *dendrogram* is used.



Building the dendrogram I

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



PROXIMITY MATRIX

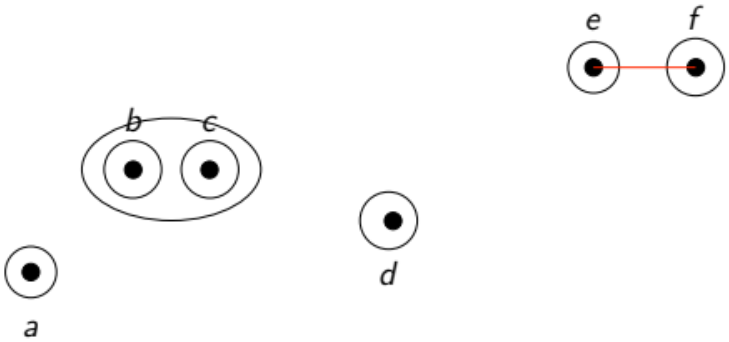
	a	b	c	d	e
b	0.7				
c	1.0	0.3			
d	1.8	.1.3	0.9		
e	2.9	2.4	1.9	1.3	
f	3.4	2.8	2.4	1.7	.5

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 $\{b, c\}$

Building the dendrogram II

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

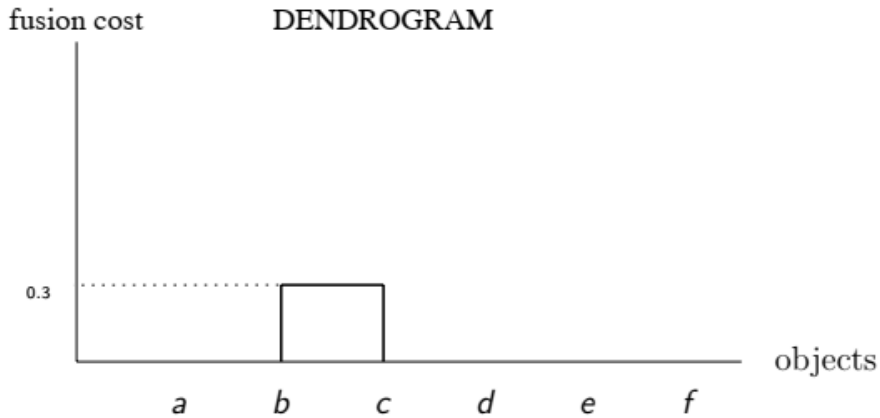


PROXIMITY MATRIX

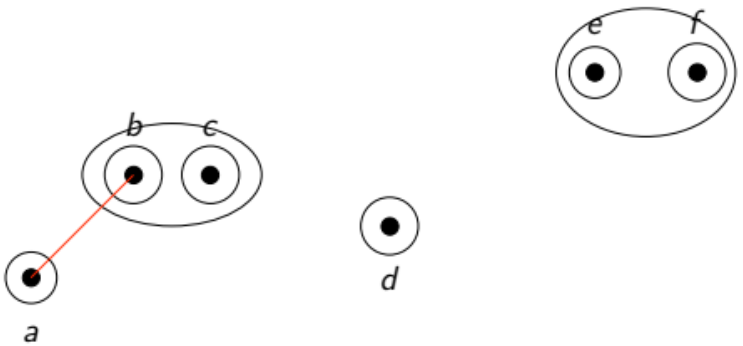
	a	$\{b,c\}$	d	e
$\{b,c\}$	0.7			
d	1.8	0.9		
e	2.9	1.9	1.3	
f	3.4	2.4	1.7	0.5

