Master-MIRI
Topics on Optimization and Machine Learning (TOML)

José M. Barceló Ordinas
Departament d'Arquitectura de Computadors (UPC)

Unsupersived Learning : Definition…

- **Unsupervised learning:** the algorithm learns patterns from untagged data, e.g. clustering algorithm (k-means) or PCA (reduction of dimensions), or anomaly detection among others,

- **Objective:** we have a vector for every measurement *i*, we observe vector $\mathbf{x_i}=(x_1, \ldots,x_M)$, but we don't have a response t, so we can not supervised our analysis using a regression or classification method. However we can do other things such as find clusters in our data or reduce feature dimensions,

- **Examples of algorithms:**
  - **Clustering:** techniques for findings subgroups, or clusters, in the data set, examples are K-means, hierarchical clustering, mixture of Gaussians, and expectation-maximization algorithm,
  - **Principal Components Analysis (PCA), explained by Jorge in Topic's 3 section:** refers to the process by which principal components are computed, and the subsequent use of these components in understanding the data,

K-means: Definition…

- We define the number of desired clusters k, and assign each observation $\mathbf{x}_i$, i=1,…N to one of the clusters. If there are K clusters, then the following properties are satisfied:
  1. $C_1 \cup C_1 \cup \ldots \cup C_K = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, it is to say, each observation belongs to at least one cluster
  2. $C_k \cap C_j = \emptyset$ for all $k \neq j$, it is to say, clusters do not overlap, or observations belong to only one cluster

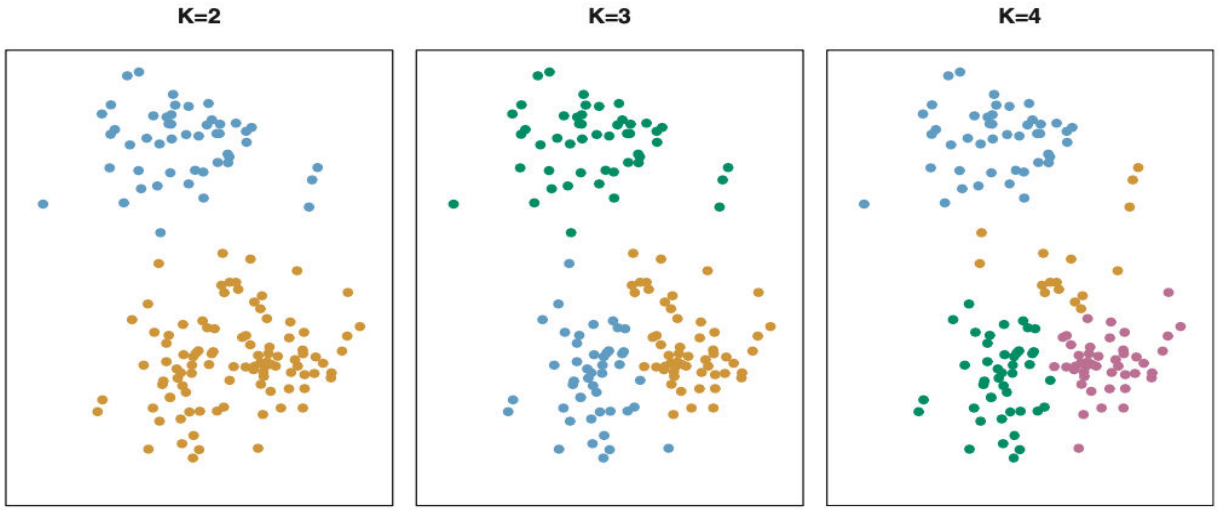


FIGURE 10.5. *A simulated data set with 150 observations in two-dimensional space. Panels show the results of applying K-means clustering with different values of K, the number of clusters. The color of each observation indicates the cluster to which it was assigned using the K-means clustering algorithm. Note that there is no ordering of the clusters, so the cluster coloring is arbitrary. These cluster labels were not used in clustering; instead, they are the outputs of the clustering procedure.*

K-means: algorithm (from JWHT "James,Witten,Hastie,Tibshirani book") …

- Objective is to minimize the **within-cluster variation**, $W(C_k)$, defined as a measure by which the observations within a cluster differ from each other. Then the idea is that we want to partition the observations in K clusters, such that the sum of within-cluster variations is as small as possible:

  $$\text{minimize}_{C1,\ldots,Ck}\ \{\ \Sigma_{k=1,\ldots,K}\ W(C_k)\ \}$$

- Within-cluster variation $W(C_k)$ can be defined in different ways, the most common as an Euclidean distance:

  $$W(C_k)= 1/|C_k|\ \Sigma_{x\neq y\in Ck}\ ||\mathbf{x}-\mathbf{y}||^2$$ where $|C_k|$ is the size of the cluster k.

  Using equality $\Sigma_{x\neq y\in Ck}\ ||\mathbf{x}-\mathbf{y}||^2 = \Sigma_{x\in Ck}\ ||\mathbf{x}-\boldsymbol{\mu}_k||^2$. This is equivalent to:

  $$W(C_k) = 1/|C_k|\ \Sigma_{x\in Ck}\ ||\mathbf{x}-\boldsymbol{\mu}_k||^2$$ where μ_k is the mean (**centroid**) of points of the cluster C_k,

- Defining a $J= \Sigma_{k=1,\ldots,K}\ W(C_k)= \Sigma_{k=1,\ldots,K}\ 1/|C_k|\Sigma_{x\neq y\in Ck}||\mathbf{x}-\mathbf{y}||^2$ as the **cost function**, the final optimization algorithm is given by minimizing the sum of all within-cluster variation:

  (OPT) $\text{minimize}_{C1,\ldots,Ck}\{\Sigma_{k=1,\ldots,K}\ 1/|C_k|\Sigma_{x\neq y\in Ck}||\mathbf{x}-\mathbf{y}||^2\}= \text{minimize}_{C1,\ldots,Ck}\{\ \Sigma_{k=1,\ldots,K}\ 1/|C_k|\Sigma_{x\in Ck}||\mathbf{x}-\boldsymbol{\mu}_k||^2\}$

K-means: final algorithm (JWHT) …

(OPT) $\text{minimize}_{C1, \ldots, Ck} \{ \sum_{k=1,\ldots,K} 1/|C_k| \sum_{x,y \in Ck} ||\mathbf{x}-\boldsymbol{\mu}_k||^2 \}$

- For minimizing this optimization algorithm we need a method to partition the observations into K clusters, and since there exists K^n ways to partition n observations into K clusters, this problem is very difficult to be solved,
- **Solution:** a greedy algorithm that obtains a local minimum.

