Unsuperivised Learning

Chapter 10

April 17, 2018

1. 10.1 Principal component analysis.

2. 10.2. Clustering methods

About this chapter

- Supervised learning aims to predicting ouput/response from features, and each training data point contains the input and output, $(\mathbf{x}_i, y_i)$.
- Unsupervised learning do not have a specific output or response to predict, i.e. only $(\mathbf{x}_i)$. It aims at discovering the structure or characteristic of the variables in study.
- This chapter focuses on two methodologies: PCA and cluster analysis.

Unsupervised learning

- Often used as part of exploratory data analysis.
- Examples: Identify the particular subgroups of stocks with close relations (clustering),
  Accurate advertising based on the customers age, profession, reading, shopping habits, (clustering) ...
  Reduce the dimension of covariates (PCA).
  ....

- We have already addressed PCA in Chapter 6.
- Linearly combine the variables to create the new variables, called principle components.
- The first few explain most of the variation.
- Achieve data reduction, without much loss of information.
- Here we summarize the results and look at the examples.

- Data: $\mathbf{x}_i = (x_{i1}, ..., x_{ip})$, $i = 1, ..., n$.
- Compute sample covariance matrix, e.g. $\mathbf{S} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \hat{\mu})^T(\mathbf{x}_i - \hat{\mu})$.
- Decompose into eigenvalue-eigenvector pairs:

$$\mathbf{S} = \hat{\mathbf{e}}\hat{\Lambda}\hat{\mathbf{e}}^T = (\hat{\mathbf{e}}_1 \vdots ... \vdots \hat{\mathbf{e}}_p)\hat{\Lambda} \begin{pmatrix} \hat{\mathbf{e}}_1 \\ \vdots \\ \hat{\mathbf{e}}_p \end{pmatrix}$$

  where $\hat{\Lambda} = diag(\hat{\lambda}_1, ..., \hat{\lambda}_p)$.
- $(\hat{\lambda}_k, \hat{\mathbf{e}}_k)$ are eigen-value-eigenvector pairs, $\hat{\lambda}_1 \geq ... \geq \hat{\lambda}_p$.

- The k-th sample PC.s:

$$Z_k = \begin{pmatrix} z_{1k} \\ \vdots \\ z_{nk} \end{pmatrix} = \mathbf{X}\hat{\mathbf{e}}_k$$

- Component-wise, $z_{ik} = x_{i1}e_{1k} + x_{i2}e_{2k} + ... + x_{ip}e_{pk}$ are the principle component scores of the i-th observation.
- $\hat{\lambda}_k$ measures the importance of the k-th PC.
- $\hat{\lambda}_k/(\hat{\lambda}_1 + ... + \hat{\lambda}_p) = \hat{\lambda}_k/trace(\mathbf{S})$ is interpreted as percentage of the total variation explained by Y_k.
- Usually retain the first few PCs.
- PCs are uncorrelated with each other.

Example: USArrests data

For each of the 50 states in the United States, the data set contains the number of arrests per 100, 000 residents for each of three crimes: Assault, Murder, and Rape.
We also record UrbanPop (the percent of the population in each state living in urban areas).
The principal component score vectors Z_k have length $n = 50$, and the principal component loading vectors ($\hat{\mathbf{e}}_k$) have length $p = 4$.
PCA was performed after standardizing each variable to have mean zero and standard deviation one.

Example: USArrests data

	PC1	PC2
Murder	0.5358995	0.4181809
Assault	0.5831836	0.1879856
UrbanPop	0.2781909	0.8728062
Rape	0.5434321	0.1673186

Table 10.1. The principal component loading vectors, $\hat{\mathbf{e}}_1$ and $\hat{\mathbf{e}}_2$, for the USArrests data. These are also displayed in Figure 10.1.

