is assumed to be uniform from 1 to 3, corresponding to more certain initial knowledge about θ. The results of this change are most pronounced when n is small. It is here also that the differences between the Bayesian and the maximum likelihood solutions are most significant. As n increases, the importance of prior knowledge diminishes, and in this particular case the curves for $n = 25$ are virtually identical. In general, one would expect the difference to be small when the number of unlabelled samples is several times the effective number of labelled samples used to determine $p(\theta)$.

6.5.4 Decision-Directed Approximations

Although the problem of unsupervised learning can be stated as merely the problem of estimating parameters of a mixture density, neither the maximum likelihood nor the Bayesian approach yields analytically simple results. Exact solutions for even the simplest nontrivial examples lead to computational requirements that grow exponentially with the number of samples. The problem of unsupervised learning is too important to abandon just because exact solutions are hard to find, however, and numerous procedures for obtaining approximate solutions have been suggested.

Since the basic difference between supervised and unsupervised learning is the presence or absence of labels for the samples, an obvious approach to unsupervised learning is to use the a priori information to design a classifier and to use the decisions of this classifier to label the samples. This is called the *decision-directed* approach to unsupervised learning, and it is subject to many variations. It can be applied sequentially by updating the classifier each time an unlabelled sample is classified. Alternatively, it can be applied in parallel by waiting until all n samples are classified before updating the classifier. If desired, this process can be repeated until no changes occur in the way the samples are labelled.* Various heuristics can be introduced to make the extent of any corrections depend upon the confidence of the classification decision.

There are some obvious dangers associated with the decision-directed approach. If the initial classifier is not reasonably good, or if an unfortunate sequence of samples is encountered, the errors in classifying the unlabelled samples can drive the classifier the wrong way, resulting in a solution corresponding roughly to one of the lesser peaks of the likelihood function. Even if the initial classifier is optimal, the resulting labelling will not in general be the same as the true class membership; the act of classification will exclude samples from the tails of the desired distribution, and will include samples from the tails of the other distributions. Thus, if there is significant

* The Basic Isodata procedure described in Section 6.4.4 is essentially a decision-directed procedure of this type.

R. Duda, P. Hart, "Pattern Classif-
ication and Scene Analysis",
Wiley, 1973.

basic conclusions are that most of these procedures work well if the parametric assumptions are valid, if there is little overlap between the component densities, and if the initial classifier design is at least roughly correct.

6.6 DATA DESCRIPTION AND CLUSTERING

Let us reconsider our original problem of learning something of use from a set of unlabelled samples. Viewed geometrically, these samples form clouds of points in a d-dimensional space. Suppose that we knew that these points came from a single normal distribution. Then the most we could learn from the data would be contained in the sufficient statistics—the sample mean and the sample covariance matrix. In essence, these statistics constitute a compact description of the data. The sample mean locates the center of gravity of the cloud. It can be thought of as the single point $\mathbf{x}$ that best represents all of the data in the sense of minimizing the sum of squared distances from $\mathbf{x}$ to the samples. The sample covariance matrix tells us how well the sample mean describes the data in terms of the amount of scatter that exists in various directions. If the data points are actually normally distributed, then the cloud has a simple hyperellipsoidal shape, and the sample mean tends to fall in the region where the samples are most densely concentrated.

Of course, if the samples are not normally distributed, these statistics can give a very misleading description of the data. Figure 6.7 shows four different data sets that all have the same mean and covariance matrix. Obviously, second-order statistics are incapable of revealing all of the structure in an arbitrary set of data.

By assuming that the samples come from a mixture of c normal distributions, we can approximate a greater variety of situations. In essence, this corresponds to assuming that the samples fall in hyperellipsoidally-shaped clouds of various sizes and orientations. If the number of component densities is not limited, we can approximate virtually any density function in this way, and use the parameters of the mixture to describe the data. Unfortunately, we have seen that the problem of estimating the parameters of a mixture

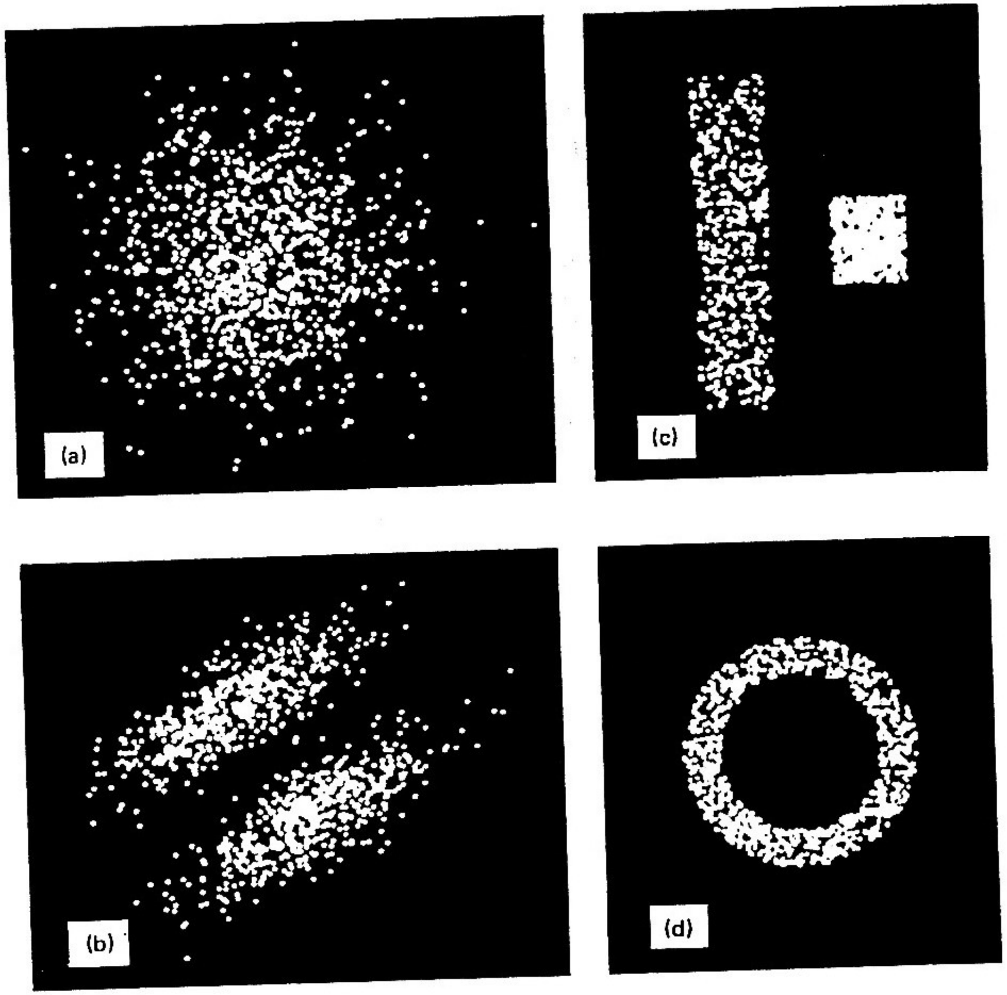


FIGURE 6.7. **Data sets having identical second-order statistics.**

density is not trivial. Furthermore, in situations where we have relatively little a priori knowledge about the nature of the data, the assumption of particular parametric forms may lead to poor or meaningless results. Instead of finding structure in the data, we would be imposing structure on it.

One alternative is to use one of the nonparametric methods described in Chapter 4 to estimate the unknown mixture density. If accurate, the resulting estimate is certainly a complete description of what we can learn from the data. Regions of high local density, which might correspond to significant subclasses in the population, can be found from the peaks or modes of the estimated density.

If the goal is to find subclasses, a more direct alternative is to use a *clustering procedure*. Roughly speaking, clustering procedures yield a data description in terms of clusters or groups of data points that possess strong internal similarities. The more formal procedures use a criterion function, such as the sum of the squared distances from the cluster centers, and seek the grouping that extremizes the criterion function. Because even this can lead to unmanageable computational problems, other procedures have been proposed that are intuitively appealing but that lead to solutions having no established properties. Their use is usually justified on the ground that they are easy to apply and often yield interesting results that may guide the application of more rigorous procedures.

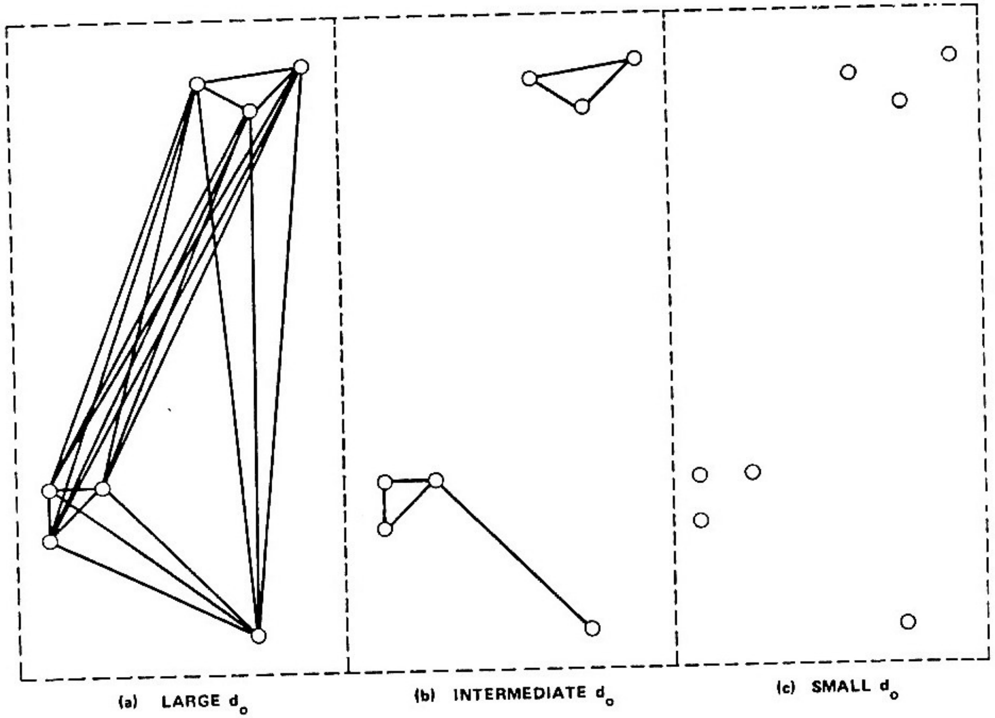
6.7 SIMILARITY MEASURES

Once we describe the clustering problem as one of finding natural groupings in a set of data, we are obliged to define what we mean by a natural grouping. In what sense are we to say that the samples in one cluster are more like one another than like samples in other clusters? This question actually involves two separate issues—how should one measure the similarity between samples, and how should one evaluate a partitioning of a set of samples into clusters? In this section we address the first of these issues.

