Lecture 10

Clustering

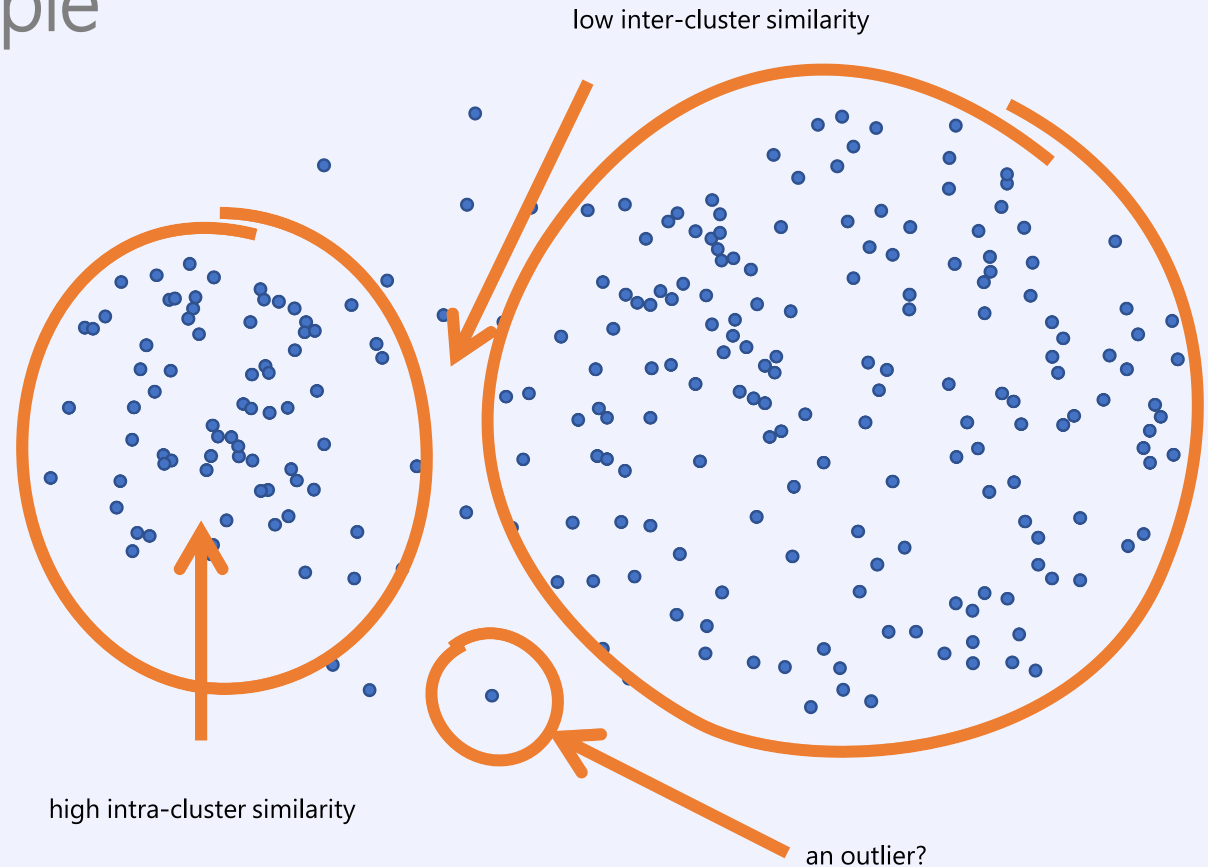
ISLR 12, ESL 14

Jilles Vreeken
Aleksandar Bojchevski

UNIVERSITÄT DES SAARLANDES

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Example



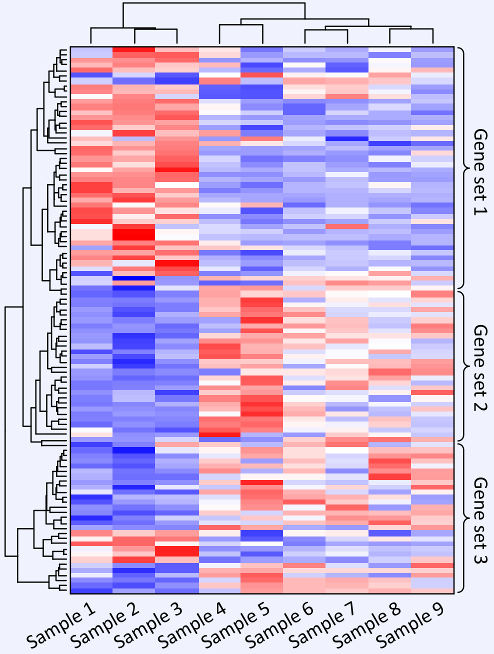
low inter-cluster similarity

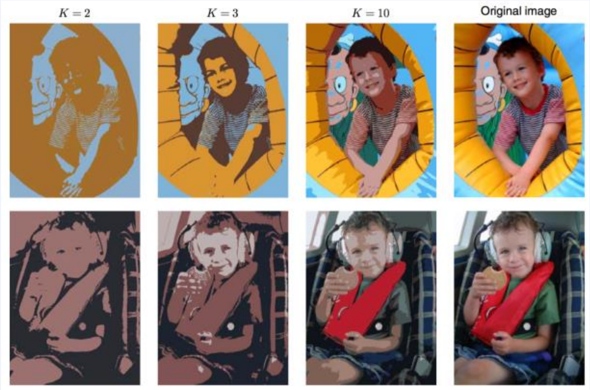
high intra-cluster similarity

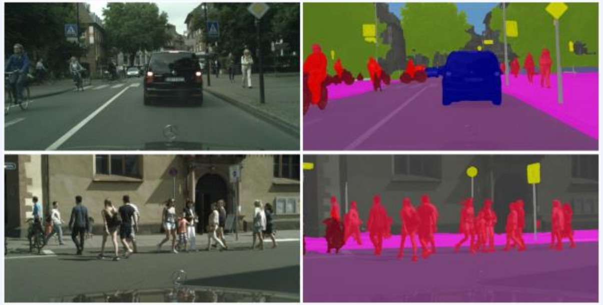
an outlier?

Clustering Apllications

- User profiling
- Gene expression analysis
- Data compression
- Image segmentation
- Visualization
- ….







The Clustering Problem

Given a set U of objects and a distance $d: U^2 \rightarrow R^+$ between objects, group the objects of U into clusters such that the

distance between points in the same cluster is low and the
distance between the points in different clusters is large

- small and large are not well defined

A clustering of U can be

- exclusive (each point belongs to exactly one cluster)
- probabilistic (each point has a probability of belonging to a cluster)
- fuzzy (each point can belong to multiple clusters)

The number of clusters can be pre-defined, or not

K-means Clustering

Iterative method for calculating disjoint clusters

- K disjoint clusters $C_1, \dots, C_K$ are subsets of the observations s.t. $C_1 \cup C_2 \cup \cdots \cup C_K = \{1, \dots, n\}$ $C_k \cap C_{k'} = \emptyset$ for all $k \neq k'$, i.e. each observation belongs to exactly one cluster
- for a good clustering the within-cluster variation $W(C_k)$ should be small $\min\limits_{C_1,\dots,C_k} \sum_{k=1}^{K} W(C_k)$
- there are many ways to define $W(C_k)$n
- requires metric data space, often we use the Euclidean distance as the underlying metric