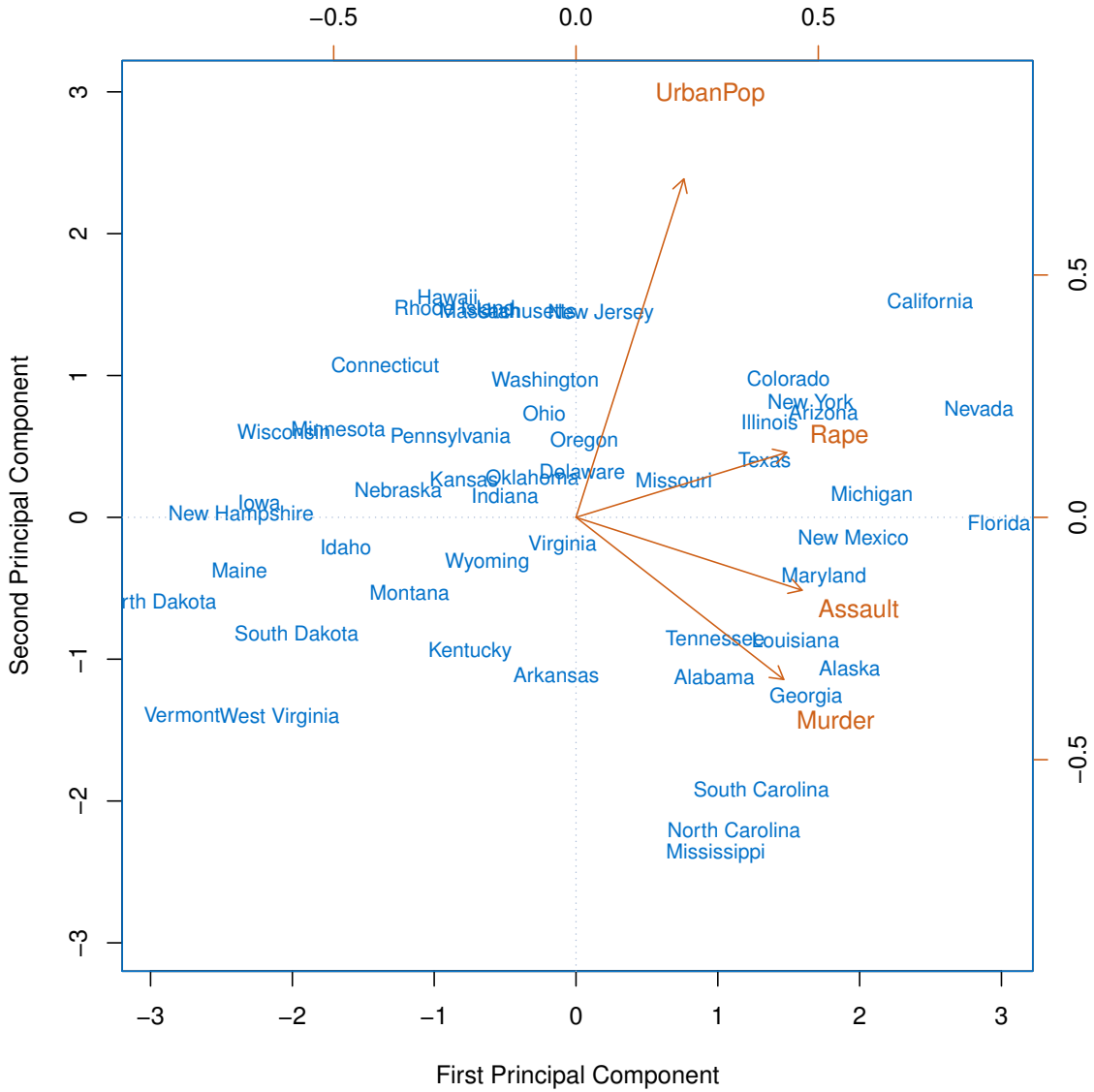


Figure: 10.1. Next page

Figure 10.1

Figure 10.1. The first two principal components for the USArrests data. The blue state names represent the scores for the first two principal components. The orange arrows indicate the first two principal component loading vectors (with axes on the top and right). For example, the loading for Rape on the first component is 0.54, and its loading on the second principal component 0.17 (the word Rape is centered at the point (0.54, 0.17)). This figure is known as a biplot, because it displays both the principal component scores and the principal component loadings.

The 1st and 2nd PCs

- The first loading vector places approximately equal weight on Assault, Murder, and Rape, with much less weight on UrbanPop
  This component roughly corresponds to a measure of overall rates of serious crimes.
- The second loading vector places most of its weight on UrbanPop and much less weight on the other three features
  This component roughly corresponds to the level of urbanization of the state.

Discussion

- The crime-related variables (Murder, Assault, and Rape) are located close to each other, and that the UrbanPop variable is far from the other three.
- This indicates that the crime-related variables are correlated with each other—states with high murder rates tend to have high assault and rape rates—-and that the UrbanPop variable is less correlated with the other three.

Discussion

Our discussion of the loading vectors (PC1 roughly about crime rates and PC2 about urbanization) suggests:
States with large positive scores on the first component, such as California, Nevada and Florida, have high crime rates;
States like North Dakota, with negative scores on the first component, have low crime rates.
California also has a high score on the second component, indicating a high level of urbanization, while the opposite is true for states like Mississippi.
States close to zero on both components, such as Indiana, have approximately average levels of both crime and urbanization. 10.2.2

Interpreation of PCs

- Seek one direction say $\mathbf{b}$ with $\|\mathbf{b}\| = \mathbf{1}$, such that

$$\sum_{i=1}^{n} \|\mathbf{x}_i - a_i \mathbf{b}\|$$

  is the smallest. This direction is $\mathbf{b} = \hat{\mathbf{e}}_1$. And $a_i = \mathbf{x}_i^T \mathbf{b} =< \mathbf{x}_i, \mathbf{b} >$ is the score of the i-th observation on the 1st PC.

- Wish to minimize rescontruction error in reconstructing all $\mathbf{x}_i$ using vectors restricted to dimension k linear space.
  Then, the this linear space is the space spanned by $\hat{\mathbf{e}}_1, ..., \hat{\mathbf{e}}_k$, the directions of the first k PCs.
  They can be interpreted as the closest k dimension linear hyperplanes to the data.

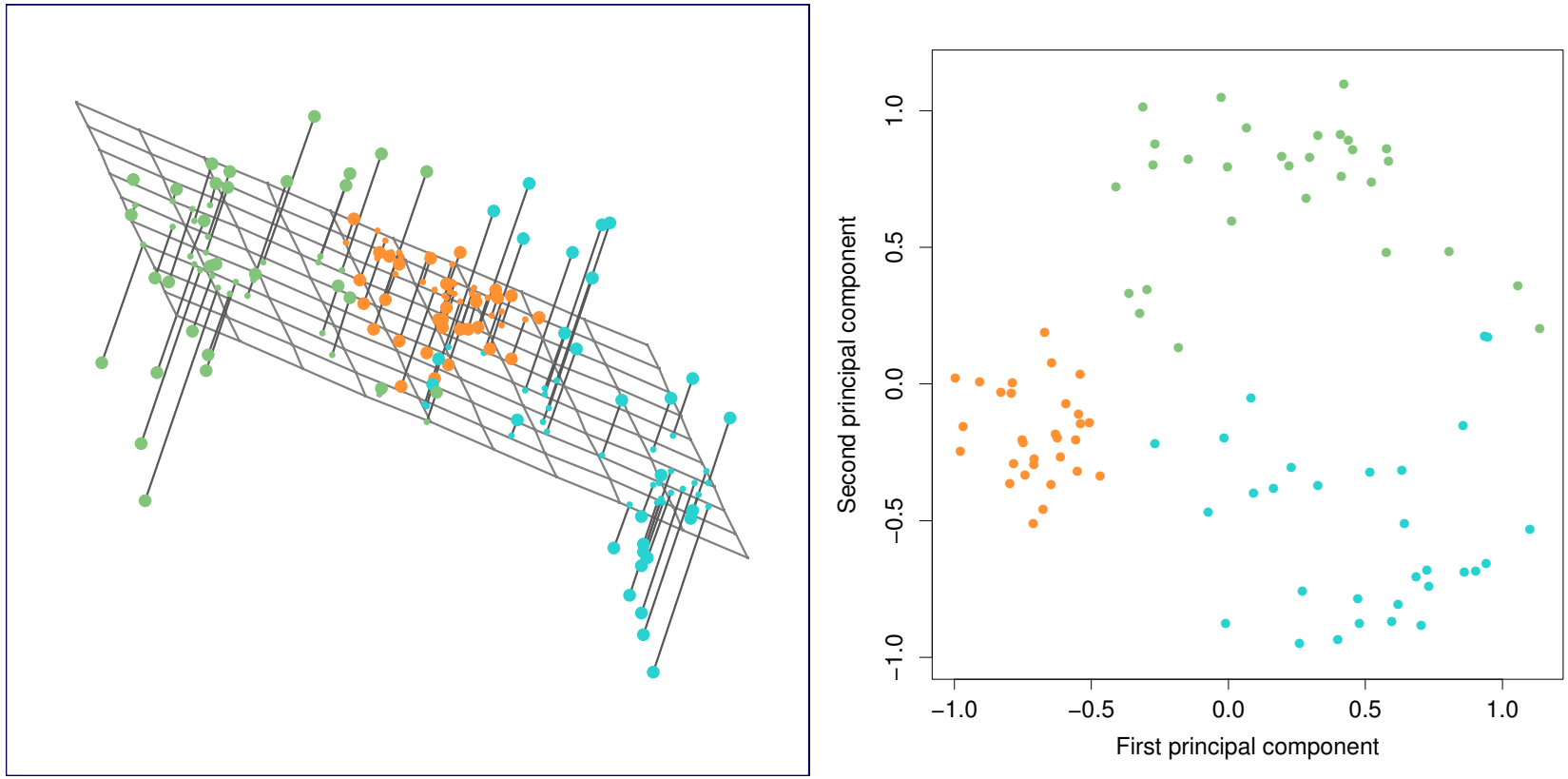


Figure: 10.2. Ninety observations simulated in three dimensions. Left: the first two principal component directions span the plane that best fits the data. It minimizes the sum of squared distances from each point to the plane. Right: the first two principal component score vectors give the coordinates of the projection of the 90 observations onto the plane. The variance in the plane is maximized.