The most obvious measure of the similarity (or dissimilarity) between two samples is the distance between them. One way to begin a clustering investigation is to define a suitable distance function and compute the matrix of distances between all pairs of samples. If distance is a good measure of dissimilarity, then one would expect the distance between samples in the same cluster to be significantly less than the distance between samples in different clusters.

Suppose for the moment that we say that two samples belong to the same cluster if the Euclidean distance between them is less than some threshold distance d_0. It is immediately obvious that the choice of d_0 is very important. If d_0 is very large, all of the samples will be assigned to one cluster. If d_0 is very small, each sample will form an isolated cluster. To obtain "natural" clusters, d_0 will have to be greater than typical within-cluster distances and less than typical between-cluster distances (see Figure 6.8).

Less obvious perhaps is the fact that the results of clustering depend on the choice of Euclidean distance as a measure of dissimilarity. This choice implies that the feature space is isotropic. Consequently, clusters defined by Euclidean distance will be invariant to translations or rotations—rigid-body motions of the data points. However, they will not be invariant to linear transformations in general, or to other transformations that distort the



(a) LARGE d_o

(b) INTERMEDIATE d_o

(c) SMALL d_o

FIGURE 6.8. The effect of a distance threshold on clustering (Lines are drawn between points closer than a distance d_0 apart).

distance relationships. Thus, as Figure 6.9 illustrates, a simple scaling of the coordinate axes can result in a different grouping of the data into clusters. Of course, this is of no concern for problems in which arbitrary rescaling is an unnatural or meaningless transformation. However, if clusters are to mean anything, they should be invariant to transformations natural to the problem.

One way to achieve invariance is to normalize the data prior to clustering. For example, to obtain invariance to displacement and scale changes, one might translate and scale the axes so that all of the features have zero mean and unit variance. To obtain invariance to rotation, one might rotate the axes so that they coincide with the eigenvectors of the sample covariance matrix. This transformation to *principal components* can be preceded and/or followed by normalization for scale.

However, the reader should not conclude that this kind of normalization is necessarily desirable. Consider, for example, the matter of translating and scaling the axes so that each feature has zero mean and unit variance. The rationale usually given for this normalization is that it prevents certain features from dominating distance calculations merely because they have large numerical values. Subtracting the mean and dividing by the standard deviation is an appropriate normalization if this spread of values is due to normal random variation; however, it can be quite inappropriate if the spread is due to the presence of subclasses (see Figure 6.10). Thus, this routine normalization may be less than helpful in the cases of greatest interest. Section 6.8.3 describes some better ways to obtain invariance to scaling.

An alternative to normalizing the data and using Euclidean distance is to use some kind of normalized distance, such as the Mahalanobis distance. More generally, one can abandon the use of distance altogether and introduce a nonmetric *similarity function* $s(\mathbf{x}, \mathbf{x}')$ to compare two vectors $\mathbf{x}$ and $\mathbf{x}'$. Conventionally, this is a symmetric function whose value is large when $\mathbf{x}$ and $\mathbf{x}'$ are similar. For example, when the angle between two vectors is a

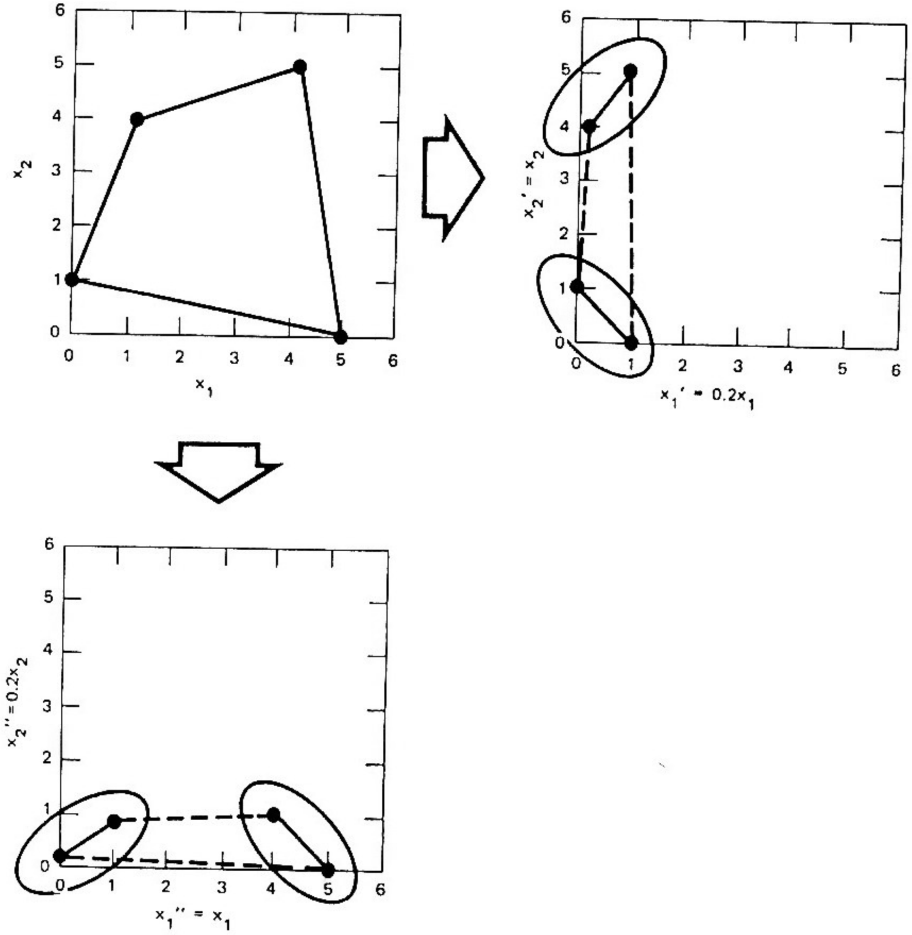
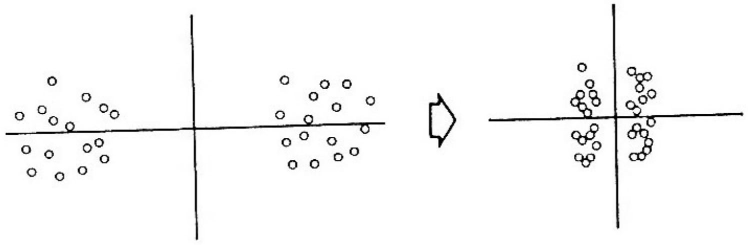


FIGURE 6.9. The effect of scaling on the apparent clustering.



(a) UNNORMALIZED

(b) NORMALIZED

FIGURE 6.10. **Undesirable effects of normalization.**

meaningful measure of their similarity, then the normalized inner product

$$s(\mathbf{x}, \mathbf{x}') = \frac{\mathbf{x}^t\mathbf{x}'}{\|\mathbf{x}\|\,\|\mathbf{x}'\|}$$

may be an appropriate similarity function. This measure, which is the cosine of the angle between $\mathbf{x}$ and $\mathbf{x}'$, is invariant to rotation and dilation, though it is not invariant to translation and general linear transformations.

When the features are binary valued (0 or 1), this similarity function has a simple nongeometrical interpretation in terms of measuring shared features or shared attributes. Let us say that a sample $\mathbf{x}$ *possesses* the ith attribute if $x_i = 1$. Then $\mathbf{x}^t\mathbf{x}'$ is merely the number of attributes possessed by $\mathbf{x}$ and $\mathbf{x}'$, and $\|\mathbf{x}\|\,\|\mathbf{x}'\| = (\mathbf{x}^t\mathbf{x}\mathbf{x}'^t\mathbf{x}')^{1/2}$ is the geometric mean of the number of attributes possessed by $\mathbf{x}$ and the number possessed by $\mathbf{x}'$. Thus, $s(\mathbf{x}, \mathbf{x}')$ is a measure of the relative possession of common attributes. Some simple variations are

$$s(\mathbf{x}, \mathbf{x}') = \frac{\mathbf{x}^t\mathbf{x}'}{d},$$

the fraction of attributes shared, and

$$s(\mathbf{x}, \mathbf{x}') = \frac{\mathbf{x}^t\mathbf{x}'}{\mathbf{x}^t\mathbf{x} + \mathbf{x}'^t\mathbf{x}' - \mathbf{x}^t\mathbf{x}'},$$

the ratio of the number of shared attributes to the number possessed by $\mathbf{x}$ or $\mathbf{x}'$. This latter measure (sometimes known as the Tanimoto coefficient) is frequently encountered in the fields of information retrieval and biological taxonomy. Other measures of similarity arise in other applications, the variety of measures testifying to the diversity of problem domains.

We feel obliged to mention that fundamental issues in measurement theory are involved in the use of any distance or similarity function. The calculation of the similarity between two vectors always involves combining the values of their components. Yet, in many pattern recognition applications the components of the feature vector measure seemingly noncomparable quantities. Using our early example of classifying lumber, how can one compare the brightness to the straightness-of-grain? Should the comparison depend on whether the brightness is measured in candles/m^2 or in foot-lamberts? How does one treat vectors whose components have a mixture of nominal, ordinal, interval, and ratio scales?* Ultimately, there is no methodological answer to these questions. When a user selects a particular similarity function or normalizes his data in a particular way, he introduces information that gives the procedure meaning. We have given examples of some alternatives that have proved to be useful. Beyond that we can do little more than alert the unwary to these pitfalls of clustering.