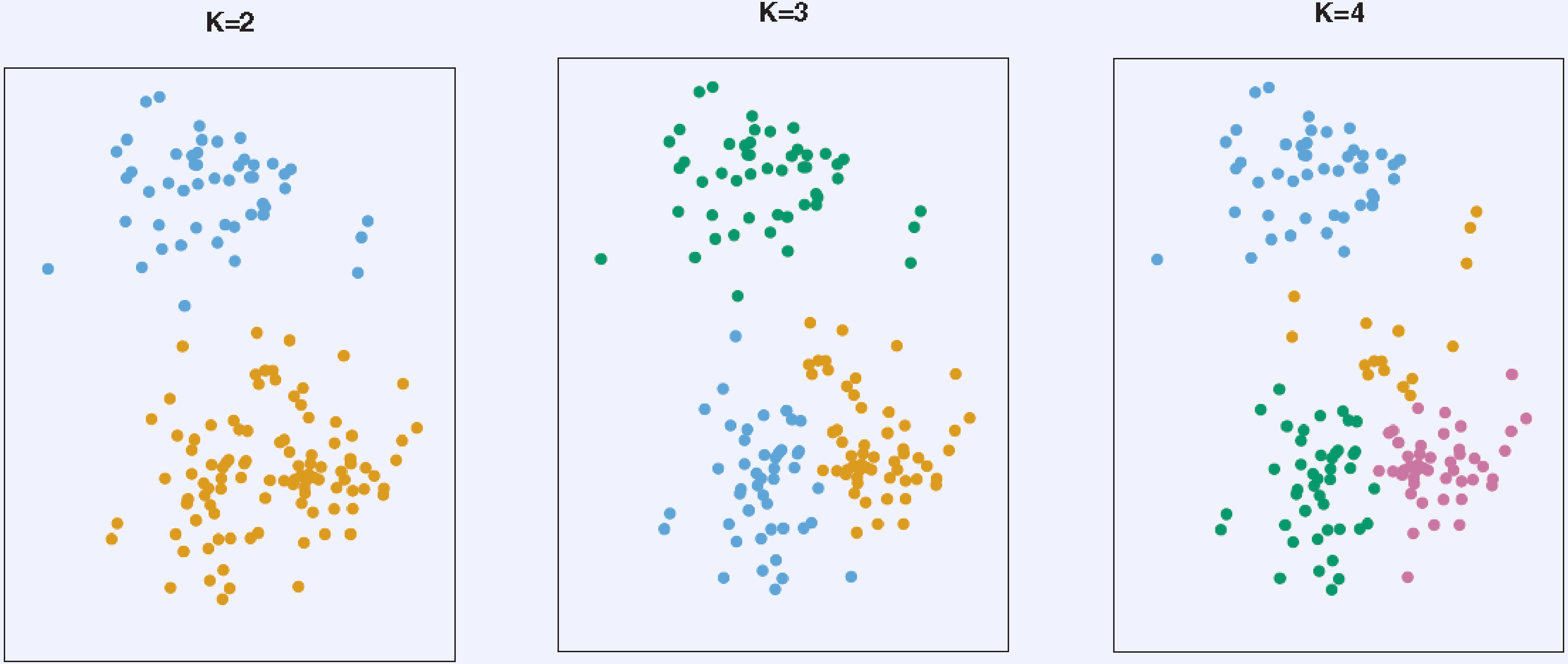
$$W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} \left(x_{ij} - x_{i'j}\right)^2 = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \|x_i - x_{i'}\|^2$$

cluster size ↗

- the minimization is very difficult because there are many $\binom{n}{k}$ partitions of the data into K clusters
- the choice of K is a difficult model decision

(ISLR 12.4.1)

K-means Clustering



K=2

K=3

K=4

Lloyd's algorithm

```
ALGORITHM 12.2 $K$-means clustering
1. randomly assign points to clusters
2. iterate until clusters stop changing
    a) for each cluster compute its centroid (i.e. the average location of its members)
    b) assign each observation to the cluster whose centroid is closest (in Euclidean distance)
```

Guaranteed to converge: finitely many configurations and $W(C_k)$ decreases at each iteration

- **proof**: observe

$$\frac{1}{|C_k|}\sum_{i,i'\in C_k}\sum_{j=1}^{p}\left(x_{ij}-x_{i'j}\right)^2 = 2\sum_{i\in C_k}\sum_{j=1}^{p}\left(x_{ij}-\bar{x}_{kj}\right)^2 = 2\sum_{i\in C_k}\|x_i-\bar{x}_k\|^2$$

- In 2a) the centroids $\bar{x}_k$ are chosen to minimize $W(C_k)$
- In 2b) the cluster assignments are chosen to minimize $W(C_k)$

(ISLR 12.4.1)

Another interpretation of $\boldsymbol{K}$-means

Let $\boldsymbol{C_k}$ be defined as before and define each cluster by its centroid $\boldsymbol{\mu_k}$

We aim to minimize the objective $\min_{\mu_k, C_k} \sum_{k=1}^{K} W(C_k) = 2 \sum_{i \in C_k} \|x_i - \mu_k\|^2$

- i.e. find the best centroids and the best cluster assignments

This joint optimization is very difficult to solve, however the two subproblems are simple:

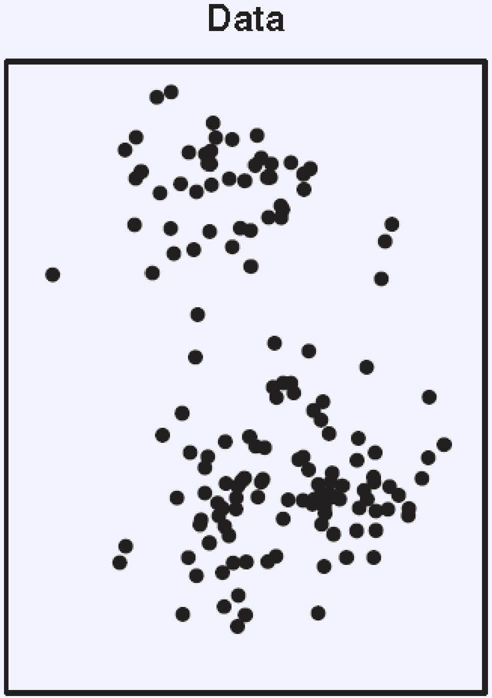
- fix $\boldsymbol{C_k}$, the best exact solution for $\mu_k = \bar{x}_k$ is just the mean
- fix $\boldsymbol{\mu_k}$, the best exact solution for $\boldsymbol{C_k}$ is assigning the observation to the closest cluster

Therefore we do alternating optimization updating $\boldsymbol{C_k}$ and $\boldsymbol{\mu_k}$ in turn

This is a general strategy that works very well in many settings (e.g. GMMs)

K-means Clustering

```
ALGORITHM 12.2 $K$-means clustering
1. randomly assign points to clusters
2. iterate until clusters stop changing
    a) for each cluster compute its centroid
    b) assign each observation to the cluster with closest centroid
```



Data

Iteratively minimize $\min_{C_k} 2\sum_{i\in C_k}\|x_i-\bar{x}_k\|^2$ (*)

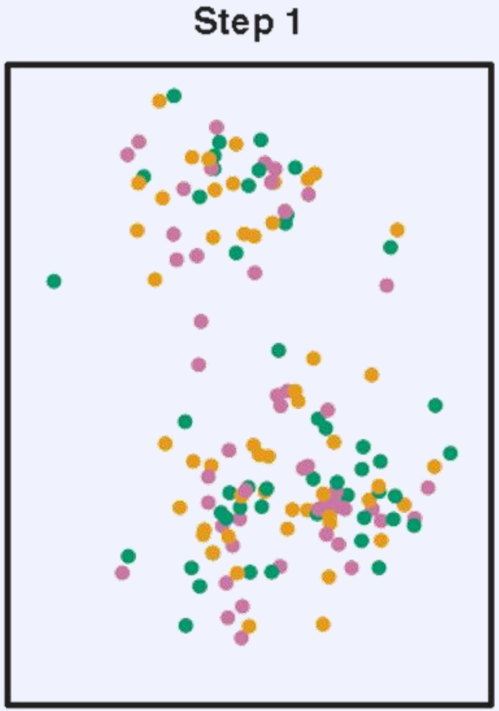
- In 2a) the centroids $\bar{x}_k$ are chosen to minimize (*)
- In 2b) the cluster assignments C_k are chosen to minimize (*)

