are similar to each other, whereas items from different clusters are quite dissimilar.

One sledgehammer method of nonhierarchical clustering would conceivably involve as a first step the total enumeration of all possible groupings of the items. Then, using some optimizing criterion, the grouping that is chosen as "best" would be that partition that optimized the criterion. Clearly, for large data sets (e.g., microarray data used for gene clustering), such a method would rapidly become infeasible, requiring incredible amounts of computer time and storage. As a result, all available clustering techniques are iterative and work on only a very limited amount of enumeration. Thus, nonhierarchical clustering methods, which do not need to store large proximity matrices, are computationally more efficient than are hierarchical methods.

This category of clustering methods includes all of the partitioning methods, (e.g., *K-means*, *partitioning around medoids*) and *mode-searching* (or *bump-hunting*) methods using parametric mixtures or nonparametric density estimates.

12.4.1 K-Means Clustering (`kmeans`*)*

The popular K-means algorithm (MacQueen, 1967) is listed in Table 12.2. Because it is extremely efficient, it is often used for large-scale clustering projects. Note that the K-means algorithm needs access to the original data.

The K-means algorithm starts either (1) by assigning items to one of K predetermined clusters and then computing the K cluster centroids, or (2) by pre-specifying the K cluster centroids. The pre-specified centroids may be randomly selected items or may be obtained by cutting a dendrogram at an appropriate height. Then, in an iterative fashion, the algorithm seeks to minimize ESS by reassigning items to clusters. The procedure stops when no further reassignment reduces the value of ESS.

The solution (a configuration of items into K clusters) will typically not be unique; the algorithm will only find a local minimum of ESS. It is recommended that the algorithm be run using different initial random assignments of the items to K clusters (or by randomly selecting K initial centroids) in order to find the lowest minimum of ESS and, hence, the best clustering solution based upon K clusters.

For the worked example, the K-means clustering solutions for $K = 2, 3, 4$ are listed in Table 12.3. For $K = 2$, ESS=23.5; for $K = 3$, ESS=8.67; and for $K = 4$, ESS=5.67. Note that, in general, we expect ESS to be a monotonically decreasing function of K, unless the solution for a given value of K turns out to be a local minimum.

TABLE 12.2. *Algorithm for K-means clustering.*

1. Input: Items $\mathcal{L} = \{\mathbf{x}_i, i = 1, 2, \ldots, n\}$, K = number of clusters.
2. Do one of the following:
   - Form an initial random assignment of the items into K clusters and, for cluster k, compute its current centroid, $\bar{\mathbf{x}}_k$, $k = 1, 2, \ldots, K$.
   - Pre-specify K cluster centroids, $\bar{\mathbf{x}}_k$, $k = 1, 2, \ldots, K$.
3. Compute the squared-Euclidean distance of each item to its current cluster centroid:
$$\text{ESS} = \sum_{k=1}^{K} \sum_{c(i)=k} (\mathbf{x}_i - \bar{\mathbf{x}}_k)^\tau (\mathbf{x}_i - \bar{\mathbf{x}}_k),$$
   where $\bar{\mathbf{x}}_k$ is the kth cluster centroid and $c(i)$ is the cluster containing $\mathbf{x}_i$.
4. Reassign each item to its nearest cluster centroid so that ESS is reduced in magnitude. Update the cluster centroids after each reassignment.
5. Repeat steps 3 and 4 until no further reassignment of items takes place.

*12.4.2 Partitioning Around Medoids (*pam*)*

This clustering method (Vinod, 1969) is a modification of the *K-medoids clustering* algorithm. Although similar to K-means clustering, this algorithm searches for K "representative objects" (or *medoids*) — rather than the centroids — among the items in the data set, and a dissimilarity-based distance is used instead of squared-Euclidean distance. Because it minimizes a sum of dissimilarities instead of a sum of (squared) Euclidean distances, the method is more robust to data anomolies such as outliers and missing values.

This algorithm starts with the proximity matrix $\mathbf{D} = (d_{ij})$, where $d_{ij} = d(\mathbf{x}_i, \mathbf{x}_j)$, either given or computed from the data set, and an initial configuration of the items into K clusters. Using $\mathbf{D}$, we find that item (called a *representative object* or *medoid*) within each cluster that minimizes the total dissimilarity to all other items within its cluster. In the K-medoids algorithm, the centroids of steps 2, 3, and 4 in the K-means algorithm (Table 12.2) are replaced by medoids, and the objective function ESS is replaced by ESS_{med}. See Table 12.4 (steps 1, 2, 3, and 4a) for the K-medoids algorithm.