6.8 CRITERION FUNCTIONS FOR CLUSTERING

Suppose that we have a set $\mathscr{X}$ of n samples $\mathbf{x}_1, \ldots, \mathbf{x}_n$ that we want to partition into exactly c disjoint subsets $\mathscr{X}_1, \ldots, \mathscr{X}_c$. Each subset is to represent a cluster, with samples in the same cluster being somehow more similar than samples in different clusters. One way to make this into a well-defined problem is to define a criterion function that measures the clustering quality of any partition of the data. Then the problem is one of finding the partition that extremizes the criterion function. In this section we examine the characteristics of several basically similar criterion functions, postponing until later the question of how to find an optimal partition.

6.8.1 The Sum-of-Squared-Error Criterion

The simplest and most widely used criterion function for clustering is the sum-of-squared-error criterion. Let n_i be the number of samples in $\mathscr{X}_i$ and let $\mathbf{m}_i$ be the mean of those samples,

$$\mathbf{m}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in \mathscr{X}_i} \mathbf{x}. \tag{25}$$

Then the sum of squared errors is defined by

$$J_e = \sum_{i=1}^{c} \sum_{\mathbf{x} \in \mathscr{X}_i} \|\mathbf{x} - \mathbf{m}_i\|^2. \tag{26}$$

This criterion function has a simple interpretation. For a given cluster $\mathscr{X}_i$, the mean vector $\mathbf{m}_i$ is the best representative of the samples in $\mathscr{X}_i$ in the sense that it minimizes the sum of the squared lengths of the "error" vectors $\mathbf{x} - \mathbf{m}_i$. Thus, J_e measures the total squared error incurred in representing the n samples $\mathbf{x}_1, \ldots, \mathbf{x}_n$ by the c cluster centers $\mathbf{m}_1, \ldots, \mathbf{m}_c$. The value of

* These fundamental considerations are by no means unique to clustering. They appear, for example, whenever one chooses a parametric form for an unknown probability density function, a metric for nonparametric density estimation, or scale factors for linear discriminant functions. Clustering problems merely expose them more clearly.

J_e depends on how the samples are grouped into clusters, and an optimal partitioning is defined as one that minimizes J_e. Clusterings of this type are often called *minimum variance* partitions.

What kind of clustering problems are well suited to a sum-of-squared-error criterion? Basically, J_e is an appropriate criterion when the clusters form essentially compact clouds that are rather well separated from one another. It should work well for the two or three clusters in Figure 6.11, but one would not expect reasonable results for the data in Figure 6.12.* A less obvious problem arises when there are great differences in the number of samples in different clusters. In that case it can happen that a partition that splits a large cluster is favored over one that maintains the integrity of the clusters merely because the slight reduction in squared error achieved is multiplied by many terms in the sum (see Figure 6.13). This situation frequently arises because of the presence of "outliers" or "wild shots," and brings up the problem of interpreting and evaluating the results of clustering. Since little can be said about that problem, we shall merely observe that if additional considerations render the results of minimizing J_e unsatisfactory, then these considerations should be used, if possible, in formulating a better criterion function.

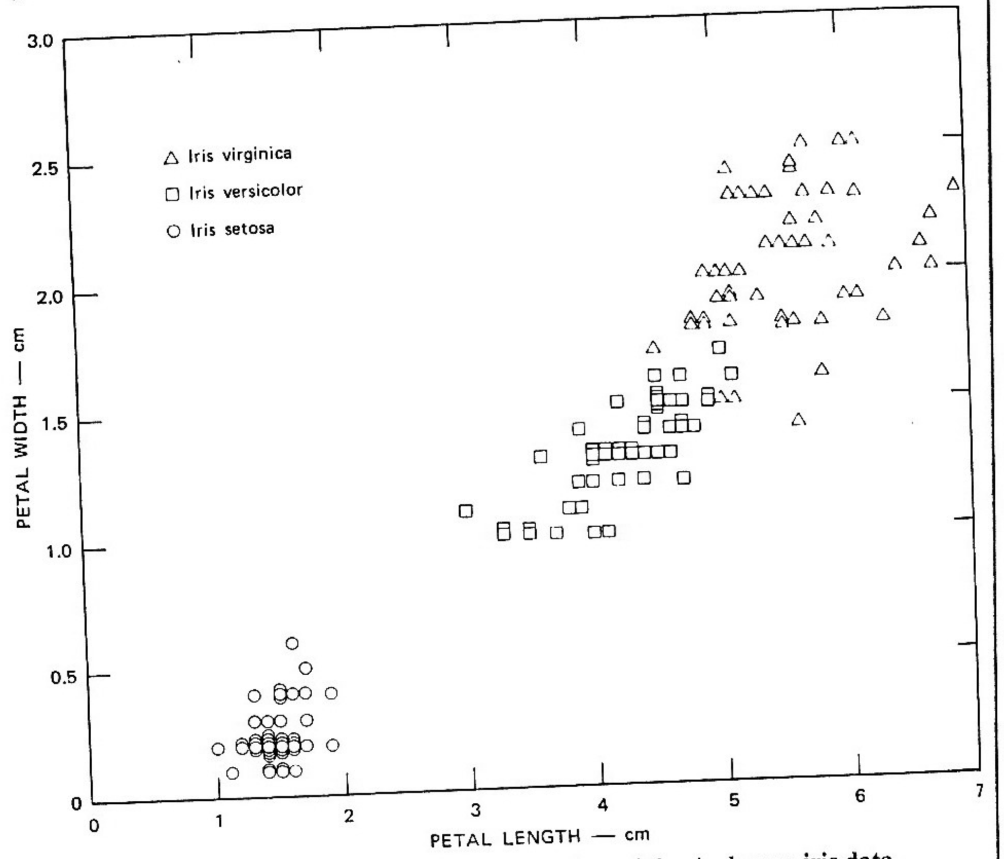


FIGURE 6.11. A two-dimensional section of the Anderson iris data.

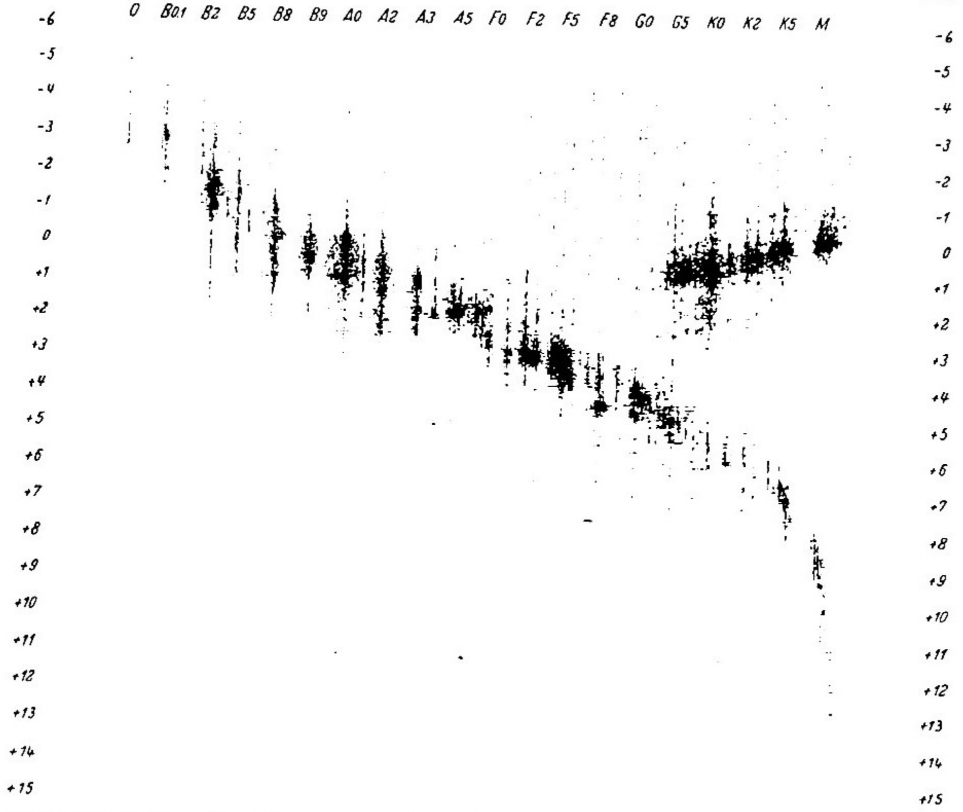


FIGURE 6.12. The Herzsprung-Russell Diagram (Courtesy Lunds Universitet Institutionen für Astronomi).

6.8.2 Related Minimum Variance Criteria

By some simple algebraic manipulation we can eliminate the mean vectors from the expression for J_e and obtain the equivalent expression

$$J_e = \tfrac{1}{2} \sum_{i=1}^{c} n_i \bar{s}_i, \tag{27}$$

* These two data sets are well known for quite different reasons. Figure 6.11 shows two of four measurements made by E. Anderson on 150 samples of three species of iris. These data were listed and used by R. A. Fisher in his classic paper on discriminant analysis (Fisher 1936), and have since become a favorite example for illustrating clustering procedures. Figure 6.12 is well known in astronomy as the Hertzsprung and Russell (or spectrum-luminosity) diagram, which led to the subdivision of stars into such categories as giants, supergiants, main sequence stars, and dwarfs. It was used by E. W. Forgey and again by D. Wishart (1969) to illustrate the limitations of simple clustering procedures.