(ALG)

1. Initial cluster assignment for the observations: randomly assign a number, 1, …, K to each of the observations.

2. Iterate until the cluster assignments stop changing:

   a) For each of the K clusters, compute the cluster centroid. The *k*th cluster centroid $\boldsymbol{\mu}_k$ is the vector of the M features means for the observations in the *k*th cluster, $\boldsymbol{\mu}_k = (\mu_{k1}, \ldots, \mu_{kM})$,

   b) Assign each observation to the cluster whose centroid is closest (defined using Euclidean distance).

- K-means: final algorithm (JWHT) …

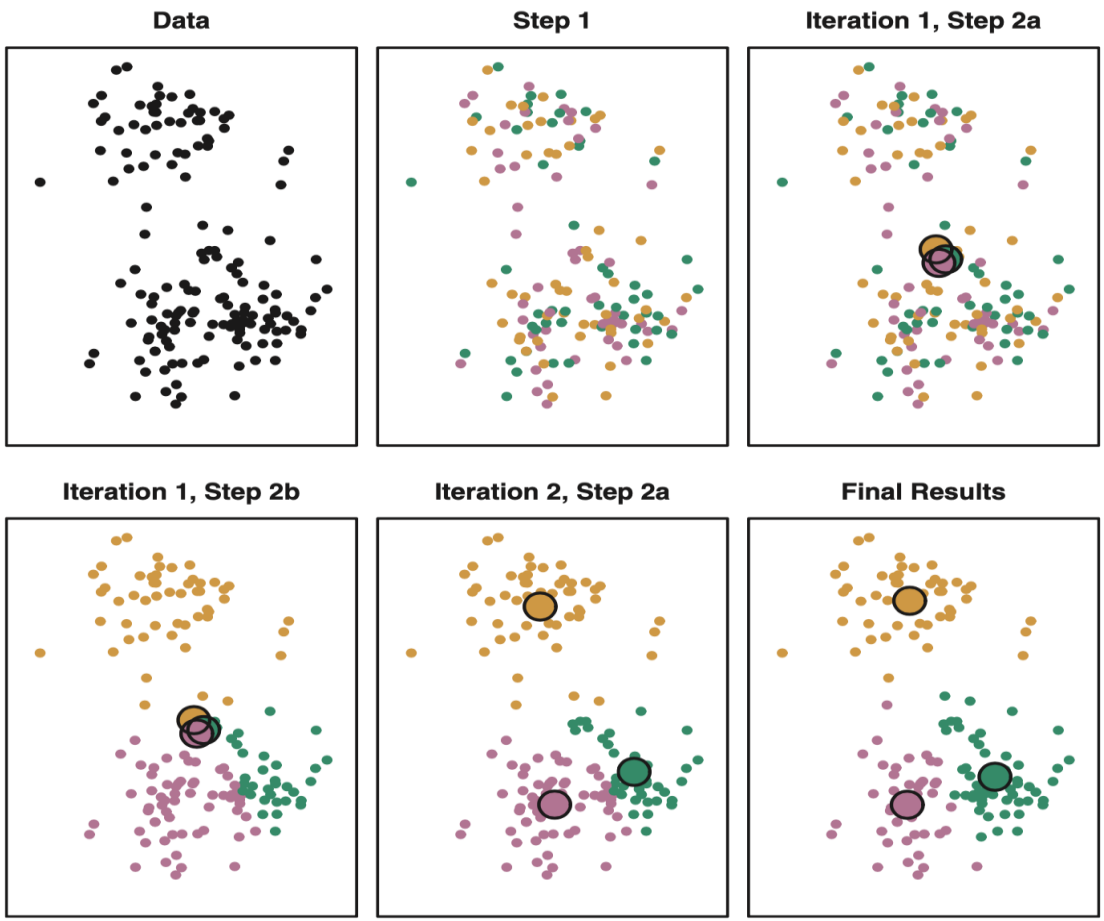
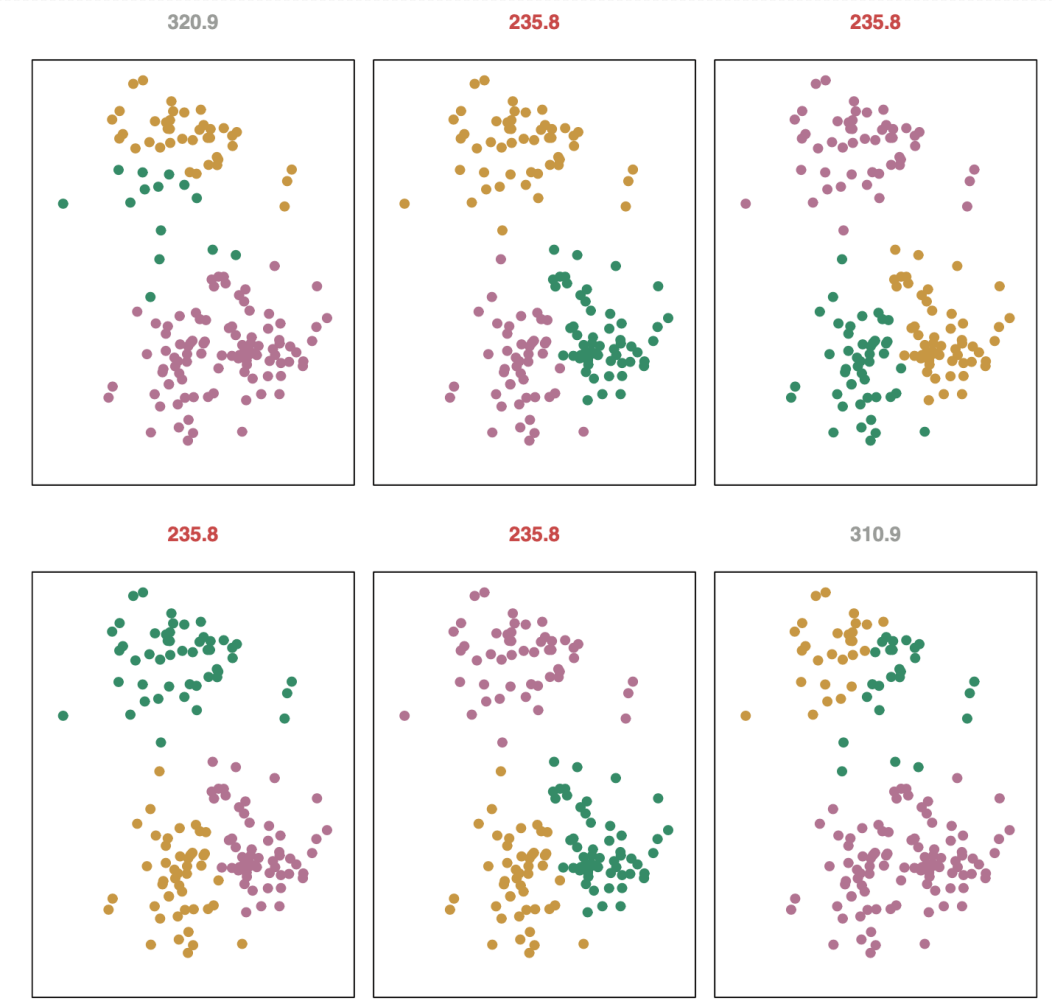