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

with fusion cost 0.5



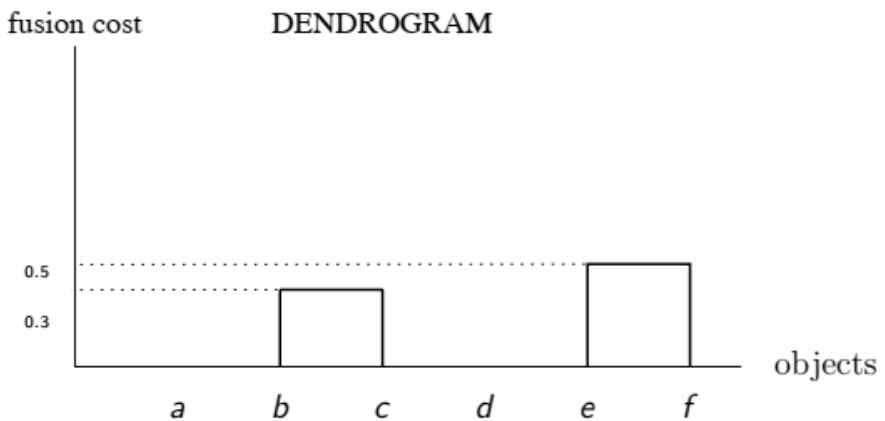
Building the dendrogram III



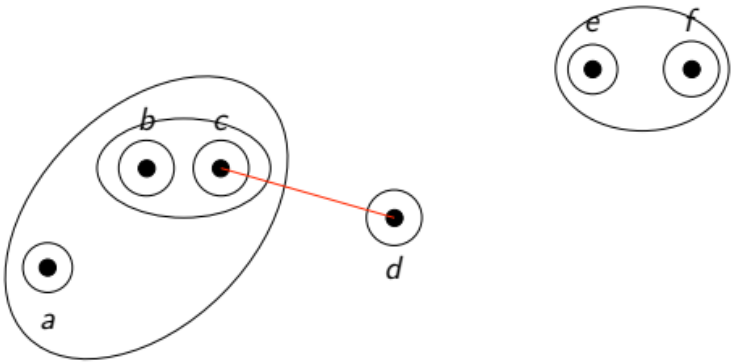
PROXIMITY MATRIX

	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



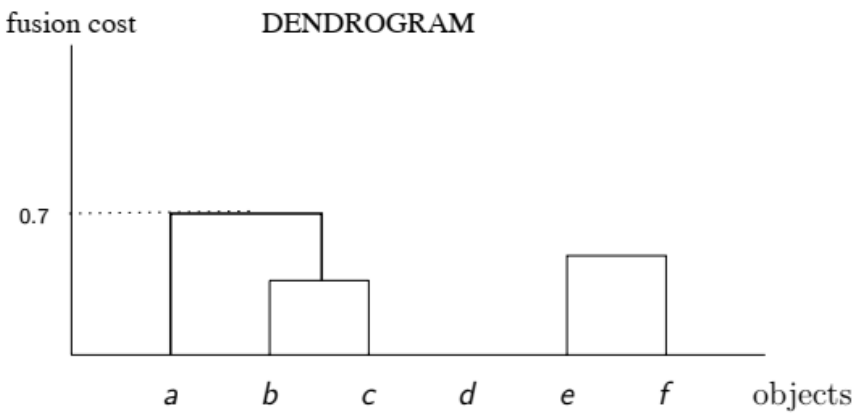
Building the dendrogram IV



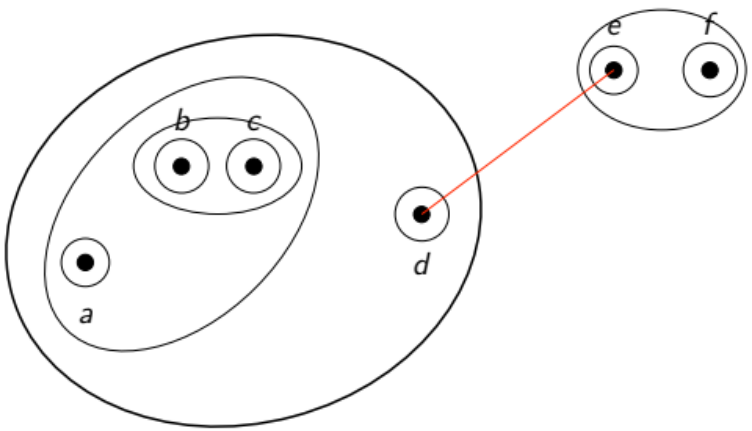
PROXIMITY MATRIX

	$\{a, b, c\}$	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



Building the dendrogram V

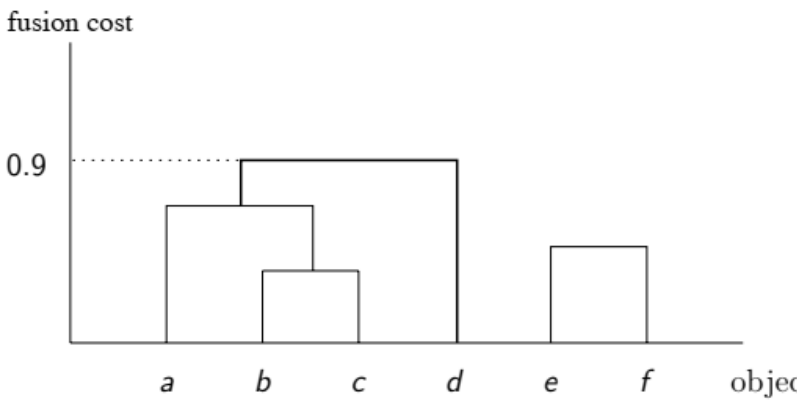


Next step is the final one and merges the clusters $\{a, b, c, d\}$ and $\{e, f\}$ with fusion cost 1.3

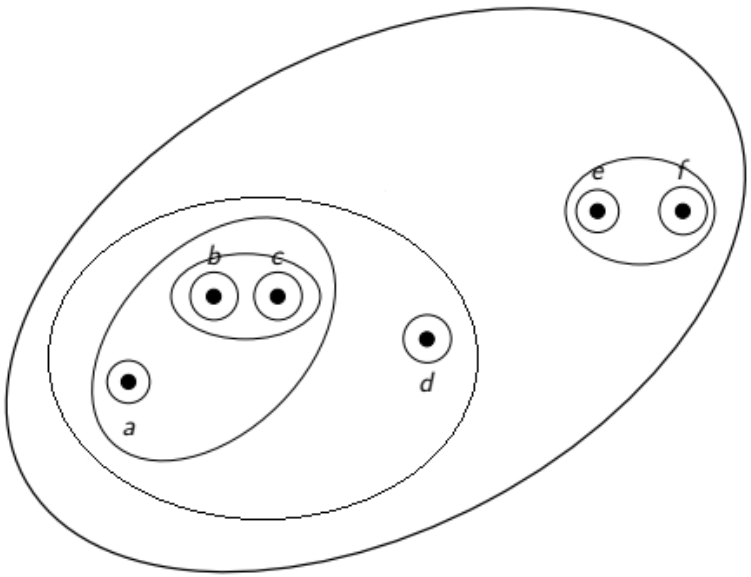
PROXIMITY MATRIX

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

DENDROGRAM



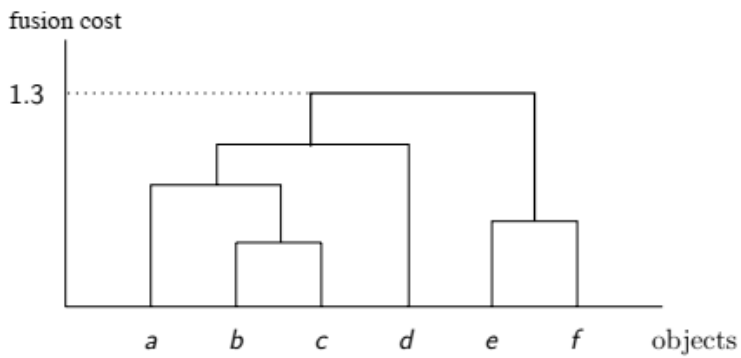
Building the dendrogram VI



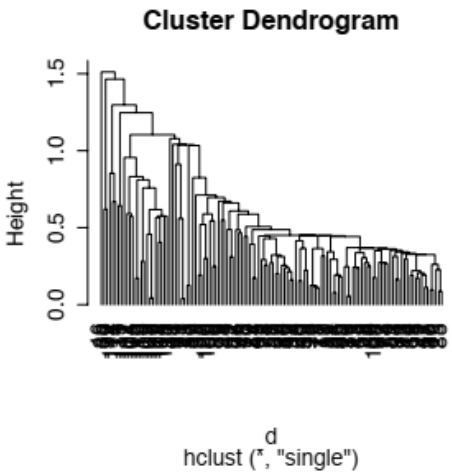
PROXIMITY MATRIX

EMPTY

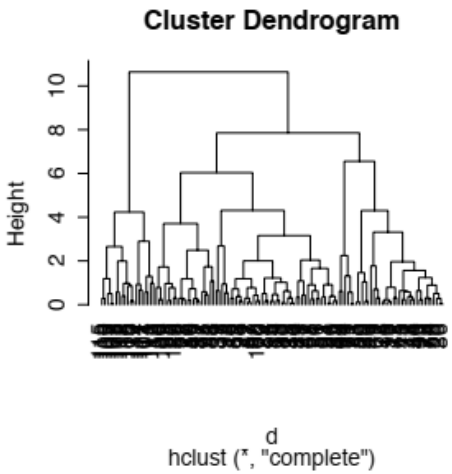
DENDROGRAM



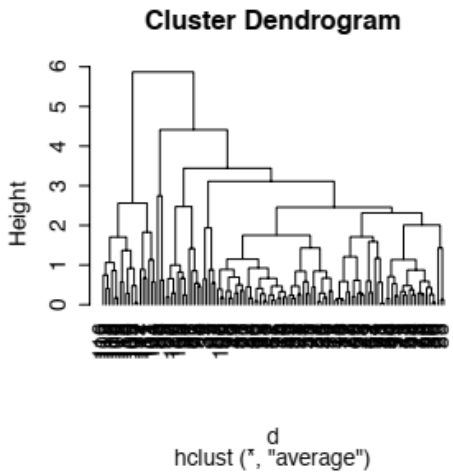
There exist the "right" dendrogram? I



contracting



dilating



conserving

There exist the "right" dendrogram? II

Some linkage methods is "dilating" with respect to the original distance, others are "contracting, still others are "coservative". How evaluate it? A possible is give a look to the cophenetic distance and corelation.