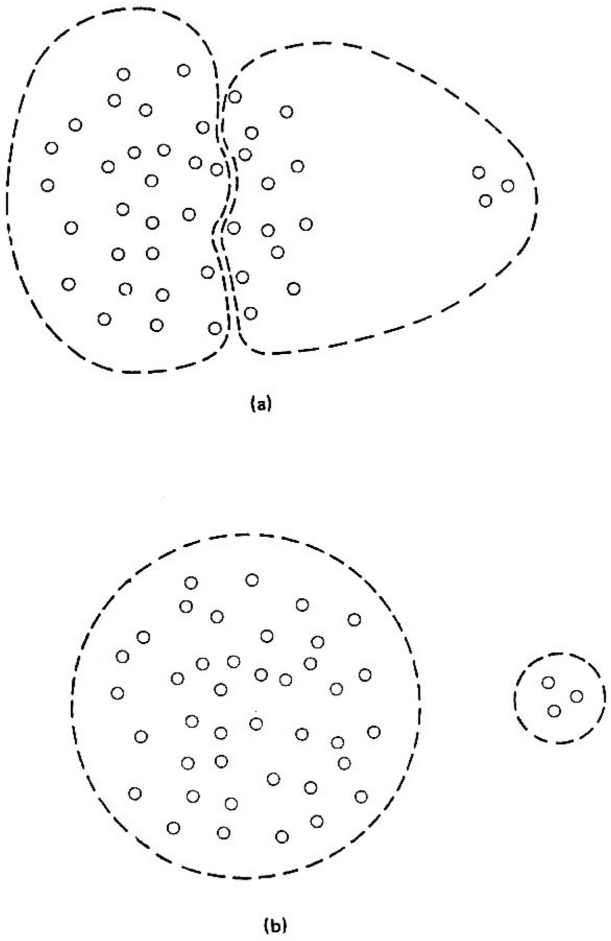


FIGURE 6.13. The problem of splitting large clusters: the sum of squared error is smaller for (a) than for (b).

where

$$\bar{s}_i = \frac{1}{n_i^2}\sum_{\mathbf{x}\in\mathscr{X}_i}\sum_{\mathbf{x}'\in\mathscr{X}_i}\|\mathbf{x}-\mathbf{x}'\|^2. \tag{28}$$

Eq. (28) leads us to interpret $\bar{s}_i$ as the average squared distance between points in the ith cluster, and emphasizes the fact that the sum-of-squared-error criterion uses Euclidean distance as the measure of similarity. It also suggests an obvious way of obtaining other criterion functions. For example, one can replace $\bar{s}_i$ by the average, the median, or perhaps the maximum distance between points in a cluster. More generally, one can introduce an appropriate similarity function $s(\mathbf{x}, \mathbf{x}')$ and replace $\bar{s}_i$ by functions such as

$$\bar{s}_i = \frac{1}{n_i^2}\sum_{\mathbf{x}\in\mathscr{X}_i}\sum_{\mathbf{x}'\in\mathscr{X}_i} s(\mathbf{x}, \mathbf{x}') \tag{29}$$

or

$$\bar{s}_i = \min_{\mathbf{x},\mathbf{x}'\in\mathscr{X}_i} s(\mathbf{x}, \mathbf{x}'). \tag{30}$$

As before, we define an optimal partitioning as one that extremizes the criterion function. This creates a well-defined problem, and the hope is that its solution discloses the intrinsic structure of the data.

6.8.3 Scattering Criteria

6.8.3.1 THE SCATTER MATRICES

Another interesting class of criterion functions can be derived from the scatter matrices used in multiple discriminant analysis. The following definitions directly parallel the definitions given in Section 4.11.

Mean vector for ith cluster:

$$\mathbf{m}_i = \frac{1}{n_i}\sum_{\mathbf{x}\in\mathscr{X}_i}\mathbf{x}. \tag{31}$$

Total mean vector:

$$\mathbf{m} = \frac{1}{n}\sum_{\mathscr{X}}\mathbf{x} = \frac{1}{n}\sum_{i=1}^{c} n_i\mathbf{m}_i. \tag{32}$$

Scatter matrix for ith cluster:

$$S_i = \sum_{\mathbf{x}\in\mathscr{X}_i}(\mathbf{x}-\mathbf{m}_i)(\mathbf{x}-\mathbf{m}_i)^t. \tag{33}$$

Within-cluster scatter matrix:

$$S_W = \sum_{i=1}^{c} S_i. \tag{34}$$

Between-cluster scatter matrix:

$$S_B = \sum_{i=1}^{c} n_i(\mathbf{m}_i-\mathbf{m})(\mathbf{m}_i-\mathbf{m})^t. \tag{35}$$

Total scatter matrix:

$$S_T = \sum_{\mathbf{x}\in\mathscr{X}}(\mathbf{x}-\mathbf{m})(\mathbf{x}-\mathbf{m})^t. \tag{36}$$

As before, it follows from these definitions that the total scatter matrix is the sum of the within-cluster scatter matrix and the between-cluster scatter matrix:

$$S_T = S_W + S_B. \tag{37}$$

Note that the total scatter matrix does not depend on how the set of samples is partitioned into clusters. It depends only on the total set of samples. The within-cluster and between-cluster scatter matrices do depend on the partitioning, however. Roughly speaking, there is an exchange between these two matrices, the between-cluster scatter going up as the within-cluster scatter goes down. This is fortunate, since by trying to minimize the within-cluster scatter we will also tend to maximize the between-cluster scatter.

To be more precise in talking about the amount of within-cluster or between-cluster scatter, we need a scalar measure of the "size" of a scatter matrix. The two measures that we shall consider are the *trace* and the *determinant*. In the univariate case, these two measures are equivalent, and we can define an optimal partition as one that minimizes S_W or maximizes S_B. In the multivariate case things are somewhat more complicated, and a number of related but distinct optimality criteria have been suggested.

6.8.3.2 THE TRACE CRITERION

Perhaps the simplest scalar measure of a scatter matrix is its trace, the sum of its diagonal elements. Roughly speaking, the trace measures the square of the scattering radius, since it is proportional to the sum of the variances in the coordinate directions. Thus, an obvious criterion function to minimize is the trace of S_W. In fact, this criterion is nothing more or less than the sum-of-squared-error criterion, since Eqs. (33) and (34) yield

$$\operatorname{tr} S_W = \sum_{i=1}^{c} \operatorname{tr} S_i = \sum_{i=1}^{c} \sum_{\mathbf{x} \in \mathscr{X}_i} \|\mathbf{x} - \mathbf{m}_i\|^2 = J_e. \tag{38}$$

Since $\operatorname{tr} S_T = \operatorname{tr} S_W + \operatorname{tr} S_B$ and $\operatorname{tr} S_T$ is independent of how the samples are partitioned, we see that no new results are obtained by trying to maximize $\operatorname{tr} S_B$. However, it is comforting to know that in trying to minimize the within-cluster criterion $J_e = \operatorname{tr} S_W$ we are also maximizing the between-cluster criterion

$$\operatorname{tr} S_B = \sum_{i=1}^{c} n_i \|\mathbf{m}_i - \mathbf{m}\|^2. \tag{39}$$

6.8.3.3 THE DETERMINANT CRITERION

In Section 4.11 we used the determinant of the scatter matrix to obtain a scalar measure of scatter. Roughly speaking, this measures the square of the scattering volume, since it is proportional to the product of the variances in the directions of the principal axes. Since S_B will be singular if the number of clusters is less than or equal to the dimensionality, $|S_B|$ is obviously a poor choice for a criterion function. S_W can also become singular, and will certainly be so if $n - c$ is less than the dimensionality d.* However, if we assume that S_W is nonsingular, we are led to consider the criterion function

$$J_d = |S_W| = \left| \sum_{i=1}^{c} S_i \right|. \tag{40}$$

The partition that minimizes J_d is often similar to the one that minimizes J_e, but the two need not be the same. We observed before that the minimum-squared-error partition might change if the axes are scaled. This does not happen with J_d. To see why, let T be a nonsingular matrix and consider the change of variables $\mathbf{x}' = T\mathbf{x}$. Keeping the partitioning fixed, we obtain new mean vectors $\mathbf{m}_i' = T\mathbf{m}_i$ and new scatter matrices $S_i' = TS_iT^t$. Thus, J_d changes to

$$J_d' = |S_W'| = |TS_WT^t| = |T|^2 J_d.$$

Since the scale factor $|T|^2$ is the same for all partitions, it follows that J_d and J_d' rank the partitions in the same way, and hence that the optimal clustering based on J_d is invariant to nonsingular linear transformations of the data.

6.8.3.4 INVARIANT CRITERIA

It is not hard to show that the eigenvalues $\lambda_1, \ldots, \lambda_d$ of $S_W^{-1}S_B$ are invariant under nonsingular linear transformations of the data. Indeed, these eigenvalues are the basic linear invariants of the scatter matrices. Their numerical values measure the ratio of between-cluster to within-cluster scatter in the direction of the eigenvectors, and partitions that yield large values are usually desirable. Of course, as we pointed out in Section 4.11, the fact that the rank of S_B can not exceed $c - 1$ means that no more than $c - 1$ of these eigenvalues can be nonzero. Nevertheless, good partitions are ones for which the nonzero eigenvalues are large.

One can invent a great variety of invariant clustering criteria by composing appropriate functions of these eigenvalues. Some of these follow naturally from standard matrix operations. For example, since the trace of a matrix is the sum of its eigenvalues, one might elect to maximize the criterion function†

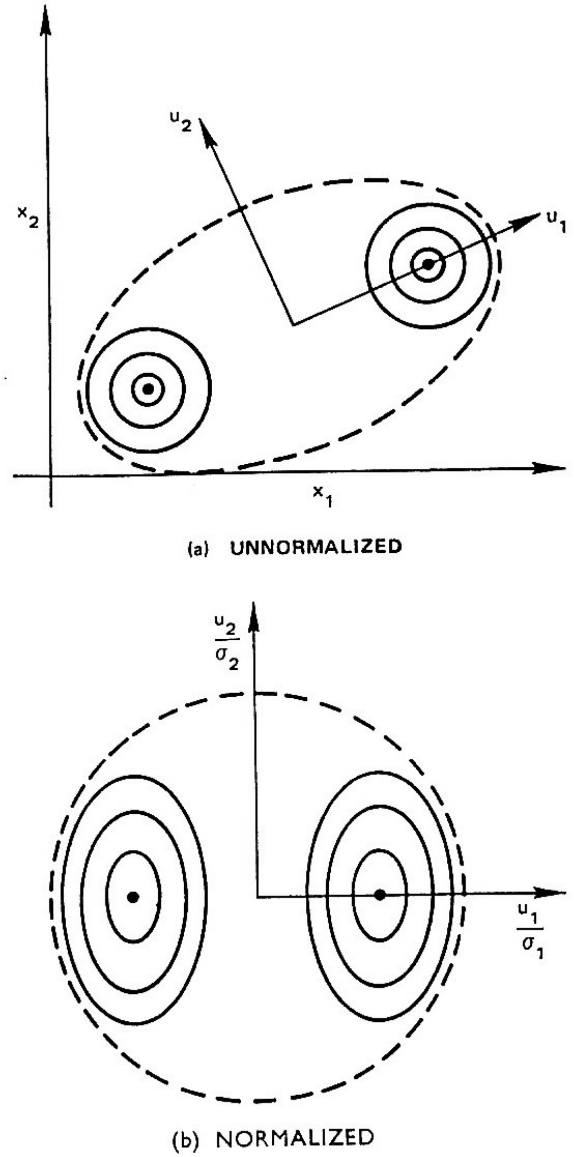
$$\operatorname{tr} S_W^{-1} S_B = \sum_{i=1}^{d} \lambda_i. \tag{41}$$

* This follows from the fact that the rank of S_i can not exceed $n_i - 1$, and thus the rank of S_W can not exceed $\Sigma(n_i - 1) = n - c$. Of course, if the samples are confined to a lower dimensional subspace it is possible to have S_W be singular even though $n - c \geq d$. In such cases, some kind of dimensionality-reduction procedure must be used before the determinant criterion can be applied (see Section 6.14).

