Unsupervised Learning (Chapter 10)

Set up: Have n independent observations on p variables (or features) $X_1, \ldots, X_p$, stored in a $n \times p$ matrix $\mathbf{X}$, but there is no response Y that *supervises* how well a method is doing.

Two possible goals: (More like **exploratory** data analysis)

- Get a low dimensional representation of data ($M < p$) while preserving much of the variation. Helps visualize the data and also offers a pre-processing before applying supervised techniques — **PCA**
- Partition observations into homogeneous subgroups so that observations within a subgroup are quite similar to each other and observations in different subgroups are quite different from each other — **clustering**

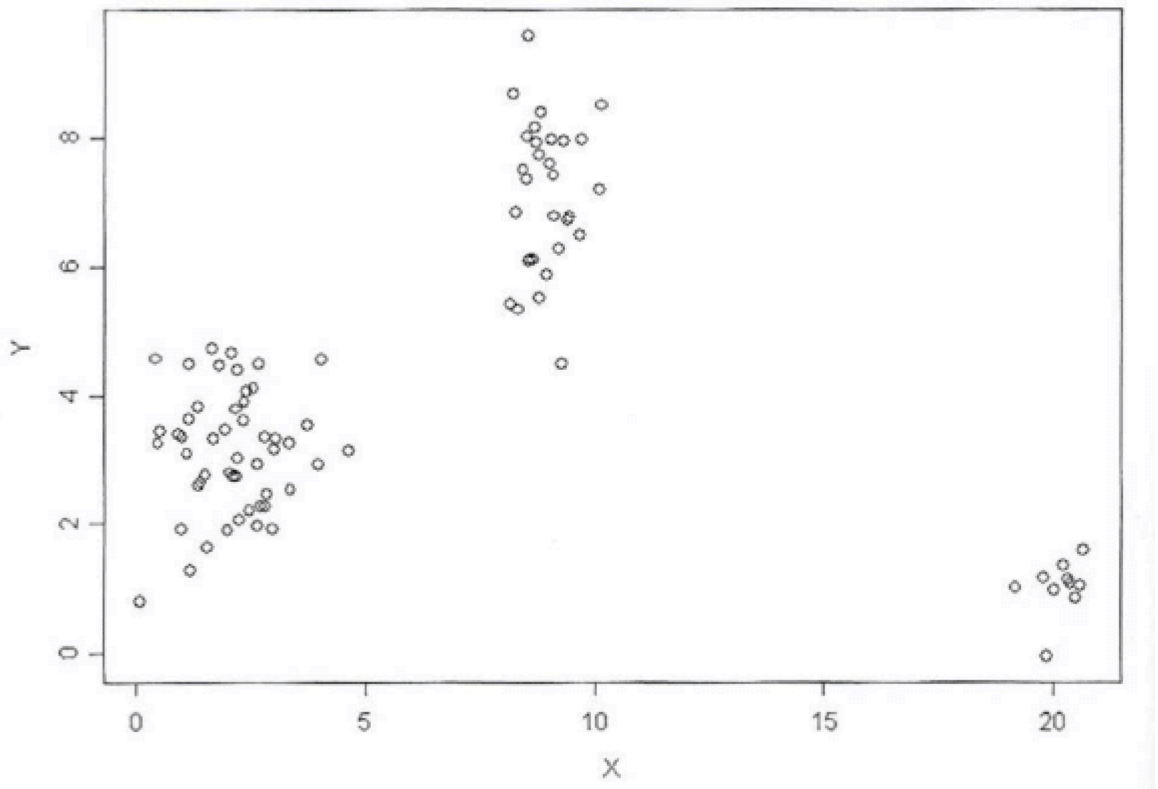
Clustering (or Segmentation)

We can cluster *observations* on the basis of features to identify subgroups among the observations, or we can cluster *features* on the basis of the observations to identify subgroups among the features. We will consider clustering observations. The converse can be performed by simply transposing the data matrix.

Classification vs clustering: In classification, the # of groups are *known* in advance (a supervised technique). With clustering, we hope to discover subgroups that may exist.

Our focus: Two of the best-known methods for clustering:

- K-means clustering — # clusters is fixed in advance
- Hierarchical clustering — # clusters is not fixed in advance, rather we get a tree-like representation of the observations, called a *dendogram*, from which we may deduce the clusters



Q: How many clusters exist in this figure? **A:** 3

K-Means Clustering

We specify K — the desired # clusters — and the method assigns each observation to exactly one of the K clusters. We would like to form the clusters in such a way that the observations within the same cluster are as **similar** as possible.

Formulation of the problem: Let $C_1, \dots, C_K$ denote sets containing indices of the observations in the K clusters. These sets satisfy two properties:

- $C_1 \cup \ldots \cup C_K = \{1, \dots, n\}$ — each observation belongs to at least one cluster
- $C_k \cap C_{k'} = \{\}$ for all $k \neq k'$ — the clusters do not overlap, i.e., no observation belongs to more than one cluster.

We will collectively refer to these sets as the partition C.

Measure of dissimilarity: Consider two observations $X_i = (X_{i1}, \ldots, X_{ip})^T$ and $X_l = (X_{l1}, \ldots, X_{lp})^T$ and let $d(X_i, X_l)$ be the **squared Euclidean distance** between them, i.e.,

$$d(X_i, X_l) = \sum_{j=1}^{p} (X_{ij} - X_{lj})^2.$$

Suppose there are n_k observations in cluster k and their sample mean — aka **centroid** of cluster k — is $\overline{X}_k = (\overline{X}_{k1}, \ldots, \overline{X}_{kp})^T$. Define a measure of **within-cluster dissimilarity** as

$$W(C_k) = \frac{1}{2n_k} \sum_{i \in C_k} \sum_{l \in C_k} d(X_i, X_l) = \sum_{i \in C_k} d(X_i, \overline{X}_k),$$

which represents the **within-cluster SS** of cluster k.

Summing over k gives **total within-cluster SS**,

$$W(C) = \sum_{k=1}^{K} W(C_k)$$

which is an **overall** measure of within-cluster dissimilarity for the given partition C. It will be small when the observations within the clusters are close together.

Next, let $\overline{X}$ be the overall sample mean. The **total SS** is

$$T = \sum_{i=1}^{n} d(X_i, \overline{X}) = \frac{1}{2n} \sum_{i=1}^{n} \sum_{l=1}^{n} d(X_i, X_l).$$

Note that T does not depend on the partition. This gives the **between-cluster SS** for the partition C as

$$B(C) = T - W(C).$$

To find the clusters that **minimize** the total within-cluster SS, we want to solve the optimization problem:

$$\min_{C_1,\ldots,C_K} W(C).$$

Since $W(C) = T - B(C)$, this is equivalent to solving:

$$\max_{C_1,\ldots,C_K} B(C).$$

Thus, the clusters that minimize the total within-cluster SS also maximize the between-cluster SS.

- Difficult as # ways to partition n observations into K clusters $\approx K^n$
- **Exact # ways:** $\frac{1}{K!}\sum_{k=1}^{K}(-1)^{K-k}\binom{K}{k}k^n$. For example, with $(n, K) = (19, 4)$, this $\approx 10^{10}$.
- ***K*-means method**: A simple algorithm to find a **local** optimum.

The minimization of $W(C) = \sum_{k=1}^{K} \sum_{i \in C_k} d(X_i, \overline{X}_k)$ wrt C is equivalent to the minimization of

$$\sum_{k=1}^{K} \sum_{i \in C_k} d(X_i, m_k) \tag{1}$$

wrt both C and $\{m_1, \ldots, m_K\}$ because, for a given C, (1) is minimized wrt $\{m_1, \ldots, m_K\}$ when $m_k = \overline{X}_k$. This suggests the following **iterative descent** method for the optimization:

1. Start with an initial cluster assignment C.
2. Iterate until the cluster assignments stop changing:
   (a) For the current C, minimize (1) wrt $\{m_1, \ldots, m_K\}$ by computing $m_k = \overline{X}_k$.
   (b) For the current $\{m_1, \ldots, m_K\}$, minimize (1) wrt C by assigning each observation to the closest cluster mean. In other words, for $i = 1, \ldots, n$, assign observation i to the cluster k for which $d(X_i, m_k)$ is smallest.