K-means Clustering

```
ALGORITHM 12.2 $K$-means clustering
1. randomly assign points to clusters
2. iterate until clusters stop changing
    a) for each cluster compute its centroid
    b) assign each observation to the cluster with closest centroid
```

Iteratively minimize $\min_{C_k} 2 \sum_{i \in C_k} \|x_i - \bar{x}_k\|^2$ (*)

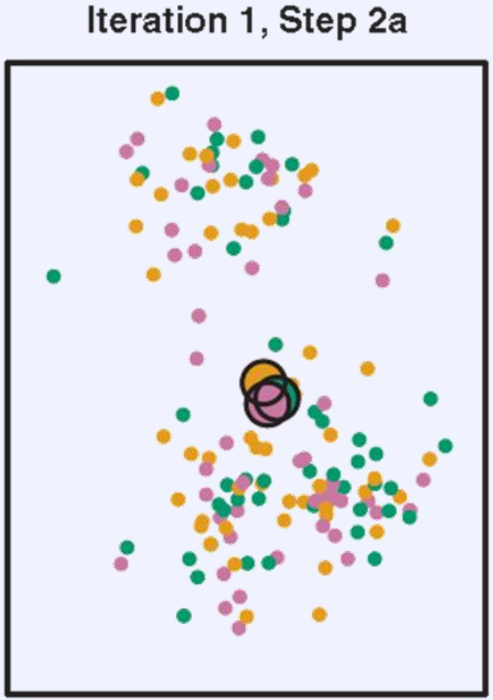
- In 2a) the centroids $\bar{x}_k$ are chosen to minimize (*)
- In 2b) the cluster assignments C_k are chosen to minimize (*)



K-means Clustering

ALGORITHM 12.2 K-means clustering

1. randomly assign points to clusters
2. iterate until clusters stop changing
    a) for each cluster compute its centroid
    b) assign each observation to the cluster with closest centroid



Iteratively minimize $\min_{C_k} 2 \sum_{i \in C_k} \|x_i - \bar{x}_k\|^2$ (*)

- In 2a) the centroids $\bar{x}_k$ are chosen to minimize (*)
- In 2b) the cluster assignments C_k are chosen to minimize (*)

K-means Clustering

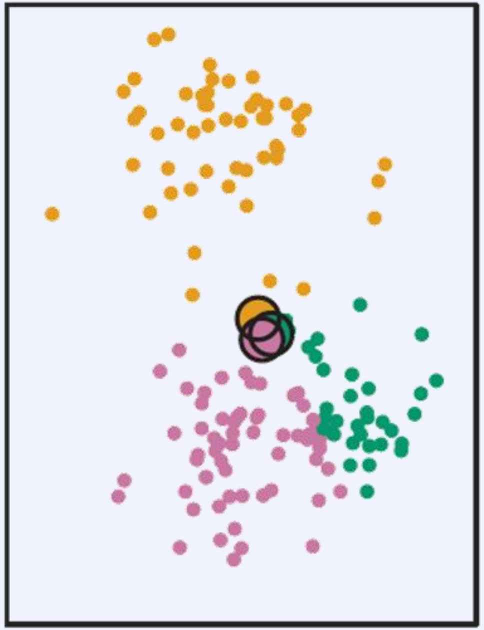
ALGORITHM 12.2 K-means clustering

1. randomly assign points to clusters
2. iterate until clusters stop changing
    a) for each cluster compute its centroid
    b) assign each observation to the cluster with closest centroid

Iteratively minimize $\min_{C_k} 2 \sum_{i \in C_k} \|x_i - \bar{x}_k\|^2$ (*)

- In 2a) the centroids $\bar{x}_k$ are chosen to minimize (*)
- In 2b) the cluster assignments C_k are chosen to minimize (*)

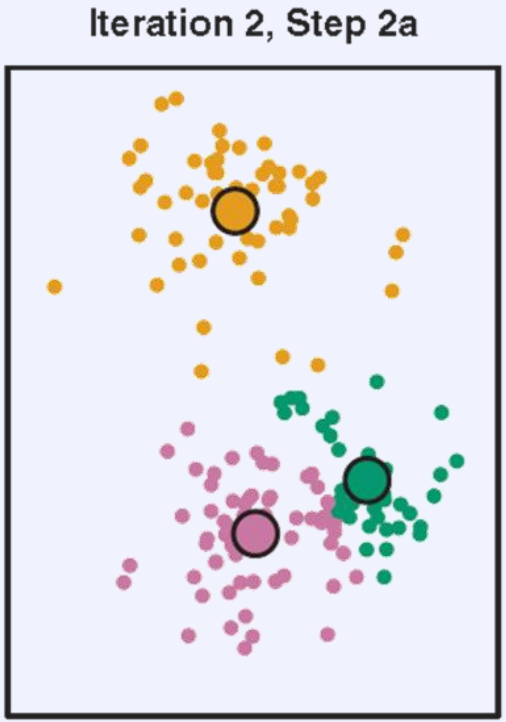
Iteration 1, Step 2b



K-means Clustering

ALGORITHM 12.2 K-means clustering

1. randomly assign points to clusters
2. iterate until clusters stop changing
   a) for each cluster compute its centroid
   b) assign each observation to the cluster with closest centroid



Iteratively minimize $\min_{C_k} 2 \sum_{i \in C_k} \|x_i - \bar{x}_k\|^2$ (*)

- In 2a) the centroids $\bar{x}_k$ are chosen to minimize (*)
- In 2b) the cluster assignments C_k are chosen to minimize (*)

K-means Clustering

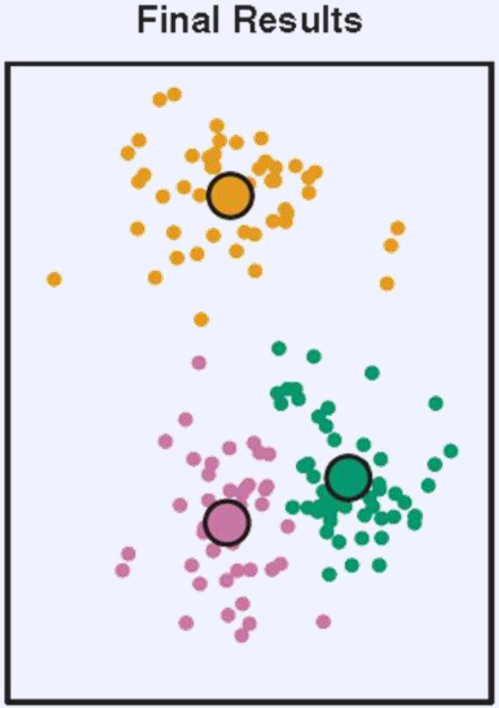
ALGORITHM 12.2 K-means clustering

```
1. randomly assign points to clusters
2. iterate until clusters stop changing
    a) for each cluster compute its centroid
    b) assign each observation to the cluster with closest centroid
```

Iteratively minimize $\min_{C_k} 2\sum_{i \in C_k} \|x_i - \bar{x}_k\|^2$ (*)

- In 2a) the centroids $\bar{x}_k$ are chosen to minimize (*)
- In 2b) the cluster assignments C_k are chosen to minimize (*)