Cophenetic matrix

The cophenetic matrix (or distance) measures the degree of fit of a partition with respect to the original data set, assessing how well the dendrogramatic (ultrametric) distance preserve the original distances.

The cophenetic matrix is obtained by filling the (original) lower triangular distance matrix with the minimum merging distance obtained with the linkage method used

Finally, the cophenetic distance is compared with the original distances with a correlation coefficient: the larger the correlation, the better is the dendrogram representation

Cophenetic correlation

The cophenetic correlation is given by

$$Coph = \frac{\sum_{i<j}^{n}(d_{ij} - \bar{d})(c_{ij} - \bar{c})}{\sqrt{\sum_{i<j}(d_{ij} - \bar{d})^2 \sum_{i<j}(c_{ij} - \bar{c})^2}},$$

where d_{ij} and c_{ij} are the pairwise distance and the pairwise cophenetic value between units i and j, respectively.

The cophenetic correlation usually ranges between 0.6 and 0.95. Cophenetic correlations above .75 are considered good.

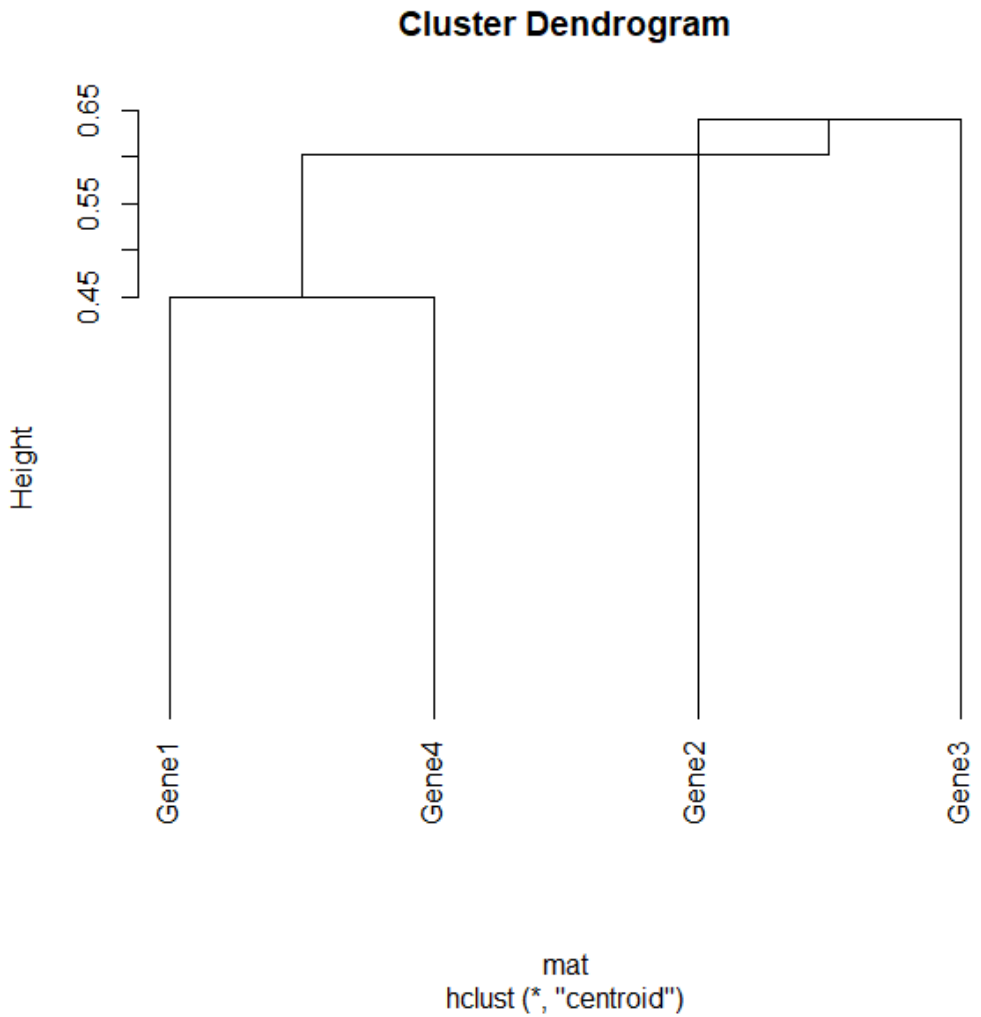
Sometimes the cophenetic correlation is expressed in terms of the Spearman correlation coefficient.

Inversion I

In both the centroid and median method the cophenetic measures may not verify the ultrametric property, giving rise to non-monotonic fusion distances with crossovers inversions in the dendrogram.

This result occurs when the distance from the union of two clusters, say r and s, to a third cluster is less than the distance between r and s. In this case, the path from a leaf to the root node takes some downward steps.

Inversion II



In this case, "gene2" and "gene 3" are joined into a new cluster, and the distance between this new cluster and cluster formed by "gene 1" and "gene 4" is less than the distance between "gene2" and "gene 3". The result is a nonmonotonic tree.

Sometimes changing the distance can solve the problem, sometimes it does not. It is better to change the linkage method.

How many clusters? part 1

Look at the within cluster similarity (ultrametric) at each stage. Cut the dendrogram at the level for which there is the maximum increase in terms of between distance joined.

Hiererchical cluster analysis: pros

- The dendrogram provides "taxonomical information" on the clusters
- 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

Hiererchical cluster analysis: 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 data sets since keeps track of a square matrix of order n (number of individuals)
- Dendrogram difficult to visualize and interpret for large data sets

Non-hierarchical clustering

K-means I

Given a data set $\mathbf{X}$, the clustering structure can be presented as a set of non-empty $K \geq 2$ subsets $\{C_1, \cdots, C_k, \cdots, C_K\}$ such that:

$$\begin{aligned} &\mathbf{X} = \bigcup_{k=1}^{K} C_k, \\ &C_k \bigcap C_{k'} = \varnothing, \quad \text{for} \quad k \neq k'. \end{aligned}$$

The idea behind K-means clustering is that a good clustering is one for which the within-cluster variation is as small as possible. This variation, for cluster C_k, is a measure $W(C_k)$ of the amount by which the observations within a cluster differ from each other. The goal is