Other issues about PCA

- Scaling:
- If the variables are measuared in different units, rescale to variables to have mean 0 and variance 1 is recommended.
- If the variables are of same unit and same nature (such as stock returns), PCA with both rescaled and original can be conducted.

USArrest data

- In USArrest data, The four variables are measured in different unites.
  Murder, Rape, and Assault are reported as the number of occurrences per 100, 000 people, and UrbanPop is the percentage of the states population that lives in an urban area. These four variables have variance 18.97, 87.73, 6945.16, and 209.5
- Assault, a more common crime than murder and rape, have much larger variance. Without standardization, it is expected to contribute much to 1st PC.

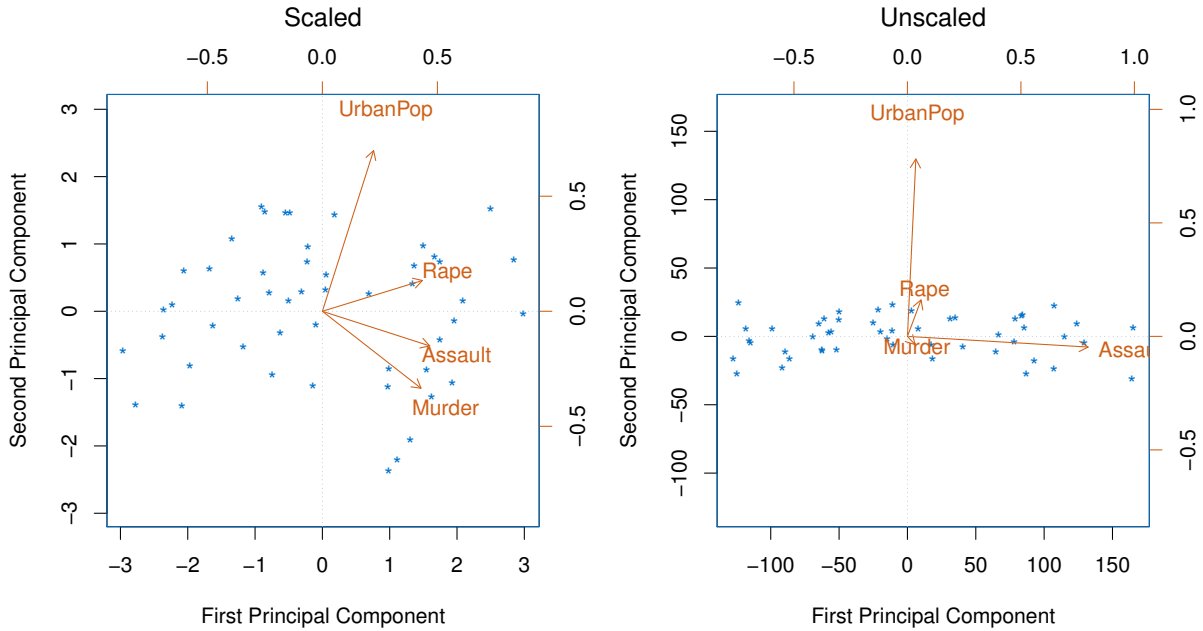


Figure: 10.3. Two principal component biplots for the USArrests data. Left: the same as Figure 10.1, with the variables scaled to have unit standard deviations. Right: principal components using unscaled data. Assault has by far the largest loading on the first principal component because it has the highest variance among the four variables. In general, scaling the variables to have standard deviation one is recommended.

- Just like eigenvectors, direction of a PC can be reversed. ($\mathbf{e}_1$ is eighvector, so is $-\mathbf{e}_1$.)
- Proportion of variance explained is an importance measure of the PCs.

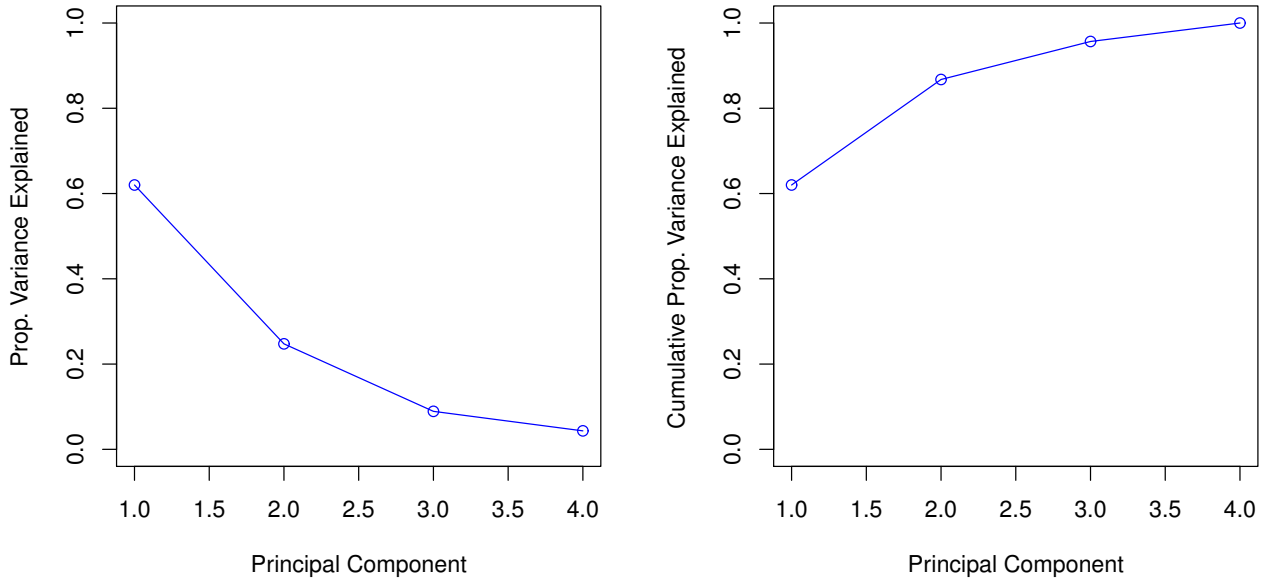


Figure: 10.4. Left: a scree plot depicting the proportion of variance explained by each of the four principal components in the USArrests data. Right: the cumulative proportion of variance explained by the four principal components in the USArrests data.