† Another invariant criterion is

$$|S_W^{-1}S_B| = \prod_{i=1}^{d} \lambda_i.$$

However, since its value is usually zero it is not very useful.



(a) UNNORMALIZED

(b) NORMALIZED

FIGURE 6.14. The effect of transforming to normalized principal components (Note: the partition that minimizes $S_T^{-1}S_W$ in (a) minimizes the sum of squared errors in (b).).

By using the relation $S_T = S_W + S_B$, one can derive the following invariant relatives of tr S_W and $|S_W|$:

$$\operatorname{tr} S_T^{-1} S_W = \sum_{i=1}^{d} \frac{1}{1+\lambda_i} \tag{42}$$

$$\frac{|S_W|}{|S_T|} = \prod_{i=1}^{d} \frac{1}{1+\lambda_i}\,. \tag{43}$$

Since all of these criterion functions are invariant to linear transformations, the same is true of the partitions that extremize them. In the special case of two clusters, only one eigenvalue is nonzero, and all of these criteria yield the same clustering. However, when the samples are partitioned into more than two clusters, the optimal partitions, though often similar, need not be the same.

With regard to the criterion functions involving S_T, note that S_T does not depend on how the samples are partitioned into clusters. Thus, the clusterings that minimize $|S_W|/|S_T|$ are exactly the same as the ones that minimize $|S_W|$. If we rotate and scale the axes so that S_T becomes the identity matrix, we see that minimizing tr $S_T^{-1}S_W$ is equivalent to minimizing the sum-of-squared-error criterion tr S_W after performing this normalization. Figure 6.14 illustrates the effects of this transformation graphically. Clearly, this criterion suffers from the very defects that we warned about in Section 6.7, and it is probably the least desirable of these criteria.

One final warning about invariant criteria is in order. If different apparent groupings can be obtained by scaling the axes or by applying any other linear transformation, then all of these groupings will be exposed by invariant procedures. Thus, invariant criterion functions are more likely to possess multiple local extrema, and are correspondingly more difficult to extremize.

The variety of the criterion functions we have discussed and the somewhat subtle differences between them should not be allowed to obscure their essential similarity. In every case the underlying model is that the samples form c fairly well separated clouds of points. The within-cluster scatter matrix S_W is used to measure the compactness of these clouds, and the basic goal is to find the most compact grouping. While this approach has proved useful for many problems, it is not universally applicable. For example, it will not extract a very dense cluster embedded in the center of a diffuse cluster, or separate intertwined line-like clusters. For such cases one must devise other criterion functions that are better matched to the structure present or being sought.

6.9 ITERATIVE OPTIMIZATION

Once a criterion function has been selected, clustering becomes a well-defined problem in discrete optimization: find those partitions of the set of samples that extremize the criterion function. Since the sample set is finite, there are only a finite number of possible partitions. Thus, in theory the clustering problem can always be solved by exhaustive enumeration. However, in practice such an approach is unthinkable for all but the simplest problems. There are approximately $c^n/c!$ ways of partitioning a set of n elements into c subsets,† and this exponential growth with n is overwhelming. For example, an exhaustive search for the best set of 5 clusters in 100 samples would require considering more than 10^{67} partitionings. Thus, in most applications an exhaustive search is completely infeasible.

The approach most frequently used in seeking optimal partitions is iterative optimization. The basic idea is to find some reasonable initial partition and to "move" samples from one group to another if such a move will improve the value of the criterion function. Like hill-climbing procedures in general, these approaches guarantee local but not global optimization. Different starting points can lead to different solutions, and one never knows whether or not the best solution has been found. Despite these limitations, the fact that the computational requirements are bearable makes this approach significant.

Let us consider the use of iterative improvement to minimize the sum-of-squared-error criterion J_e, written as

$$J_e = \sum_{i=1}^{c} J_i,$$

where

$$J_i = \sum_{\mathbf{x}\in\mathscr{X}_i} \|\mathbf{x} - \mathbf{m}_i\|^2$$

and

$$\mathbf{m}_i = \frac{1}{n_i} \sum_{\mathbf{x}\in\mathscr{X}_i} \mathbf{x}.$$

Suppose that a sample $\hat{\mathbf{x}}$ currently in cluster $\mathscr{X}_i$ is tentatively moved to $\mathscr{X}_j$. Then $\mathbf{m}_j$ changes to

$$\mathbf{m}_j^* = \mathbf{m}_j + \frac{\hat{\mathbf{x}} - \mathbf{m}_j}{n_j + 1}$$

and J_j increases to

$$\begin{aligned} J_j^* &= \sum_{\mathbf{x}\in\mathscr{X}_j} \|\mathbf{x} - \mathbf{m}_j^*\|^2 + \|\hat{\mathbf{x}} - \mathbf{m}_j^*\|^2 \\ &= \sum_{\mathbf{x}\in\mathscr{X}_j} \left\| \mathbf{x} - \mathbf{m}_j - \frac{\hat{\mathbf{x}} - \mathbf{m}_j}{n_j + 1} \right\|^2 + \left\| \frac{n_j}{n_j + 1} (\hat{\mathbf{x}} - \mathbf{m}_j) \right\|^2 \\ &= J_j + \frac{n_j}{n_j + 1} \|\hat{\mathbf{x}} - \mathbf{m}_j\|^2. \end{aligned}$$

Under the assumption that $n_i \neq 1$ (singleton clusters should not be destroyed), a similar calculation shows that $\mathbf{m}_i$ changes to

$$\mathbf{m}_i^* = \mathbf{m}_i - \frac{\hat{\mathbf{x}} - \mathbf{m}_i}{n_i - 1}$$

and J_i decreases to

$$J_i^* = J_i - \frac{n_i}{n_i - 1} \|\hat{\mathbf{x}} - \mathbf{m}_i\|^2.$$

These equations greatly simplify the computation of the change in the criterion function. The transfer of $\hat{\mathbf{x}}$ from $\mathscr{X}_i$ to $\mathscr{X}_j$ is advantageous if the decrease in J_i is greater than the increase in J_j. This is the case if

$$n_i/(n_i - 1)\ \|\hat{\mathbf{x}} - \mathbf{m}_i\|^2 > n_j/(n_j + 1)\ \|\hat{\mathbf{x}} - \mathbf{m}_j\|^2,$$

which typically happens whenever $\hat{\mathbf{x}}$ is closer to $\mathbf{m}_j$ than $\mathbf{m}_i$. If reassignment is profitable, the greatest decrease in sum of squared error is obtained by selecting the cluster for which $n_j/(n_j + 1)\ \|\hat{\mathbf{x}} - \mathbf{m}_j\|^2$ is minimum. This leads to the following clustering procedure:

```
Procedure: Basic Minimum Squared Error
           1. Select an initial partition of the $n$ samples into clusters and
              compute $J_e$ and the means $\mathbf{m}_1, \ldots, \mathbf{m}_c$.
Loop:      2. Select the next candidate sample $\hat{\mathbf{x}}$. Suppose that $\hat{\mathbf{x}}$ is currently
              in $\mathscr{X}_i$.
                                   cluster
           3. If $n_i = 1$ go to Next; otherwise compute
                 $\rho_j = \begin{cases} \frac{n_j}{n_j + 1} \|\hat{\mathbf{x}} - \mathbf{m}_j\|^2 & j \neq i \\ \frac{n_i}{n_i - 1} \|\hat{\mathbf{x}} - \mathbf{m}_i\|^2 & j = i. \end{cases}$
           4. Transfer $\hat{\mathbf{x}}$ to $\mathscr{X}_k$ if $\rho_k \leq \rho_j$ for all $j$.
           5. Update $J_e$, $\mathbf{m}_i$, and $\mathbf{m}_k$.
Next:      6. If $J_e$ has not changed in $n$ attempts, stop; otherwise go to Loop.
```

† The reader who likes combinatorial problems will enjoy showing that there are exactly

$$\frac{1}{c!} \sum_{i=1}^{c} \binom{c}{i} (-1)^{c-i} i^n$$

partitions of n items into c nonempty subsets. (see W. Feller, *An Introduction to Probability Theory and Its Applications*, Vol. I, p. 58 (John Wiley, New York, Second Edition, 1959)). If $n \gg c$, the last term is the most significant.

If this procedure is compared to the Basic Isodata procedure described in Section 6.4.4, it is clear that the former is essentially a sequential version of the latter. Where the Basic Isodata procedure waits until all n samples have been reclassified before updating, the Basic Minimum Squared Error procedure updates after each sample is reclassified. It has been experimentally observed that this procedure is more susceptible to being trapped at a local minimum, and it has the further disadvantage of making the results depend on the order in which the candidates are selected. However, it is at least a stepwise optimal procedure, and it can be easily modified to apply to problems in which samples are acquired sequentially and clustering must be done in real time.