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \sum_{k=1}^{K} W(C_k)$$

K-means II

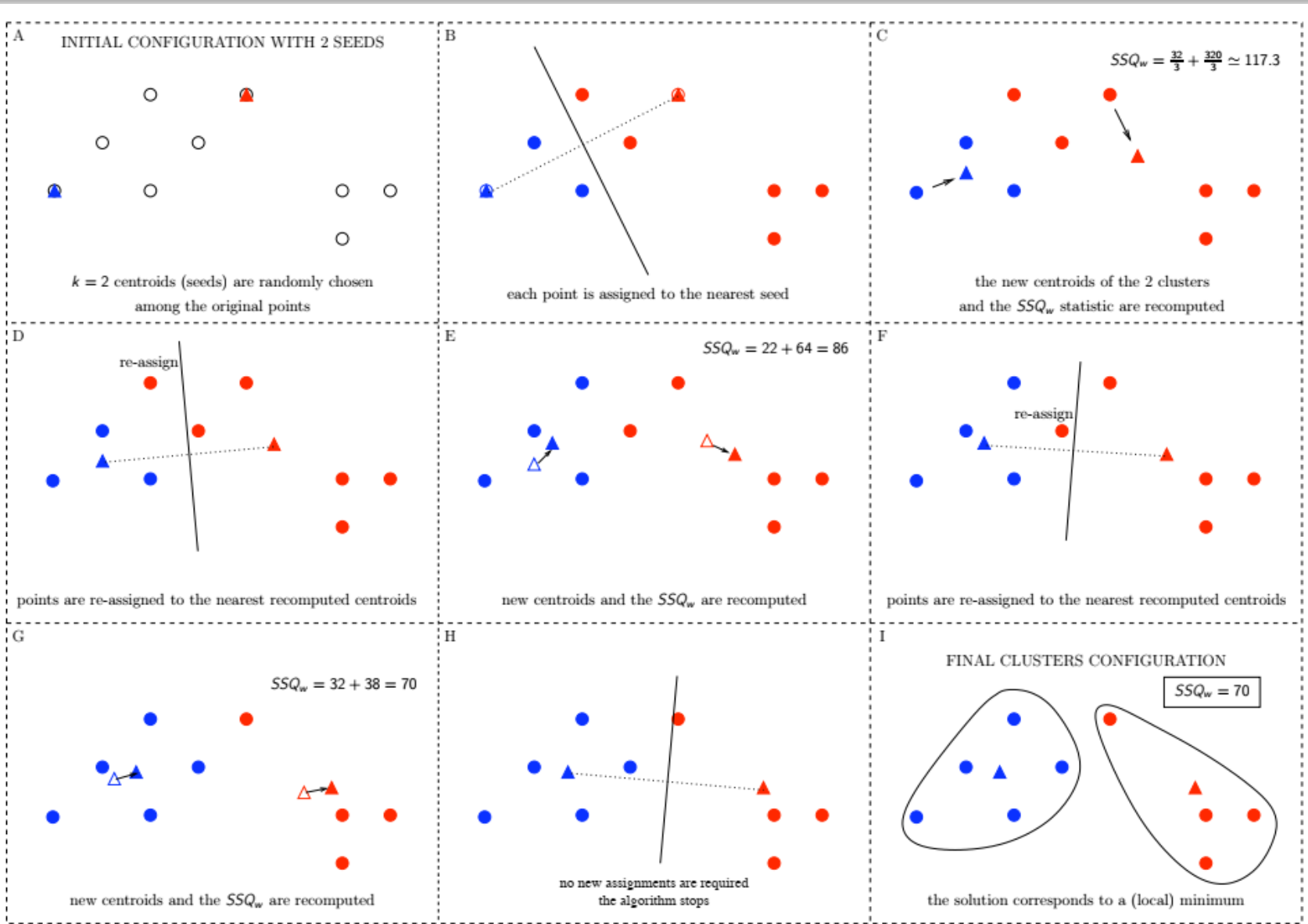
The K-means algorithm uses as variation measure the squared Euclidean distance. Other distance measures can be used, but in this case the name should be K-means-like algorithm.

1. Randomly assign each observation to one of the K clusters
2. Until the clustering assignment stop changing
   - For each of the K clusters, compute the *cluster centroid*, or bari-center, or cluster center.
   - Assign each observation to the cluster whose centroid is closest.

K-means III

When the algorithm finds a solution, there is no guarantee that the achieved solution is optimum. K-means algorithm can return a local optimum. For this reason, usually the procedure is repeated several time with different (random) starting points.

K-means IV


A
INITIAL CONFIGURATION WITH 2 SEEDS
$k = 2$ centroids (seeds) are randomly chosen among the original points
B
each point is assigned to the nearest seed
C
$SSQ_w = \frac{32}{3} + \frac{320}{3} \simeq 117.3$
the new centroids of the 2 clusters and the SSQ_w statistic are recomputed
D
re-assign
points are re-assigned to the nearest recomputed centroids
E
$SSQ_w = 22 + 64 = 86$
new centroids and the SSQ_w are recomputed
F
re-assign
points are re-assigned to the nearest recomputed centroids
G
$SSQ_w = 32 + 38 = 70$
new centroids and the SSQ_w are recomputed
H
no new assignments are required
the algorithm stops
I
FINAL CLUSTERS CONFIGURATION
$SSQ_w = 70$
the solution corresponds to a (local) minimum

K-means: summary

- The optimizing function (SSQw)within Sum of Squares) is always monotonic decreasing
- The number of iterations required to converge to an optimum is usually small, but
- Finding an optimal solution is NP-hard.
- Tend to form convex clusters. In particular cannot detect arbitrary cluster shapes.
- Nearby points can end in distinct classes.
- Assumes the squared Euclidean distance (Huygens theorem)
- It is not correct calling "k-means" algorithms that do not use Euclidean distance. It is better call them "k-means like" algorithms.

Non-hierarchical clusterig: K-medoids I

K-medoids clustering is a partitioning method commonly used in domains that require robustness to outlier data, arbitrary distance metrics, or ones for which the mean or median does not have a clear definition.

In the K-means algorithm, the center of the subset is the mean of measurements in the subset, called centroid. In the K-medoids algorithm, the center of the subset is a member of the subset, called *medoid*.

The K-medoids algorithm returns medoids which are the actual data points in the data set.