Number of PCs to retain

- No universal rule.
- Problem specific.
- Recommended:
  1. Look for elbow in the scree plot.
  2. Enough PCs to explain 90% of total variation.
  3. PCs with variance larger than average.
  4. Horn's Parallel Analysis with random permutations (Random Matrix Theory)
- This is useful to regression, classification and cluster analysis to work with the first few PCs rather than all the p inputs. p could be too large and contain many noisy or useless inputs.

About cluster analysis

- Techniques for finding subgroups or data points, or clusters, in a data set, so that the observations within each group are quite similar to each other.
- An unsupervised problem: to discover structure — in this case, distinct clusters — on the basis of a data set.
- Both clustering and PCA seek to simplify the data via a small number of summaries, but their mechanisms are different:
  1. PCA looks to find a low-dimensional representation of the observations that explain a good fraction of the variance;
  2. Clustering looks to find homogeneous subgroups among the observations.

Market segmentation

- The goal is to perform market segmentation by identifying subgroups of people who might be more receptive to a particular form of advertising, or more likely to purchase a particular product.
- The task of performing market segmentation amounts to clustering the people in the data set.

K-means clustering and hierarchical clustering

- Two best-known clustering approaches: K-means clustering and hierarchical clustering.
  1. K-means clustering: seek to partition the observations into a pre-specified number of clusters.
  2. Hierarchical clustering: a tree-like visual representation of the observations, called a dendrogram, that allows us to view at once the clusterings obtained for each possible number of clusters, from 1 to n.
- Cluster observations on the basis of the features in order to identify subgroups among the observations;
  Or cluster features on the basis of the observations in order to discover subgroups

K-Means clustering

- Partition the data set of n observations into K distinct, non-overlapping subsets.
  Each set, denoted as C_k, $k = 1, .., K$, is called a cluster.
- Good clustering: the within-cluster variation is as small as possible
- Let $W(C_k)$ be a measure of the within-cluster variation for cluster C_k.
- We wish to minimize the total within-cluster variations

$$\text{minmize}_{C_1,\ldots,C_K} \left\{ \sum_{i=1}^{K} W(C_K) \right\}$$

K-Means clustering

- Several different ways to define $W(C_k)$.
- Using squared Euclidan distance, we define

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,j \in C_k} \|\mathbf{x}_i - \mathbf{x}_j\|^2$$

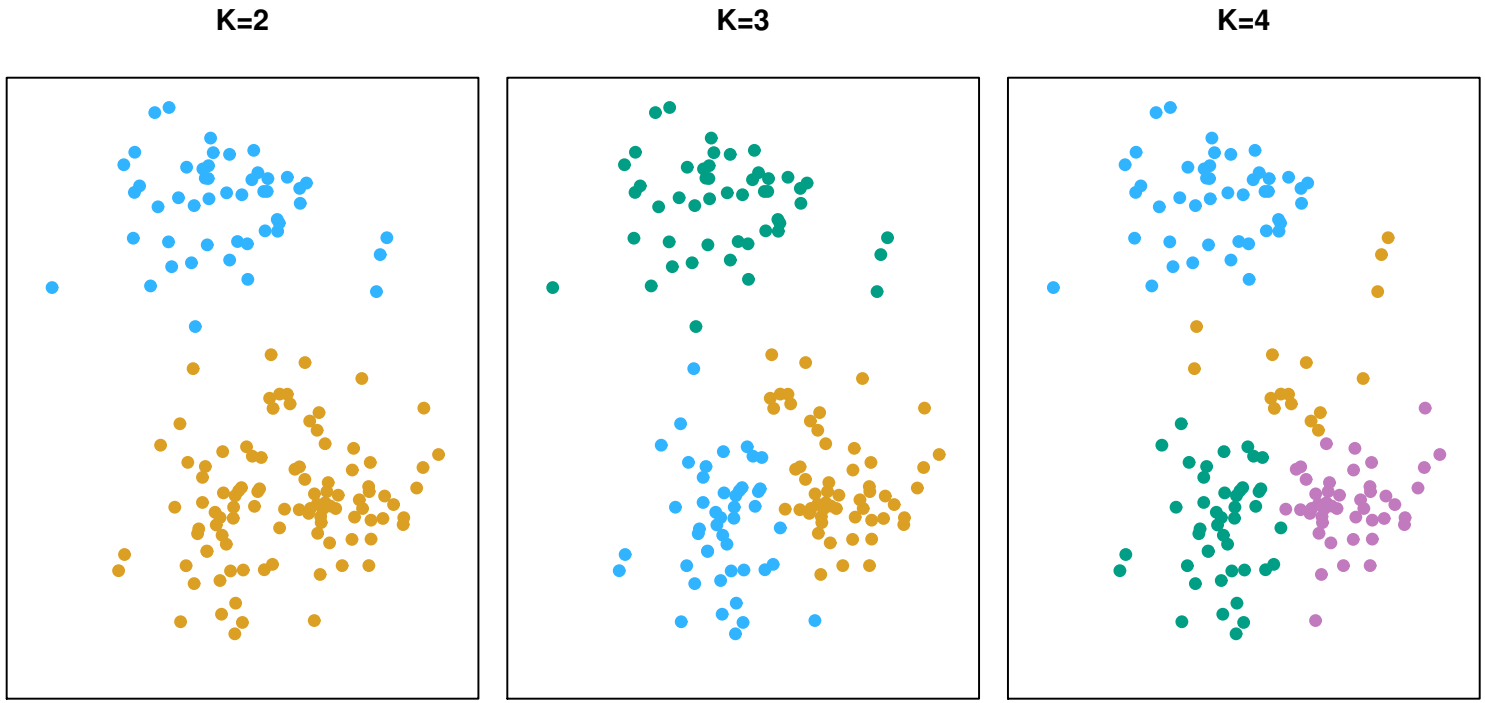


Figure: 10.5. A simulated data set with 150 observations in two-dimensional space. Panels show the results of applying K-means clustering with different values of K, the number of clusters. The color of each observation indicates the cluster to which it was assigned using the K-means clustering algorithm. 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 cluster algorithm

Algorithm 10.1 K-Means Clustering

- 1. Randomly assign a number, from 1 to K, to each of the observations. These serve as initial cluster assignments for the observations.
- 2. Iterate until the cluster assignments stop changing:
  1. For each of the K clusters, compute the cluster centroid. The kth cluster centroid is the vector of the p feature means for the observations in the kth cluster.
  2. Assign each observation to the cluster whose centroid is closest (where closest is defined using Euclidean distance).