This is the ***K*-means method**. Each of Steps 2a and 2b reduces the value of (1) — convergence is assured.

Q: How to make the initial assignment C?

***K*-means clustering**: (A simpler way to write the algorithm)

1. Randomly assign a number, from 1 to K, to each of the observations. This serves as the initial cluster assignment.
2. Iterate until the cluster assignments stop changing:
   (a) For each of the K clusters, compute the cluster mean
   (b) Proceed through the list of observations, assigning an observation to the cluster whose mean is nearest.

- This method yields a **local** optimum — local in the sense that the final result depends on the initial assignment.
- Run this algorithm with a large number of different random initial assignments, and choose the **best** solution, i.e., for which the objective (1) is smallest.
- How to choose K? Run with different K and choose the *elbow* in the plot of optimal value of the objective (1). Alternatively, look at the resulting clusters for different K.

Ex: Suppose we have the following $n = 4$ observations on $p = 2$ features and we would like to group them into $K = 2$ clusters.

ID	x_1	x_2
A	5	3
B	-1	1
C	1	-2
D	-3	-2

Iteration 1:

Step 1: Suppose the initial clusters are (AB) and (CD).

Step 2(a): Compute the cluster means

cluster	$\overline{x}_1$	$\overline{x}_2$
(AB)	$\frac{5-1}{2} = 2$	$\frac{3+1}{2} = 2$
(CD)	$\frac{1-3}{2} = -1$	$\frac{-2-2}{2} = -2$

Step 2(b): Proceed through each observation to check its squared distance with the cluster means and reassign if needed

Observation 1:

$$d(A,(AB)) = (5-2)^2 + (3-2)^2 = 10$$
$$d(A,(CD)) = (5+1)^2 + (3+2)^2 = 61$$

$\Longrightarrow$ Keep A in (AB) since A is closer to (AB) than (CD).

Observation 2:

$$d(B,(AB)) = (-1-2)^2 + (1-2)^2 = 10$$
$$d(B,(CD)) = (-1+1)^2 + (1+2)^2 = 9$$

$\Longrightarrow$ Reassign B to (CD).

Observation 3:

$$d(C,(AB)) = (1-2)^2 + (-2-2)^2 = 17$$
$$d(C,(CD)) = (1+1)^2 + (-2+2)^2 = 4$$

$\Longrightarrow$ Keep C in (CD).

Observation 4:

$$d(D,(AB)) = (-3-2)^2 + (-2-2)^2 = 41$$
$$d(D,(CD)) = (-3+1)^2 + (-2+2)^2 = 4$$

$\Longrightarrow$ Keep D in (CD).

Thus, at the end of iteration 1, the two clusters are (A) and (BCD).

Iteration 2:

Step 2(a): Update the cluster means

cluster	$\overline{x}_1$	$\overline{x}_2$
(A)	5	3
(BCD)	-1	-1

Step 2(b): Get the squared distances to the cluster means

cluster	A	B	C	D
(A)	0	40	41	89
(BCD)	52	4	5	5

$\Longrightarrow$ Stop iterating since each observation is assigned to the nearest cluster mean. Thus, the final clusters are (A) and (BCD).

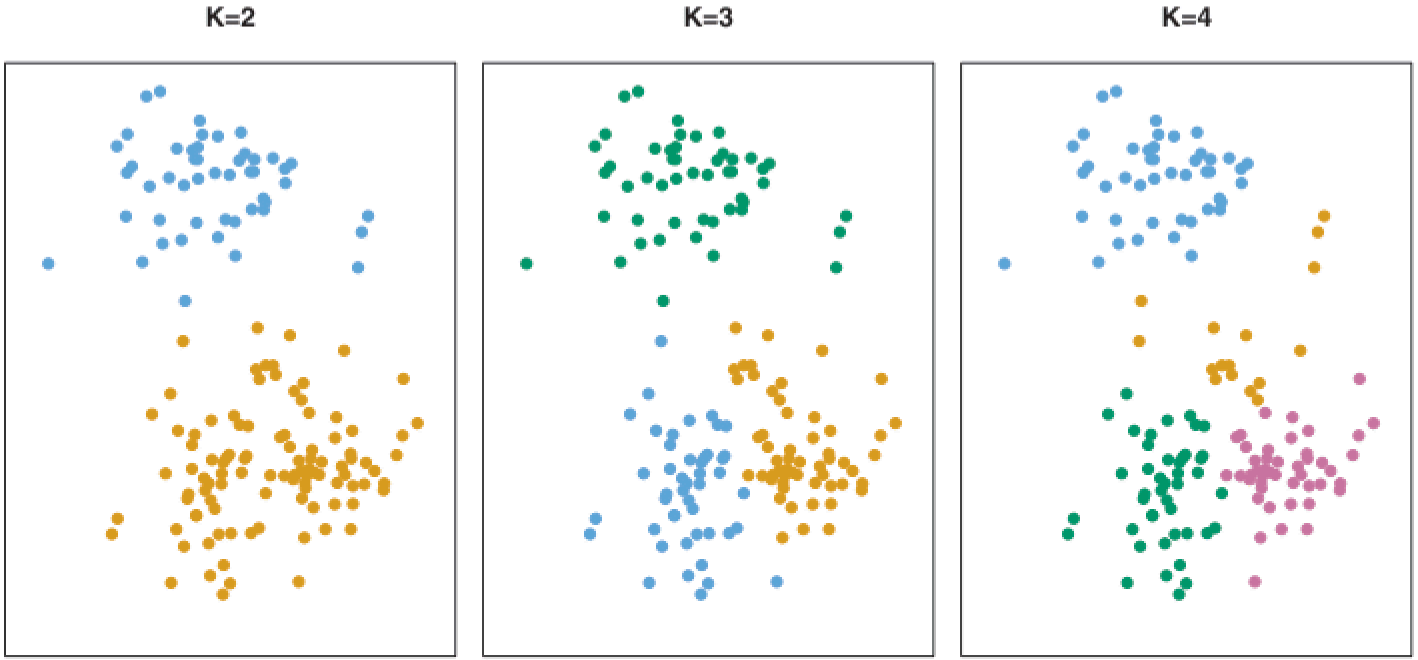


FIGURE 10.5. *A simulated data set with 150 observations in two-dimensional space. Panels show the results of applying K-means clustering with different values of K, the number of clusters. The color of each observation indicates the cluster to which it was assigned using the K-means clustering algorithm.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