Non-hierarchical clusterig: K-medoids II

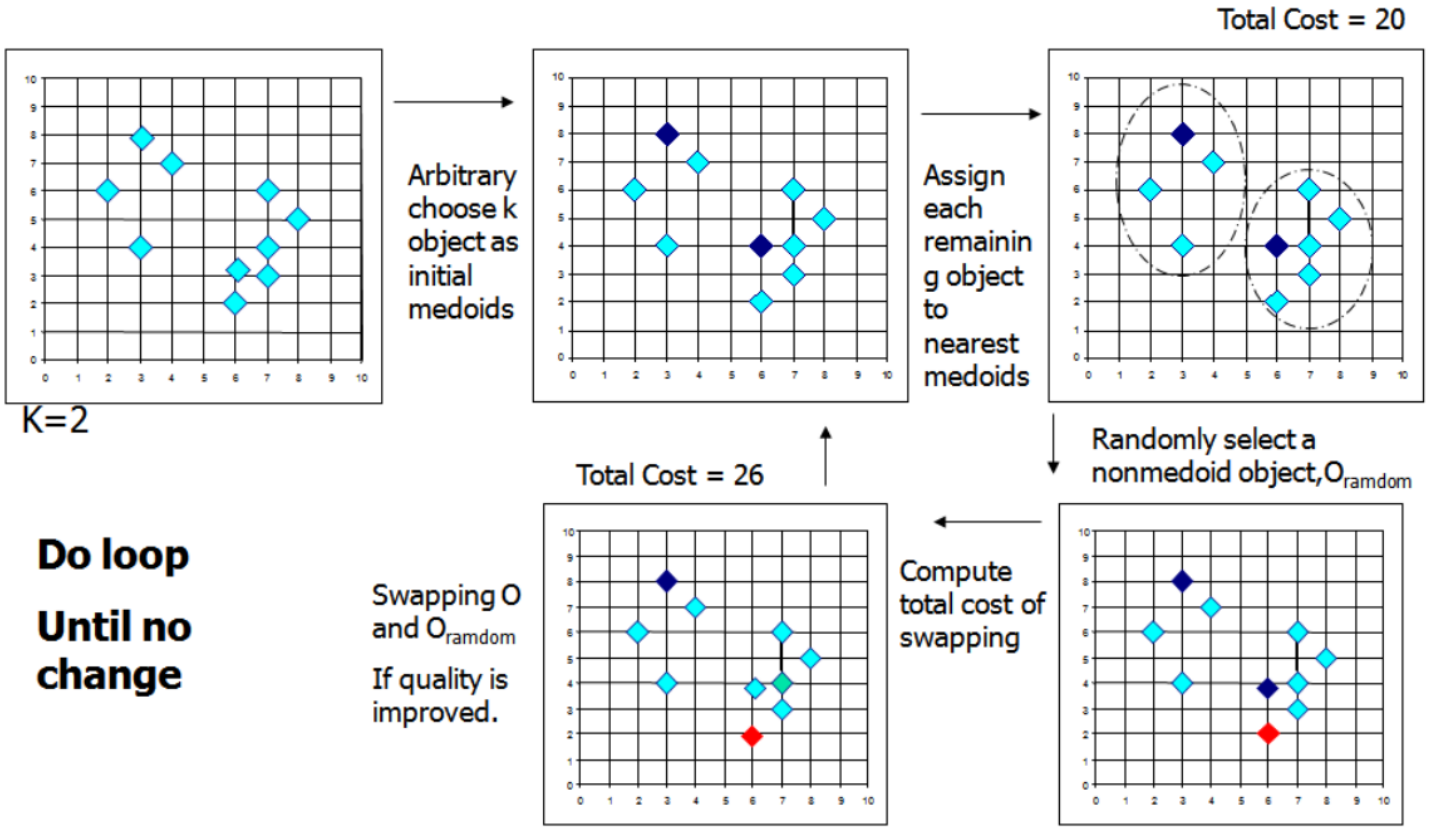
This allows you to use the algorithm in situations where the mean of the data does not exist within the data set.

K-medoids is useful for clustering categorical data where a mean is impossible to define or interpret.

K-medoids is useful when the distance among data points is really hard to compute and computationally time consuming (i.e., Dinamic Warping Distance for time series).

K-medoids: PAM algorithm

Figure from `https://slidewiki.org/deck/1374-1/partitioning-methods/slide/11560-2/1376-1:3;1372-1:2;11560-2:2/view`



Fuzzy partitions I

Fuzzy clustering methods determine a degree of membership, which indicates the degree to which objects belong to each cluster. In this way, objects that are on the boundary between different clusters are not forced to belong to one specific cluster, but they present a different degree of membership in each cluster.

Fuzzy partitions II

More formally, these methods partition the elements of $\mathbf{X}$ in K fuzzy clusters, with respect to some defined criterion, and they return both a set of cluster centers and a partition matrix of the following form

$$\mathbf{W} = \{w_{i,k}\}_{(n \times K)} \in [0,1]; \quad \sum_{k=1}^{K} w_{i,k} = 1 \quad \forall i \in \{1, \ldots, n\},$$

where $w_{i,k}$ represents the degree to which the element $\mathbf{x}_i$ belongs to the cluster C_k.
The most popular algorithm for performing fuzzy clustering is the fuzzy C-means algorithm

Fuzzy C-means I

The Fuzzy C-means algorithm aims to

$$\underset{C_1,\ldots,C_C}{\text{minimize}} \sum_{i=1}^{n}\sum_{j=1}^{C} w_{ic}^{m}\|X_i - C_c\|^2,$$

where $w_{ic} = \left(\sum_{k=1}^{C}\left(\frac{\|X_i - C_c\|}{\|X_i - C_k\|}\right)^{\frac{2}{m-1}}\right)^{-1}$ and $m \geq 1$.

Fuzzy C-means II

The quantity m is called *fuzzyfier* and governs the degree of fuzzyness.

If $m = 1$, then w_{ic} converges to zero or one. In this case *crisp* partitions are achieved.

If m is large, then w_{ic} will contain small membership values.

In general, m is set equal to 2.

Fuzzy C-means III

1. Assign randomly to each point coefficients for being in the C clusters
2. Repeat until convergence
   - Compute the cluster centers $C_c = \left(\sum_{i=1}^{n} w_{ic}^m x_i\right) / \left(\sum_{i=1}^{n} w_{ic}^m\right)$
   - Update w_{ic}

Probabilistic-distance clustering I

The Probabilistic-Distance (PD) clustering allows for a probabilistic allocation of cases to classes or clusters. It is a form of fuzzy clustering that is independent on the specification of fuzzifiers. It is based on the principle that probability and distance are inversely related

$$p_k(\mathbf{x})d_k(\mathbf{x}) = constant, \text{ depending on } \mathbf{x},$$

in which $d_k(\mathbf{x}) = d(\mathbf{x}_j, \mathbf{c}_k)$ is a distance measure between the j-th individual and the k-th cluster center and $p_k(\mathbf{x}) = p(\mathbf{x}_j \in k)$ denotes the probability of the j-th individual to belong to the k-th cluster, for $k = 1, \ldots, K$.

Probabilistic-distance clustering II

The membership probabilities are defined as

$$p_k(\mathbf{x}) = \frac{\prod_{j \neq k} d(\mathbf{x}_j, \mathbf{c}_k)}{\sum_{t=1}^{K} \prod_{j \neq t} d(\mathbf{x}_j, \mathbf{c}_t)}, \; k = 1, \cdots, K.$$

Probabilistic-distance clustering III

Given a data matrix $\mathbf{X}$ and K clusters. The problem is summarized as

$$\begin{array}{ll} \text{minimize} & \sum_{k=1}^{K} d_k(\mathbf{X}) p_k^2 \\ \text{subject to} & \sum_{k=1}^{k} P_k = 1 \\ & P_k \geq 0 \ \forall k = 1, \ldots, K \end{array}$$

Probabilistic-distance clustering IV

Repeat until convergence:

1. compute the the $n \times K$ matrix of distances $d(\mathbf{X})$ of each individuals from each cluster center;
2. update the cluster centers in this way:

$$c_k = \sum_{i=1}^{n} \left(\frac{u_k(\mathbf{x}_i)}{\sum_{j=1}^{n} u_k(\mathbf{x}_j)} \right) \mathbf{x}_i$$

where $u_k(\mathbf{x}_i) = \left(p_k(\mathbf{x}_i)^2\right) / \left(d_k(\mathbf{x}_i, c_k)\right)$.

Model-based clustering

Model-based clustering in a nutshell

Model-based clustering assumes that the data were generated by a model and tries to recover the original model from the data (the original mixture of distributions generating the data).

It is assumed that the distribution in the population in known.

Most of the time the EM (Expectation-Maximization) algorithm is used

EM approach I

Let X be the data set, let Z be the latent variables (the clusters), let Θ be a set of parameters. We have that

$$ln p(X|\Theta) = ln \left(\sum_Z p(X, Z|\Theta) \right).$$

Theoretically, the "complete" data set is X, Z, but we observe only X. The knowledge of Z is possible only through the posterior probability

$$p(Z|X, \Theta).$$

EM approach II

We can compute the expectation of the likelihood of the complete data set under the posterior (E step), and then maximize this value (M step).