K-Means cluster algorithm

- The objective function always decreases at each step.
- $$\frac{1}{|C_k|}\sum_{i,j\in C_k}\|\mathbf{x}_i-\mathbf{x}_j\|^2 = 2\sum_{i\in C_k}\|\mathbf{x}_i-\bar{\mathbf{x}}\|^2$$

  where $\bar{\mathbf{x}}$ is the sample mean $\mathbf{x}_i$ for $i \in C_k$
- K-means algorithm finds a local minimum.
- The result depends on the initial (random) cluster assignment.
- Try several different initials, and select the best result (the smallest objective function).

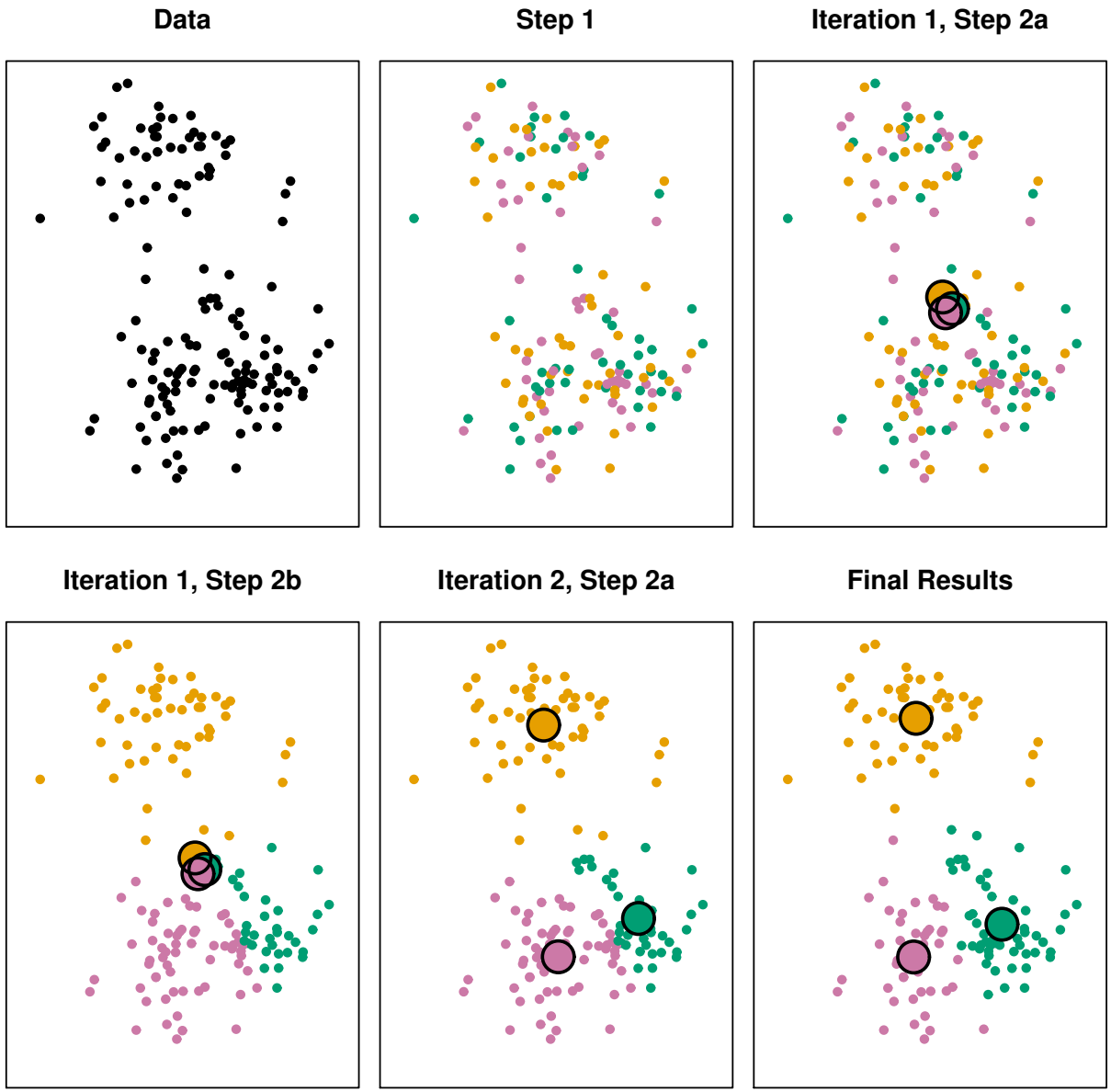

Data
Step 1
Iteration 1, Step 2a
Iteration 1, Step 2b
Iteration 2, Step 2a
Final Results

Figure: 10.6

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.

Figure: 10.7

FIGURE 10.7. K-means clustering performed six times on the data from Figure 10.5 with $K = 3$, each time with a different random assignment of the observations in Step 1 of the K-means algorithm. Above each plot is the value of the objective (10.11). Three different local optima were obtained, one of which resulted in a smaller value of the objective and provides better separation between the clusters. Those labeled in red all achieved the same best solution, with an objective value of 235.8.

Hierarchical clustering

- K-means clustering requres pre-specified number of clusters, a disadvantage.
- Hierarchical clustering does not require that.
- It results in a tree-based representation of the observations, called a dendrogram.
- *bottom-up* or *agglomerative* clustering