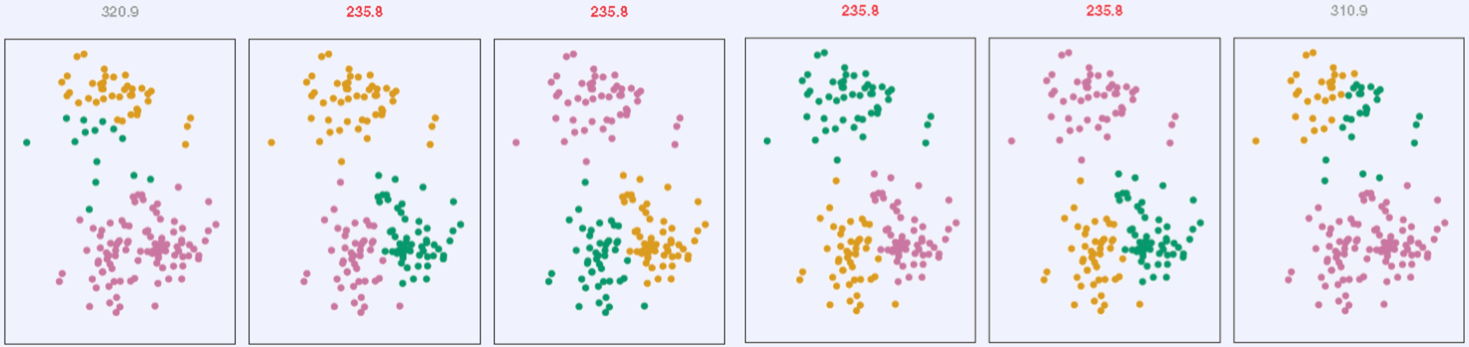
- Final result after 10 iterations



K-means initialization

K-means clustering is greedy and thus only finds a local optimum

- thus it is important to run the algorithm multiple times each with different starting solutions
- here results for six random starting solutions, with K=3
- the smallest within-cluster variation is 235.8



In practice K-means++ is the most popular algorithm for choosing the initial centroids

(ISLR 12.4.1)

K-medoids

Limitation of K-means

- needs a metric space (when the centroids of the clusters are chosen)
- sensitive to outliers

K-medoids clustering algorithm proceeds iteratively, just like K-means

- for a given cluster assignment C find medoids
- given a set of medoids, minimize total error by assigning each observation to the closest medoid

Medoid: the observation that is closest (least dissimilar) to all other observations in the cluster

$$i_k^* = \arg\min_{i:C_i=k} \sum_{j:C(j)=k} d(x_i, x_j)$$

- can also be used if only dissimilarity matrices are given (does not need the metric space)
- computation of a cluster center increases from N to N^2

(ESL 13.3.10)

Hierarchical Clustering

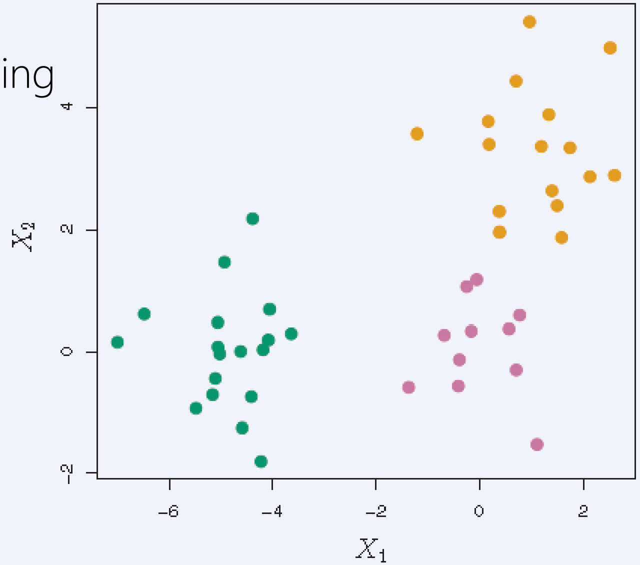
The need to choose a K is a problem with K-means clustering

Hierarchical clustering does not have this requirement

There are top-down and bottom-up versions

- top-down (divisive):
  recursively bisect the dataset into clusters
- bottom-up (agglomerative):
  start with singleton clusters and iteratively merge clusters

Both methods produce tree-like dendrograms



Simulated dataset with three classes depicted by color. The class labels are **unknown** to the clustering algorithm.

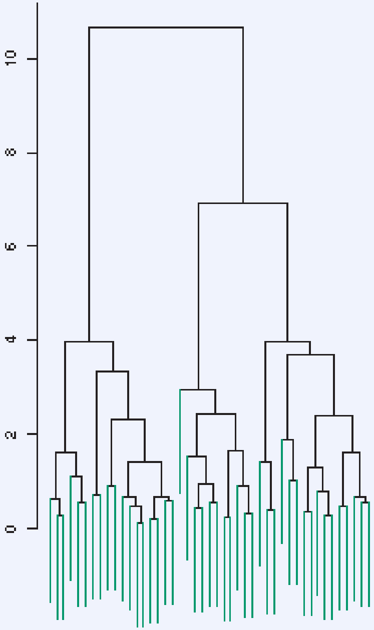
(ISLR 12.4.2)

Interpreting a Dendrogram

A tree-like structure where

- each leaf represents an observation
- each internal node is the root of a subtree that can be considered a cluster

- y-coordinate shows the dissimilarity of the two clusters joined by a node
- ordering along x-axis is arbitrary (as long as it obeys the tree topology)
  - often secondary criteria are used to select this ordering
  - distance along the horizontal axis does not reflect similarity of observations



Dendogram for hierarchical clustering with complete linkage

(ISLR 12.4.2)

Interpreting a Dendrogram

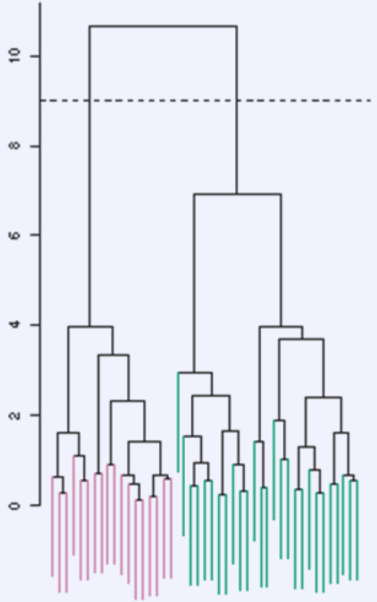
A tree-like structure where

- each leaf represents an observation
- each internal node is the root of a subtree that can be considered a cluster

- y-coordinate shows the dissimilarity of the two clusters joined by a node
- ordering along x-axis is arbitrary (as long as it obeys the tree topology)
  - often secondary criteria are used to select this ordering
  - distance along the horizontal axis does not reflect similarity of observations

Horizontal cuts in the dendrogram result in disjoint clusters

- cut at 9 results in two clusters



Dendogram for hierarchical clustering with complete linkage

(ISLR 12.4.2)

Interpreting a Dendrogram

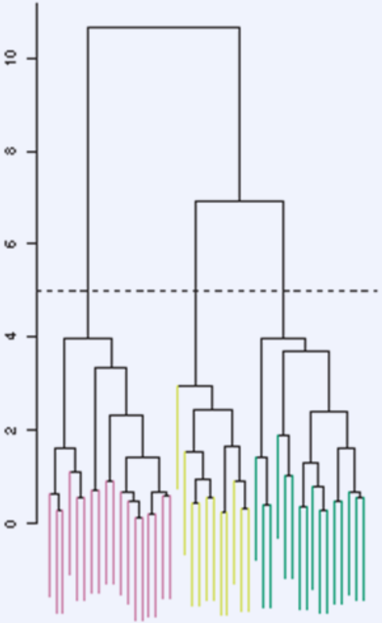
A tree-like structure where

- each leaf represents an observation
- each internal node is the root of a subtree that can be considered a cluster