FIGURE 10.6. *The progress of the K-means algorithm on the example of Figure 10.5 with K=3.* Top left: *the observations are shown.* Top center: *in Step 1 of the algorithm, each observation is randomly assigned to a cluster.* Top right: *in Step 2(a), the cluster centroids are computed. These are shown as large colored disks. Initially the centroids are almost completely overlapping because the initial cluster assignments were chosen at random.* Bottom left: *in Step 2(b), each observation is assigned to the nearest centroid.* Bottom center: *Step 2(a) is once again performed, leading to new cluster centroids.* Bottom right: *the results obtained after ten iterations.*

K-means: final algorithm (JWHT) …

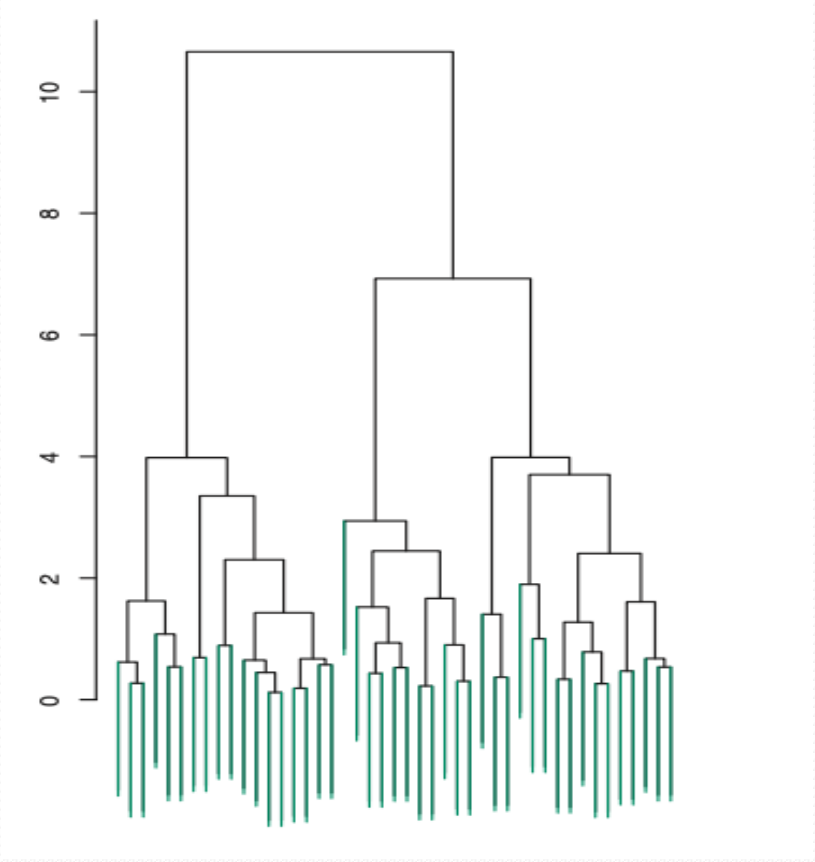
- **Be careful:** since point 1 of the algorithm initiates randomly the assignment of observations to clusters, the final result (is a local minimum and not a global minimum) depends on the initial assignment. Then: run the algorithm multiple times with different initial assignments and choose that one with minimum cost function (*best solution*).



K-means performed 6 times with different initial random assignments, with K=3. The number above is the result of the cost function ($\Sigma_{k=1,\dots,K}\ W(C_k)$). Four of the six result in the same assignment, with minimum cost, two of them end with highest cost , and different assignment.

Hierarchical clustering (JWHT): main idea …

- Disadvantage of k-means (how many clusters k do we have to use?),
- hierarchical clustering does not commit to a pre-fixed number of clusters k,
- **dendograms:** a tree-based representation of the observations,



Example that corresponds to a data set with N=48 observations. Each observations ends with a leaf in the tree.

