Chapter 6. Fuzzy Clustering

• ***Clustering:***

Clustering essentially deals with the task of splitting a set of patterns into a number of more-or-less homogeneous classes (clusters) with respect to a suitable similarity measure such that the patterns belonging to any one of the clusters are similar and the patterns of different clusters are as dissimilar as possible.

~ Depending on the structure of the partition, two different kinds of clustering can be distinguished:

(1) Hard clustering:

The partition is a set of disjoint subsets of patterns in such a way that each object belongs to exactly one cluster.

(2) Fuzzy clustering:

Each object belongs to a cluster to a certain degree according to the membership function of the cluster. The partition is fuzzy in the sense that a single object can simultaneously belong to multiple clusters.

• ***Hard clustering***.

(1) k – means algorithm :

~ " k" simply refers to the number of desired clusters and can be replaced by any desired index .

~ Feature vectors are continuously reassigned to clusters, and the cluster centroids are updated until no further reassignment is necessary.

~ Algorithm :

Initialization :

randomly generate $\bar{K}$ mean vectors (centroid) of each cluster, $\bar{X}_k$, k = 1 , 2 … $\bar{K}$.

Recursion :

Step 1. For each feature vector, X , in the training set , assign X to C_{k^*},

where $k^* = \arg\min_k d\left(X, \bar{X}_k\right)$.

d(· , ·) represents some distance measure in the feature space.

e.g. Euclidean distance :

$$d\left(X, \overline{X}\right) = (X - \overline{X})^T (X - \overline{X})$$

Mahalanobis distance

$$d\left(X, \overline{X}\right) = \left(X - \overline{X}\right)^T C_{\overline{X}}^{-1} (X - \overline{X}),$$

$C_{\overline{X}}$: covariance matrix

Step 2. Recompute the cluster centroids and return to step 1 if any of the centroids change from the last iteration.

$$\overline{X}_{k^*}^{(t+1)} = \overline{X}_{k^*}^{(t)} + \frac{1}{N+1}\left(X - \overline{X}_{k^*}^{(t)}\right), \qquad \boxed{\overline{X}_{K^*}^{(t+1)} = \frac{X + N \cdot \overline{X}_{K^*}^{(t)}}{N+1}}$$

N : number of samples in cluster k^* at t .

(2). ***Kohonen learning rule*** *or* ***winner–take–all learning rule***

Similarity matching :

$$k^* = \arg\min_{k} d\left(X, \overline{X}_k\right).$$

Updating

$$\overline{X}_{k^*}^{(t+1)} = \overline{X}_{k^*}^{(t)} + \alpha^{(t)}\left(X - \overline{X}_{k^*}^{(t)}\right)$$

$$\overline{X}_{k}^{(t+1)} = \overline{X}_{k}^{(t)} , \quad k = 1, 2, \cdots, K , \quad k \neq k^*.$$

$\alpha^{(t)}$: a suitable learning constant at the t-th time step.

• ***Formulation of fuzzy clustering***

~ Given $X = \{\vec{x}_1, \vec{x}_2 \cdots \vec{x}_n\}$, $\vec{x}_i \in R^p$, perform a partition of X into C fuzzy sets w.r.t. a given criterion.

C: number of clusters

Criterion: optimization of an objective function

Result: a partition matrix U s.t.

$$U = [u_{ij}]_{i=1\sim c,\, j=1\sim n} \qquad u_{ij} \in [0,\ 1]$$

u_{ij}:expresses the degree to which the element $\vec{x}_j$ belongs to the *i*-th cluster.

additional constraints:

$$\sum_{i=1}^{C} u_{ij} = 1, \quad \text{for } j = 1 \sim n \qquad \cdots(1)$$

$$0 < \sum_{j=1}^{n} u_{ij} < n, \quad \text{for } i = 1\sim\text{C} \qquad \cdots(2)$$

~ A general form of the objective function:

$$J(u_{ij}, \vec{v}_i) = \sum_{i=1}^{C} \sum_{j=1}^{n} g[w(\vec{x}_i),\ u_{ij}]\, d(\vec{x}_j, \vec{v}_i)$$

$w(\vec{x}_j)$: priori weight for $\vec{x}_i$

$d(\vec{x}_j, \vec{v}_i)$: degree of similarity between $\vec{x}_i$ and $\vec{v}_i$ (central vector of the ith cluster), and should satisfy two axioms:

$(a.1)$ $d(\vec{x}_j, \vec{v}_i) \ge 0$

$(a.2)$ $d(\vec{x}_j, \vec{v}_i) = d(\vec{v}_i, \vec{x}_j)$

~ Fuzzy clustering can be formulated as:

Minimize $J(u_{ij}, \vec{v}_i)$ $\quad i = 1, 2 \cdots C \quad j = 1, 2 \cdots n \qquad (3)$

subject to Eqs. (1) and (2)

• Fuzzy c-means (FCM) : [Bezdek , 1981]

Based on Eq.(3) and

$$J(u_{ij}, \vec{v}_i) = \sum_{i=1}^{C} \sum_{j=1}^{n} u_{ij}^m \| \vec{x}_j - \vec{v}_i \|^2 \qquad m > 1$$

m: exponential weight , influences the degree of fuzziness of the membership (partition) matrix.

~Derivation of $\vec{v}_i$ and u_{ij} :

1.$\vec{v}_p$,(for fixed u_{ij}) $\quad 0=\dfrac{\partial J}{\partial \vec{v}_p}=\dfrac{\partial}{\partial \vec{v}_p}\left(\sum_{i=1}^{C}\sum_{j=1}^{n}u_{ij}^{m}\parallel \vec{x}_j-\vec{v}_i \parallel^2\right)$

$$=\sum_{j=1}^{n}u_{pj}^{m}\frac{\partial}{\partial \vec{v}_p}\parallel \vec{x}_j-\vec{v}_p \parallel^2$$

$$=\sum_{j=1}^{n}u_{pj}^{m}\frac{\partial}{\partial \vec{v}_p}\left(\vec{x}_j-\vec{v}_p\right)^T\left(\vec{x}_j-\vec{v}_p\right)$$

$$=2\sum_{j=1}^{n}u_{pj}^{m}\left(\vec{x}_j-\vec{v}_p\right)$$

$$\boxed{\vec{v}_p=\frac{1}{\sum_{j=1}^{n}\left(u_{pj}\right)^m}\sum_{j=1}^{n}\left(u_{pj}\right)^m\vec{x}_j \quad , \quad p=1\ ,\ 2\ ,\cdots C \quad (4)}$$

2 u_{ij}(for fixed $\vec{v}_i$): The constraint of Eq. (1) can be expressed by introducing a Lagrange multiplier λ .

$$J(u_{ij},\vec{v}_i,\lambda)=\sum_{i=1}^{C}\sum_{j=1}^{n}u_{ij}^{m}\parallel\vec{x}_j-\vec{v}_i\parallel^2-\lambda\left(\sum_{i=1}^{C}u_{ij}-1\right)$$

$$0=\frac{\partial J}{\partial u_{pq}}=m\cdot\left(u_{pq}\right)^{m-1}\parallel \vec{x}_q-\vec{v}_p \parallel^2-\lambda\ ,\ u_{pq}=\left(\frac{\lambda}{m}\right)^{\frac{1}{m-1}}\left(\frac{1}{\parallel\vec{x}_q-\vec{v}_p\parallel^2}\right)^{\frac{1}{m-1}} \quad (5)$$

$$0=\frac{\partial J}{\partial \lambda} \quad \Rightarrow \quad 1=\sum_{i=1}^{C}u_{iq}=\left(\frac{\lambda}{m}\right)^{\frac{1}{m-1}}\sum_{i=1}^{C}\left(\frac{1}{\parallel\vec{x}_q-\vec{v}_i\parallel^2}\right)^{\frac{1}{m-1}}$$

$$\left(\frac{\lambda}{m}\right)^{\frac{1}{m-1}}=\frac{1}{\sum_{i=1}^{C}\left(\parallel\vec{x}_q-\vec{v}_i\parallel^2\right)^{\frac{1}{m-1}}},\ \text{substitute into Eq. (5)}$$

$$\boxed{u_{pq}=\frac{\left(1/\parallel\vec{x}_q-\vec{v}_p\parallel^2\right)^{\frac{1}{m-1}}}{\sum_{i=1}^{C}\left(1/\parallel\vec{x}_q-\vec{v}_i\parallel^2\right)^{\frac{1}{m-1}}},\quad p=1,2,\ldots,C,\ \ q=1,2,\ldots,n \qquad (6)}$$

~ The system described by Eqs.(4) and (6) can not be solved analytically

~ Solution : by iterative approach

Algorithm FCM : Fuzzy C-Means Algorithm

Step1 : Select $C \ (2 \le C \le n),\ m \ (1 < m < \infty)$,

initial partition $U^{(0)}$ and termination ε

set the iteration index l to 0.