- y-coordinate shows the dissimilarity of the two clusters joined by a node
- ordering along x-axis is arbitrary (as long as it obeys the tree topology)
    - often secondary criteria are used to select this ordering
    - distance along the horizontal axis does not reflect similarity of observations

Horizontal cuts in the dendrogram result in disjoint clusters

- cut at 9 results in two clusters
- cut at 5 results in three clusters
- the lower the cut, the more clusters

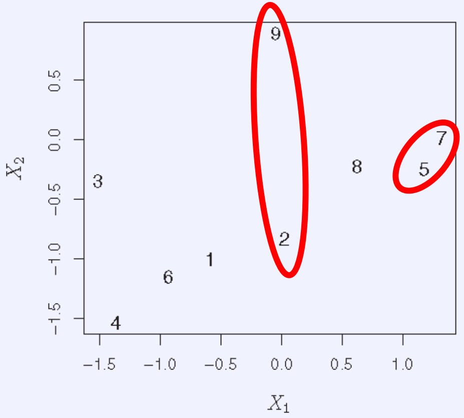


Dendogram for hierarchical clustering with complete linkage

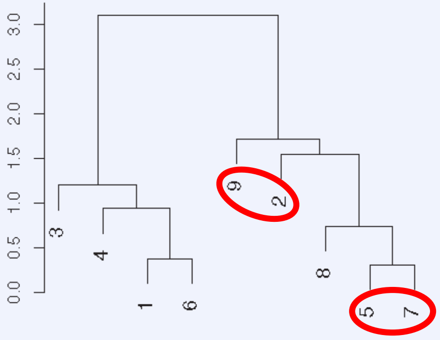
(ISLR 12.4.2)

Interpreting a Dendrogram

Distance along the horizontal axis does not reflect similarity of observations



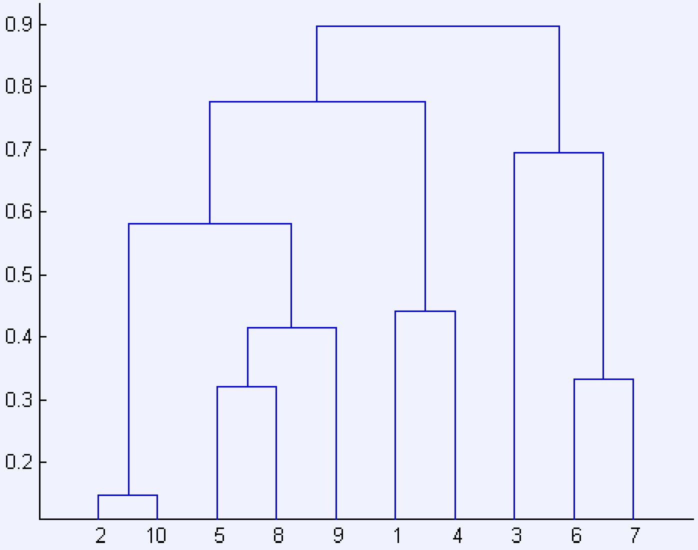
raw data



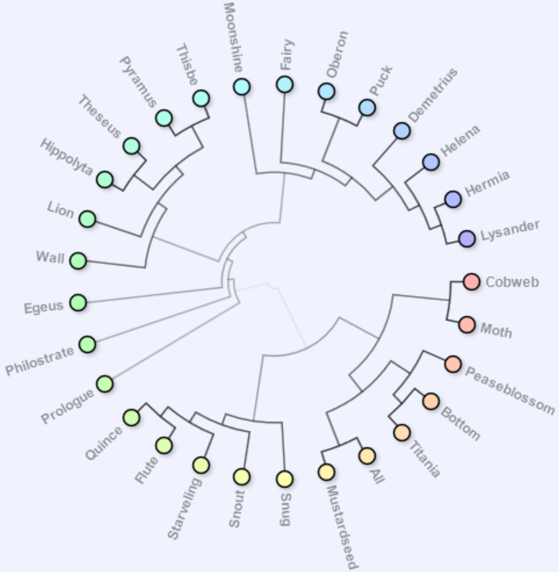
dendrogram

(ISLR 12.4.2)

Other Visualizations of Dendrograms



All leaf prongs are drawn at the zero y-coordinate

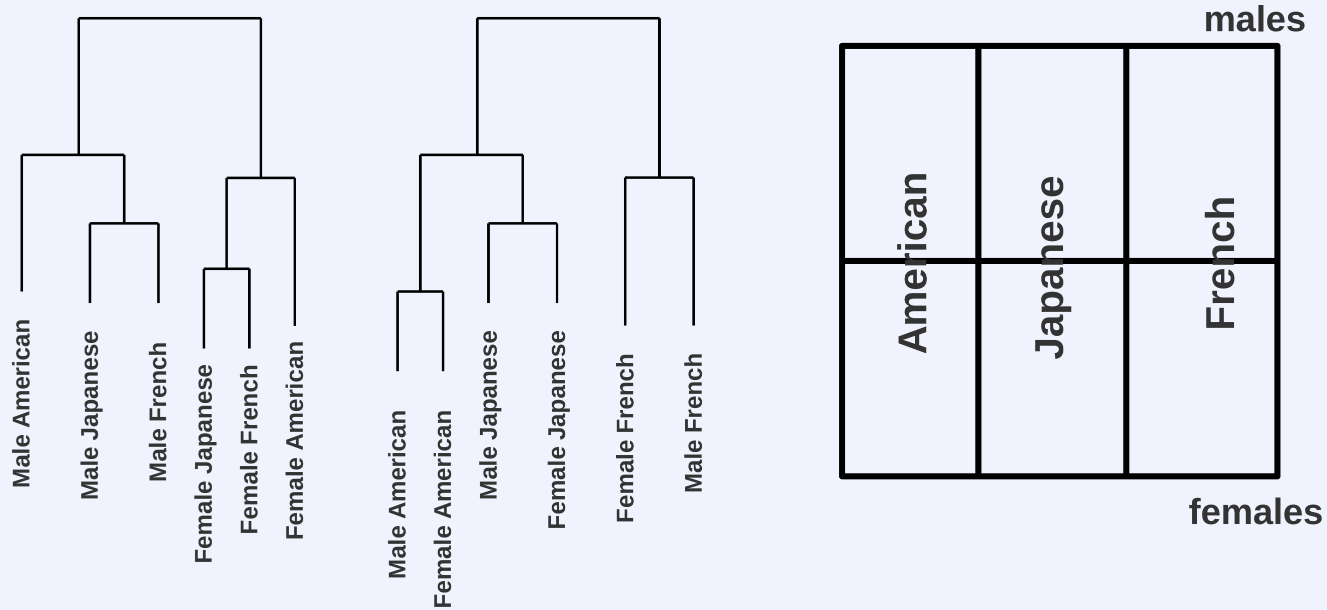


(ISLR 12.4.2)

Hierarchical Clustering

A dendrogram is not always appropriate for structuring the cluster structure of a data set

- some datasets do not have hierarchical structure
- in such case hierarchical clustering leads to worse results than K-means in terms of cluster coherence (the inverse of within-cluster variance)



(ISLR 12.4.2)

Agglomerative Clustering

Agglomerative clustering is superior to divisive clustering

- divisive clustering is at risk of forming wrong partitions early on that cannot be rectified
- agglomerative clustering repeatedly joins the two most similar (least dissimilar) clusters

ALGORITHM 12.3 Agglomerative Clustering

```
1. each observation is its own singleton cluster, compute all pairwise dissmilarities between observations
2. For $i = n, n-1, \dots, 2$ do:
    a) fuse the two most similar clusters and set the height of the respective node in
       the dendrogram as the dissimilarity between these two clusters
    b) compute new pairwise dissimilarities between the clusters
```

Several notions of **cluster dissimilarity** are available

- all are based on the matrix of pairwise dissimilarities of the observations

(ISLR 12.4.2)