Idea: observations that fuse near the leaves, are similar (same cluster), while those ones that fuse at higher level, are dissimilar, and possible belong to different clusters,

The height of this fusion, measured as vertical axe, give us how different the two observations are.

Hierarchical clustering (JWHT): dendogram interpretation ...

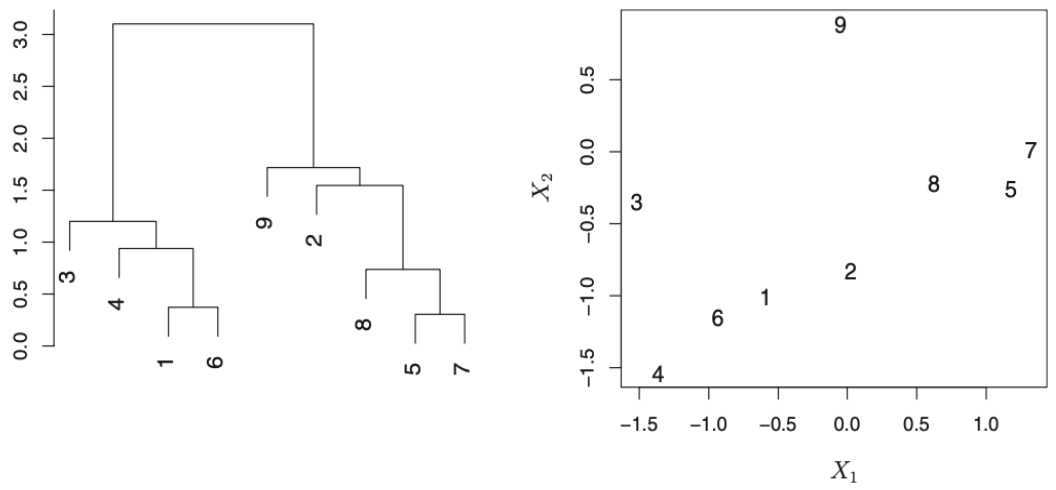


FIGURE 10.10. *An illustration of how to properly interpret a dendrogram with nine observations in two-dimensional space.* Left: *a dendrogram generated using Euclidean distance and complete linkage. Observations* 5 *and* 7 *are quite similar to each other, as are observations* 1 *and* 6. *However, observation* 9 *is* no more similar to *observation* 2 *than it is to observations* 8, 5, *and* 7, *even though observations* 9 *and* 2 *are close together in terms of horizontal distance. This is because observations* 2, 8, 5, *and* 7 *all fuse with observation* 9 *at the same height, approximately* 1.8. Right: *the raw data used to generate the dendrogram can be used to confirm that indeed, observation* 9 *is no more similar to observation* 2 *than it is to observations* 8, 5, *and* 7.

There are 2^{N-1} possible reorderings of the dendogram, with *N* the number of leaves.

Then, we **can't** get conclusions about similarity of two observations based on the horizontal axis (observation 2 and 9 are so dissimilar as observation 9 and 5 or 9 and 7).

We draw conclusions of similarities of two observations based on vertical axes, where branches containing those two observations first are fused.

- **Hierarchical clustering (JWHT): main idea …**
  - **dendograms:** a tree-based representation of the observations,
  - for creating clusters we make a cut in the vertical axe. Each branch forms a cluster. Cutting at 9 forms k=2 clusters, while cutting at 5 forms K=3 clusters. Cutting at 0 forms N clusters. The height in the cut has the same meaning as K in the K-means algorithm.