Step2 : Calculate the fuzzy cluster centers

$\{ \vec{v}_p^{\,(l)} \mid p = 1,2,\cdots,C \}$ by using $U^{(l)}$ and Eq. (4).

Step3 : Calculate $U^{(l+1)}$ by using $\{ \vec{v}_p^{\,(l)} \mid p = 1,2,\cdots,C \}$ and Eq. (6).

Step4 : Calculate $\Delta = \left\| U^{(l+1)} - U^{(l)} \right\| = \max\limits_{i,j} \left| u_{ij}^{(l+1)} - u_{ij}^{(l)} \right|$

If $\Delta > \varepsilon$, then set $l = l + 1$ and go to Step2

if $\Delta \le \varepsilon$, then stop.

EX : The data for the butterfly shown below are processed

by a fuzzy 2-means algorithm with

$$U^{(0)} = \begin{bmatrix} 0.854 & 0.146 & 0.854 & 0.854 & \cdots & 0.854 \\ 0.146 & 0.854 & 0.146 & 0.146 & \cdots & 0.146 \end{bmatrix}_{2\times 15}$$

$$\varepsilon = 0.01 \ , \ \mathrm{m} = 1.25$$

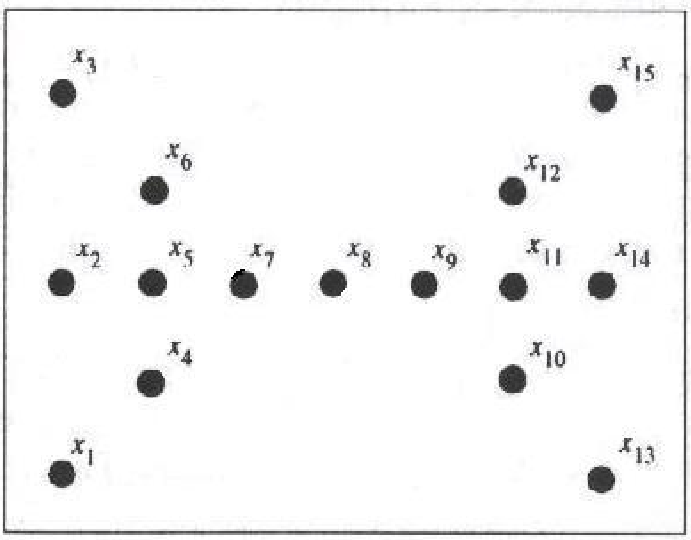

Figure 8.3 Data for the "butterfly" in Example 8.3

Result :

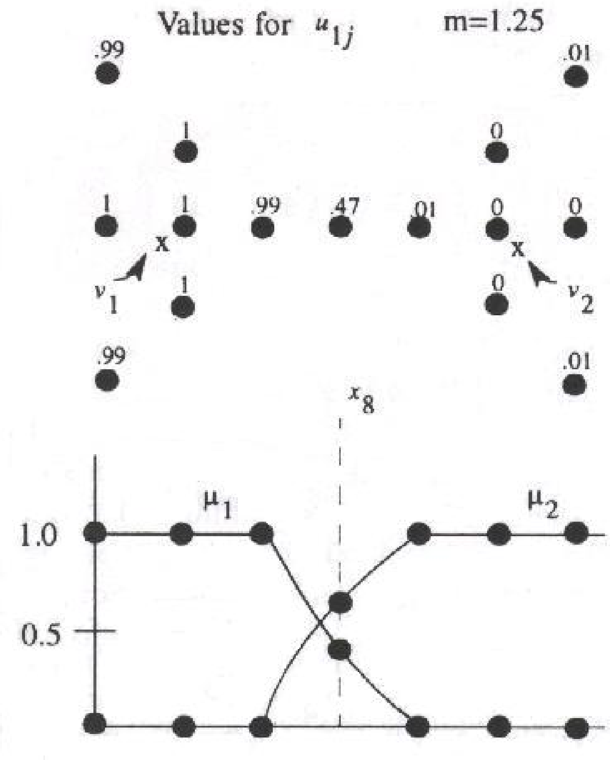

Figure 8.4 Clustering result for $m = 1.25$ in Example 8.3.