- E step
  Use the current parameter Θ^* to compute the posterior $P(Z|X,\Theta^*)$, then find the expectation of the complete log-likelihood

$$Q(\Theta^*,\Theta) = \sum_Z p(Z|X,\Theta^*)\, ln(p(X,Z|\Theta)).$$

- M-step
  Update the estimate of Θ^*

$$\Theta^{new} = \underset{\Theta}{argmax}\, Q(\Theta,\Theta^*)$$

Mix of categorical and numerical data

Mix of categorical and numerical data

Hierarchical cluster analysis, K-medoids, K-means-like algorithms and PD clustering can be used also for data bearing both categorical and numerical data.

When it is not possible (or it is not convenient) recode categorical variables or discretize numerical variables, dissimilarity measures for mixed variables can be adopted.

Gower similarity index (Gower, 1971) I

Let X a data set with n objects and p variables of mixed type. The Gower similarity index is

$$g(x_i, x_j) = \frac{\sum_{v=1}^{p} w_v(x_i, x_j) s_v(x_i, x_j)}{\sum_{v=1}^{p} w_v(x_i, x_j)},$$

where $w_v(x_i, x_j)$ is the weight of the vth variable for the pair x_i, x_j and $s_v(x_i, x_j)$ is the similarity between x_i and x_j in v.

Gower similarity index (Gower, 1971) II

For continuous variables, $s_v(x_i, x_j) = \frac{|x_i - x_j|}{R_v}$, where R_v is the sample range of variable v.

For categorical variables, $s_v(x_i, x_j)$ is the simple matching coefficient.

For ordinal categorical variables, the original values are replaced by their associated rank values.

Once s_v is converted to dissimilarity with the option $1 - s_v$, any feasible clustering algorithms can be used.

K-prototypes (Huang, 1998) I

K-prototypes is a variant of K-means that is based on the weighted combination of the squared Euclidean distance for continuous variables and the matching distance for categorical variables. The distance measure is defined as

$$d(x_i, Q_k) = d_{cont}(x_i, Q_k) + \lambda d_{cat}(x_i, Q_k),$$

where Q_k is the centroid (or prototype) of cluster k, d_{cont} is the Euclidean distance to be computed for the numerical part of the data matrix, d_{cat} is the distance associated to the categorical part of the data matrix (1-simple matching coefficient), and λ is a user-defined weight of the significance of the entire group of categorical variables in cluster k.

K-prototypes (Huang, 1998) II

The cost function to be minimized is

$$\sum_{k=1}^{K}\sum_{i=1}^{n} z_{ik}\, d(x_i, Q_k),$$

where z_{ik} is an element of the $n \times K$ partition membership matrix Z.

Convex K-means (Modha and Spangler, 2003)

The convex K-means algorithm which considers a weighted combination of the squared Euclidean distance and the cosine distance for continuous and dummy-coded categorical variables, respectively.

The overall dissimilarity between an object and a cluster centroid is defined similar to K-prototypes, with the important difference that the weight λ is automatically determined inside the algorithm.

K-means for mixed data (Ahmad and Dey, 2007)

K-means for mixed data combines the squared Euclidean distance for continuous with a special distance for categorical variables, where the distance between two categories is computed as a function of their co-occurrence with other categories.

The overall dissimilarity between an object and a cluster centroid is defined similar to K-prototypes

Dimension reduction and cluster analysis I

Extant dimension reduction techniques all result in new numerical scores (coordinates) for the observations. Hence, an obvious approach is to perform a two-step analysis where cluster 9analysis is applied to the results of dimension reduction. Such an approach is often referred to as a "tandem analysis" (Hubert and Arabie, 1985).

- Principal Component Analysis, only numerical variables
- (Multiple) Correspondence analysis, only categorical variables
- For data reduction of mixed data, FAMD/PCAMIX, was originally proposed independently by several authors (de Leeuw and van Rijckevorsel, 1980; Hill and Smith, 1976; Kiers, 1991; Pagés, 2004). It can be seen as a compromise between PCA and Multiple Correspodence Analysis.

Dimension reduction and cluster analysis II

- Reduced K-means clustering (RKM) (De Soete and Carroll, 1994). In RKM the simultaneous dimension reduction and cluster analysis problem is tackled in such a way that the cluster allocation and dimension reduction maximizes the between variance of the clusters in the reduced space.
- Factorial K-means (FKM) (Vichi and Kiers, 2001).
- ...

Internal Validation Criteria

How many clusters? part 2 I

The number K of clusters must be known a-priori. If we do not know it, we can proceed with some tools, such as

- Silhouette
- GAP
- Scree-plot
- Calinski-Harabasz index
- Davies-Boudin criterion

Silhouette

Silhouette I

The silhouette value for each point is a measure of how similar that point is to points in its own cluster, when compared to points in other clusters. The silhouette value for the i-th point, S_i, is defined as

$$S_i = (b_i - a_i)/max(b_i, a_i)$$

where a_i is the average distance from the i-th point to the other points in the same cluster as i, and b_i is the minimum average distance from the i-th point to points in a different cluster, minimized over clusters.