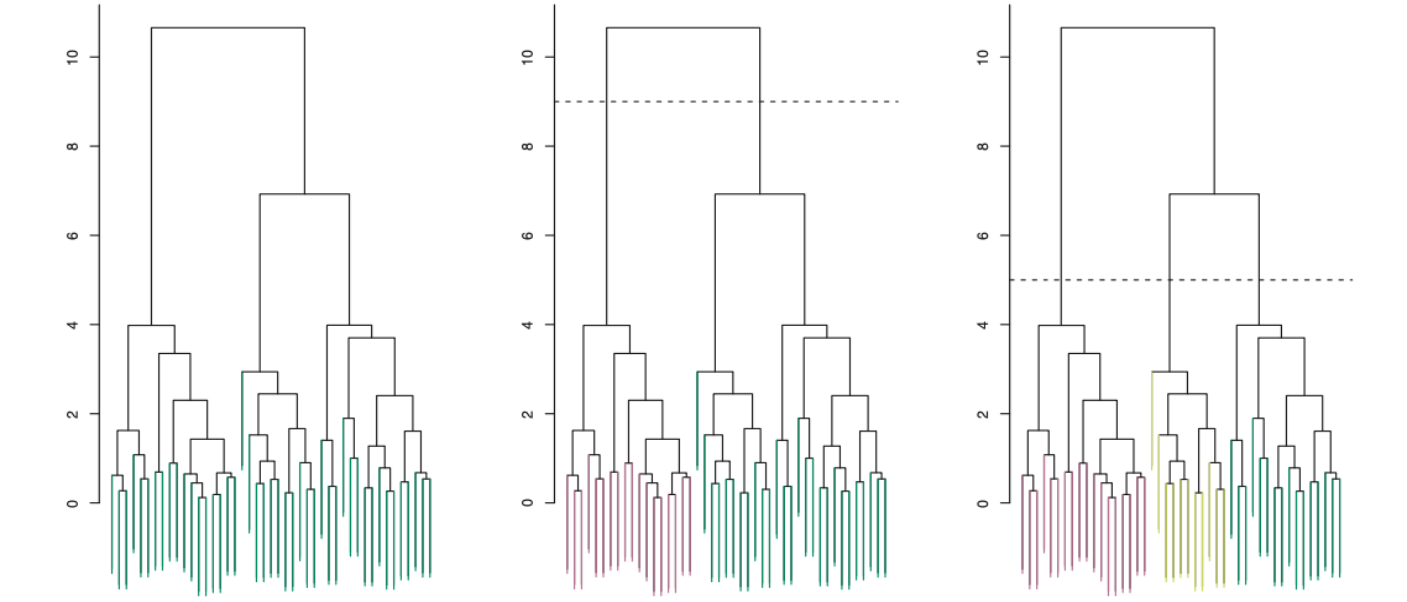


FIGURE 10.9. Left: *dendrogram obtained from hierarchically clustering the data from Figure 10.8 with complete linkage and Euclidean distance.* Center: *the dendrogram from the left-hand panel, cut at a height of nine (indicated by the dashed line). This cut results in two distinct clusters, shown in different colors.* Right: *the dendrogram from the left-hand panel, now cut at a height of five. This cut results in three distinct clusters, shown in different colors. Note that the colors were not used in clustering, but are simply used for display purposes in this figure.*

- **Hierarchical clustering (JWHT): obtaining the dendogram …**
  - We first choose a dissimilarity measure (e.g. Euclidean distance, correlation-based distance),
  - **Bottom-up approach:** the N observations form N clusters. Now form N-1 clusters by fusing the two most similar clusters (the two most similar observations), so we have 1 cluster of two observations and a N-2 clusters of a single observation. Iterate until there is a single cluster.
  - We have a definition of dissimilarity between two points, but what happens when we have to compare clusters with more than one observation ? → extend the definition to groups of observations,
  - **Linkage:** defines the dissimilarity between groups of observations, with four groups of linkage (complete, average, single and centroid),

Linkage	*Description*
Complete	Maximal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *largest* of these dissimilarities.
Single	Minimal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *smallest* of these dissimilarities. Single linkage can result in extended, trailing clusters in which single observations are fused one-at-a-time.
Average	Mean intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *average* of these dissimilarities.
Centroid	Dissimilarity between the centroid for cluster A (a mean vector of length p) and the centroid for cluster B. Centroid linkage can result in undesirable *inversions*.

Average, complete and single are used by statisticians, while centroid are used by genomics.

Average and complete give better results than single (better balanced dendograms).

Centroid suffers from inversion (two clusters fused below the height of individual heights).

- **Hierarchical clustering (JWHT): obtaining the dendogram …**
  - **Linkage:** defines the dissimilarity between groups of observations, with four groups of linkage (complete, average, single and centroid),

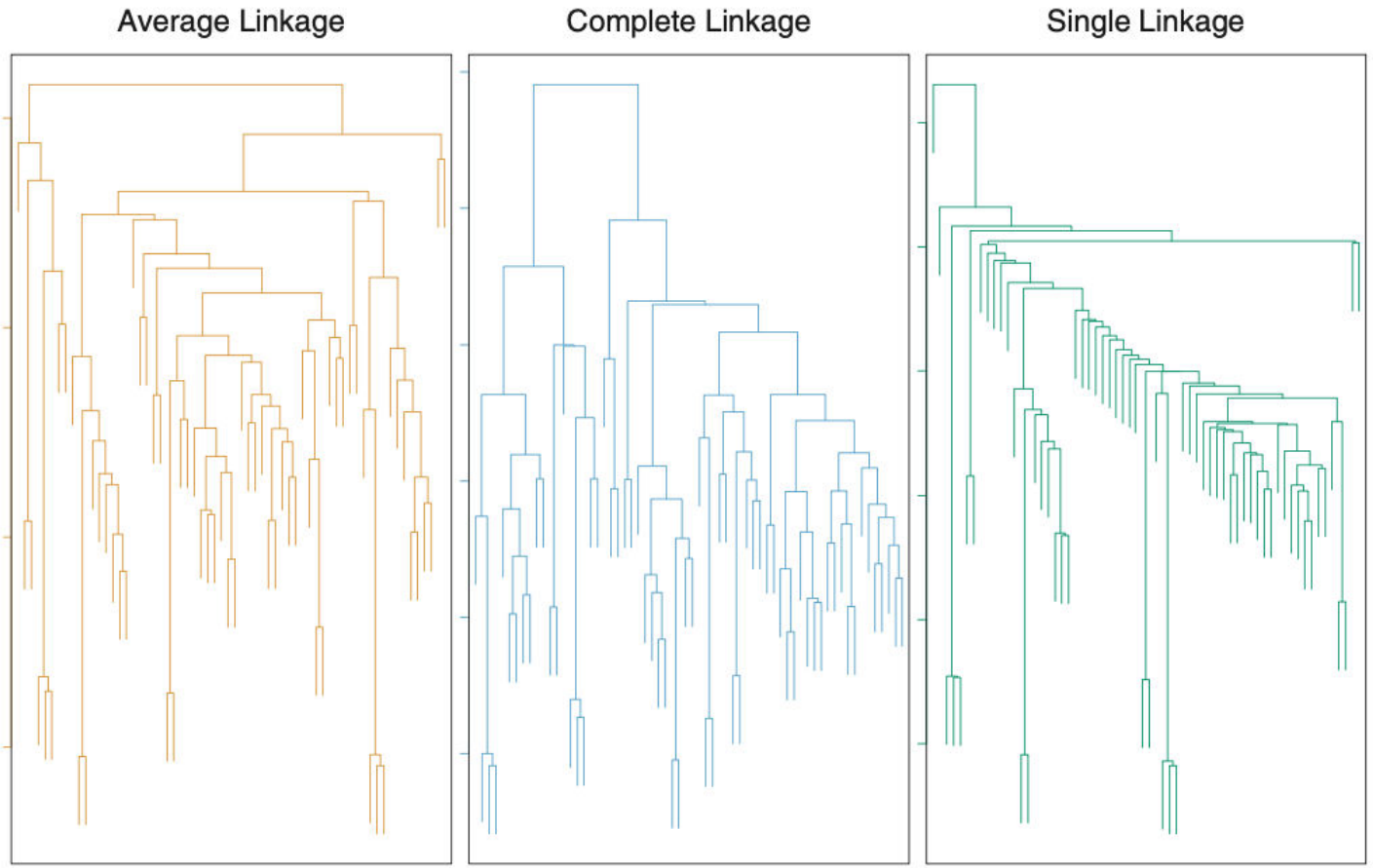


FIGURE 10.12. *Average, complete, and single linkage applied to an example data set. Average and complete linkage tend to yield more balanced clusters.*

Hierarchical clustering (JWHT): obtaining the dendogram …

- Algorithm for obtaining the dendogram,

(ALG)

1. Begin with N observations and a measure (e.g. Euclidean distance), and obtain the N(N-1)/2 pairwise dissimilarities. Treat each observation as a cluster.