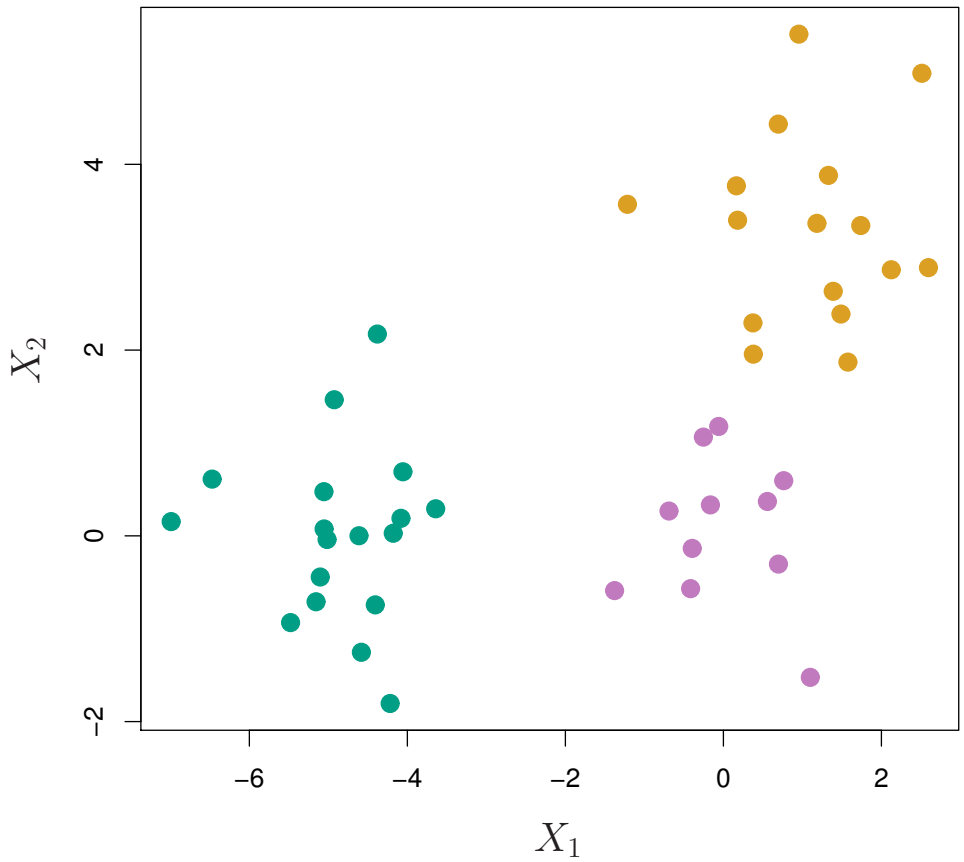


Figure: 10.8. Forty-five observations generated in two-dimensional space. In reality there are three distinct classes, shown in separate colors. However, we will treat these class labels as unknown and will seek to cluster the observations in order to discover the classes from the data.

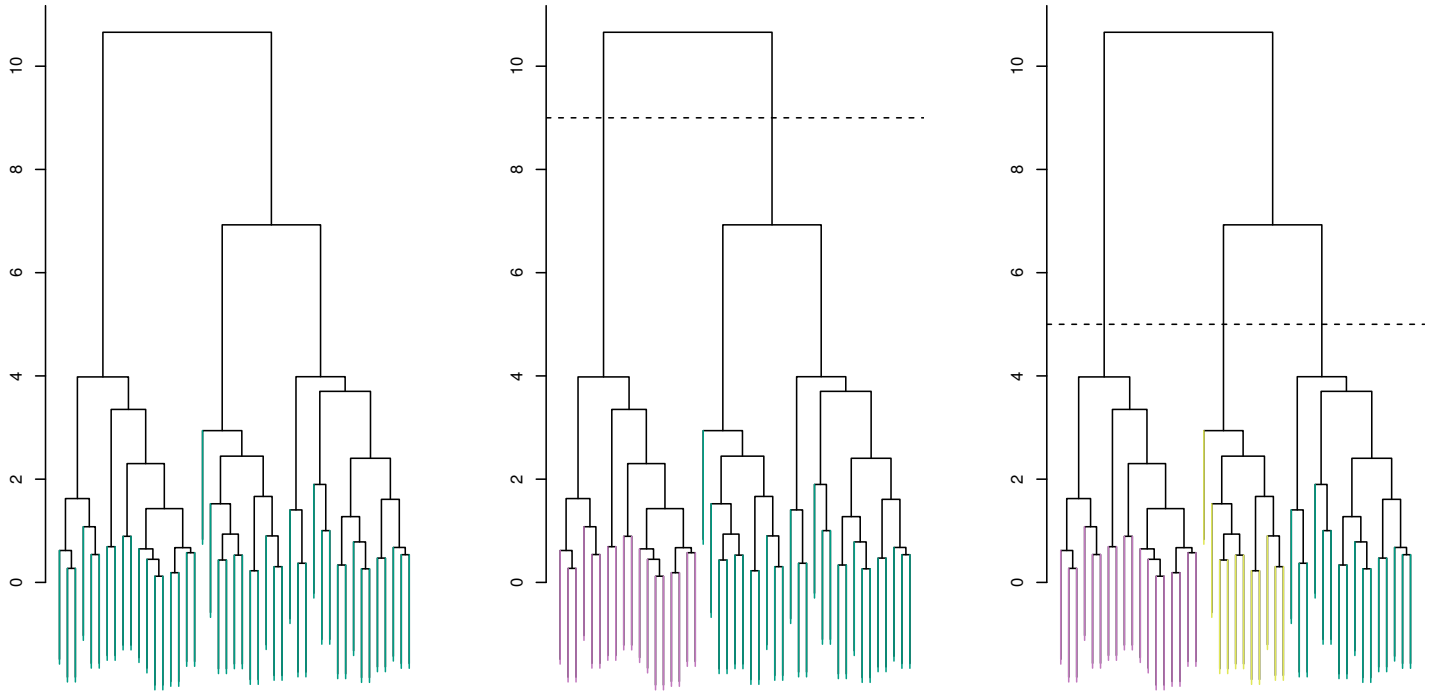


Figure: 10.9. Left: dendrogram obtained from hierarchically clustering the data from Figure 10.8 with complete linkage and Euclidean distance. Center: the dendrogram from the left-hand panel, cut at a height of nine (indicated by the dashed line). This cut results in two distinct clusters, shown in different 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.

Interpreting a dendrogram

- Each leaf of the dendrogram represents one of the 45 observations in Figure 10.8.
- However, as we move up the tree, some leaves begin to fuse into branches. These correspond to observations that are similar to each other.
- As we move higher up the tree, branches themselves fuse, either with leaves or other branches.
- The earlier (lower in the tree) fusions occur, the more similar the groups of observations are to each other.
- Observations that fuse later (near the top of the tree) can be quite different

A rough closeness measure

- For any two observations, we can look for the point in the tree where branches containing those two observations are first fused. The height of this fusion, as measured on the vertical axis, indicates how different the two observations are
- Observations that fuse at the very bottom of the tree are quite similar to each other, whereas observations that fuse close to the top of the tree will tend to be quite different.

