Exercise 7.5.5
Repeat Example 7.5.5 using the GMDAS algorithm.

Hint
See the hint in the previous exercise to obtain a hard clustering from the a posteriori probabilities.

7.5.2 Nonhard Clustering Algorithms

In contrast to the previously examined algorithms, the algorithms in this category assume that each data vector may belong to (or may be compatible with) more than one cluster up to a certain number.

Fuzzy c-Means Algorithm

In the fuzzy c-means (FCM) algorithm each (compact) cluster is represented by a parameter vector θ_j, $j = 1,\ldots,m$. Also, it is assumed that a vector x_i of the data set X does *not necessarily* belong exclusively to a single cluster C_j. Rather, it may belong simultaneously to more than one cluster up to some degree. The variable u_{ij} quantifies the "grade of membership" of x_i in cluster C_j, and it is required that $u_{ij} \in [0, 1]$ and $\sum_{j=1}^{m} u_{ij} = 1$ for all x_i. Once more, the number of clusters, m, is assumed to be known.

The aim of FCM is to move each of the m available l-dimensional parameter vector (representative) θ_j, $j = 1,\ldots,m$, toward regions in the data space that are dense in data points. Finally, the algorithm involves an additional parameter q (>1) called the *fuzzifier*.

FCM is one of the most popular algorithms. It is iterative, starting with some initial estimates, $\theta_1(0),\ldots,\theta_m(0)$, for $\theta_1,\ldots,\theta_m$, respectively, and at each iteration t:

- The grade of membership, $u_{ij}(t-1)$, of the data vector x_i in cluster C_j, $i = 1,\ldots,N$, $j = 1,\ldots,m$, is computed, taking into account the (squared Euclidean) distances of x_i from all θ_j's, $j = 1,\ldots,m$.
- The representatives θ_j's are updated as the weighted means of all data vectors (each data vector x_i is weighted by $u_{ij}^q(t-1)$).

The algorithm terminates when the difference in the values of θ_j's between two successive iterations is small enough. It returns the values of the parameter vectors (representatives) θ_j's and the u_{ij}'s, $i = 1,\ldots,N, j = 1,\ldots,m$. If a hard clustering is required, we can define C_j as the cluster containing all x_i for which $u_{ij} > u_{ik}$, $k \neq j$.

To apply the FCM algorithm, type

$$[theta, U, obj\_fun] = fuzzy\_c\_means(X, m, q)$$

where

X contains the data vectors in its columns,

m is the number of clusters,

q is the fuzzifier,

theta contains the cluster representatives in its columns,

U is an $N \times m$ matrix containing in its ith row the grade of membership of x_i in the m clusters,

obj_fun is a vector whose tth coordinate is the value of the cost function, J, for the clustering produced at the tth iteration.

Remarks

- Like all previously presented cost function optimization algorithms, FCM imposes a clustering structure on X, even if this is not physically justified.
- FCM stems from the minimization of the cost function

$$J(\theta, U) = \sum_{i=1}^{N} \sum_{j=1}^{m} u_{ij}^{q} ||x_i - \theta_j||^2$$

  where $\theta = [\theta_1^T, \ldots, \theta_m^T]^T$, subject to the constraints $u_{ij} \in [0, 1]$ and $\sum_{j=1}^{m} u_{ij} = 1$. That is, $J(\theta, U)$ is a weighted sum of the distances of all x_i's from all θ_j's.
- The involvement of q is critical in fuzzy clustering. Typical values of q are in the range [1.5,3] [Theo 09, Section 14.3].
- The algorithm is sensitive in the presence of outliers because of the requirement that $\sum_{j=1}^{m} u_{ij} = 1$ for all x_i.
- Other fuzzy clustering algorithms where hypercurves of the second degree or hyperplanes are used as representatives have also been proposed. These are mainly useful in image processing applications [Theo 09, Section 14.3.2].

Exercise 7.5.6

Repeat Example 7.5.1, using FCM with $q = 2$.

Exercise 7.5.7

Repeat Example 7.5.3, using FCM with $q = 2$.

Exercise 7.5.8

Repeat Example 7.5.5, using FCM with $q = 2$.

The next exercise shows the influence of the fuzzifier parameter q in the resulting clustering.

Exercise 7.5.9

Apply the FCM on the data set X_3 generated in Example 7.5.1 for $q = 2$, $q = 10$, and $q = 25$. Define and plot the three corresponding hard clusterings, as discussed previously. Compare the u_{ij} parameters and the θ_j's for the three cases and draw conclusions.

Hint

For low values of q (e.g., $q = 2$), each data vector turns out to belong almost exclusively to a single cluster [Theo 09, Section 14.3]. That is, for each x_i, only a single u_{ij} has a very high value (above 90%) among $u_{i1}, \ldots, u_{im}$. However, as q increases, the u_{ij}'s for each data vector x_i tend to become equal to $\frac{1}{m} = 0.25$. Especially in the case where $q = 25$, this leads to a clustering that does not correspond to the true underlying clustering structure of X_3.

The next example shows the effect of outliers on the performance of the FCM.

Example 7.5.7. Apply the FCM algorithm on the data set X_7 generated in Example 7.5.6. Produce a hard clustering, as previously discussed, and plot the results. Comment on the grade of memberships of

the data points in the two obtained clusters. Compare the resulting representatives with those obtained from the application of k-means and PAM on X_7.