Notions of Cluster Dissimilarity

Let $(d_{ij})_{i,j=1,\dots,n}$ be the pairwise dissimilarity matrix, often using the Euclidian distance and let $d(G,H)$ be the dissimilarity between two clusters G and H

Complete linkage (CL) $$d_{CL}(G,H) = \max_{i\in G, j\in H} d_{ij}$$

- leads to compact clusters

Single linkage (SL) $$d_{SL}(G,H) = \min_{i\in G, j\in H} d_{ij}$$

- can lead to snake-like clusters

(Group) average linkage (GA) $$d_{GA}(G,H) = \frac{1}{N_G N_H} \sum_{i\in G} \sum_{j\in H} d_{ij}$$

- compromise between the previous two extremes

(ISLR 12.4.2)

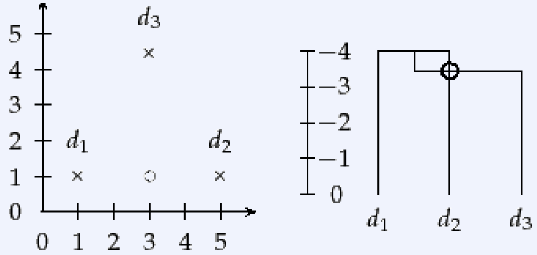
The Case Against Centroid Linkage

Let $(d_{ij})_{i,j=1,\dots,n}$ be the dissimilarity matrix, often using the Euclidian distance and let $d(G,H)$ be the dissimilarity between two clusters G and H

Centroid linkage (CTL) $$d_{CTL}(G,H) = \left|\left|\bar{G} - \bar{H}\right|\right|^2$$

- where $\bar{G}$ and $\bar{H}$ are the centroids of the two clusters

Can result in undesired inversions



► **Figure 17.9** Centroid clustering is not monotonic. The documents d_1 at $(1+\epsilon, 1)$, d_2 at $(5,1)$, and d_3 at $(3, 1+2\sqrt{3})$ are almost equidistant, with d_1 and d_2 closer to each other than to d_3. The non-monotonic inversion in the hierarchical clustering of the three points appears as an intersecting merge line in the dendrogram. The intersection is circled.

http://nlp.stanford.edu/IR-book/html/htmledition/centroid-clustering-1.html

Notions of Cluster Dissimilarity

Diameter of a cluster G *is defined as* $D_G = \max_{i,j \in G} d_{ij}$

- CL clusters have small diameter
- SL cluster can have large diameter
- GA and CTL are in between

Group average dissimilarity is a sound estimate of mean distance

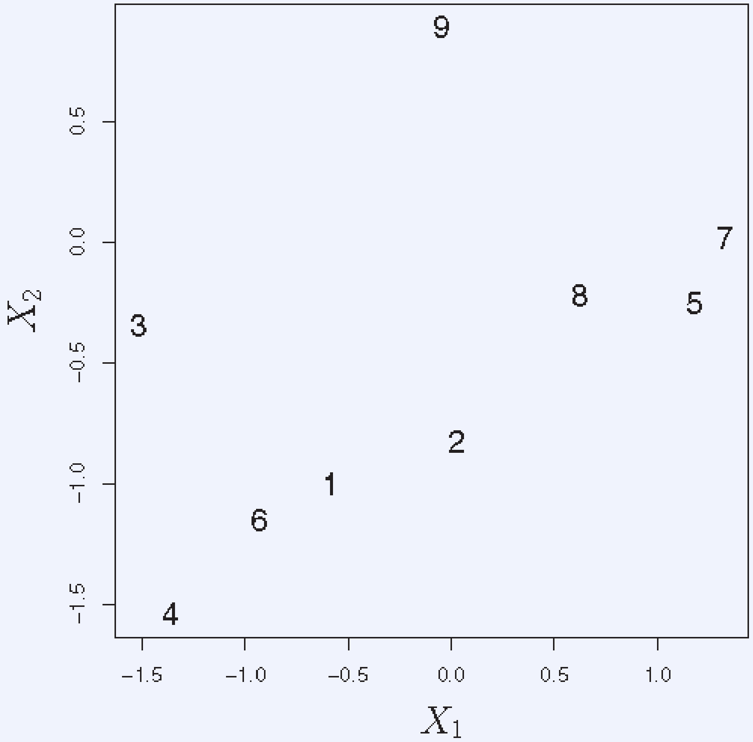
$$d_{GA}(G,H) = \int\int d(x,x')p_G(x)p_H(x')dxdx'$$

- the mean is taken over distances in a continuous data space
- as $n \to \infty$ we have that $d_{GA}(G,H) = \frac{1}{N_G N_H}\sum_{i\in G}\sum_{j\in H} d_{ij}$ approaches the equation above

- $d_{SL}(G,H)$ approaches $\mathbf{0}$
- $d_{CL}(G,H)$ approaches ∞

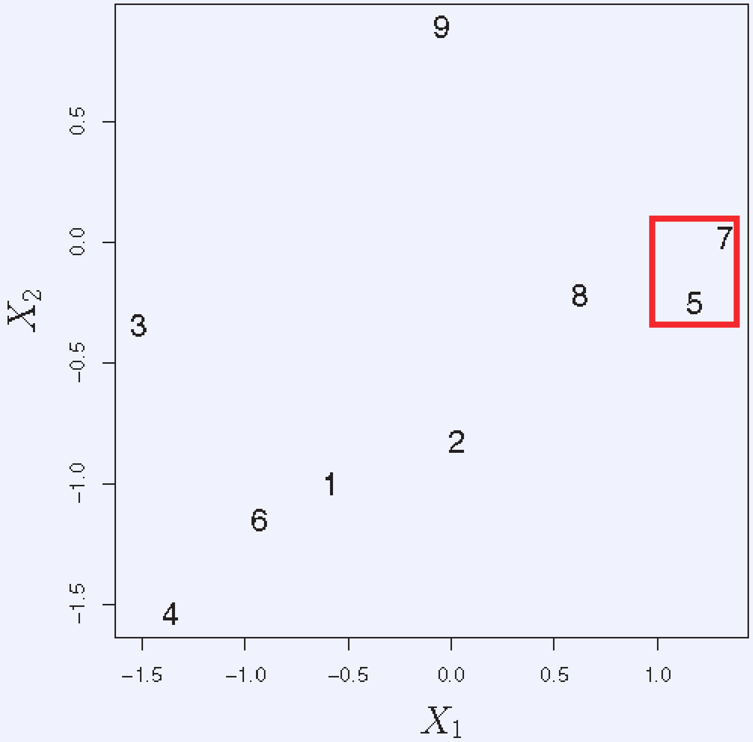
(ISLR 12.4.2)

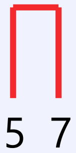
Agglomerative Hierarchical Clustering Example



(ISLR 12.4.2)