Source: ISL

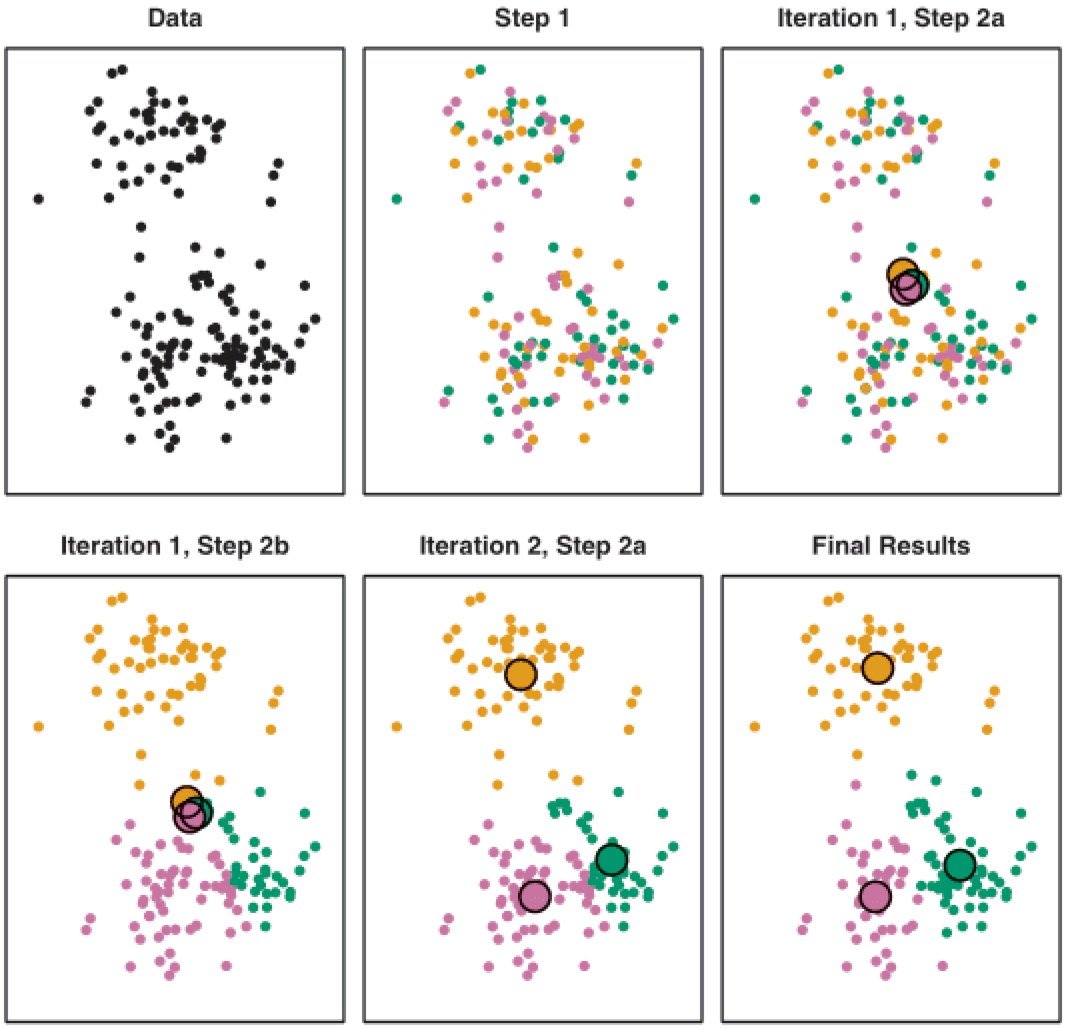


FIGURE 10.6. *The progress of the K-means algorithm on the example of Figure 10.5 with K=3.* Top left: *the observations are shown.* Top center: *in Step 1 of the algorithm, each observation is randomly assigned to a cluster.* Top right: *in Step 2(a), the cluster centroids are computed. These are shown 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.*

Hierarchical Clustering

***K*-means clustering**: K needs to be specified in advance; sensitive to initial assignment

Hierarchical clustering:

- No need to commit to a choice of K
- Results summarized in a *dendogram* — a tree-like representation that allows easy interpretation of data and choosing any number of clusters
- **Our focus**: **Bottom-up or agglomerative clustering** — start with individual observations (aka **leaves**) as clusters, move up by merging (or fusing) leaves into **branches**, then by merging branches with other leaves or branches, and eventually reach the top when everything is merged in into one cluster

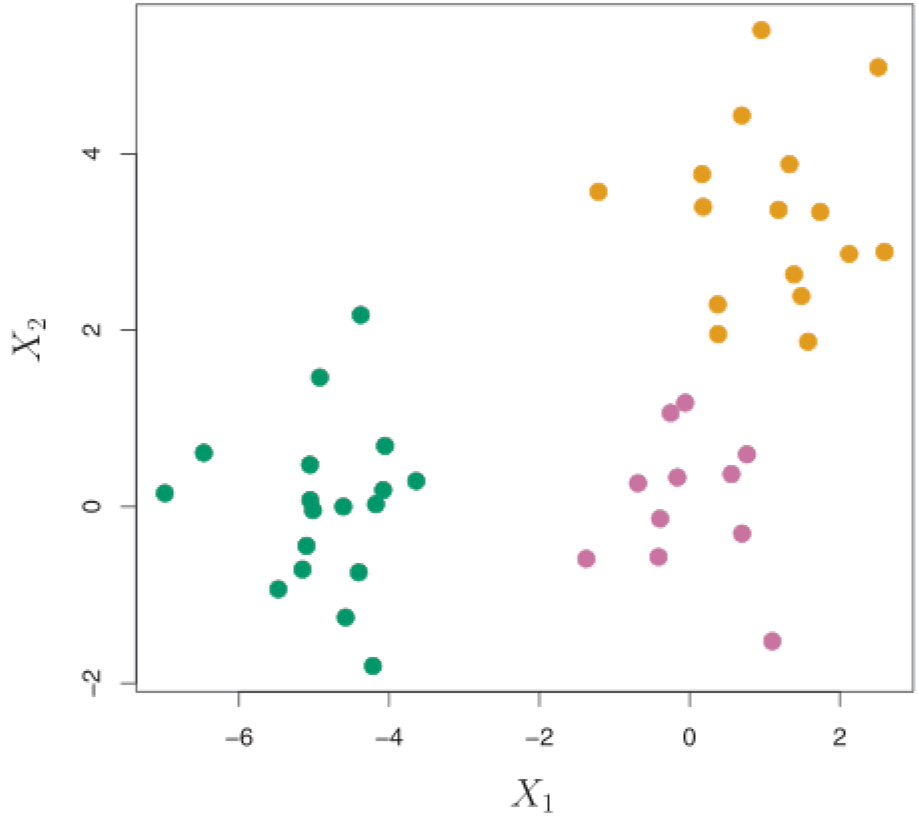


FIGURE 10.8. *Forty-five observations generated in two-dimensional space. In reality there are three distinct classes, shown in separate colors. However, we will treat these class labels as unknown and will seek to cluster the observations in order to discover the classes from the data.*