One question that plagues all hill-climbing procedures is the choice of the starting point. Unfortunately, there is no simple, universally good solution to this problem. One approach is to select c samples randomly for the initial cluster centers, using them to partition the data on a minimum-distance basis. Repetition with different random selections can give some indication of the sensitivity of the solution to the starting point. Another approach is to find the c-cluster starting point from the solution to the $(c - 1)$-cluster problem. The solution for the one-cluster problem is the total sample mean; the starting point for the c-cluster problem can be the final means for the $(c - 1)$-cluster problem plus the sample that is furthest from the nearest cluster center. This approach leads us directly to the so-called hierarchical clustering procedures, which are simple methods that can provide very good starting points for iterative optimization.

6.10 HIERARCHICAL CLUSTERING

6.10.1 Definitions

Let us consider a sequence of partitions of the n samples into c clusters. The first of these is a partition into n clusters, each cluster containing exactly one sample. The next is a partition into $n - 1$ clusters, the next a partition into $n - 2$, and so on until the nth, in which all the samples form one cluster. We shall say that we are at level k in the sequence when $c = n - k + 1$. Thus, level one corresponds to n clusters and level n to one. Given any two samples $\mathbf{x}$ and $\mathbf{x}'$, at some level they will be grouped together in the same cluster. If the sequence has the property that whenever two samples are in the same cluster at level k they remain together at all higher levels, then the sequence is said to be a *hierarchical clustering*. The classical examples of hierarchical clustering appear in biological taxonomy, where individuals are grouped into species, species into genera, genera into families, and so on. In fact, this kind of clustering permeates classificatory activities in the sciences.

For every hierarchical clustering there is a corresponding tree, called a *dendrogram*, that shows how the samples are grouped. Figure 6.15 shows a dendrogram for a hypothetical problem involving six samples. Level 1 shows the six samples as singleton clusters. At level 2, samples x_3 and x_5 have been grouped to form a cluster, and they stay together at all subsequent levels. If it is possible to measure the similarity between clusters, then the dendrogram is usually drawn to scale to show the similarity between the clusters that are grouped. In Figure 6.15, for example, the similarity between the two groups of samples that are merged at level 6 has a value of 30. The similarity values are often used to help determine whether the groupings are natural or forced. For our hypothetical example, one would be inclined to say that the groupings at levels 4 or 5 are natural, but that the large reduction in similarity needed to go to level 6 makes that grouping forced. We shall see shortly how such similarity values can be obtained.

Because of their conceptual simplicity, hierarchical clustering procedures are among the best-known methods. The procedures themselves can be divided into two distinct classes, agglomerative and divisive. *Agglomerative* (bottom-up, clumping) procedures start with n singleton clusters and form

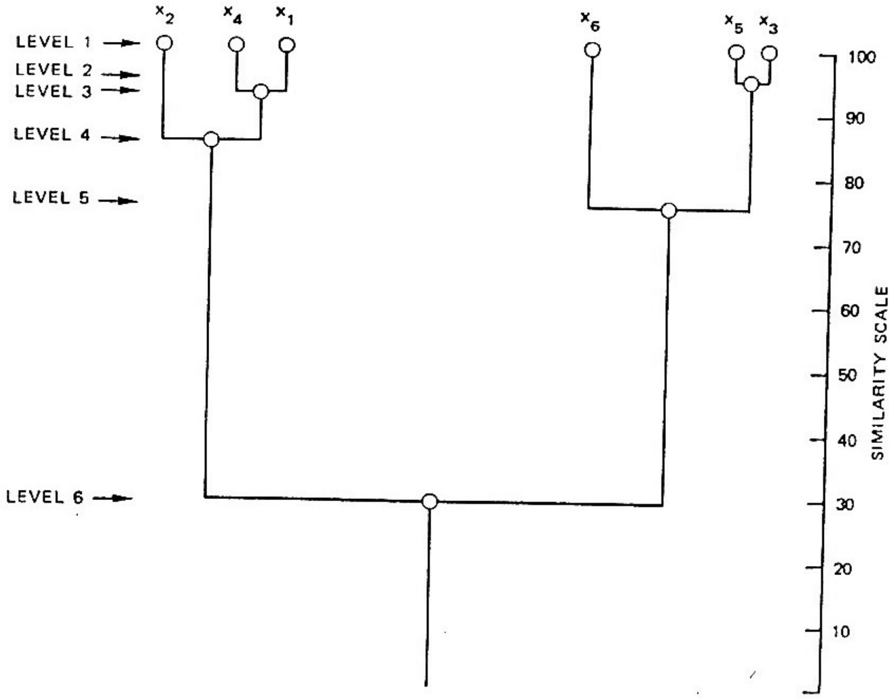


FIGURE 6.15. A dendrogram for hierarchical clustering.

the sequence by successively merging clusters. *Divisive* (top-down, splitting) procedures start with all of the samples in one cluster and form the sequence by successively splitting clusters. The computation needed to go from one level to another is usually simpler for the agglomerative procedures. However, when there are many samples and one is interested in only a small number of clusters, this computation will have to be repeated many times. For simplicity, we shall limit our attention to the agglomerative procedures, referring the reader to the literature for divisive methods.

6.10.2 Agglomerative Hierarchical Clustering

The major steps in agglomerative clustering are contained in the following procedure:

```
Procedure: Basic Agglomerative Clustering
        1. Let $\hat{c} = n$ and $\mathcal{X}_i = \{\mathbf{x}_i\}$, $i = 1, \ldots, n$.
Loop:   2. If $\hat{c} \leq c$, stop.
        3. Find the nearest pair of distinct clusters, say $\mathcal{X}_i$ and $\mathcal{X}_j$.
        4. Merge $\mathcal{X}_i$ and $\mathcal{X}_j$, delete $\mathcal{X}_j$, and decrement $\hat{c}$ by one.
        5. Go to Loop.
```

As described, this procedure terminates when the specified number of clusters has been obtained. However, if we continue until $c = 1$ we can produce a dendrogram like that shown in Figure 6.15. At any level the distance between nearest clusters can provide the dissimilarity value for that level. The reader will note that we have not said how to measure the distance between two clusters. The considerations here are much like those involved in selecting a criterion function. For simplicity, we shall restrict our attention to the following distance measures, leaving extensions to other similarity measures to the reader's imagination:

$$d_{\min}(\mathcal{X}_i, \mathcal{X}_j) = \min_{\mathbf{x}\in\mathcal{X}_i, \mathbf{x}'\in\mathcal{X}_j} \|\mathbf{x} - \mathbf{x}'\|$$

$$d_{\max}(\mathcal{X}_i, \mathcal{X}_j) = \max_{\mathbf{x}\in\mathcal{X}_i, \mathbf{x}'\in\mathcal{X}_j} \|\mathbf{x} - \mathbf{x}'\|$$

$$d_{\text{avg}}(\mathcal{X}_i, \mathcal{X}_j) = \frac{1}{n_i n_j} \sum_{\mathbf{x}\in\mathcal{X}_i} \sum_{\mathbf{x}'\in\mathcal{X}_j} \|\mathbf{x} - \mathbf{x}'\|$$

$$d_{\text{mean}}(\mathcal{X}_i, \mathcal{X}_j) = \|\mathbf{m}_i - \mathbf{m}_j\|.$$

All of these measures have a minimum-variance flavor, and they usually yield the same results if the clusters are compact and well separated. However, if the clusters are close to one another, or if their shapes are not basically hyperspherical, quite different results can be obtained. We shall use the two-dimensional point sets shown in Figure 6.16 to illustrate some of the differences.

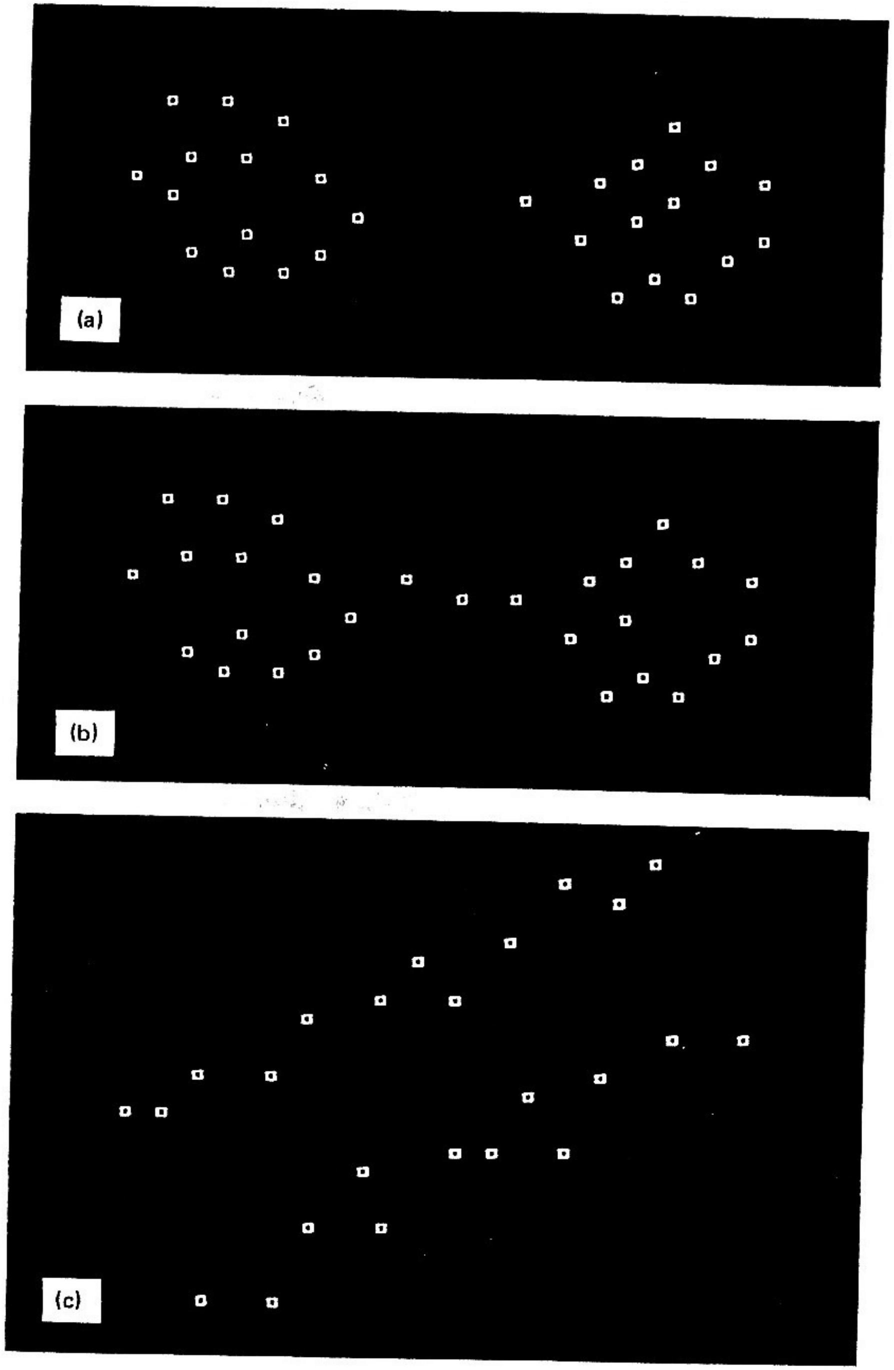


FIGURE 6.16. Three illustrative examples.

