$$trace(S_{intra}) + trace(S_{inter}) = 3.3133 + 703.2924 \cong 706.6057 = trace(S)$$

where the approximation symbols are used because of numerical round-off errors.

8.3.3 Partitional Clustering

By partitional clustering (also called non-hierarchical clustering), it is usually meant that the clusters are obtained as a definite partition of the feature space with respect to a fixed number of clusters. A simple partitional clustering algorithm can be immediately obtained in terms of the trace-based dispersion measures introduced in the previous section, which can be used to implement the similarity clustering criterion, in the sense that a good clustering should exhibit low intraclass dispersion and high interclass dispersion. However, as the overall dispersion is preserved, these two possibilities become equivalent. A possible clustering algorithm based on such criteria is

Algorithm: ***Clustering***

```
Assign random classes to each object;

While unstable
   Randomly select an object and randomly change its
      class, avoiding to leave any class empty;
   If the intraclass dispersion, measured for
      instance in terms of the trace of the
      intraclass scatter matrix, increased, reassign
      the original class.
```

The termination condition involves identifying when the clusters have stabilized, which is achieved, for instance, when the number of unchanged successive classifications exceeds a pre-specified threshold (typically two). An important point concerning this algorithm is that the number of clusters usually be pre-specified. This is a consequence of the fact that the intraclass dispersion tends to decrease with larger numbers of clusters (indeed, in the extreme situation where each object becomes a cluster, the scattering becomes null), which tends to decrease the number of clusters if the latter is allowed to vary.

Figure 8.28 presents the progression of decreasing intraclass configurations (the intermediate situations leading to increased intraclass dispersion are not shown) obtained by the above algorithm, together with the respective total, inter and intraclass dispersions. Although the convergence is usually fast, as just a few interactions are usually required, this methodology unfortunately is not guaranteed to converge to the absolute minimal intraclass dispersion (the local minimum problem), a problem that

can be minimized by using simulated annealing (see, for instance, [Press *et al*., 1989; Rose et al., 1993]). In addition, if the trace of the scatter matrices is used, different clusters can be obtained in case the coordinate axes of the feature space are scaled [Jain and Dubes, 1988]. It can also be shown [Jain and Dubes, 1988] that the quantification of the intraclass dispersion in terms of the trace of the respective scatter matrix corresponds to a popular partitional clustering technique known as *square-error method*, which tries to minimize the sum of the squared Euclidean distances between the feature vectors representing the objects in each cluster and the respective mean feature vectors. This can be easily perceived by observing that the trace of the intraclass scatter matrix corresponds to the sum of the squared distances.

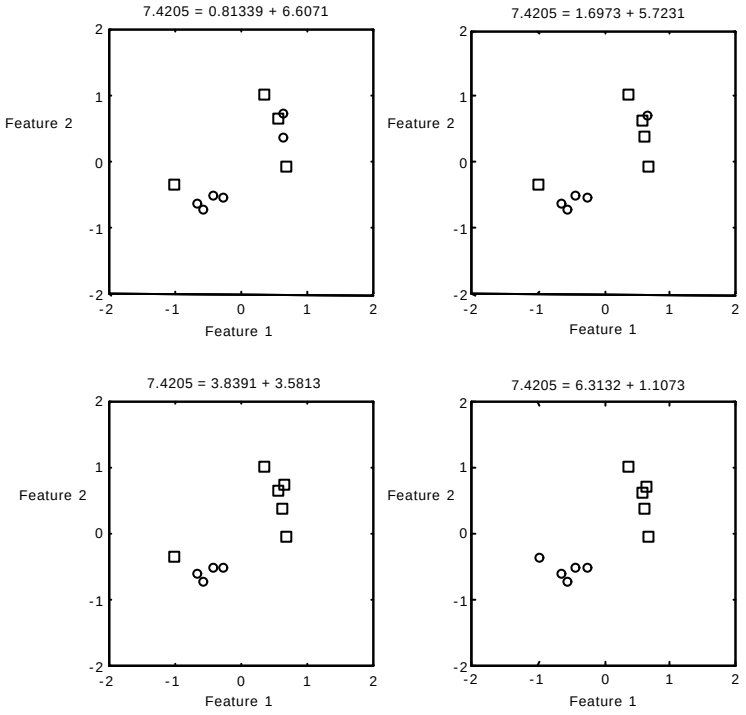

Figure 8.28: *The traces of the scatter matrices ("trace(S)=trace(S_{inter})+trace(S_{intra})") for a sequence of cluster configurations. The last clustering allows the smallest intracluster scattering.*

An alternative clustering technique based on the minimal intraclass dispersion criterion is commonly known as *k-means*, which can be implemented in increasing degrees of sophistication. Here we present one of

terms of the trace of the intraclass scattering matrix. Several additional variations and enhancements of this basic technique have been reported in the literature, including the possibility of merging the clusters corresponding to centroids that are too close (with respect to some supplied threshold) and splitting in two a cluster exhibiting too high a dispersion (this parameter has also to be determined a priori). Both strategies are used in the well-known ISODATA clustering algorithm [Gose, 1996].