Solution. To apply the FCM algorithm on X_7, type

```
[theta,U,obj_fun] = fuzzy_c_means(X7,m,q)
```

To obtain a hard clustering using U, type

```
[qw,bel]=max(U');
```

where *bel* contains the cluster labels of the data vectors.
Plot the clustering results, using different symbols and colors for vectors that belong to different clusters, as in step 1 of Example 7.5.1 (see Figure 7.8).

Observation of the grade of memberships reveals that

- For the first 100 points, the grade of memberships in cluster C_1 is significantly higher (>89.4%) than that in cluster C_2 (<10.6%) (see Figure 7.8).
- For the next 100 points, the grade of memberships in cluster C_2 is significantly higher (>97.2%) than that in cluster C_1 (<2.8%).
- For the last 16 points (outliers), the grade of memberships in clusters C_1 and C_2 are significant (>66.62% for C_1 and >30.10% for C_2), so their effect on the computation of both θ_1 and θ_2 is not negligible.

Comparing the results shown in Figure 7.8 with those in Figure 7.7, we observe that the estimates of θ_2 (the representative of the upper right cluster) are better for k-means and PAM than for FCM (this is because the outliers have no effect on the estimation of θ_2 in k-means and PAM, which is not the case in FCM), and that the estimates of θ_1 (the representative of the other cluster) are better in PAM and FCM than in k-means, in the sense that in PAM and FCM θ_1 remains close to the main volume of the data set.

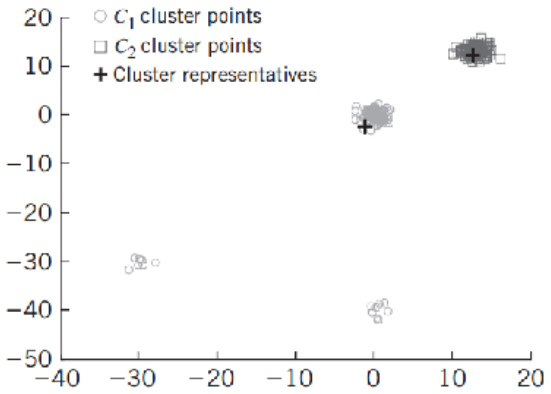

FIGURE 7.8

Clustering obtained by FCM on data set X_7 in Example 7.5.7. The three lower left groups of points are from cluster C_1; the upper right group of points constitute cluster C_2.

This happens because in FCM the outliers contribute to the estimation of θ_1 by (at least) 30%, while in k-means they contribute by 100% (since in the hard clustering case a vector belongs exclusively (100%) to a single cluster). ■

Possibilistic c-Means Algorithm

This algorithm (known as PCM) is also appropriate for unraveling compact clusters. The framework here is similar to the one used in FCM: Each data vector x_i is associated with a cluster C_j via a scalar u_{ij}. However, the constraint that all u_{ij}'s for a given x_i sum up to 1 is removed (it is only required that they lie in the interval [0, 1]). As a consequence, the u_{ij}'s (for a given x_i) are not interrelated anymore and they cannot be interpreted as "grade of membership" of vector x_i in cluster C_j, since this term implies that the summation of u_{ij}'s for each x_i should be constant. Rather, u_{ij} is interpreted as the "degree of compatibility" between x_i and C_j. The degree of compatibility between x_i and C_j is independent of that between x_i and the remaining clusters.

As with FCM, a parameter q (> 1) is involved in PCM. However it does not act as a fuzzifier as it was the case in FCM. Also, in contrast to FCM, PCM is less sensitive in knowing the "exact" number of clusters. Rather, an overestimated value of m can be used (see also the remarks given below). A set of parameters $\eta_j, j = 1,\ldots,m$, each one corresponding to a cluster, is also required (loosely speaking, they are estimates of the "size" of the cluster [Theo 09, Section 14.4]). Like k-means and FCM, PCM's goal is to move the θ_j's to regions of space that are dense in data points.

PCM is iterative. It starts with some initial estimates, $\theta_1(0),\ldots,\theta_m(0)$, for $\theta_1,\ldots,\theta_m$, respectively, and at each iteration,

- The "degree of compatibility", $u_{ij}(t-1)$, of the data vector x_i to cluster $C_j, i = 1,\ldots,N, j = 1,\ldots,m$, is computed, taking into account the (squared Euclidean) distance of x_i from θ_j and the parameter η_j.
- The representatives, θ_j's, are updated, as in FCM, as the weighted means of all data vectors (each data vector x_i is weighted by $u_{ij}^q(t-1)$).

The algorithm terminates when the difference in the values of θ_j's between two successive iterations is small enough. It returns the values of the parameter vectors (representatives) θ_j's and the "compatibility coefficients" u_{ij}'s, $i = 1,\ldots,N, j = 1,\ldots,m$.

To apply PCM on a data set X, type

$$[U, theta] = possibi(X, m, eta, q, sed, init\_proc, e\_thres)$$

where

X contains the data vectors in its columns,

m is the number of clusters,

eta is an m-dimensional array whose jth coordinate is the η_j parameter for the cluster C_j,

q is the "q" parameter of the algorithm,

sed is a scalar integer used as the seed for the built-in MATLAB function *rand*,

init_proc is an integer taking values 1, 2, or 3, with 1 corresponding to the *rand_init* initialization procedure, which chooses randomly m vectors from the smallest hyper-rectangular that contains all vectors of X and its sides are parallel to the axes; 2 corresponding to *rand_data_init*, which