Source: ISL

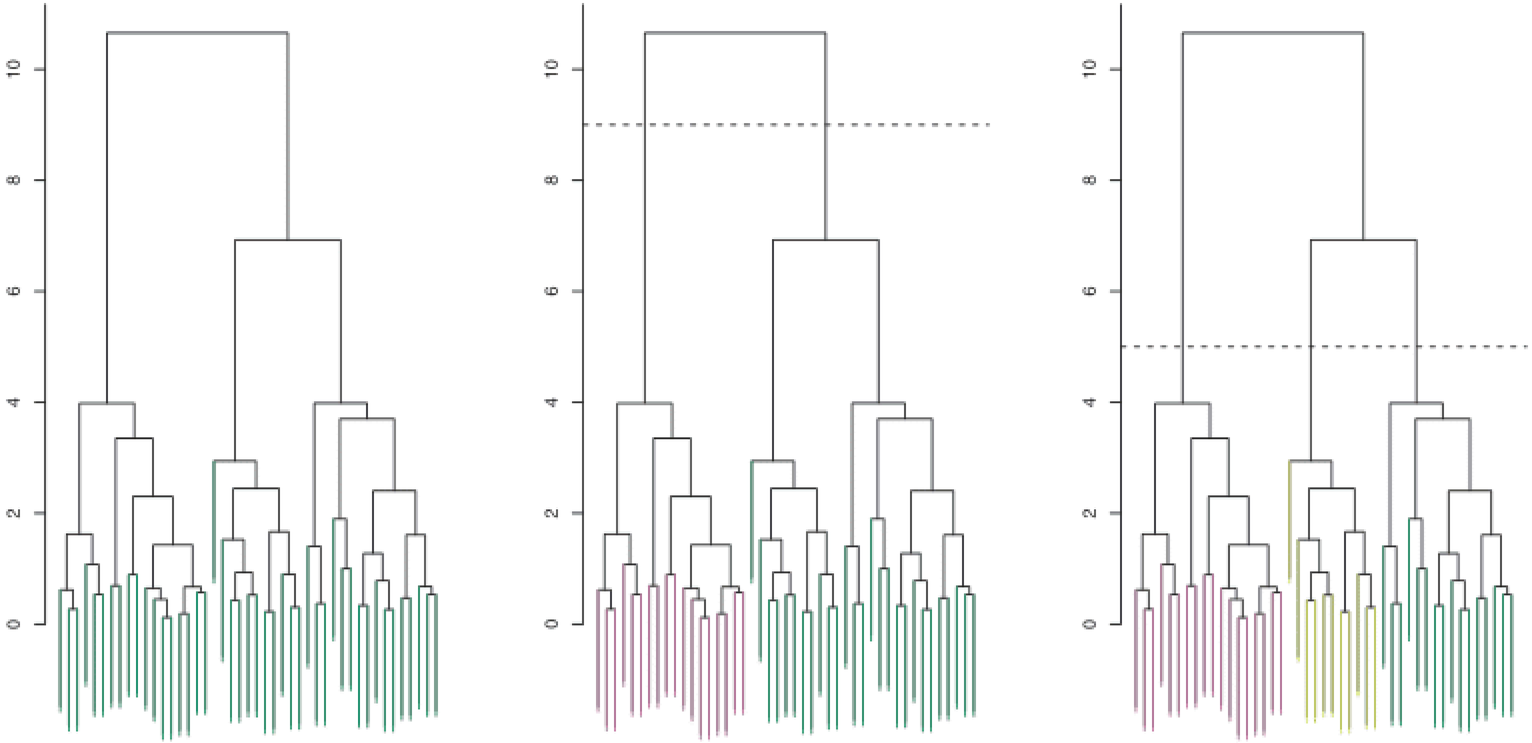


FIGURE 10.9. Left: *dendrogram obtained from hierarchically clustering the data from Figure 10.8 with complete linkage and Euclidean distance.* Center: *the dendrogram from the left-hand panel, cut at a height of nine (indicated by the dashed line). This cut results in two distinct clusters, shown in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

Source: ISL

Hierarchical property: There are n levels of clusters and the clusters any given level are formed by merging two clusters at the previous level. More precisely:

Level 1: n clusters, each containing a single observation
Level 2: Merge the two most similar level 1 clusters to get $(n-1)$ clusters
$\vdots$
Level $(n-1)$: Merge the two most similar level $(n-2)$ clusters to get 2 clusters
Level n: Merge the two level $(n-1)$ clusters to get 1 cluster

Issue: How to measure *dissimilarity* between two clusters?

Dissimilarity between two observations: Use a distance measure $d_{il} = d(X_i, X_l)$, e.g., the Euclidean distance, or a correlation-based distance (more on this later)

Dissimilarity between two clusters: Use the notion of **linkage**. Let A and B denote two clusters and $d(A, B)$ denote their dissimilarity (or distance).

- **Complete linkage**: (*furthest-neighbor* method)

$$d(A, B) = \max_{i \in A, l \in B} d_{il}$$

- **Single linkage**: (*nearest-neighbor* method)

$$d(A, B) = \min_{i \in A, l \in B} d_{il}$$

- **Average linkage**:

$$d(A, B) = \frac{1}{n_A n_B} \sum_{i \in A} \sum_{l \in B} d_{il}$$

- **Centroid linkage**:

$$d(A, B) = d(\overline{X}_A, \overline{X}_B)$$

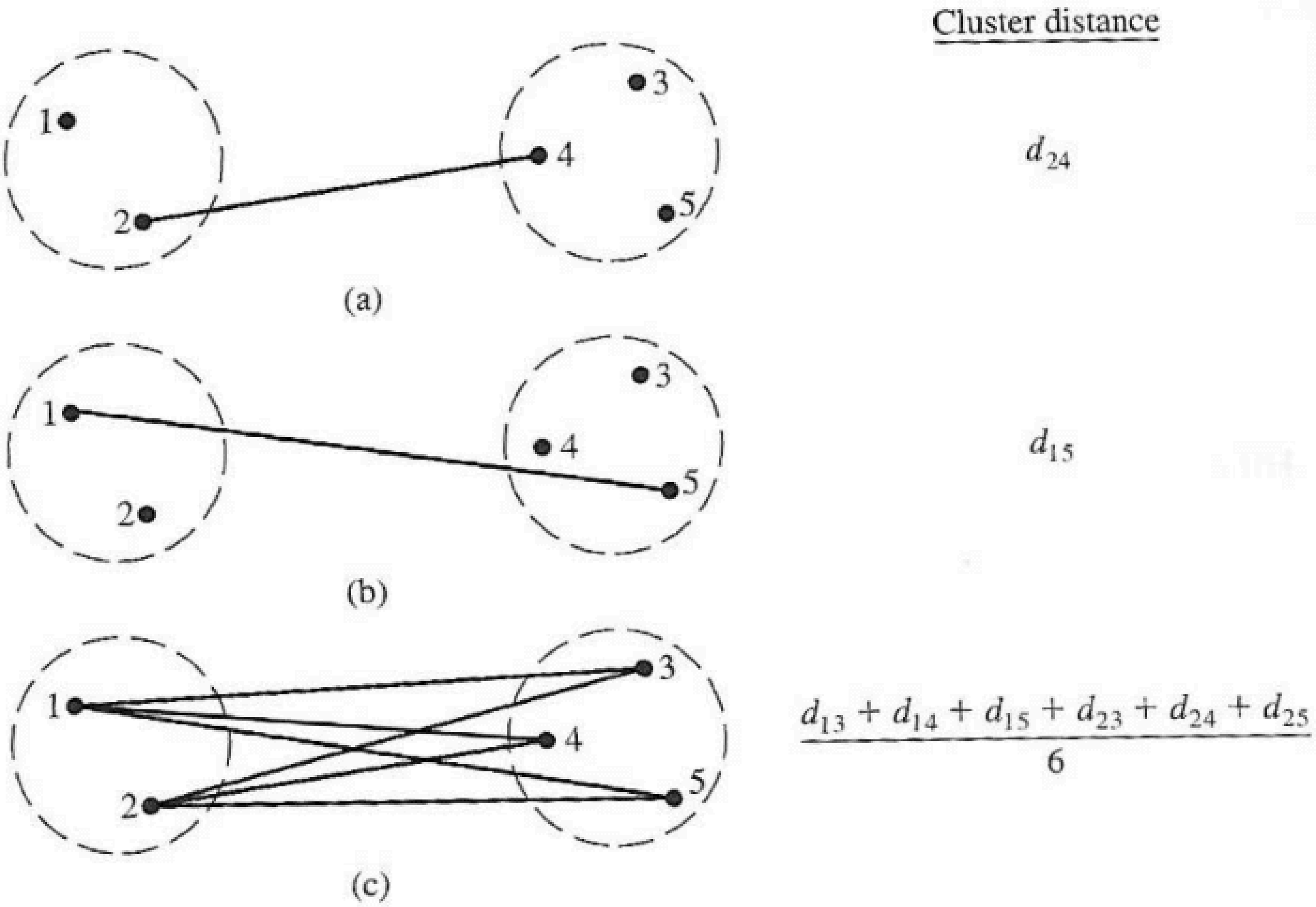
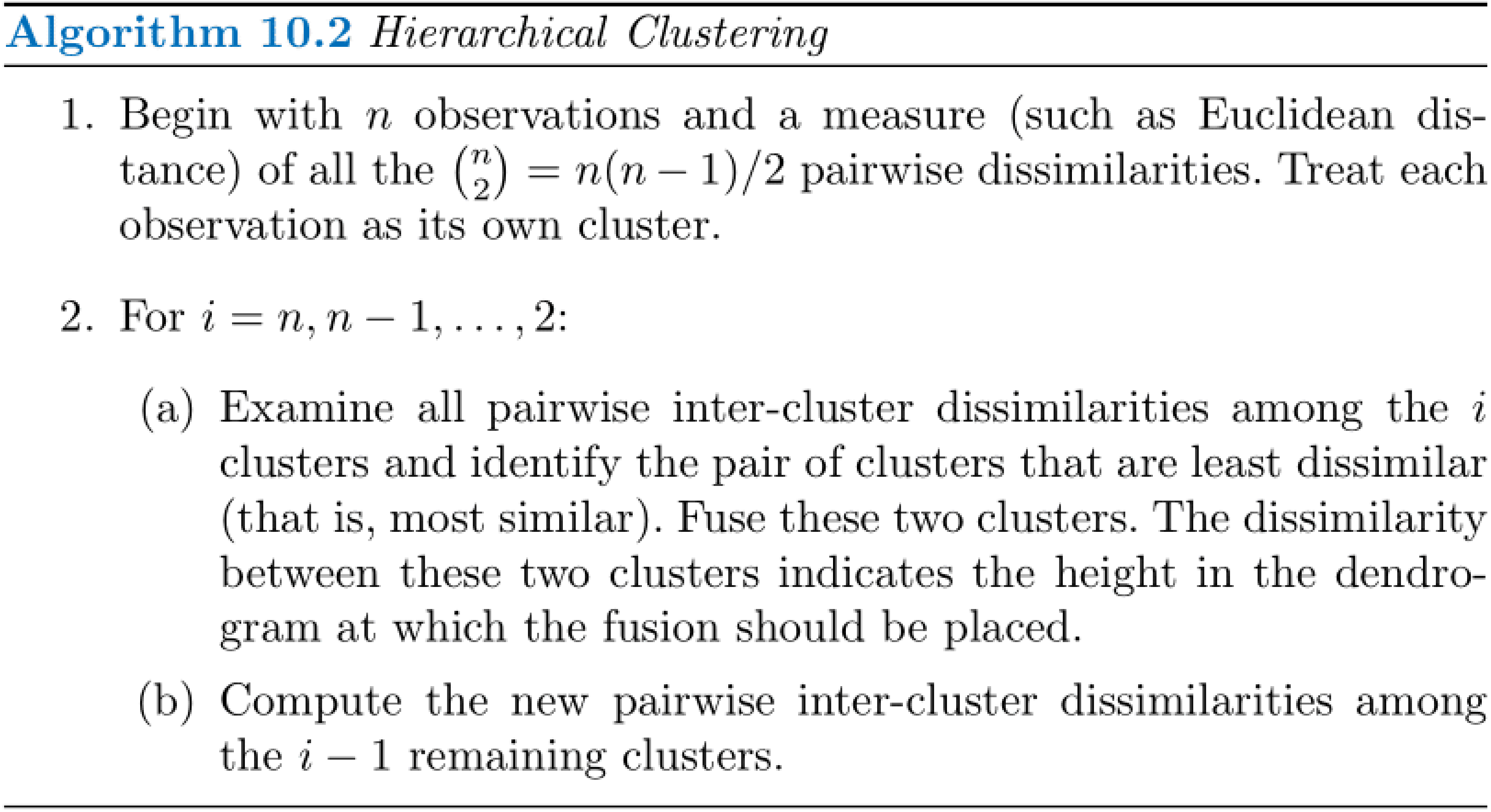