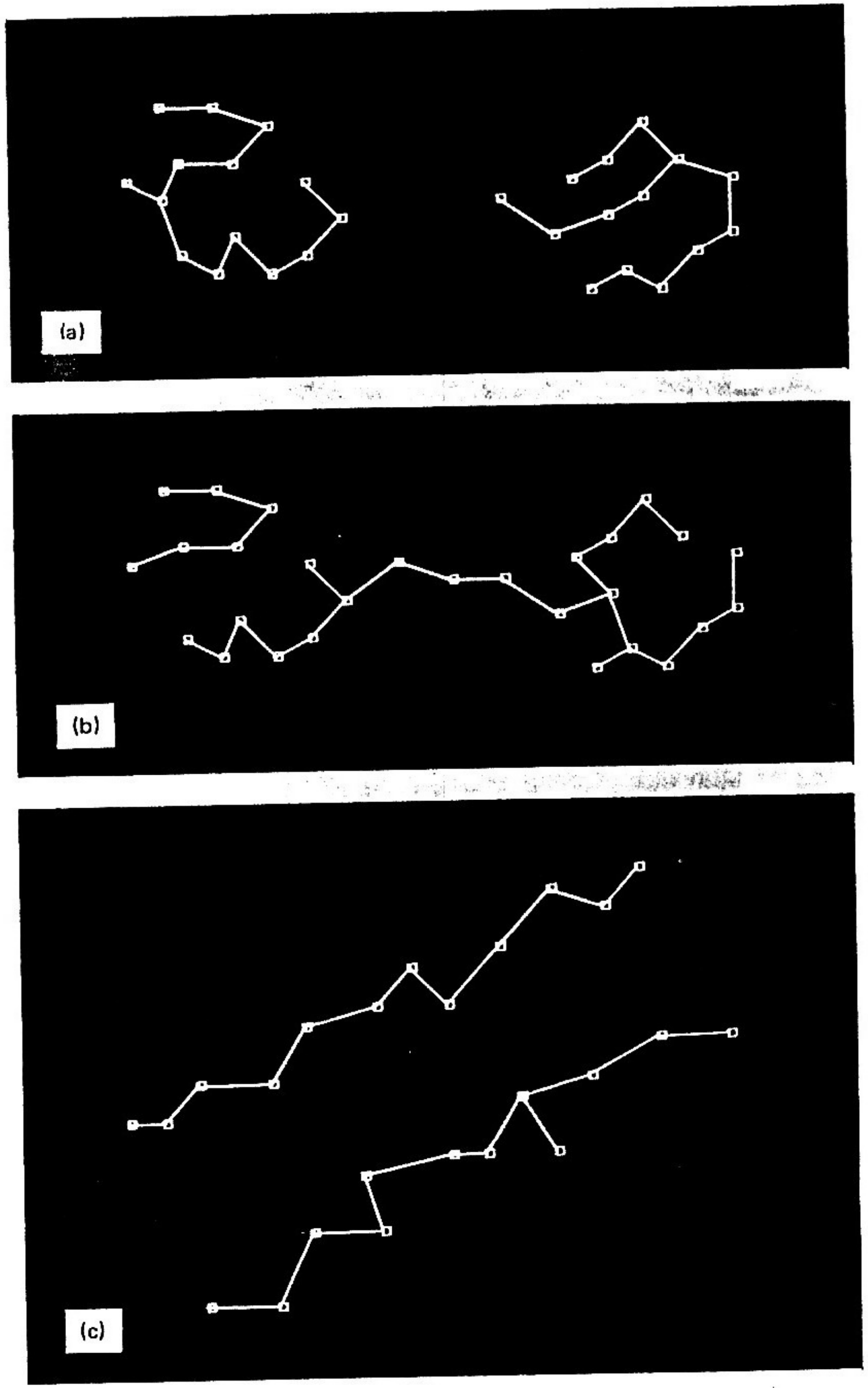


FIGURE 6.17. Results of the nearest-neighbor algorithm.

6.10.2.1 THE NEAREST-NEIGHBOR ALGORITHM

Consider first the behavior when $d_{\min}$ is used.* Suppose that we think of the data points as being nodes of a graph, with edges forming a path between nodes in the same subset $\mathscr{X}_i$.† When $d_{\min}$ is used to measure the distance between subsets, the nearest neighbors determine the nearest subsets. The merging of $\mathscr{X}_i$ and $\mathscr{X}_j$ corresponds to adding an edge between the nearest pair of nodes in $\mathscr{X}_i$ and $\mathscr{X}_j$. Since edges linking clusters always go between distinct clusters, the resulting graph never has any closed loops or circuits; in the terminology of graph theory, this procedure generates a *tree*. If it is allowed to continue until all of the subsets are linked, the result is a *spanning tree*, a tree with a path from any node to any other node. Moreover, it can be shown that the sum of the edge lengths of the resulting tree will not exceed the sum of the edge lengths for any other spanning tree for that set of samples. Thus, with the use of $d_{\min}$ as the distance measure, the agglomerative clustering procedure becomes an algorithm for generating a *minimal spanning tree*.

Figure 6.17 shows the results of applying this procedure to the data of Figure 6.16. In all cases the procedure was stopped at $c = 2$; a minimal spanning tree can be obtained by adding the shortest possible edge between the two clusters. In the first case where the clusters are compact and well separated, the obvious clusters are found. In the second case, the presence of a few points located so as to produce a bridge between the clusters results in a rather unexpected grouping into one large, elongated cluster, and one small, compact cluster. This behavior is often called the "chaining effect," and is sometimes considered to be a defect of this distance measure. To the extent that the results are very sensitive to noise or to slight changes in position of the data points, this is certainly a valid criticism. However, as the third case illustrates, this very tendency to form chains can be advantageous if the clusters are elongated or possess elongated limbs.

6.10.2.2 THE FURTHEST-NEIGHBOR ALGORITHM

When $d_{\max}$ is used to measure the distance between subsets, the growth of elongated clusters is discouraged.‡ Application of the procedure can be thought of as producing a graph in which edges connect all of the nodes in

* In the literature, the resulting procedure is often called the *nearest-neighbor* or the *minimum* algorithm. If it is terminated when the distance between nearest clusters exceeds an arbitrary threshold, it is called the *single-linkage* algorithm.

† Although we will not make deep use of graph theory, we assume that the reader has a general familiarity with the subject. A clear, rigorous treatment is given by O. Ore, *Theory of Graphs* (American Math. Soc. Colloquium Publ., Vol. 38, 1962).

‡ In the literature, the resulting procedure is often called the *furthest neighbor* or the *maximum* algorithm. If it is terminated when the distance between nearest clusters exceeds an arbitrary threshold, it is called the *complete-linkage* algorithm.