Interpreting dendrogram

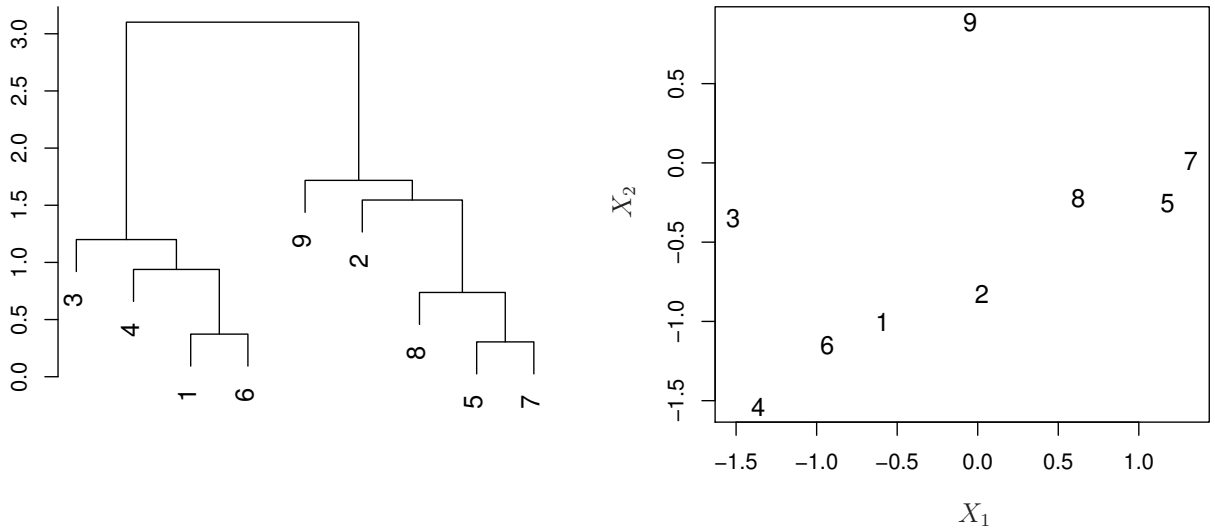


Figure: 10.10. An illustration of how to properly interpret a dendrogram with nine observations in two-dimensional space. Left: a dendrogram generated using Euclidean distance and complete linkage. Observations 5 and 7 are quite similar to each other, as are observations 1 and 6. However, observation 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. Now

Identifying clusters

- Make a horizontal cut across the dendrogram, as Figure 10.9
- The distinct sets of observations beneath the cut can be interpreted as clusters.
- The lower cuts create more clusters. The higher cuts create less clusters.
- One single dendrogram can be used to obtain any number of clusters.
- Choice of cuts can even be done by visual judgment of the dendrogram.
- When hierarchical structure does not exist in data, the hirarchical clustering could be worse than K-means clustering.

Hierarchical clustering algorithm

- 1. Begin with n observations and a measure (such as Euclidean distance) of all the $\binom{n}{2} = n(n-1)/2$ pairwise dissimilarities. Treat each observation as its own cluster.
- 2. For $i = n, n-1, ...2$:
  1. 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 two clusters indicates the height in the dendrogram at which the fusion should be placed.
  2. Compute the new pairwise inter-cluster dissimilarities among the $i-1$ remaining clusters.

Linkage: the dissimilarity measure between two clusters

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.

TABLE 10.2. A summary of the four most commonly-used types of linkage

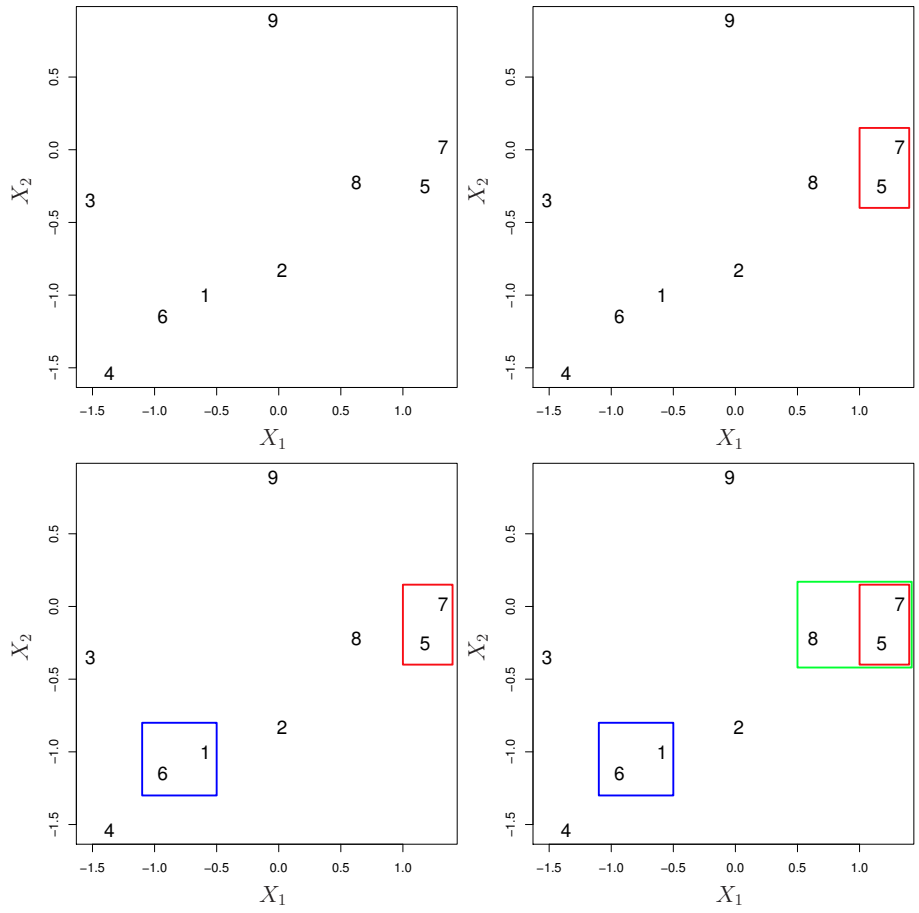


Figure: 10.11. An illustration of the first few steps of the hierarchical clustering algorithm, using the data from Figure 10.10, with complete linkage and Euclidean distance. Top Left: initially, there are nine distinct clusters, $\{1\}, \{2\}, \ldots, \{9\}$. Top Right: the two clusters that are closest 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.