Cluster distance
1
2
3
4
5
(a)
d_{24}
(b)
d_{15}
(c)
$\frac{d_{13} + d_{14} + d_{15} + d_{23} + d_{24} + d_{25}}{6}$

Figure 12.3 Intercluster distance (dissimilarity) for (a) single linkage, (b) complete linkage, and (c) average linkage.

Source: Applied Multivariate Statistical Analysis, 5th edition



Algorithm 10.2 *Hierarchical Clustering*

```
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 $i = n, n-1, \ldots, 2$:
    (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 two clusters indicates the height in the dendrogram at which the fusion should be placed.
    (b) Compute the new pairwise inter-cluster dissimilarities among the $i-1$ remaining clusters.
```

- This also describes how to construct a dendogram.

Source: ISL

Ex: (Clustering using complete linkage) Consider $n = 5$ observations with the following 5×5 distance matrix (**Step 1**):

$$\mathbf{D} = (d_{il}) = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & & & & \\ 9 & 0 & & & \\ 3 & 7 & 0 & & \\ 6 & 5 & 9 & 0 & \\ 11 & 10 & \mathbf{2} & 8 & 0 \end{bmatrix} \end{array}$$

Step 2: **Take** $i = 5$. In Step 2a, merge observations 3 and 5 at height 2 to get cluster (35). Now there are 4 clusters left. In Step 2b, compute the new 4×4 distance matrix. We have:

$$d((35), 1) = \max\{d_{31}, d_{51}\} = \max\{3, 11\} = 11$$
$$d((35), 2) = \max\{d_{32}, d_{52}\} = 10$$
$$d((35), 4) = \max\{d_{34}, d_{54}\} = 9$$

Therefore, the updated distance matrix is

$$\begin{array}{cc} & \begin{matrix} (35) & 1 & 2 & 4 \end{matrix} \\ \begin{matrix} (35) \\ 1 \\ 2 \\ 4 \end{matrix} & \begin{bmatrix} 0 & & & \\ 11 & 0 & & \\ 10 & 9 & 0 & \\ 9 & 6 & \mathbf{5} & 0 \end{bmatrix} \end{array}$$

Take $i = 4$. In Step 2a, merge observations 2 and 4 at height 5 to get cluster (24). Now there are 3 clusters left. In Step 2b, compute the new 3×3 distance matrix. We have:

$$d((24), (35)) = \max\{d(2, (35)), d(4, (35))\} = \max\{10, 9\} = 10$$
$$d((24), 1) = \max\{d_{21}, d_{41}\} = 9$$

The updated distance matrix is

$$\begin{array}{cc} & \begin{array}{ccc} (35) & (24) & 1 \end{array} \\ \begin{array}{c} (35) \\ (24) \\ 1 \end{array} & \begin{bmatrix} 0 & & \\ 10 & 0 & \\ 11 & \mathbf{9} & 0 \end{bmatrix} \end{array}$$

Take $i = 3$. In Step 2a, merge observation 1 and cluster (24) at height 9 to get cluster (124). Now there are 2 clusters left. In Step 2b, compute the new 2 × 2 distance matrix. We have:

$$d((35),(124)) = \max\{d((35),1), d((35),(24))\} = \max\{11, 10\} = 11.$$

The updated distance matrix is:

$$\begin{array}{cc} & \begin{array}{cc} (35) & (124) \end{array} \\ \begin{array}{c} (35) \\ (124) \end{array} & \begin{bmatrix} 0 & \\ \mathbf{11} & 0 \end{bmatrix} \end{array}$$

Take $i = 2$. In Step 2, merge the remaining two clusters at height 11 to get the cluster (12435) that contains all the observations. The process stops. The resulting dendogram is:

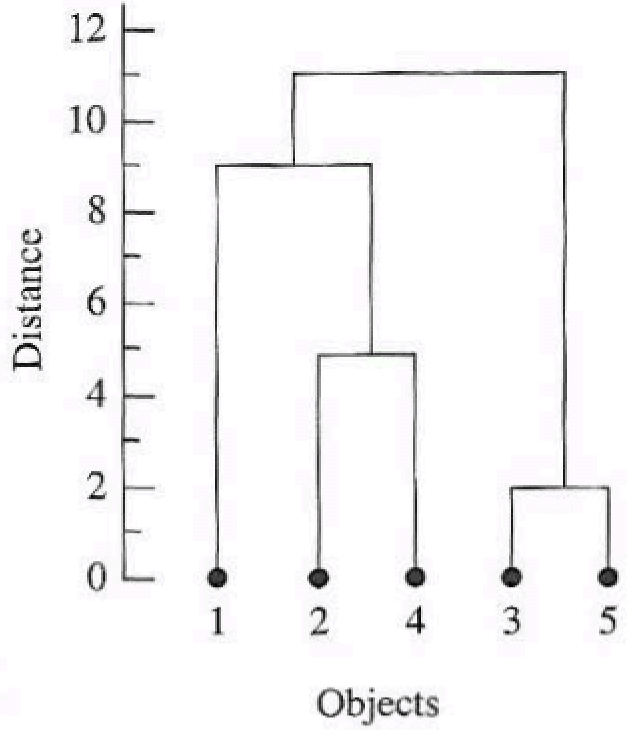


Figure 12.7 Complete linkage dendrogram for distances between five objects.

