Lecture 4: Finish PCA, Clustering

Reading: Sections 10.3, 10.5

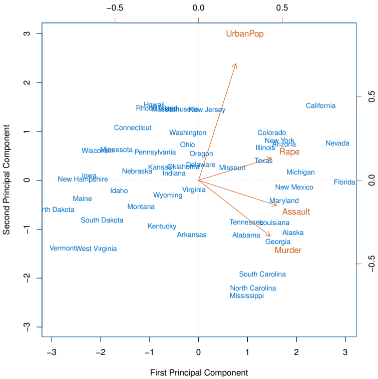
STATS 202: Data mining and analysis

September 30, 2019

PCA: Summary of last lecture

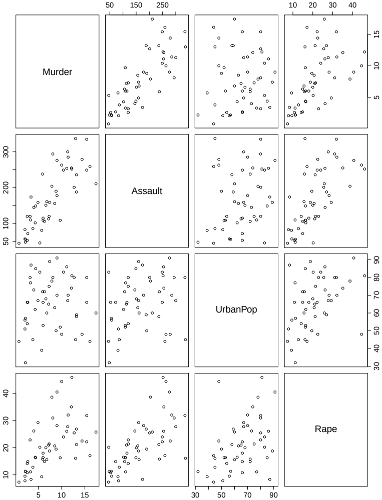
- The first principal component direction ϕ_1 is a unit vector of length p, which maximizes the variance of the projections or *scores* $z_{i,1} = x_i \cdot \phi_1$ for $i = 1, \ldots, n$. (x_i is the p-vector of the ith sample.)
- The second principal component direction ϕ_2 is a unit vector, orthogonal to ϕ_1, which maximizes the variance of the scores $z_{i,2}$, $i = 1, \ldots, n$.
- The third principal component direction ϕ_3 is orthogonal to ϕ_1 and ϕ_2, and so on...
- If $\mathbf{X} = \mathbf{U\Sigma\Phi}^T$ is the *singular value decomposition* of $\mathbf{X}$, the principal components are the columns of Φ and the matrix of scores is given by $\mathbf{U\Sigma}$.

PCA in practice: The biplot



- j-th variable represented by its PC loadings (ϕ_{1j}, ϕ_{2j}) on the top and right axes
- i-th datapoint represented by its scores on the bottom and left axes

How many principal components are enough?



We said 2 principal components capture most of the relevant information. But how can we tell?

The proportion of variance explained

We can think of the top **principal components** as directions in space in which the data vary the most.

The jth **score vector** $(z_{1j}, \ldots, z_{nj})$ can be interpreted as a *new* variable. The variance of this variable decreases as we take j from 1 to p. However, the total variance of the score vectors is the same as the total variance of the original variables:

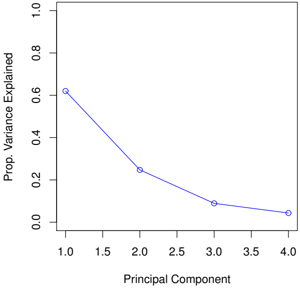
$$\sum_{j=1}^{p} \frac{1}{n} \sum_{i=1}^{n} z_{ij}^2 = \sum_{j=1}^{p} \frac{1}{n} \sum_{i=1}^{n} x_{ij}^2.$$

We can quantify how much of the variance is captured by the first m principal components/score variables.

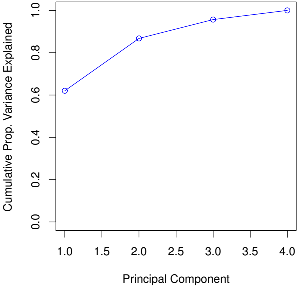
The proportion of variance explained

The variance of the mth score variable is (using $\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\boldsymbol{\Phi}^T$):

$$\frac{1}{n}\sum_{i=1}^{n} z_{im}^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{jm}x_{ij}\right)^2 = \frac{1}{n}\boldsymbol{\Sigma}_{mm}^2.$$



Scree plot



Clustering

As in **classification**, we assign a class to each sample in the data matrix. However, the class *is not an output variable*; we only use input variables.

Clustering is an **unsupervised** procedure, whose goal is to find homogeneous subgroups among the observations.

We will discuss 2 algorithms:

- K-means clustering
- Hierarchical clustering

K-means clustering

- K is the number of clusters and must be fixed in advance.
- The goal of this method is to minimize the dissimilarity of samples $W(C_\ell)$ within each cluster C_ℓ:

$$\min_{C_1,\dots,C_K} \sum_{\ell=1}^{K} W(C_\ell) \quad ; \quad W(C_\ell) = \frac{1}{|C_\ell|} \sum_{i,j\in C_\ell} \text{Distance}^2(x_{i,:}, x_{j,:}).$$

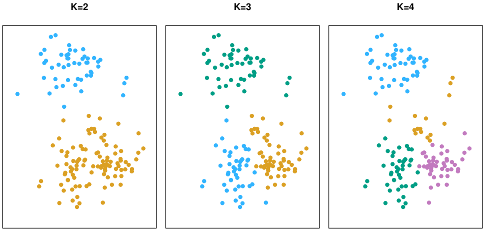


Figure 10.5

K-means clustering algorithm

1. Assign each sample to a cluster from 1 to K arbitrarily, e.g. at random.

2. Iterate these two steps until the clustering is constant:

   - Find the *centroid* of each cluster ℓ; i.e. the average $\overline{x}_{\ell,:}$ of all the samples in the cluster:

     $$\bar{x}_{\ell,j} = \frac{1}{|C_\ell|} \sum_{i \in C_\ell} x_{i,j} \quad \text{for } j = 1, \ldots, p.$$

   - Reassign each sample to the nearest centroid.