2. For *i*=N, N-1, ..., 2

   a) Examine all pairwise inter-cluster dissimilarities among the *i* clusters and identify the pair of clusters that are least dissimilar (that is, most similar). Fuse these two clusters. The dissimilarity between these clusters give the height in the dendogram at which the fusion took place,

   b) Compute the new pairwise inter-cluster dissimilarities among the *i*-1 remaining clusters.

- **Hierarchical clustering (JWHT): obtaining the dendogram …**

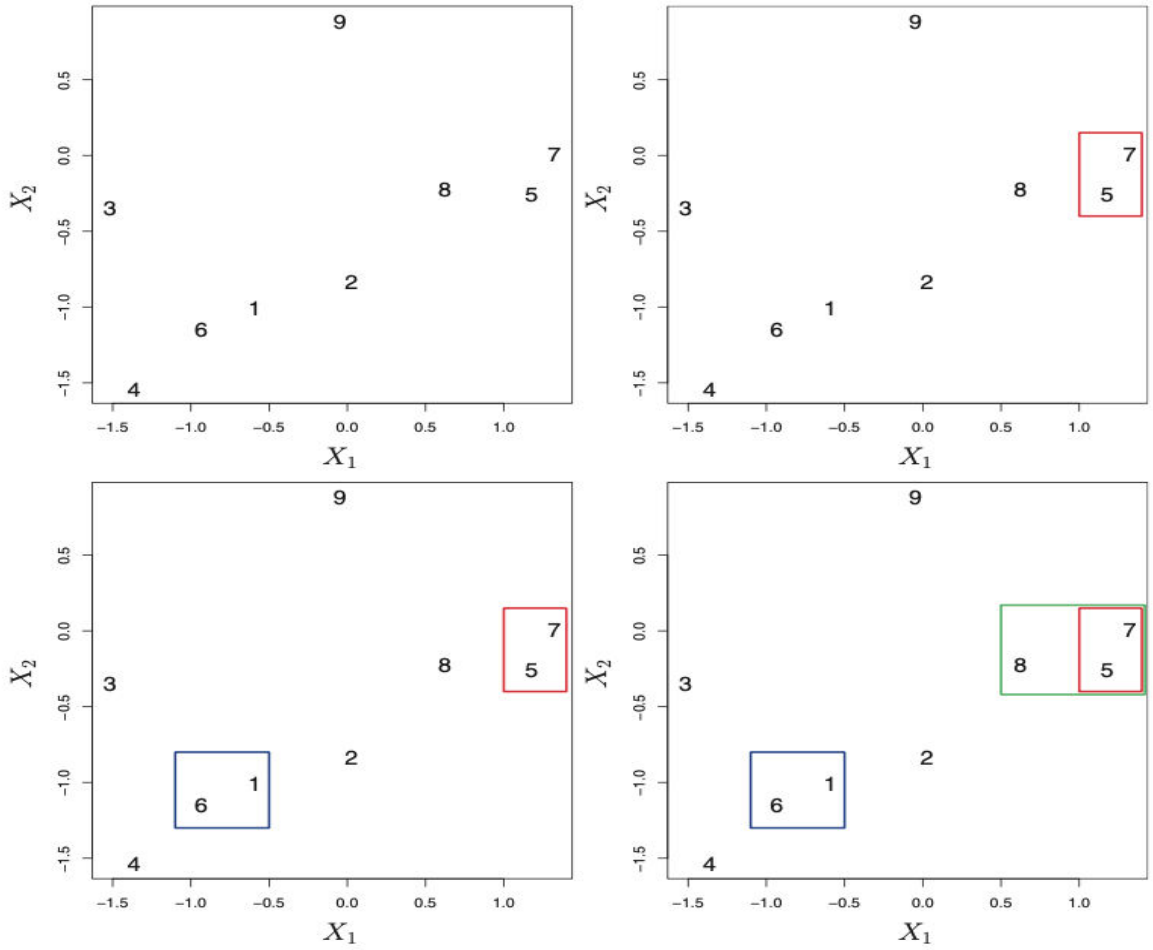


FIGURE 10.11. *An illustration of the first few steps of the hierarchical clustering algorithm, using the data from Figure 10.10, with complete linkage and Euclidean distance.* Top Left: *initially, there are nine distinct clusters,* $\{1\}, \{2\}, \ldots, \{9\}$. Top Right: *the two clusters that are closest together,* $\{5\}$ *and* $\{7\}$, *are fused into a single cluster.* Bottom Left: *the two clusters that are closest together,* $\{6\}$ *and* $\{1\}$, *are fused into a single cluster.* Bottom Right: *the two clusters that are closest together using* complete linkage, $\{8\}$ *and the cluster* $\{5, 7\}$, *are fused into a single cluster.*

- **K-means or hierarchical clustering (JWHT) ?**
  - Depends on the data. Sometimes the hierarchical clustering does not capture the true clusters and K-means performs better.
  - Let's assume data that splits men and women (50% of data of each class), and that these data are taken from Americans, Japanese and French. Maybe hierarchical clustering makes two divisions (gender) but the data is best represented by 3 clusters (nationality)

- **Decisions (JWHT) in K-means and hierarchical clustering ...**
  - Should observations or features be standardize in some way (e.g. zero mean ands scale to standard deviation) ?
  - How to choose K in K-means ?
  - In the case of hierarchical clustering: (i) which dissimilarity function use? (ii) what type of linkage use ?, (iii) where do we cut the dendogram to form clusters ?