Source: Applied Multivariate Statistical Analysis, 5th edition

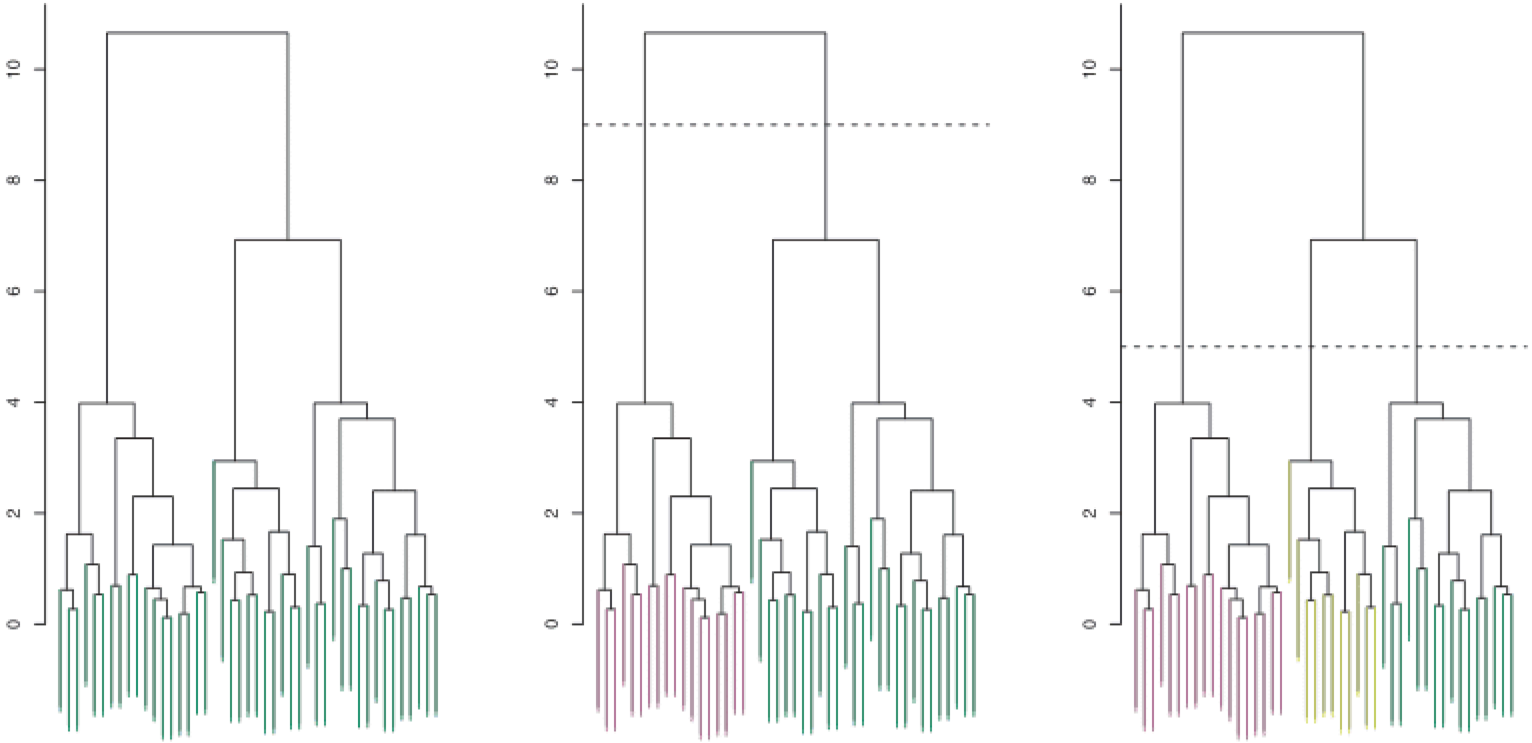


FIGURE 10.9. Left: *dendrogram obtained from hierarchically clustering the data from Figure 10.8 with complete linkage and Euclidean distance.* Center: *the dendrogram from the left-hand panel, cut at a height of nine (indicated by the dashed line). This cut results in two distinct clusters, shown in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

Source: ISL

Interpreting a Dendogram

- Cut at an appropriate height to get the desired # clusters
- **Vertical axis:** Dissimilarity measure (or distance) — the height where two clusters merge
- Draw conclusions about similarity of two observations by looking at the height at which the clusters containing them merge — **lower height** $\implies$ **more similarity**
- Proximity along horizontal axis **does not** indicate similarity. Look at any point where a merger occurs. The positions of the branches merging there can be swapped (i.e., left and right branches can be swapped to become right and left branches) without affecting the meaning of the dendogram.

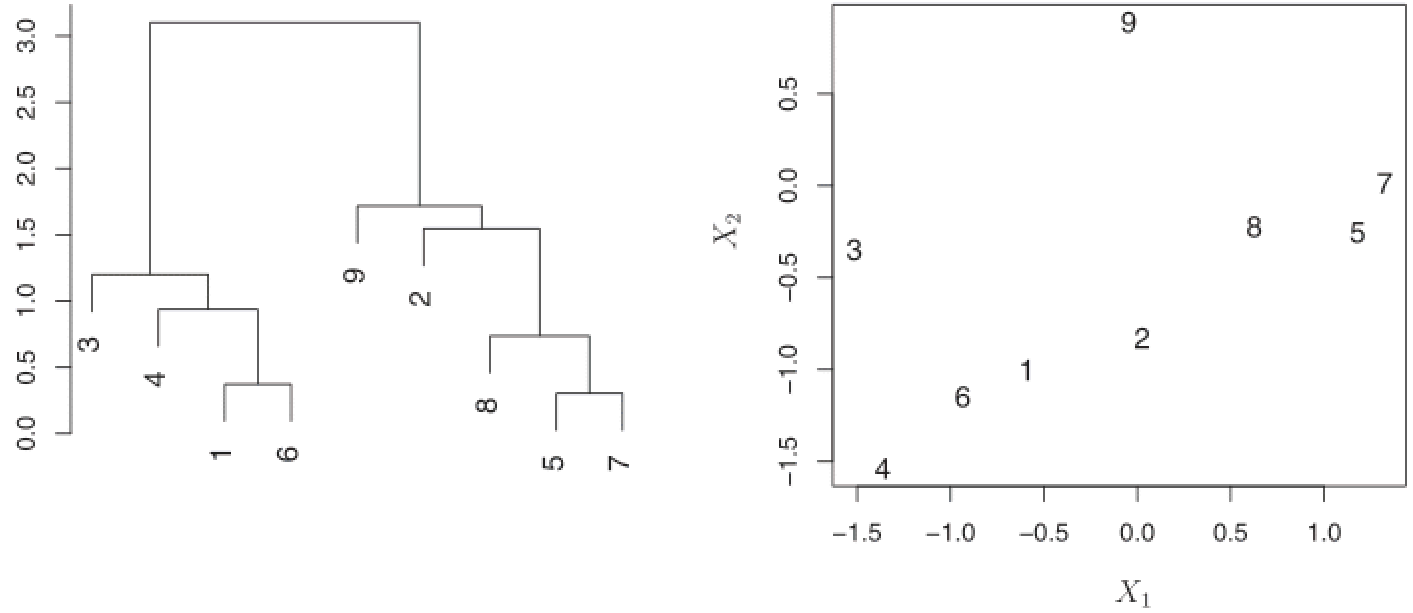


FIGURE 10.10. *An illustration of how to properly interpret a dendrogram with nine observations in two-dimensional space.* Left: *a dendrogram generated using Euclidean distance and complete linkage. Observations* 5 *and* 7 *are quite similar to each other, as are observations* 1 *and* 6*. However, observation*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