K-means clustering algorithm

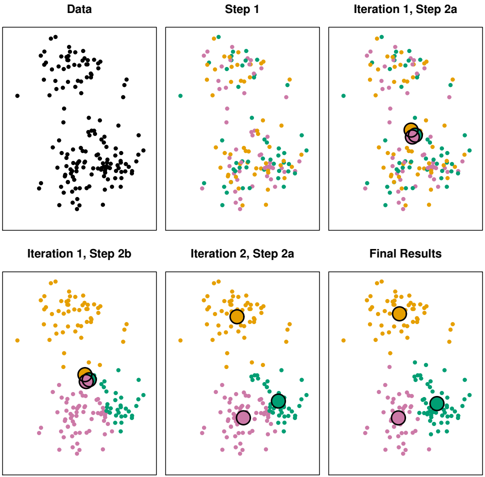


Figure 10.6

Properties of K-means clustering

- The algorithm always converges to a local minimum of

$$\min_{C_1,\dots,C_K} \sum_{\ell=1}^{K} W(C_\ell) \quad ; \quad W(C_\ell) = \frac{1}{|C_\ell|} \sum_{i,j \in C_\ell} \text{Distance}^2(x_{i,:}, x_{j,:}).$$

Why?

$$\frac{1}{|C_\ell|} \sum_{i,j \in C_\ell} \text{Distance}^2(x_{i,:}, x_{j,:}) = 2 \sum_{i \in C_\ell} \text{Distance}^2(x_{i,:}, \overline{x}_{\ell,:})$$

This side can only be reduced in each iteration.

- Each initialization could yield a different minimum, so there is no guarantee that we are at a global minimum.

Example: K-means output with different initializations

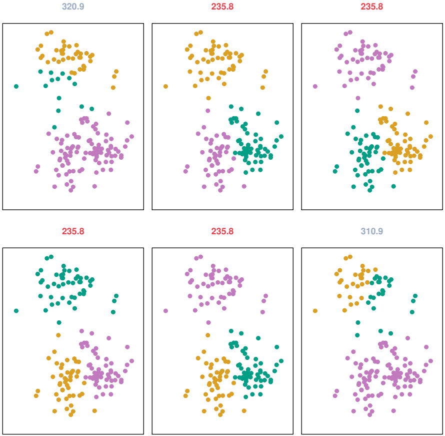
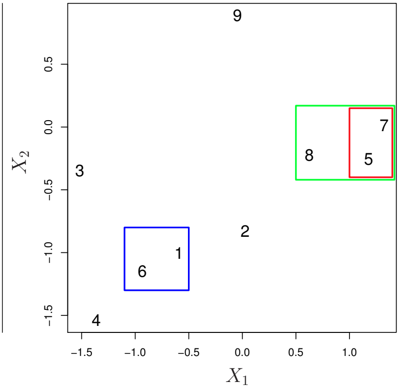


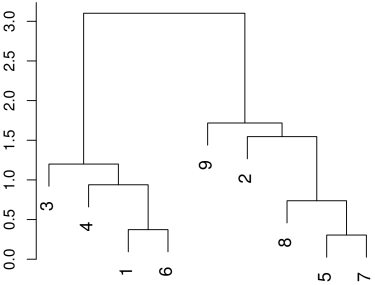
Figure 10.7

In practice, we start from many random initializations and choose the output which minimizes the objective function.

Hierarchical clustering

Most algorithms for hierarchical clustering are *agglomerative*.





The output of the algorithm is a *dendogram*. We must be careful about how we interpret the dendogram.

Hierarchical clustering

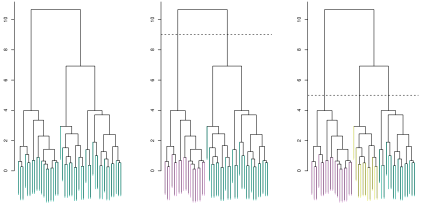


Figure 10.9

- The number of clusters is not fixed.
- Clusterings are nested.
- Dendogram summarizes relationships among clusters.
- Hierarchical clustering is not always appropriate.

  e.g. Market segmentation for consumers of 3 different nationalities.