The *partitioning around medoids* (pam) modification of the K-medoids algorithm (Kaufman and Rousseeuw, 1990, Section 2.4) introduces a swapping strategy by which the medoid of each cluster is replaced by another item in that cluster, but only if such a swap reduces the value of the objec-

TABLE 12.3. *K-means clustering solutions ($K = 2, 3, 4$) for the worked example.*

K	k	Indexes	Centroid	Within-Cluster SS
2	1	1,2,3,4	(3.5, 8.5)	13.5
	2	5,6,7,8	(2.25, 4.25)	10.0
3	1	1,2,3	(1.33, 4.0)	2.67
	2	4,5	(5.0, 6.0)	2.0
	3	6,7,8	(3.0, 9.0)	4.0
4	1	1,2,3	(1.33, 4.0)	2.67
	2	4,5	(5.0, 6.0)	2.0
	3	6,8	(3.5, 9.5)	1.0
	4	7	(2.0, 8.0)	0.0

tive function. The `pam` algorithm is listed in Table 12.4 (steps 1, 2, 3, and 4b).

A disadvantage of both the K-medoids and the `pam` algorithms is that, although they run well on small data sets, they are not efficient enough to use for clustering large data sets.

12.4.3 Fuzzy Analysis (`fanny`)

The idea behind *fuzzy clustering* is that items to be clustered can be assigned probabilities of belonging to each of the K clusters (Kaufman and Rousseeuw, 1990, Section 4.4). Let u_{ik} denote the *strength of membership* of the ith item for the kth cluster. For the ith item, we require that the $\{u_{ik}\}$ behave like probabilities; that is, $u_{ik} \geq 0$, for all i and $k = 1, 2, \ldots, K$, and $\sum_{k=1}^{K} u_{ik} = 1$ for each i. This contrasts with the partitioning methods of `kmeans` or `pam`, where each item is assigned to one and only one cluster.

Given a proximity matrix $\mathbf{D} = (d_{ij})$ and number of clusters K, the unknown membership strengths, $\{u_{ik}\}$, are found by minimizing the objective function,

$$\sum_{k=1}^{K} \frac{\sum_i \sum_j u_{ik}^2 u_{jk}^2 d_{ij}}{2 \sum_\ell u_{\ell k}^2}. \tag{12.1}$$

The objective function is minimized subject to the nonnegativity and unit sum restrictions by using an iterative algorithm.

For the worked example, the solution (after 90 iterations) is given in Table 12.5, where the most likely cluster memberships are as follows: cluster 1: items 1, 2, 3; cluster 2: items 4, 5; cluster 3: items 6, 7, 8. The minimum of the objective function is 3.428.

TABLE 12.4. *Algorithms for K-medoid and partitioning-around-medoids clustering.*

1. Input: proximity matrix $\mathbf{D} = (d_{ij})$; K = number of clusters.
2. Form an initial assignment of the items into K clusters.
3. Locate the *medoid* for each cluster. The medoid of the kth cluster is defined as that item in the kth cluster that minimizes the total dissimilarity to all other items within that cluster, $k = 1, 2, \ldots, K$.

4a. For K-medoids clustering:

- For the kth cluster, reassign the i_kth item to its nearest cluster medoid so that the objective function,

$$\text{ESS}_{\text{med}} = \sum_{k=1}^{K} \sum_{c(i)=k} d_{ii_k},$$

  is reduced in magnitude, where $c(i)$ is the cluster containing the ith item.
- Repeat step 3 and the reassignment step until no further reassignment of items takes place.

4b. For partitioning-around-medoids clustering:

- For each cluster, swap the medoid with the non-medoid item that gives the largest reduction in ESS_{med}.
- Repeat the swapping process over all clusters until no further reduction in ESS_{med} takes place.

12.4.4 Silhouette Plot

A useful feature of partitioning methods based upon the proximity matrix $\mathbf{D}$ (e.g., `kmeans`, `pam`, and `fanny`) is that the resulting partition of the data can be graphically displayed in the form of a *silhouette plot* (Rousseeuw, 1987).

Suppose we are given a particular clustering, $\mathcal{C}_K$, of the data into K clusters. Let $c(i)$ denote the cluster containing the ith item. Let a_i be the average dissimilarity of that ith item to all other members of the same cluster $c(i)$. Also, let c be some cluster other than $c(i)$, and let $d(i, c)$ be the average dissimilarity of the ith item to all members of c. Compute $d(i, c)$ for all clusters c other than $c(i)$. Let $b_i = \min_{c \neq c(i)} d(i, c)$. If $b_i = d(i, C)$, then, cluster C is called the *neighbor* of data point i and is regarded as the second-best cluster for the ith item.