Source: ISL

Desirable Properties for Clusters

- **Compactness of a cluster A**: Measured using its **diameter**, $D_A = \max_{i \in A, l \in A} d_{il}$ — largest dissimilarity among its members. Smaller is better.
- **Closeness of two clusters A and B**: Two clusters are considered close if all of the observations in their union are relatively similar. Closer is better.
- **Invariance to monotone transformation of distance**: The clusters should not change if d_{il} is replaced by $h(d_{il})$ where h is an increasing function. In other words, the clusters should depend only on the ordering of the d_{il}.

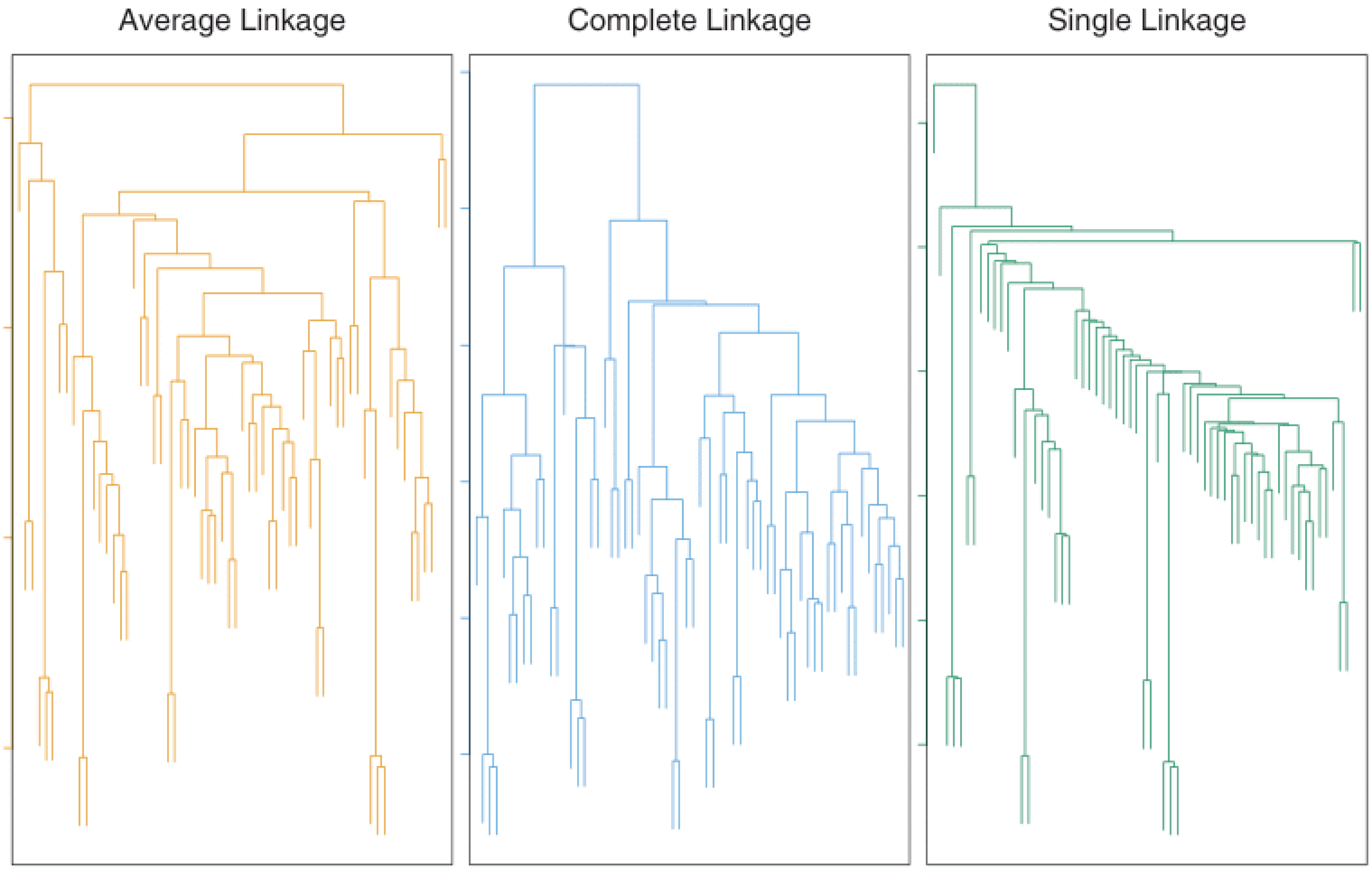


FIGURE 10.12. *Average, complete, and single linkage applied to an example data set. Average and complete linkage tend to yield more balanced clusters.*

Source: ISL

Effect of Linkage

Single linkage:

- Has the closeness and invariance property
- Violates the compactness property as its clusters tend to have large diameters. This occurs because the fusion heights often increase in small increments, producing extended trailing clusters — **chaining**.

Complete linkage:

- Has the compactness and invariance property
- Violates the closeness property — some members of a cluster may be much closer to members of another cluster than to its own cluster. This occurs because the fusion heights often increase in large increments

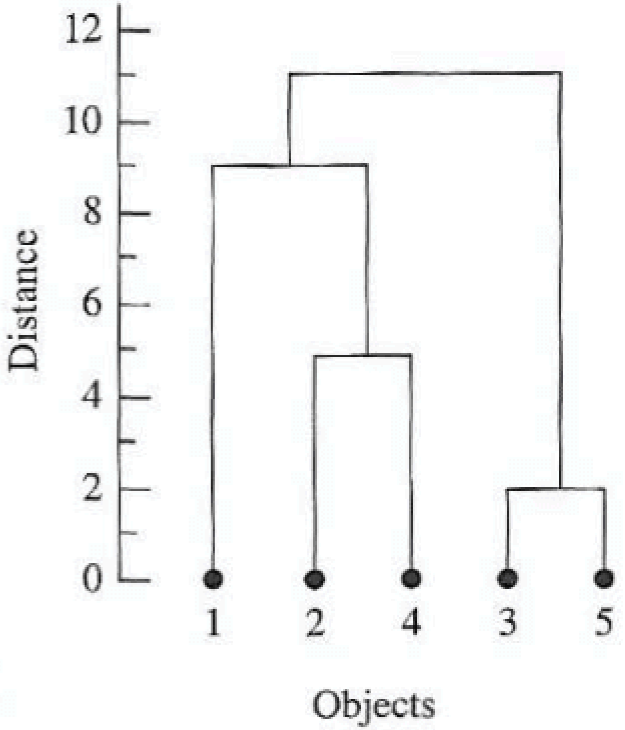


Figure 12.7 Complete linkage dendrogram for distances between five objects.