Remark :

(1) $m\uparrow$, fuzziness of the partition matrix $\uparrow$.

(2) $m \to \infty$, $\mathrm{U} \to \left[\frac{1}{\mathrm{c}}\right]$

(3) In practice , choose m from the range [1.5 , 30].

- Determination of the "correct" number of clusters, C

~ no exact solution to this question.

~ some scalar measures of partitioning fuzziness have been used as validity indicators, e.g.

1. Partitioning entropy :

$$H(U, C) = \frac{1}{n} \sum_{j=1}^{n} \sum_{i=1}^{c} \left| u_{ij} \ln u_{ij} \right|$$

2. Partition coefficient :

$$F(U, C) = \frac{1}{n} \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^{2}$$

3. Proportion exponent :

$$P(U, C) = -\ln \left\{ \prod_{j=1}^{n} \left[\sum_{k=1}^{u_j^{-1}} (-1)^{k+1} \binom{c}{k} (1 - k u_j)^{(c-1)} \right] \right\}$$

where $\mathrm{u}_{\mathrm{j}} = \max\limits_{1 \le i \le c} u_{ij}$, $\left[\mathrm{u}_{\mathrm{j}}^{-1} \right]$ = greatest integer $\le \left(\frac{1}{\mathrm{u}_{\mathrm{j}}} \right)$

Remarks: (1) $H = 0$ and $F = 1$ if $u_{ij} \in \{0,1\}$ (hard partitioning)

(2) $H = \ln \mathrm{C}$ and $F = \frac{1}{\mathrm{C}}$ if $\mathrm{u}_{\mathrm{ij}} = \frac{1}{C}$, for all i , j

(3) $0 \le H \le \ln C$, $\frac{1}{C} \le F \le 1$, $0 \le P \le \infty$

~ Assume that the clustering structure is better identified when more points concentrate around the cluster centers i.e. the crisper the partitioning matrix is .

~Based on the above assumption , the heuristic rules for selecting the best partitioning number C are:

$$\min_{c=2}^{n-1} \left\{ \min_{U \in \Omega_C} \left[H(U, C) \right] \right\}$$

$$\max_{c=2}^{n-1} \left\{ \max_{U \in \Omega_C} \left[F(U, C) \right] \right\}$$

$$\max_{c=2}^{n-1} \left\{ \max_{U \in \Omega_C} \left[P(U, C) \right] \right\}$$

Ω_C : The set of all optimal solution for a given C

Ex : Iris data .

It contains three categories of the species Iris .

$\in = 0.05$, $m = 40$.

only one guess is used for U^0

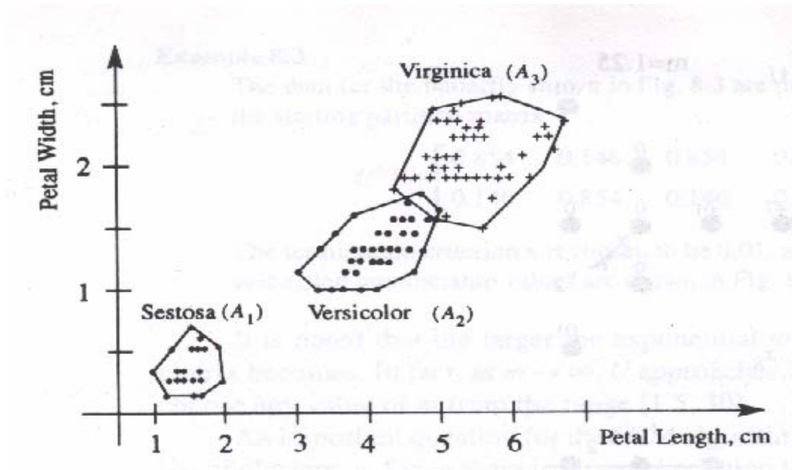

Fig. Iris data set in example

Table is Validity indicators in example

C	Partition entropy $H(U;c)$	Partition coefficient $F(U;c)$	Proportion exponent $P(U;c)$
2	0.55 ←	0.63 ←	109
3	0.92	0.45	113 ←
4	1.21	0.35	111
5	1.43	0.27	62
6	1.60	0.23	68