TABLE 12.5. *Fuzzy clustering for the worked example with $K = 3$. The boldfaced entries show the most probable cluster memberships for each item.*

	Cluster k		
i	1	2	3
1	**0.799**	0.117	0.083
2	**0.828**	0.107	0.065
3	**0.735**	0.146	0.119
4	0.116	**0.790**	0.094
5	0.102	**0.715**	0.183
6	0.072	0.146	**0.782**
7	0.196	0.239	**0.565**
8	0.064	0.097	**0.839**

The ith *silhouette value* (or *width*) is given by

$$s_i(\mathcal{C}_K) = s_{iK} = \frac{b_i - a_i}{\max\{a_i, b_i\}}, \tag{12.2}$$

so that $-1 \leq s_{iK} \leq 1$. Large positive values of s_{iK} (i.e., $a_i \approx 0$) indicate that the ith item is well-clustered, large negative values of s_{iK} (i.e., $b_i \approx 0$) indicate poor clustering, and $s_{iK} \approx 0$ (i.e., $a_i \approx b_i$) indicates that the ith item lies between two clusters. If $\max_i\{s_{iK}\} < 0.25$, this indicates either that there are no definable clusters in the data or that, even if there are, the clustering procedure has not found it. Negative silhouette widths tend to attract attention: the items corresponding to these negative values are considered to be borderline allocations; they are neither well-clustered nor are they assigned by the clustering process to an alternative cluster.

A *silhouette plot* is a bar plot of all the $\{s_{iK}\}$ after they are ranked in decreasing order, where the length of the ith bar is s_{iK}. For the worked example, where we used the `pam` clustering method with $K = 3$ clusters, the silhouette plot is displayed in Figure 12.5.

The *average silhouette width*, $\bar{s}_K$, is the average of all the $\{s_{iK}\}$. For the worked example with $K = 3$, the overall average silhouette width is $\bar{s}_3 = 0.51$. (For $K = 2$, $\bar{s}_2 = 0.44$, and for $K = 4$, $\bar{s}_4 = 0.41$.) The statistic $\bar{s}_K$ has been found to be a very useful indicator of the merit of the clustering $\mathcal{C}_K$. The average silhouette width has also been used to choose the value of K by finding K to maximize $\bar{s}_K$.

As a clustering diagnostic, Kaufman and Rousseeuw defined the *silhouette coefficient*, $SC = \max_K\{\bar{s}_K\}$, and gave subjective interpretations of its value:

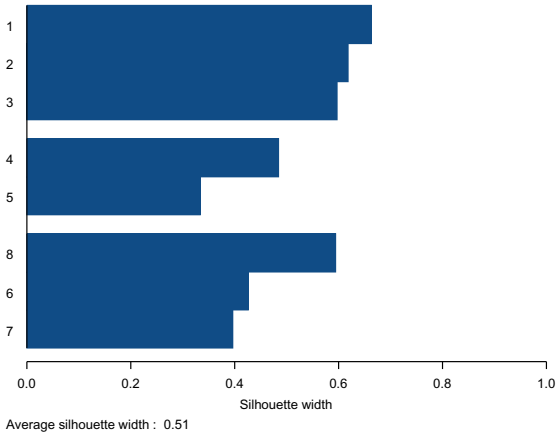

FIGURE 12.5. *Silhouette plot for the worked example using the partitioning around medoids (*`pam`*) clustering method with $K = 3$ clusters.*

SC	Interpretation
0.71–1.00	A strong structure has been found
0.51–0.70	A reasonable structure has been found
0.26–0.50	The structure is weak and could be artificial
≤ 0.25	No substantial structure has been found

12.4.5 Example: Landsat Satellite Image Data

Since 1972, Landsat satellites orbiting the Earth have used a combination of scanning geometry, satellite orbit, and Earth rotation to collect high-resolution multispectral digital information for detecting and monitoring different types of land surface cover characteristics. The Landsat data in this example were generated from a Landsat Multispectral Scanner (MSS) image database used in the European STATLOG Project for assessing machine-learning methods.[3] The following description of the data is taken from the STATLOG website:

> One frame of Landsat MSS imagery consists of four digital images of the same scene in different spectral bands. Two of these are in the visible region (corresponding approximately to green and red regions of the visible spectrum) and two are in the (near) infrared. Each pixel is an 8-bit word, with 0 correspond-

[3]These data, which are available in the file `satimage` at the book's website, can also be downloaded from `http://www.niaad.liacc.up.pt/old/statlog/`. For information on the Landsat satellites, see `http://edc.usgs.gov/guides/landsat_mss.html`.