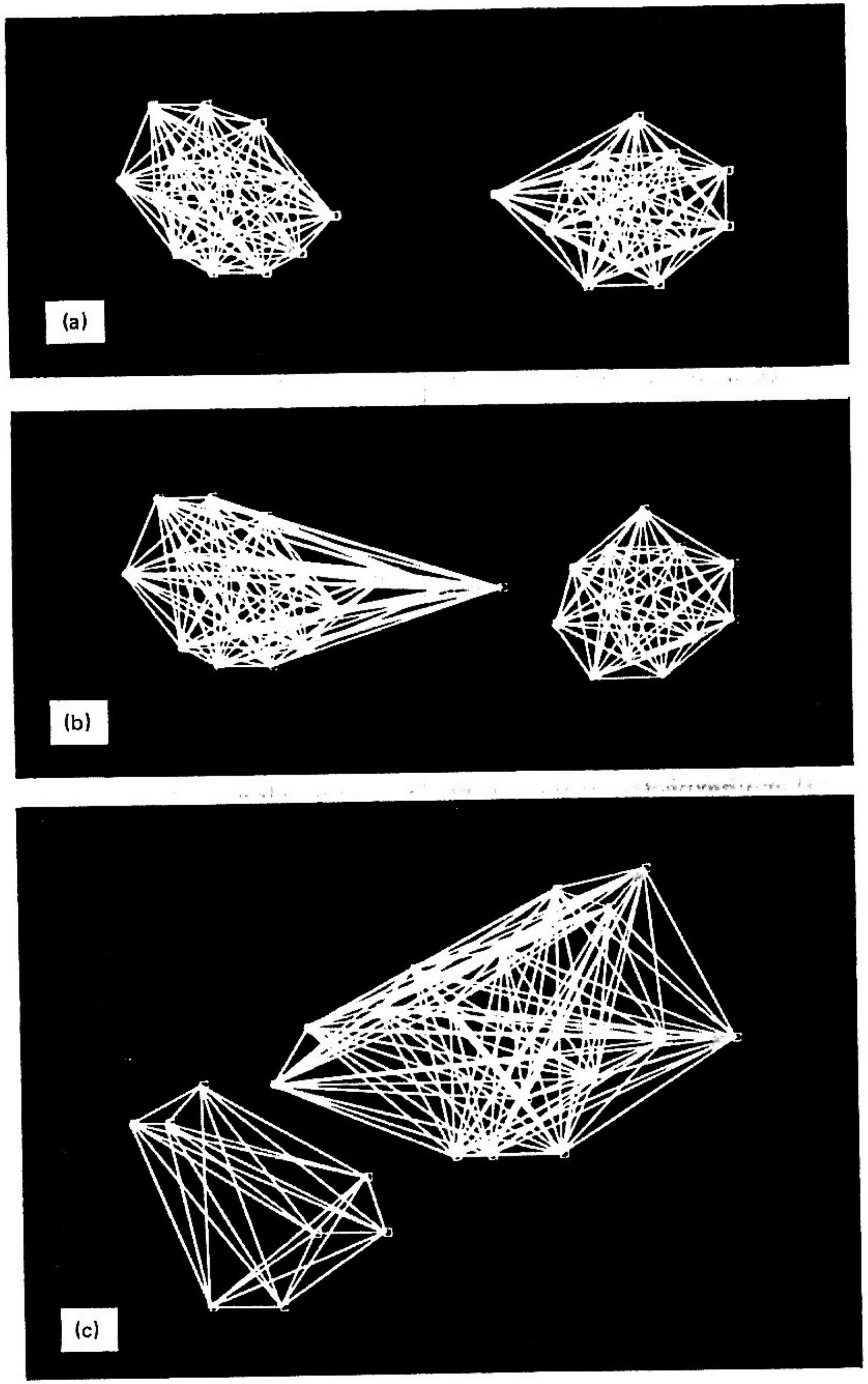


FIGURE 6.18. Results of the furthest-neighbor algorithm.

a cluster. In the terminology of graph theory, every cluster constitutes a *complete* subgraph. The distance between two clusters is determined by the most distant nodes in the two clusters. When the nearest clusters are merged, the graph is changed by adding edges between every pair of nodes in the two clusters. If we define the *diameter* of a cluster as the largest distance between points in the cluster, then the distance between two clusters is merely the diameter of their union. If we define the diameter of a partition as the largest diameter for clusters in the partition, then each iteration increases the diameter of the partition as little as possible. As Figure 6.18 illustrates, this is advantageous when the true clusters are compact and roughly equal in size. However, when this is not the case, as happens with the two elongated clusters, the resulting groupings can be meaningless. This is another example of imposing structure on data rather than finding structure in it.

6.10.2.3 COMPROMISES

The minimum and maximum measures represent two extremes in measuring the distance between clusters. Like all procedures that involve minima or maxima, they tend to be overly sensitive to "mavericks" or "sports" or "outliers" or "wildshots." The use of averaging is an obvious way to ameliorate these problems, and $d_{\rm avg}$ and $d_{\rm mean}$ are natural compromises between $d_{\rm min}$ and $d_{\rm max}$. Computationally, $d_{\rm mean}$ is the simplest of all of these measures, since the others require computing all n_in_j pairs of distances $\|\mathbf{x} - \mathbf{x}'\|$. However, a measure such as $d_{\rm avg}$ can be used when the distances $\|\mathbf{x} - \mathbf{x}'\|$ are replaced by similarity measures, where the similarity between mean vectors may be difficult or impossible to define. We leave it to the reader to decide how the use of $d_{\rm avg}$ or $d_{\rm mean}$ might change the way that the points in Figure 6.16 are grouped.

6.10.3 Stepwise-Optimal Hierarchical Clustering

We observed earlier that if clusters are grown by merging the nearest pair of clusters, then the results have a minimum variance flavor. However, when the measure of distance between clusters is chosen arbitrarily, one can rarely assert that the resulting partition extremizes any particular criterion function. In effect, hierarchical clustering defines a cluster as whatever results from applying the clustering procedure. However, with a simple modification it is possible to obtain a stepwise-optimal procedure for extremizing a criterion function. This is done merely by replacing Step 3 of the Basic Agglomerative Clustering Procedure (Section 6.10.2) by

3′. Find the pair of distinct clusters $\mathcal{X}_i$ and $\mathcal{X}_j$ whose merger would increase (or decrease) the criterion function as little as possible.

This assures us that at each iteration we have done the best possible thing, even if it does not guarantee that the final partition is optimal.

We saw earlier that the use of $d_{\max}$ causes the smallest possible stepwise increase in the diameter of the partition. Another simple example is provided by the sum-of-squared-error criterion function J_e. By an analysis very similar to that used in Section 6.9, we find that the pair of clusters whose merger increases J_e as little as possible is the pair for which the "distance"

$$d_e(\mathscr{X}_i, \mathscr{X}_j) = \sqrt{\frac{n_i n_j}{n_i + n_j}}\, \|\mathbf{m}_i - \mathbf{m}_j\|$$

is minimum. Thus, in selecting clusters to be merged, this criterion takes into account the number of samples in each cluster as well as the distance between clusters. In general, the use of d_e tends to favor growth by adding singletons or small clusters to large clusters over merging medium-sized clusters. While the final partition may not minimize J_e, it usually provides a very good starting point for further iterative optimization.

6.10.4 Hierarchical Clustering and Induced Metrics

Suppose that we are unable to supply a metric for our data, but that we can measure a *dissimilarity* value $\delta(\mathbf{x}, \mathbf{x}')$ for every pair of samples, where $\delta(\mathbf{x}, \mathbf{x}') \geq 0$, equality holding if and only if $\mathbf{x} = \mathbf{x}'$. Then agglomerative clustering can still be used, with the understanding that the nearest pair of clusters is the least dissimilar pair. Interestingly enough, if we define the dissimilarity between two clusters by

$$\delta_{\min}(\mathscr{X}_i, \mathscr{X}_j) = \min_{\mathbf{x}\in\mathscr{X}_i, \mathbf{x}'\in\mathscr{X}_j} \delta(\mathbf{x}, \mathbf{x}')$$

or

$$\delta_{\max}(\mathscr{X}_i, \mathscr{X}_j) = \max_{\mathbf{x}\in\mathscr{X}_i, \mathbf{x}'\in\mathscr{X}_j} \delta(\mathbf{x}, \mathbf{x}'),$$

then the hierarchical clustering procedure will induce a distance function for the given set of n samples. Furthermore, the ranking of the distances between samples will be invariant to any monotonic transformation of the dissimilarity values.

To see how this comes about, we begin by defining the *value* v_k for the clustering at level k. For level 1, $v_1 = 0$. For all higher levels, v_k is the minimum dissimilarity between pairs of distinct clusters at level $k - 1$. A moment's reflection will make it clear that with both $\delta_{\min}$ and $\delta_{\max}$ the value v_k either stays the same or increases as k increases. Moreover, we shall assume that no two of the n samples are identical, so that $v_2 > 0$. Thus, $0 = v_1 < v_2 \leq v_3 \leq \cdots \leq v_n$.

We can now define the *distance* $d(\mathbf{x}, \mathbf{x}')$ between $\mathbf{x}$ and $\mathbf{x}'$ as the value of the lowest level clustering for which $\mathbf{x}$ and $\mathbf{x}'$ are in the same cluster. To show that this is a legitimate distance function, or *metric*, we need to show three things:

(1) $d(\mathbf{x}, \mathbf{x}') = 0 \Leftrightarrow \mathbf{x} = \mathbf{x}'$
(2) $d(\mathbf{x}, \mathbf{x}') = d(\mathbf{x}', \mathbf{x})$
(3) $d(\mathbf{x}, \mathbf{x}'') \leq d(\mathbf{x}, \mathbf{x}') + d(\mathbf{x}', \mathbf{x}'')$.

It is easy to see that the first requirement is satisfied. The lowest level for which $\mathbf{x}$ and $\mathbf{x}$ are in the same cluster is level 1, so that $d(\mathbf{x}, \mathbf{x}) = v_1 = 0$. Conversely, if $d(\mathbf{x}, \mathbf{x}') = 0$, the fact that $v_2 > 0$ implies that $\mathbf{x}$ and $\mathbf{x}'$ must be in the same cluster at level 1, and hence that $\mathbf{x} = \mathbf{x}'$. The truth of the second requirement follows immediately from the definition of $d(\mathbf{x}, \mathbf{x}')$. This leaves the third requirement, the triangle inequality. Let $d(\mathbf{x}, \mathbf{x}') = v_i$ and $d(\mathbf{x}', \mathbf{x}'') = v_j$, so that $\mathbf{x}$ and $\mathbf{x}'$ are in the same cluster at level i and $\mathbf{x}'$ and $\mathbf{x}''$ are in the same cluster at level j. Because of the hierarchical nesting of clusters, one of these clusters includes the other. If $k = \max(i, j)$, it is clear that at level k $\mathbf{x}$, $\mathbf{x}'$, and $\mathbf{x}''$ are all in the same cluster, and hence that

$$d(\mathbf{x}, \mathbf{x}'') \leq v_k.$$

But since the values v_k are monotonically nondecreasing, it follows that $v_k = \max(v_i, v_j)$ and hence that

$$d(\mathbf{x}, \mathbf{x}'') \leq \max(d(\mathbf{x}, \mathbf{x}'), d(\mathbf{x}', \mathbf{x}'')).$$

This is known as the *ultrametric inequality*. It is even stronger than the triangle inequality, since $\max(d(\mathbf{x}, \mathbf{x}'), d(\mathbf{x}', \mathbf{x}'')) \leq d(\mathbf{x}, \mathbf{x}') + d(\mathbf{x}', \mathbf{x}'')$. Thus, all the conditions are satisfied, and we have created a bona fide metric for comparing the n samples.

6.11 GRAPH THEORETIC METHODS

In two or three instances we have used linear graphs to add insight into the nature of certain clustering procedures. Where the mathematics of normal mixtures and minimum-variance partitions seems to keep returning us to the picture of clusters as isolated clumps of points, the language and concepts of graph theory lead us to consider much more intricate structures. Unfortunately, few of these possibilities have been systematically explored, and there is no uniform way of posing clustering problems as problems in graph theory. Thus, the effective use of these ideas is still largely an art, and the reader who wants to explore the possibilities should be prepared to be creative.

We begin our brief look into graph-theoretic methods by reconsidering the simple procedure that produced the graphs shown in Figure 6.8. Here a