Silhouette II

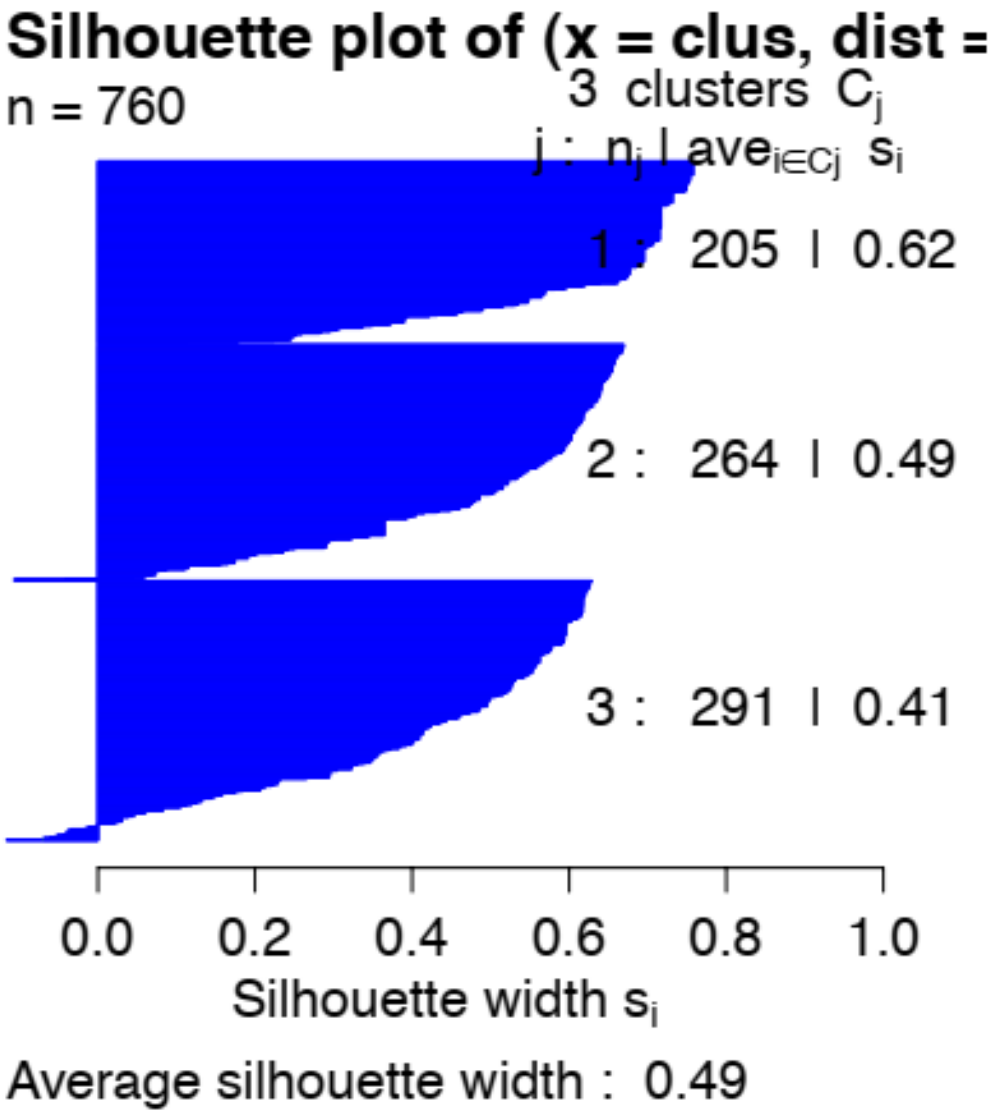
The silhouette value ranges from -1 to $+1$. A high silhouette value indicates that i is well-matched to its own cluster, and poorly-matched to neighboring clusters.
If most points have a high silhouette value, then the clustering solution is appropriate.
If many points have a low or negative silhouette value, then the clustering solution may have either too many or too few clusters.

Silhouette III

A way to choose the "optimal" K is perform the K-means with different levels of K, then compute the clusters silhouette for each solution and choose the clustering solution with the largest averaged silhouette.

The silhouette clustering evaluation criterion can be used with any distance metric.

Silhouette IV



GAP

GAP I

A common graphical approach to cluster evaluation involves plotting an error measurement versus several proposed numbers of clusters, and locating the *elbow* of this plot. The gap criterion formalizes this approach by estimating the "elbow" location as the number of clusters with the largest gap value. Therefore, under the gap criterion, the optimal number of clusters occurs at the solution with the largest local or global gap value within a tolerance range.

GAP II

The Gap statistic is

$$Gap_n(k) = E_n^* \{log(W_k)\} - log(W_k),$$

where E_n^* denotes the expected value under a sample size n from the reference distribution and W_k is the pooled within-cluster dispersion measurement

$$W_k = \sum_{r=1}^{k} \frac{1}{2n_r} D_r.$$

Above, n_r is the number of data points in cluster r, and D_r is the sum of the pairwise distances for all points in cluster r.

GAP III

The expected value $E_n^* \{log(W_k)\}$ is determined by Monte Carlo sampling from a reference distribution, and $log(W_k)$ is computed from the sample data.

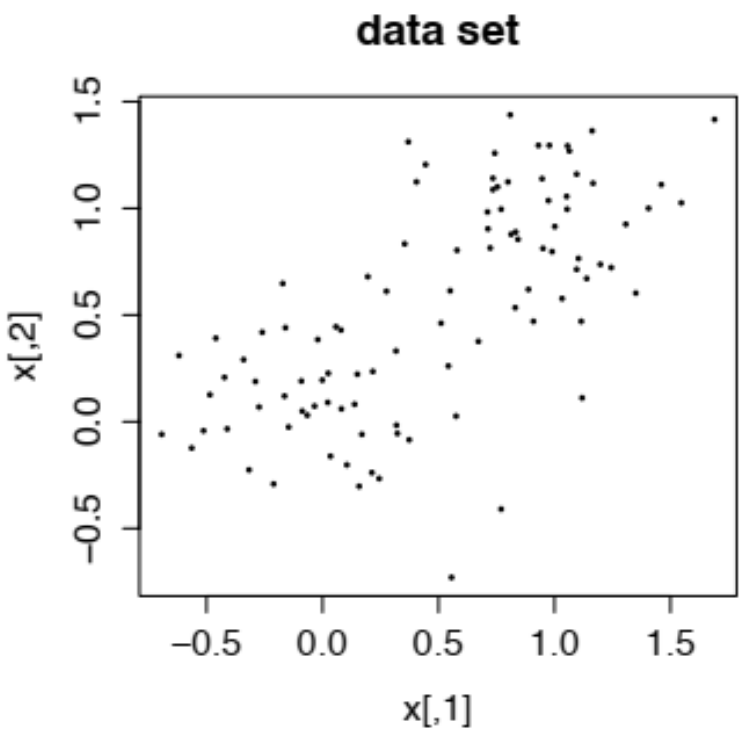
The GAP is then the "distance" between random data clustered in k clusters and the actual data.

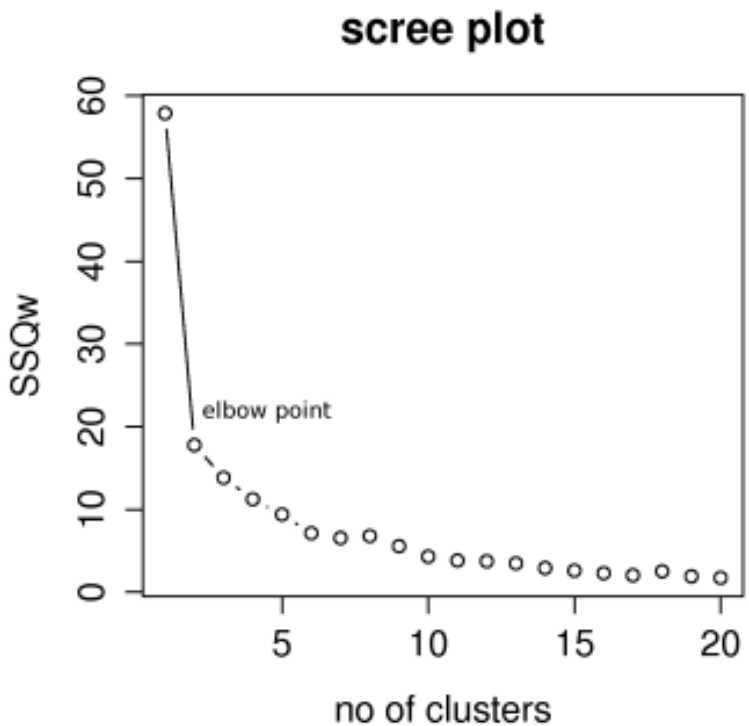
Scree-plot

Scree-plot I

A simple method consists in analysing the variation of the within cost (say SSQw) statistic, or equivalently, the percentage of explained variance SSQb/SSQt, against the number of clusters in a scree plot. This statistic is usually monotonically decreasing as the number of cluster increases. An elbow point in this plot indicating a high decrease in the SSQw statistic such that increasing the number of clusters only marginally improves (i.e., lowers) this statistics, may provide a good estimate for the number of clusters

Scree-plot II





Kalinski-Harabasz index

Kalinski-Harabasz index

The Calinski-Harabasz criterion is sometimes called the variance ratio criterion (VRC). The Calinski-Harabasz index is defined as

$$VRC_k = \frac{SSB}{SSW}\frac{N-K}{N-1}.$$

Well-defined clusters have a large between-cluster variance (SSB) and a small within-cluster variance (SSW). The larger the VRCk ratio, the better the data partition. To determine the optimal number of clusters, maximize VRCk with respect to k. The optimal number of clusters is the solution with the highest Calinski-Harabasz index value.
The Calinski-Harabasz criterion is best suited for k-means clustering solutions with squared Euclidean distances.