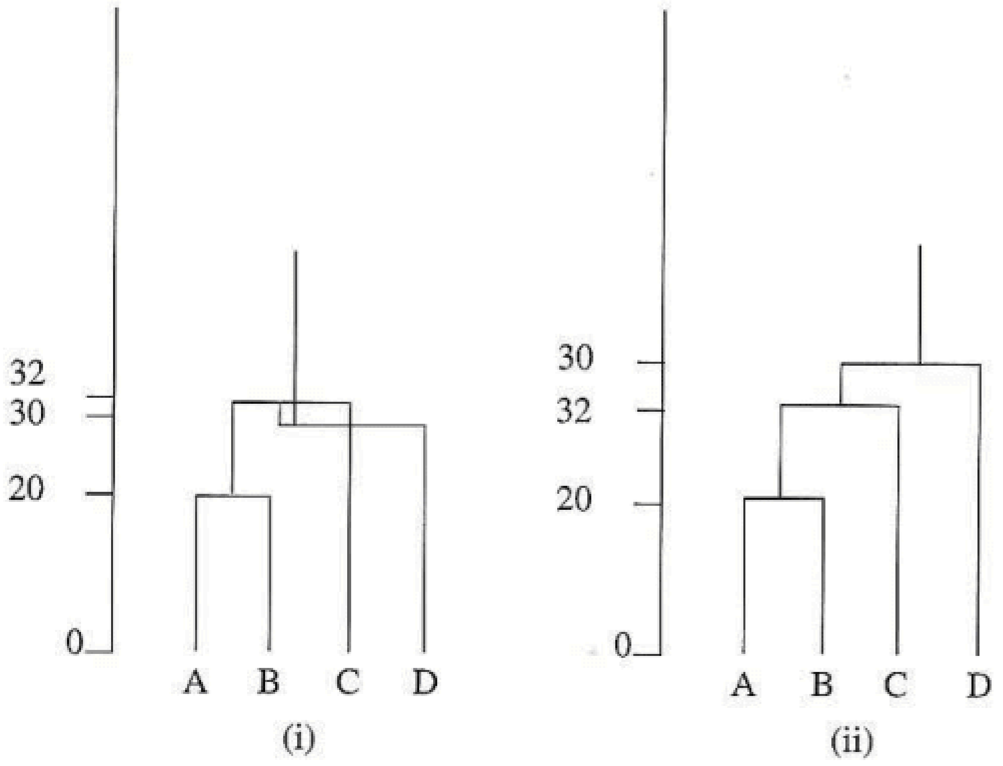
- From the original **D**, see that $(d_{12}, d_{14}, d_{13}) = (9, 6, 3)$, imply that observation 1 is much closer to observation 3 (members of another cluster) than to observations 2 and 4 (members of its own cluster)

Source: Applied Multivariate Statistical Analysis, 5th edition

Average linkage:

- Offers a compromise between the two extremes of single and complete linkage by producing relatively compact clusters that are also moderately close
- Violates the invariance property

Centroid linkage: Not popular because it may produce **inversion** — occurs when an observation joins an existing cluster at a *smaller* height than that of a previous merger. The other linkage methods are not prone to inversion.



- C joins (AB) at height 32 to form (ABC) but (D) joins (ABC) at a smaller height of 30
- Inversion is indicated by a crossover in (i) and by a non-monotonic scale on the vertical axis in (ii)

Source: Applied Multivariate Statistical Analysis, 5th edition

Choosing a Dissimilarity Measure

There are various ways to define dissimilarity between two observations X_i and X_l. Consider the following two. The choice between them depends on the application.

Metric-based measure: (Minkowski metric)

$$d(X_i, X_l) = \left(\sum_{j=1}^{p} |X_{ij} - X_{lj}|^q \right)^{1/q}$$

- $q = 1$: "city-block" or Manhattan distance
- $q = 2$: Euclidean distance (most common)

Correlation-based measure:

$$\rho(X_i, X_l) = \frac{\sum_{j=1}^{p} (X_{ij} - \overline{X}_i)(X_{lj} - \overline{X}_l)}{\sqrt{\sum_{j=1}^{p} (X_{ij} - \overline{X}_i)^2 \sum_{j=1}^{p} (X_{lj} - \overline{X}_l)^2}}$$

- Correlation between X_i and X_l, averaged over variables.

- $d(X_i, X_l) = 1 - \rho(X_i, X_l)$, i.e., distance = one − correlation
- Does not satisfy triangle inequality
- Standardized inputs: $\sum_{j=1}^{p}(X_{ij} - X_{lj})^2 \propto 2(1 - \rho(X_i, X_l))$, implying that clustering based on correlation is equivalent to that based on Euclidean distance.

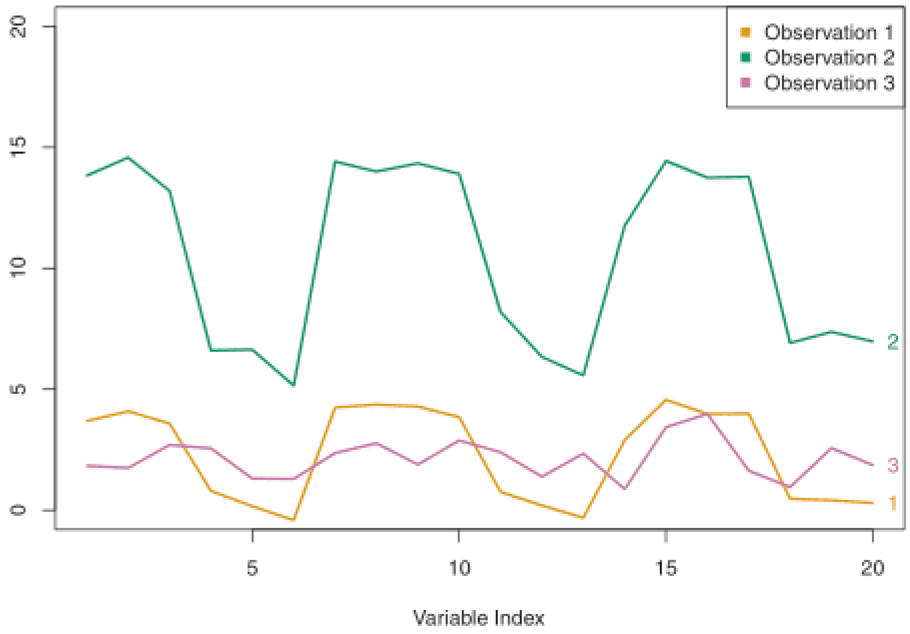


FIGURE 10.13. *Three observations with measurements on 20 variables are shown. Observations 1 and 3 have similar values for each variable and so there is a small Euclidean distance between them. But they are very weakly correlated, so they have a large correlation-based distance. On the other hand, observations 1 and 2 have quite different values for each variable, and so there is a large Euclidean distance between them. But they are highly correlated, so there is a small correlation-based distance between them.*

A drawback of Hierarchical Clustering

Hierarchical clustering implies that the clusters obtained by cutting a dendogram at given height are necessarily nested within the clusters obtained by cutting the dendogram at any greater height. However, this assumption of hierarchical structure may not hold for the data at hand.

Ex: : Consider a dataset on $p = 2$ features — gender (male and female) and race (White, AA, others), where the observations are equally split among the two genders and also among the three races. Imagine that the best division in two groups involves splitting by gender and the best division in three groups involves splitting by race. These clusters are not nested in that the best division in three groups does not result from taking the best division in two groups and splitting up one of those groups. In this case, hierarchical clustering will be less accurate than K-means clustering.

Some Practical Issues in Clustering

- Clustering observations or features
- Standardize or not
- Distance-based or correlation-based dissimiliarity measure
- Qualitative features — use zero-one dissimiliarity
- Type of linkage
- How many clusters?
- Validating the clusters
- Ties among the distances — depending upon how the ties are resolved, the dendograms may not look identical
- Clusters are generally not robust — small changes in data may produce very different clusters
- K-means and hierarchical clustering will force every observation into a cluster — not a good idea if some observations in the data do not truly belong to any subgroups. Use methods based on mixture models
- Be careful in reporting results of a cluster analysis

Unsupervised learning (Clustering & PCA)

- Python applications
  - Principal Component Analysis
  - K-Means Clustering
  - Hierarchical Clustering
  - NC160 Data Example