Example*:* ***k-means classification***

Apply the *k*-means algorithm in order to cluster into two classes the points characterized in terms of the following features:

Object	Feature 1	Feature 2
X_1	1	1
X_2	3	4
X_3	5	4

Consider as initial prototype points the vectors $P_1 = (0,0)$ and $P_2 = (3,3)$ and use 0.25 as minimum value for the termination criterion.

Solution:

(a) The initial distance matrix is

$D = \begin{bmatrix} \sqrt{2} & 5 & \sqrt{41} \\ 2\sqrt{2} & 1 & \sqrt{5} \end{bmatrix}$, and $L_1 = (X_1)$ and $L_2 = (X_2, X_3)$. Hence:

$P_1 = mean\{X_1\} = X_1$ and $P_2 = mean\{X_2, X_3\} = (1, 4)$, and

$m = \max\left\{\left\|\tilde{\tilde{P}}_1 - \vec{P}_1\right\|, \left\|\tilde{\tilde{P}}_2 - \vec{P}_2\right\|\right\} = \max\left\{\sqrt{2}, 2\sqrt{2}\right\} = 5$

(b) As m>0.25, we have a new interaction:

$D = \begin{bmatrix} 0 & \sqrt{13} & 5 \\ \sqrt{18} & 1 & 1 \end{bmatrix}$, and $L_1 = (X_1)$ and $L_2 = (X_2, X_3)$. Hence

practice. The basic idea of the fuzzy k-means algorithm is described in the following.

Let the probability that an object p_j (recall that $j = 1, 2, \ldots, N$) belongs to the class C_i; $i = 1, 2, \ldots, K$; be represented as $P(C_i \mid p_j)$. At each step of the algorithm, the probabilities are normalized in such a way that for each object p_j we have

$$\sum_{i=1}^{K} P(C_i \mid p_j) = 1$$

The mean for each class at any stage of the algorithm is calculated as

$$P_i = \frac{\sum_{j=1}^{N} [P(C_i \mid p_j)]^a p_j}{\left[\sum_{j=1}^{N} P(C_i \mid p_j)\right]^a}$$

where a is a real parameter controlling the interaction between each observation and the respective mean value. After all the new means P_i have been obtained by using the above equation, the new probabilities are calculated as follows:

$$P(C_i \mid p_j) = \frac{\|p_j - P_i\|^{\frac{2}{1-a}}}{\sum_{q=1}^{K} \|p_j - P_q\|^{\frac{2}{1-a}}}$$

As in the classical k-means, this algorithm stops once the mean values stabilize.

8.3.4 Hierarchical Clustering

By hierarchical clustering it is usually meant that the grouping of M objects into K classes is performed *progressively* according to some parameter, typically the distance or similarity between the feature vectors representing the objects. In other words, the objects that are more similar to one another (e.g., the distance between them is smaller) are grouped into subclasses before objects that are less similar, and the process ends once all the objects have been joined into a single cluster. Observe that, unlike the partitional clustering methodology, which produces a single partition of the objects, hierarchical clustering provides several possible partitions, which can be selected in terms of a distance (or similarity) parameter. Although it is also possible to start with a single cluster and proceed by splitting it into

of the hierarchical clustering technique based on the typical algorithm to be described in the next section.

Table 8.6: *Four definitions of possible distances between two sets A and B.*

Distance between two sets A and B	Comments	Hierarchical clustering
$dist\{A,B\} = \min_{\substack{x\in A,\\ y\in B}}(dist\{x,y\})$	Minimal distance between any of the points of A and any of the points of B	Single linkage
$dist\{A,B\} = \max_{\substack{x\in A,\\ y\in B}}(dist\{x,y\})$	Maximum distance between any of the points of A and any of the points of B	Complete linkage
$dist\{A,B\} = \frac{1}{N_A N_B}\sum_{\substack{x\in A,\\ y\in B}} dist(x,y)$	Average of the distances between each of the N_A points of A and each of the N_B points of B	Group average
$dist\{A,B\} = dist\{C_A, C_B\}$	Distance between the centers of mass (centroids) of the points in set A (i.e., C_A) and B (i.e., C_B)	Centroid

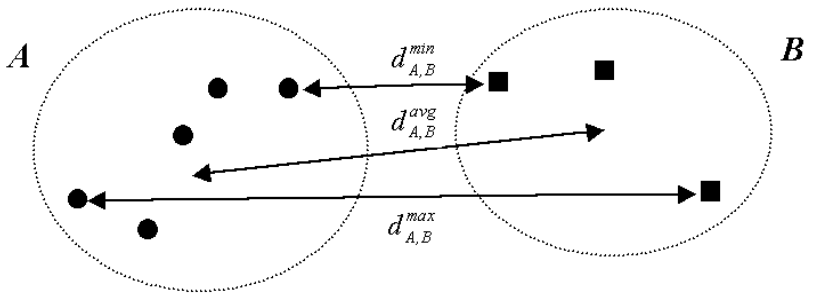

Figure 8.32: *Minimal ($d^{\min}_{A,B}$), maximal ($d^{\max}_{A,B}$), and average ($d^{avg}_{A,B}$) distances between the sets A and B.*