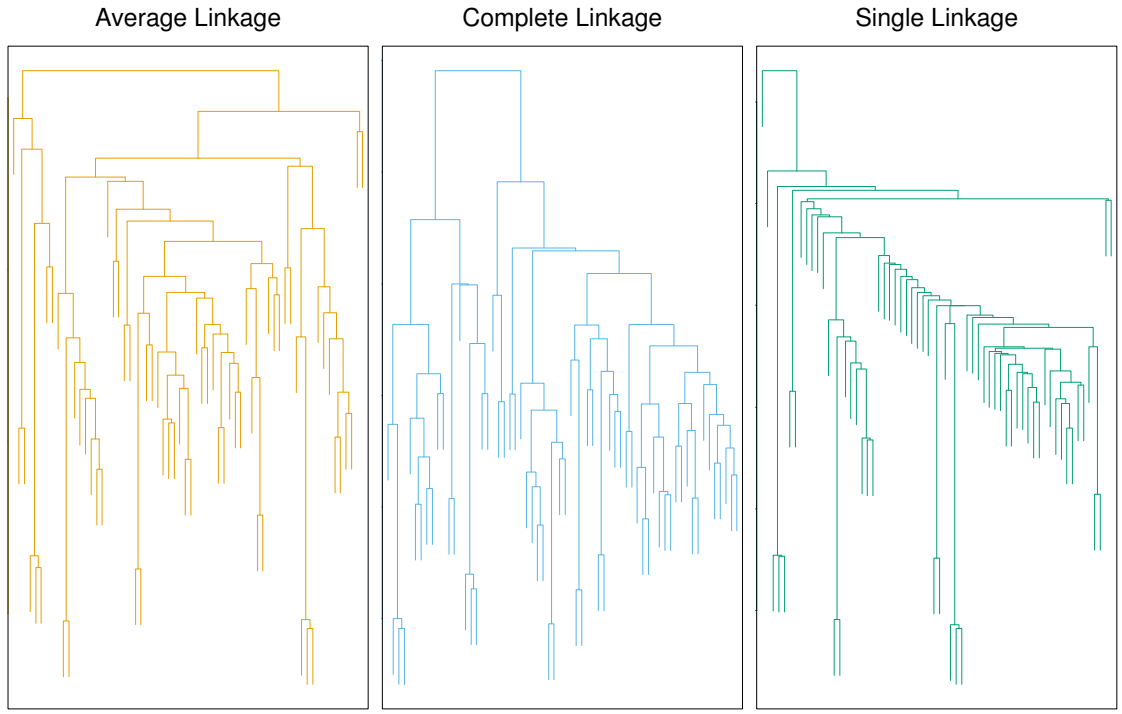


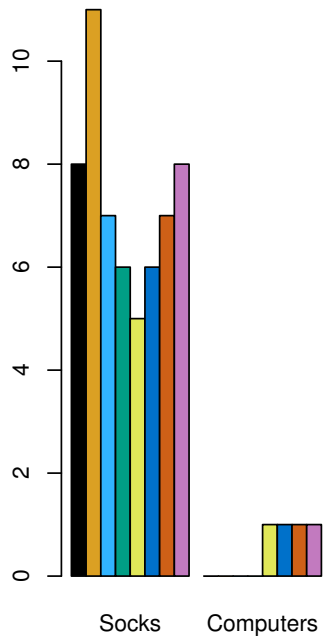
Figure: 10.12. Average, complete, and single linkage applied to an example data set. Average and complete linkage tend to yield more balanced clusters.

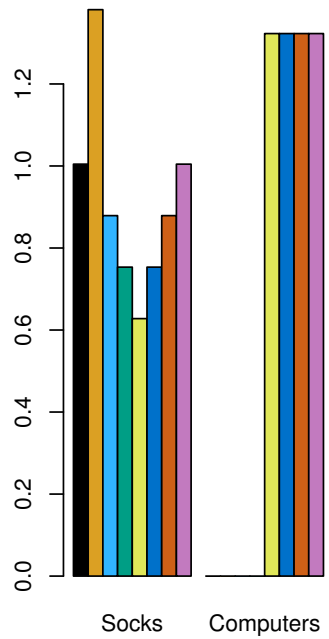
Dissimilarity measure

- Very important and can greatly affect the final result.
- Euclidean distrance.
- Correlation based distrance: if two observations have high correlation, the distance is closer.
  (Caution: this is not correlation between two variables, but between two observations.)
- Different problem may need different dissimilarity measure.

The online shopping example

- Using Euclidean distance may not be appropriate.
  Those with same shopping habit but different shopping volume should be but may not be clustered together.
- Using correlation based distrance is more appropriate.
- Variable scaling, as in PCA, whether the variables should be standardized is problem specific.
  Example: High-frequency purchases like socks therefore tend to have a much larger effect on the inter-shopper dissimilarities, and hence on the clustering ultimately obtained, than rare purchases like computers. This may not be desirable.
- Variables measured in different units should be standardized.


0
2
4
6
8
10
Socks
Computers


0.0
0.2
0.4
0.6
0.8
1.0
1.2
Socks
Computers

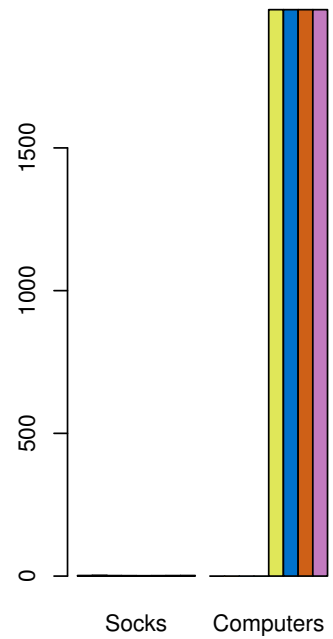

0
500
1000
1500
Socks
Computers

FIGURE 10.14. An online retailer sells two items: socks and computers. Left: the number of pairs of socks, and computers, purchased by eight online shoppers is displayed. Each shopper is shown in a different color. If inter-observation dissimilarities are computed using Euclidean distance on the raw variables, then the number of socks purchased by an individual will drive the dissimilarities obtained, and the number of computers purchased will have little effect. This might be undesirable, since (1) computers are more expensive than socks and so the online retailer may be more interested in encouraging shoppers to buy computers than socks, and (2) a large difference in the number of socks purchased by two shoppers may be less informative about the shoppers overall shopping preferences than a small difference in the number of computers purchased. Center: the same data is shown, after scaling each variable by its standard deviation. Now the number of computers purchased will have a much greater effect on the inter-observation dissimilarities obtained. Right: the same data are displayed, but now the y-axis represents the number of dollars spent by each online shopper on socks and on computers. Since computers are much more expensive than socks, now computer purchase history will drive the inter-observation dissimilarities obtained.

- Should the observations or features first be standardized in some way? For instance, maybe the variables should be centered to have mean zero and scaled to have standard deviation one.
- In the case of hierarchical clustering,
  What dissimilarity measure should be used?
  What type of linkage should be used?
  Where should we cut the dendrogram in order to obtain clusters?
- In the case of K-means clustering, how many clusters should we look for in the data?

- Forcing every observation, including outliers, into clusters, can distort the final outcome. (A soft version of K-means clustering by mixture model may help)
- Clustering methods generally are not very robust to perturbations to the data.
- Performing clustering with different choices of these parameters (linkage, standardization or not, etc), and looking at the full set of results
- Clustering subsets of the data in order to get a sense of the robustness

Exercises

Run the R-Lab codes in Section 10.4 of ISLR
Exercises 1-3 and 10 of Section 10.7 of ISLR

End of Chapter 10.