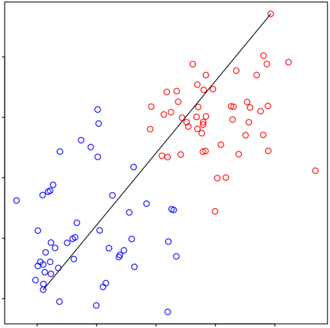
  - Natural 2 clusters: gender
  - Natural 3 clusters: nationality

  These clusterings are not nested or hierarchical.

Notion of distance between clusters

At each step, we link the 2 clusters that are "closest" to each other.

Hierarchical clustering algorithms are classified according to the notion of distance between clusters.

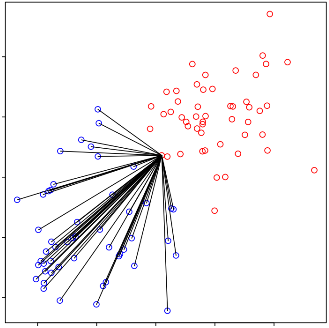


Complete linkage:
The distance between 2 clusters is the *maximum* distance between any pair of samples, one in each cluster.

Notion of distance between clusters

At each step, we link the 2 clusters that are “closest” to each other.

Hierarchical clustering algorithms are classified according to the notion of distance between clusters.

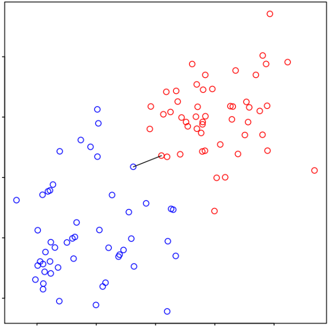


Average linkage:
The distance between 2 clusters is the average of all pairwise distances.

Notion of distance between clusters

At each step, we link the 2 clusters that are "closest" to each other.

Hierarchical clustering algorithms are classified according to the notion of distance between clusters.



Single linkage:
The distance between 2 clusters is the *minimum* distance between any pair of samples, one in each cluster.

Suffers from the *chaining phenomenon*: May produce long, spread out clusters with dissimilar points in same cluster.

Example

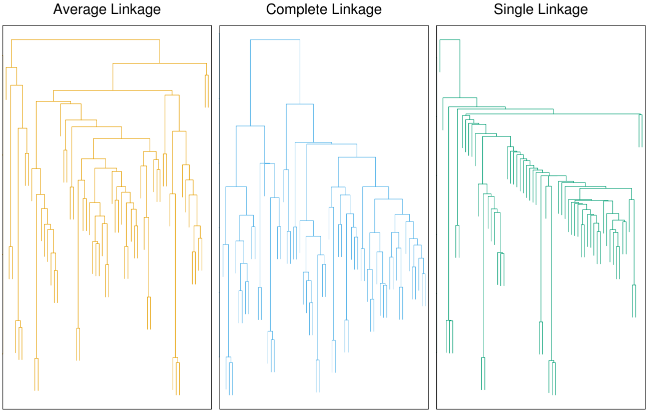


Figure 10.12

Clustering is riddled with questions and choices

- Is clustering appropriate? i.e. Could a sample belong to more than one cluster?
  - Mixture models, soft clustering, topic models.
- How many clusters are appropriate?
  - Choose subjectively — depends on the application.
  - There are formal methods based on gap statistics, mixture models, etc.
- Are the clusters robust?
  - Run the clustering on different random subsets of the data. Is the structure preserved?
  - Try different clustering algorithms. Are the conclusions consistent?
  - Most important: temper your conclusions.

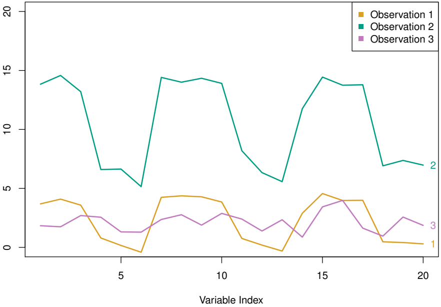
Clustering is riddled with questions and choices

- Should we scale the variables before doing the clustering.
  - Variables with larger variance have a larger effect on the Euclidean distance between two samples.
- Does Euclidean distance capture dissimilarity between samples?

Correlation distance

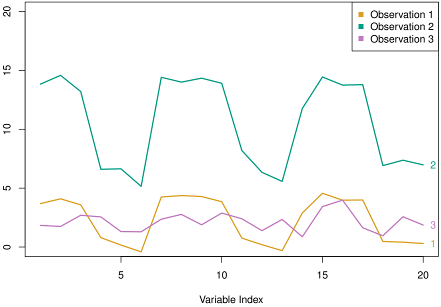
Example: Suppose that we want to cluster customers at a store for market segmentation.

- Samples are customers
- Each variable corresponds to a specific product and measures the number of items bought by the customer during a year.



Correlation distance

- Euclidean distance would cluster all customers who purchase few things (orange and purple).
- Perhaps we want to cluster customers who purchase *similar* things (orange and teal).
- Then, the **correlation distance** may be a more appropriate measure of dissimilarity between samples.


Observation 1
Observation 2
Observation 3
Variable Index