  Each of these decisions have a huge impact in the results. **Solution:** try several and take the one that gives best results (e.g. best interpretation),
  - K-means and hierarchical clustering assign each data point to a cluster. Situations in which some data points are not appropriate to be assigned to a cluster (outliers) → use of *soft clustering* (mixture models).

K-means (Bishop): a different interpretation

- Let us rename the cost function as $J= \sum_{n=1,\dots,N} \sum_{k=1,\dots,K} r_{nk}||\mathbf{x_n}-\boldsymbol{\mu_k}||^2$ - called **distortion measure** - that represents the sum of square distance of each point to its assigned vector $\boldsymbol{\mu_k}$,
- variable $r_{nk}=\{0,1\}$ is a binary indicator variable that takes values $r_{nk}=1$ if observation *n* belongs to cluster k, and 0 otherwise.
- **Goal:** find r_{nk}, and $\boldsymbol{\mu_k}$ such as minimizes the distortion measure J.
- **Solution:** iterative process with two steps that corresponds to successive optimizations with respect to r_{nk} and $\boldsymbol{\mu_k}$. Choose some initial values to $\boldsymbol{\mu_k}$.
  1. **Step E:** minimize J with respect r_{nk} keeping $\boldsymbol{\mu_k}$ fixed. Since J is linear on r_{nk}, this step has a closed solution: the n data points are independent, and we can assign a value to r_{nk} separately; $r_{nk}=1$ if $k=\text{argmin}_j \{||\mathbf{x_n}-\boldsymbol{\mu_k}||^2\}$, 0 otherwise. In other words, we assign observation *n* to the closest cluster center,
  2. **Step M:** minimize J with respect $\boldsymbol{\mu_k}$ keeping r_{nk} fixed. The problem is quadratic on $\boldsymbol{\mu_k}$, and we can obtain the solution by setting the gradient equal to zero:

     $\boldsymbol{\mu_k} = (\sum_{k=1,\dots,K} r_{nk}\mathbf{x_n})/(\sum_{k=1,\dots,K} r_{nk})$ → is the mean with respect cluster members
- Repeat until the method converges. Convergence is assured since at each step J is reduced,
- This method corresponds to the steps **E** (Expectation) and **M** (maximization) of the **EM algorithm** that we will study later.

- **K-means (Bishop): a different interpretation. Improvements and variations ...**
  - Algorithm is slow when computing step E, since implies obtaining the Euclidean distance for all points. Then there are schemes to speed up the calculation,

  - There also are on-line schemes for updating $\boldsymbol{\mu}_k$ when computing each point n:

    $\boldsymbol{\mu}_k^{new} = \boldsymbol{\mu}_k^{old} + \eta_n(\mathbf{x}_n - \boldsymbol{\mu}_k^{old})$

Where η_n is the learning rate which is made to decrease as more data points are considered.

  - **Generalization of the K-means algorithm:** K-means uses square of Euclidean distance as measure of dissimilarity, other methods considered a different dissimilarity measure. Assuming that $\upsilon(x, x')$ is a measure of dissimilarity, then $J = \sum_{n=1,\ldots,N} \sum_{k=1,\ldots,K} r_{nk} \upsilon(\mathbf{x}, \boldsymbol{\mu}_k)$, which gives the **K-medoids** algorithms,

- **Mixture of Gaussians (Bishop) …**
  - **Goal:** obtain a clustering (K clusters) algorithm using the linear superposition of K Gaussian distributions (mixture of Gaussians) in terms of *latent variables*,

  - Let $\mathbf{z}$ be a K-dimensional variable, $\mathbf{z}=(z_1, \ldots, z_K)$, with $z_k=\{0,1\}$, and $\sum_{k=1,\ldots,K} z_k=1$, so only one position of the vector $\mathbf{z}$ is equal to 1, with probability $p(z_k=1)= \pi_k$. We can write the probability of $\mathbf{z}$ as $p(\mathbf{z})=\prod_{k=1,\ldots,K} \pi_k^{z_k}$. We call $\mathbf{z}$ a latent variable,
  - In a similar way, $p(\mathbf{x}|z_k=1)=N(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ and this can also be re-written as $p(\mathbf{x}|\mathbf{z})= \prod_{k=1,\ldots,K} N(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_k}$,
  - The joint distribution is given by $p(\mathbf{x},\mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$, and $p(\mathbf{x})$ is the marginal summing over all $\mathbf{z}$:

    $$p(\mathbf{x})= \sum_z p(\mathbf{x}, \mathbf{z})= \sum_z p(\mathbf{x}|\mathbf{z})p(\mathbf{z})= \sum_{k=1,\ldots,K} \pi_k N(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

  - Other parameter that we will need is:

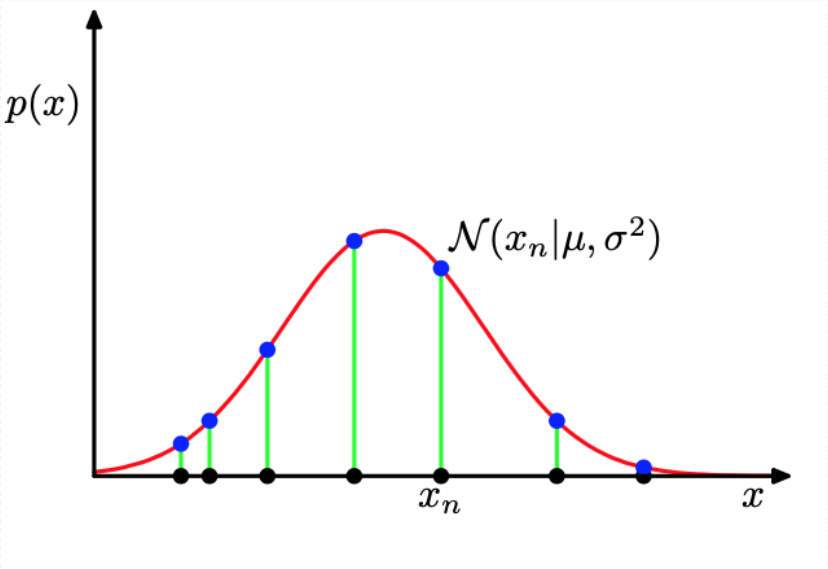
    $$\gamma(z_k)= p(z_k=1|\mathbf{x}) = p(\mathbf{x}|z_k=1)/p(\mathbf{x}) = \pi_k N(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)/(\sum_{k=1,\ldots,K} \pi_k N(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k))$$

We can see π_k as the prior and $\gamma(z_k)$ as the posterior probability once we have observed $\mathbf{x}$,

- **Maximum likelihood (ML) reminder …**
  - Assume a variable x~N(μ, σ^2) (Normal distributed with mean μ and variance σ^2, where in this example, x is 1-D for simplicity),
  - Now we want to characterize $\{\mu,\sigma^2\}$ from a sampling of N data observations (samples), x_1, …, x_N, taken from this distribution, the **maximum likelihood** maximizes the log of the joint distribution, that since the data points are taken independently and identically distributed p(**x**)= p(x_1, …, x_N)= $\Pi_{n=1,\dots,N}$ p(x_n), and then:

$$\ln\{p(\mathbf{x})\} = \ln\{p(x_1,\dots,x_N)\} = \ln\{\Pi_{n=1,\dots,N}\, p(x_n)\} = \Sigma_{n=1,\dots,N}\, p(x_n) = \Sigma_{n=1,\dots,N}\, N(\mu,\sigma^2),$$

In general, ML underestimates the variance of the distribution due to overfitting, except when N→∞.

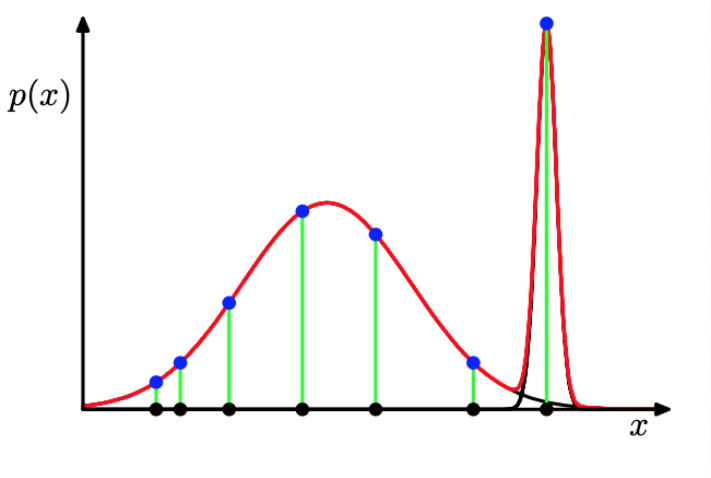


The black points are the data set points $\{x_n\}$, and the maximum likelihood is given by the product of the the blue values. Maximizing this product allows to obtain the parameters $\{\mu,\sigma^2\}$ that govern the normal distribution.

- **Mixture of Gaussians (Bishop) ...**
  - If we have N observations, $\mathbf{x}_1, \ldots, \mathbf{x}_N$ (each observation has M features), then we will have N latent variables $\mathbf{z}_1, \ldots, \mathbf{z}_N$. If we want to use maximum likelihood, and we want to classify them using K clusters, and knowing that $p(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \Pi_{n=1,\ldots,N}\, p(\mathbf{x}_n)$, we take logarithms, and

$$\ln\{\mathbf{x}\} = \ln\{p(\mathbf{x}_1, \ldots, \mathbf{x}_N)\} = \ln\{\Pi_{n=1,\ldots,N}\, p(\mathbf{x}_n)\} = \Sigma_{n=1,\ldots,N}\, \ln\{p(\mathbf{x}_n)\} = \Sigma_{n=1,\ldots,N}\, \ln\{\Sigma_{k=1,\ldots,K}\, \pi_k N(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\}$$

  - now we have to maximize $\ln\{p(\mathbf{x})\}$ to find the parameters $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, that characterize the joint distribution, and π_k that characterize the assignment to ca cluster k,
  - the main problem with this maximization problem is that appear singularities, and it is not a well posed problem,
  - for example, assume that $\boldsymbol{\Sigma}_k = \sigma_k^2 \mathbf{I}$, and $\boldsymbol{\mu}_j = \mathbf{x}_n$ for some value n, then this point contributes to the likelihood with a term of the form $N(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = N(\mathbf{x}_n, \sigma_k^2 \mathbf{I}) = 1/((2\pi)^{1/2}\sigma_j^2)$, as $\sigma_j \rightarrow 0$, then the log likelihood tends to infinity,



- **Mixture of Gaussians (Bishop) …**
  - Moreover having the problem of singularities and overfitting, we have the problem of having a K-component mixture that will produce K! equivalent solutions, a problem that is called *identifiability*, that consists in that there are K! of assigning K sets of parameters to K components,
  - The introduction of a latent variables lead to significant simplifications since instead of working with marginal distributions p(**x**), we will be able to work with joint distributions p(**x**,**z**) through the introduction of a very efficient an elegant algorithm called **E-M** (**Expectation-Maximization**) designed for obtaining maximum likelihoods,
  - **EM** is an algorithm that although applied here to the mixture of Gaussians, is a very powerful tool that appears in many algorithms and models,