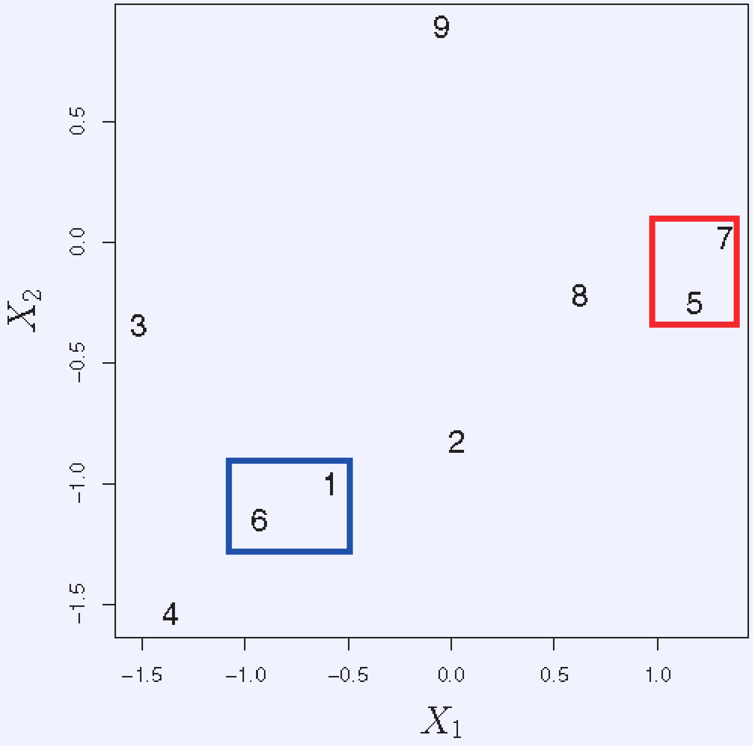
Agglomerative Hierarchical Clustering Example

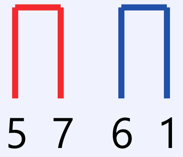




(ISLR 12.4.2)

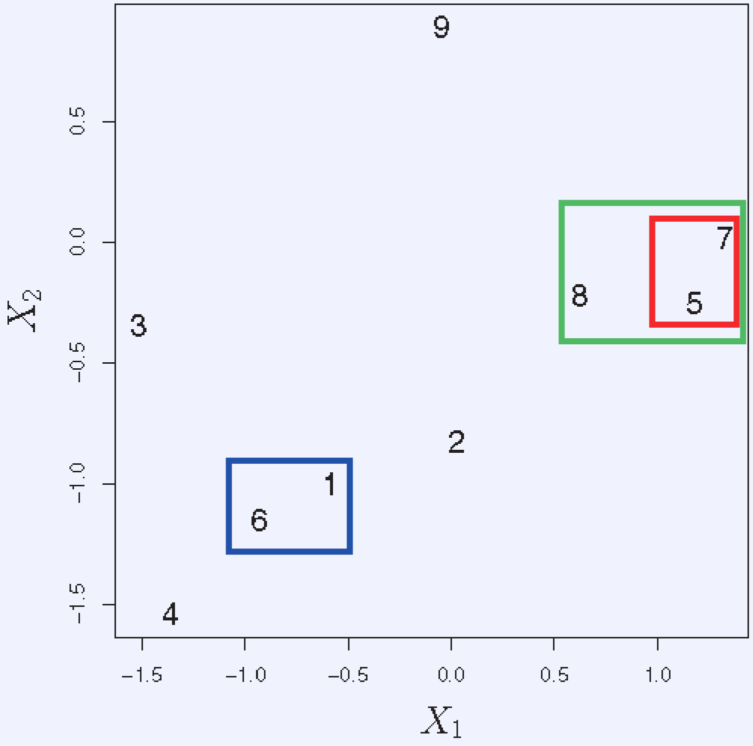
Agglomerative Hierarchical Clustering Example

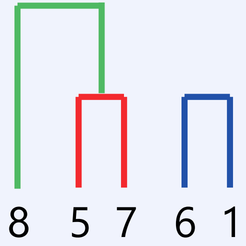




(ISLR 12.4.2)

Agglomerative Hierarchical Clustering Example

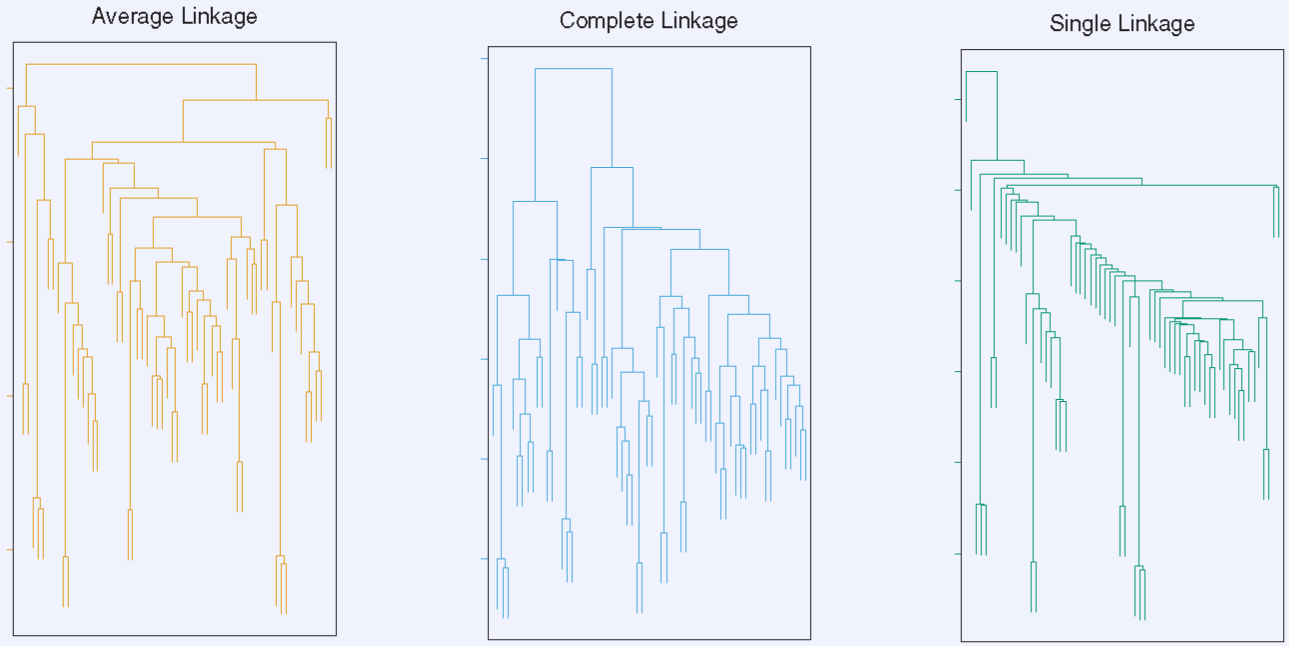




(ISLR 12.4.2)

Dendograms Vary with the Linkage Methods

Example of hierarchical clustering on a human tumor microarray data set

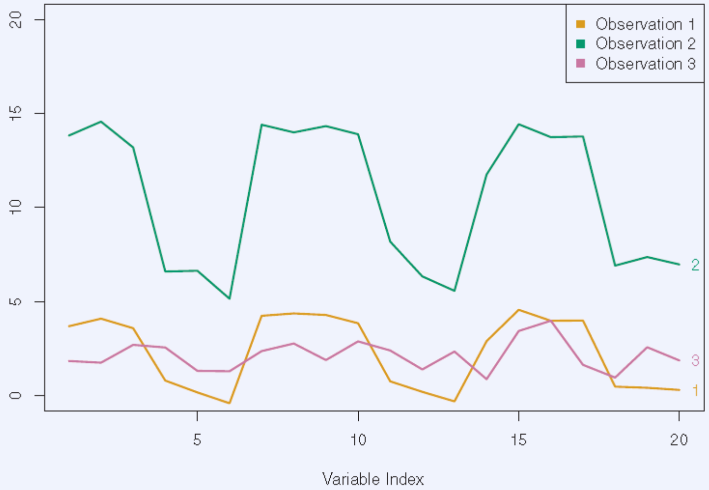


(ISLR 12.4.2)

Choice of dissimilarity matrix

So far we used Euclidean distance, **correlation**-based distance is sometimes more appropriate

- similar observations have feature vectors with high correlation, even if the are far in Euclidean distance
- focuses on shapes of observation profiles rather than their magnitudes



- observations 1 and 3 are similar w.r.t. Euclidean distance but not w.r.t. correlation-based distance
- observations 1 and 2 are similar w.r.t. correlation-based distance but not w.r.t. Euclidean distance
- observations 2 and 3 are not similar w.r.t. either

(ISLR 12.4.2)

Example: Shoppers buying profiles

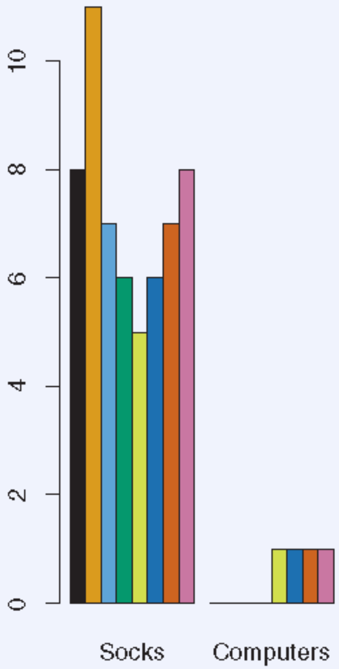
When to use which distance matrix?