Davies-Bouldin criterion

Davies-Bouldin criterion

The Davies-Bouldin criterion is based on a ratio of within-cluster and between-cluster distances:

$$DB = \frac{1}{K}\sum_{k=1}^{K} max_{i \neq j} D_{i,j},$$

where $D_{i,j}$ is the within-to-between cluster distance ratio for the ith and jth clusters:

$$D_{i,j} = \frac{\bar{d}_i + \bar{d}_j}{d_{i,j}},$$

where $\bar{d}_i$ ($\bar{d}_j$) is the average distance between each point in the ith (jth) cluster and the centroid of the ith (jth) cluster, abd $d_{i,j}$ is the Euclidean distance between the centroids of the ith and jth clusters.

The optimal clustering solution has the smallest Davies-Bouldin index value.

External Validation Criteria

External vaidation criteria I

Let $\mathbf{X}$ the $(n \times p)$ data matrix, where n is the number of objects and p the number of variables. Let $\mathbf{P} = \{\mathbf{P}_1, , ..., \mathbf{P}_K\}$ be a K partition of the n objects. We say that two elements of $\mathbf{X}$, i.e. $(\mathbf{x}, \mathbf{x}')$ are paired in $\mathbf{P}$ if they belong to the same cluster. Let $\mathbf{P}$ and $\mathbf{Q}$ be two partitions of the objects set $\mathbf{X}$.

External vaidation criteria II

Let us define the following quantities:

- a: the number of pairs $(\mathbf{x}, \mathbf{x}')$ that are paired in $\mathbf{P}$ and in $\mathbf{Q}$;
- b: the number of pairs $(\mathbf{x}, \mathbf{x}')$ that are paired in $\mathbf{P}$ but not paired in $\mathbf{Q}$;
- c: the number of pairs $(\mathbf{x}, \mathbf{x}')$ that are not paired in $\mathbf{P}$ but paired in $\mathbf{Q}$;
- d: the number of pairs $(\mathbf{x}, \mathbf{x}')$ that are neither paired in $\mathbf{P}$ nor in $\mathbf{Q}$.

We will focus the attention only on the indices that can be built with these quantities.

External vaidation criteria III

Starting from these quantities, several indexes have been defined:

- Rand index (RI): $(a+d)/(a+b+c+d)$;
- Adjusted Rand index (ARI): $[2(ad-bc)]/[b^2+c^2+2ad+(a+d)(c+b)]$;
- Jaccard index: $a/(a+b+c)$;
- Fowlkes and Mallows index: $a/\sqrt{(a+b)(a+c)}$;
- Yule index: $[(ad)-(bc)]/[ab+cd]$;
- Dice's coefficient: $2a/(2a+b+c)$;
- ...

External validation criteria for fuzzy partitions I

There exist several adjustments of these indices for fuzzy partitions. The following is the Normalized Degree of Concordance index by Hullermeier:
Let $\mathbf{W} = \{\mathbf{W}_1, ..., \mathbf{W}_K\}$ be a fuzzy partition of the data matrix $\mathbf{X}$. Given any pair $(\mathbf{x}, \mathbf{x}') \in \mathbf{X}$, these membership vectors define a fuzzy equivalence relation on $\mathbf{X}$ in terms of similarity measure as:

$$E_{\mathbf{W}} = 1 - \|\mathbf{W}(\mathbf{x}) - \mathbf{W}(\mathbf{x}')\|,$$

where $\|\cdot\| = 0.5 \sum_{k=1}^{K} |\mathbf{x}_k - \mathbf{x}'_k|$.

External validation criteria for fuzzy partitions II

Consider now two fuzzy partitions, $\mathbf{P}$ and $\mathbf{Q}$. The *degree of concordance* is:

$$conc(\mathbf{x}, \mathbf{x}') = 1 - \|E_{\mathbf{P}}(\mathbf{x}, \mathbf{x}') - E_{\mathbf{Q}}(\mathbf{x}, \mathbf{x}')\| \in [0, 1],$$

and the *degree of discordance* is:

$$disc(\mathbf{x}, \mathbf{x}') = \|E_{\mathbf{P}}(\mathbf{x}, \mathbf{x}') - E_{\mathbf{Q}}(\mathbf{x}, \mathbf{x}')\|.$$

External validation criteria for fuzzy partitions III

A distance measure is defined by the normalized sum of concordant pairs:

$$d(\mathbf{P}, \mathbf{Q}) = \frac{\sum_{(\mathbf{x},\mathbf{x}')\in\mathbf{X}} \|E_{\mathbf{P}}(\mathbf{x}, \mathbf{x}') - E_{\mathbf{Q}}(\mathbf{x}, \mathbf{x}')\|}{n(n-1)/2}.$$

Finally, the normalized degree of concordance is defined as

$$R_E(\mathbf{P}, \mathbf{Q}) = 1 - d(\mathbf{P}, \mathbf{Q}),$$

External validation criteria for fuzzy partitions IV

The Adjusted Concordance Index is the fuzzy variant of the Adjusted Rand Index. Let $\mathbf{P}$ and $, \mathbf{Q}$ two probabilistic (fuzzy) partitions. The ACI is defined as

$$ACI = \frac{R_E(\mathbf{P}, \mathbf{Q}) - \bar{R}_E(\mathbf{P}, \mathbf{Q})}{1 - \bar{R}_E(\mathbf{P}, \mathbf{Q})},$$

where $\bar{R}_E(\mathbf{P}, \mathbf{Q})$ is the mean value of the R_E over all permutations of (say) $\mathbf{P}$ keeping fixed $\mathbf{Q}$.
When determining all of the permutations is not practical, for example, when $n > 20$, the expected value is estimated by taking into account a very large number h of randomly selected permutations of the total $n!$ permutations.

Non-hierarchical clustering: pros

Non-hierarchical cluster analysis

- Can reallocate an individual that was misplaced in its cluster
- It is generally Computationally efficient (e.g. K-means)
- Can improve the objective function obtained with some hierarchical methods (e.g., K-means and Ward)

Non-hierarchical clustering: cons

- The number of clusters (or some other parameters, for model-based clustering) has to be known (or estimated) a priori
- No taxonomic type of relationship between clusters is obtained and no dendrogram is produced
- Some methods work only with "geometric" data and may require the euclidean distance