- goal: suggest items that shoppers are likely to want to buy
- feature values are quantity of each item bought
- here we are more interested in shape than in magnitude
- so correlation-based distance appears more appropriate

When should we standardize the data?

- shoppers may tend to buy more socks than computers
- without standardization socks will dominate the dissimilarity values, even though
  - computers might be the more interesting item for the retailer
  - socks may be less informative about the customer than the number of computers bought
- standardization gives each variable equal importance
- standardization is also good, if different variables are measured in different scales

#items bought for 8 customers



(ISLR 12.4.2)

Example: Shoppers buying profiles

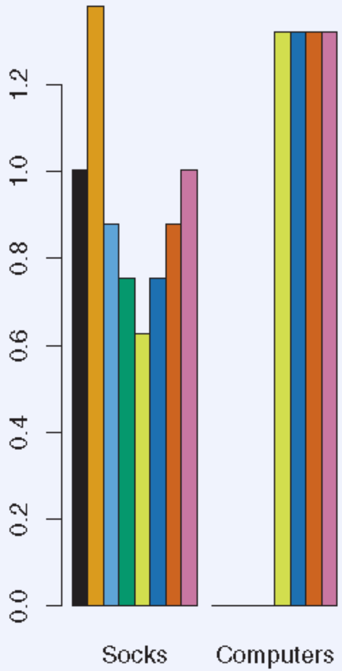
When to use which distance matrix?

- goal: suggest items that shoppers are likely to want to buy
- feature values are quantity of each item bought
- here we are more interested in shape than in magnitude
- so correlation-based distance appears more appropriate

When should we standardize the data?

- shoppers may tend to buy more socks than computers
- without standardization socks will dominate the dissimilarity values, even though
  - computers might be the more interesting item for the retailer
  - socks may be less informative about the customer than the number of computers bought
- standardization gives each variable equal importance
- standardization is also good, if different variables are measured in different scales

#items after standardization



(ISLR 12.4.2)

Example: Shoppers buying profiles

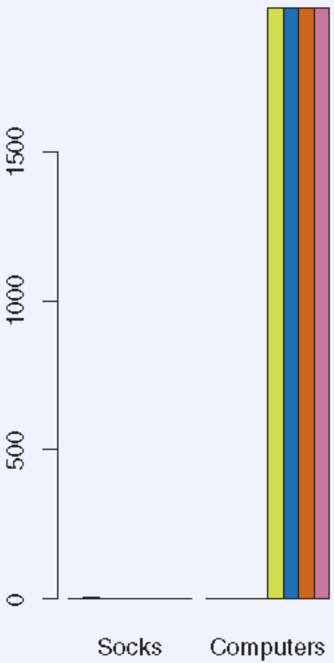
When to use which distance matrix?

- goal: suggest items that shoppers are likely to want to buy
- feature values are quantity of each item bought
- here we are more interested in shape than in magnitude
- so correlation-based distance appears more appropriate

When should we standardize the data?

- shoppers may tend to buy more socks than computers
- without standardization socks will dominate the dissimilarity values, even though
  - computers might be the more interesting item for the retailer
  - socks may be less informative about the customer than the number of computers bought
- standardization gives each variable equal importance
- standardization is also good, if different variables are measured in different scales

#dollars spent



(ISLR 12.4.2)

Practical Issues in Clustering

Small decisions with big consequences

- should we standardize the data?
- for hierarchical clustering
  - which dissimilarity matrix?
  - which type of linkage?
  - where to place the dendrogram cut?
- for K-means clustering
  - how to set K?

- Validating the clusters
  - difficult topic
  - if we have labels for at least some observations we can assess class purity
  - otherwise we can use the bootstrap to analyze the robustness of clusters

(ISLR 12.4.3)

Clustering Both Features and Observations

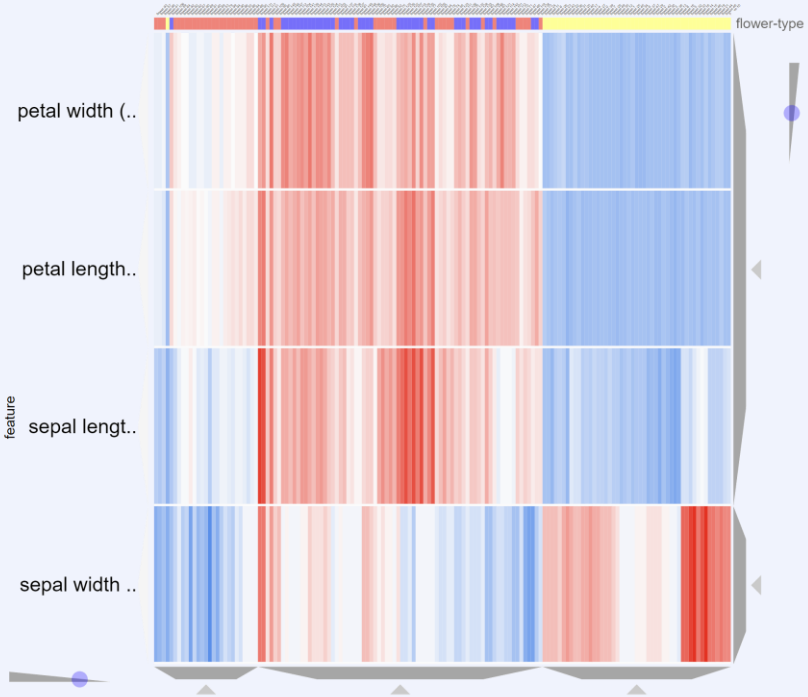
Apply hierarchical separately on both the

- observartions using distance between features
- features using distance between observations

Visualize the entire dataset as a matrix where rows and columns are sorted by the clusters

Can reveal interesting patterns in the data

Link to an interactive tool



clustering of the Iris dataset from clustergrammer

Different Variants of Clustering

Here, we have assigned each observation to exactly one cluster

- often, it is desirable to give a preference of observations to several clusters
  - a probability that the observation belongs to the cluster
- there are “soft” versions of K-means based on this principle
- often clusters are not very robust to changes in the data

Sometimes it is desirable to assign observations to multiple clusters instead of a single one

- see also community detection in e.g. social networks

Be cautious and thoughtful when you cluster data!

Summary

K-means

- find a predefined number of clusters such that each observation is assigned to the closest centroid
- greedy iterative algorithm that alternatingly updates the clusters assignments and centroids

Hierarchical clustering

- find a hierarchy of potential clusterings visually represented with a dendogram
- agglomerative clustering merges clusters in a bottom-up manner based on cluster dissimilarity metrics

Other Unsupervised Learning Methods

- dependency discovery
- pattern mining
- graph mining
- causal inference
- anomaly detection
- and many, many, many more…