Fuzzy C-Means Clustering Algorithm

- FCM is an iterative algorithm. The aim is to find cluster centers (centroids) that minimize a dissimilarity function.
- To initiate the fuzzy partitioning, the membership matrix (U) is randomly initialized accordingly

$$\sum_{i=1}^{c} u_{ij} = 1, \forall j = 1,\ldots\ldots,n$$

- The algorithm minimizes a dissimilarity (or distance) function which is given below

$$J(U, c_1, c_2, \ldots\ldots, c_c) = \sum_{i=1}^{c} j_i = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^{m} d_{ij}^{2}$$

- where u_{ij} is between 0 and 1, c_i is the centroid of cluster i, d_{ij} is the Euclidian distance between i^{th} centroid and j^{th} data point and m is a weighting exponent on each membership (1 for hard clustering and increasing for fuzzy clustering)

FCM Clustering Algorithm

- To reach a minimum of dissimilarity function there are two conditions. These are given in (2.5) and (2.6)

$$c_i = \frac{\sum_{j=1}^{n} u_{ij}^m x_j}{\sum_{j=1}^{n} u_{ij}^m} \tag{2.5}$$

Point *j*'s membership of cluster *i*

Fuzziness exponent

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{kj}} \right)^{2/(m-1)}} \tag{2.6}$$

Distance from point *j* to current cluster centre *i*

Distance from point *j* to other cluster centres *k*

- By iteratively updating the cluster centers and the membership grades for each data point, FCM iteratively moves the cluster centers to the optimal location within a data set.

FCM Algorithm

Algorithm: FCM

Input: X, c, t, m

X is a data set, c is the number of clusters, t is the convergence threshold and m is the exponential weight

Output: U – membership matrix

1: Randomly initialize matrix U with c clusters

2: **repeat**

3: Calculate c_i

4: Compute dissimilarity between centroids and data points

5: Compute a new U

6: **until** the improvement over previous iteration is below t.

Fuzzy C-Means Steps

- **Steps:**
  - **Step1:** choose random centroid at least 2
  - **Step2:** compute membership matrix.

$$u_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|}\right)^{\frac{2}{m-1}}}$$

$$= \frac{1}{\left(\frac{\|x_i - c_j\|}{\|x_i - c_1\|}\right)^{2/(m-1)} + \left(\frac{\|x_i - c_j\|}{\|x_i - c_2\|}\right)^{2/(m-1)} + \cdots + \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|}\right)^{2/(m-1)}}$$

where $\|x_i - c_j\|$ is the Distance from point *i to current cluster centre j*, $\|x_i - c_k\|$ is the Distance from point *i to other cluster centers k.(note if we have 2D* data we use euclidean distance).

Fuzzy C-Means Steps

- **Step3:** calculate the c cluster centers.

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} \cdot x_i}{\sum_{i=1}^{N} u_{ij}^{m}}$$

Fuzzy C-Means Example

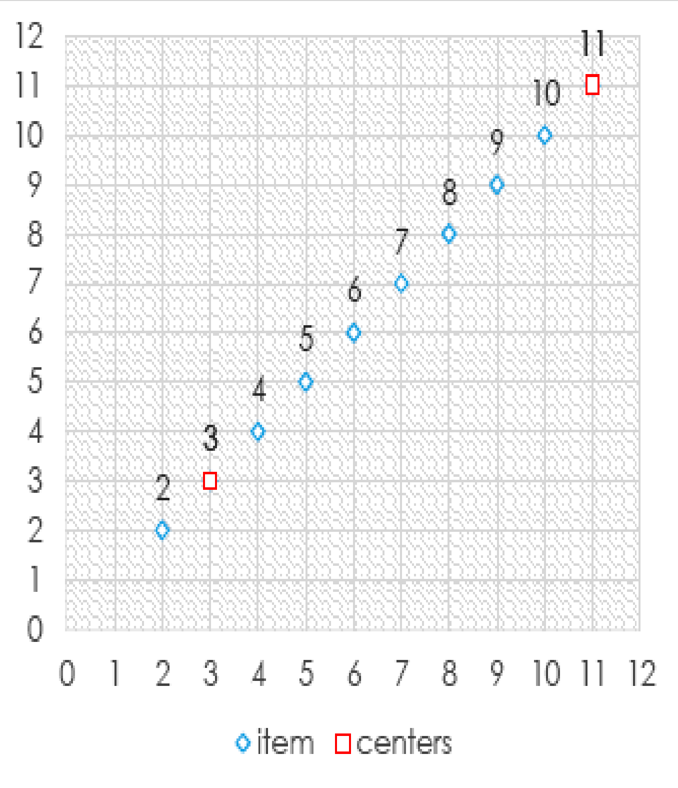
- Let x=[2 3 4 5 6 7 8 9 10 11] , m=2, number of cluster C=2, c_1=3 , c_2=11.
- Step 1: for first iteration calculate membership matrix.
- For node 2 (1st element):

$$U11 = \frac{1}{\left(\frac{2-3}{2-3}\right)^{\frac{2}{2-1}} + \left(\frac{2-3}{2-11}\right)^{\frac{2}{2-1}}} = \frac{1}{1+\frac{1}{81}} = \frac{81}{82} = 98.78\%$$

The membership of first node to first cluster

$$U12 = \frac{1}{\left(\frac{2-11}{2-3}\right)^{\frac{2}{2-1}} + \left(\frac{2-11}{2-11}\right)^{\frac{2}{2-1}}} = \frac{1}{81+1} = \frac{1}{82} = 1.22\%$$

The membership of first node to second cluster



Fuzzy C-Means Example

- For node 3 (2^{nd} element):

  U21 = 100%

  The membership of second node to first cluster

  U22 = 0%

  The membership of second node to second cluster

Fuzzy C-Means Example

- For node 4 (3rd element):

$$U31 = \frac{1}{\left(\frac{4-3}{4-3}\right)^{\frac{2}{2-1}} + \left(\frac{4-3}{4-11}\right)^{\frac{2}{2-1}}} = \frac{1}{1+\frac{1}{49}} = \frac{1}{\frac{50}{49}} = 98\%$$

The membership of first node to first cluster

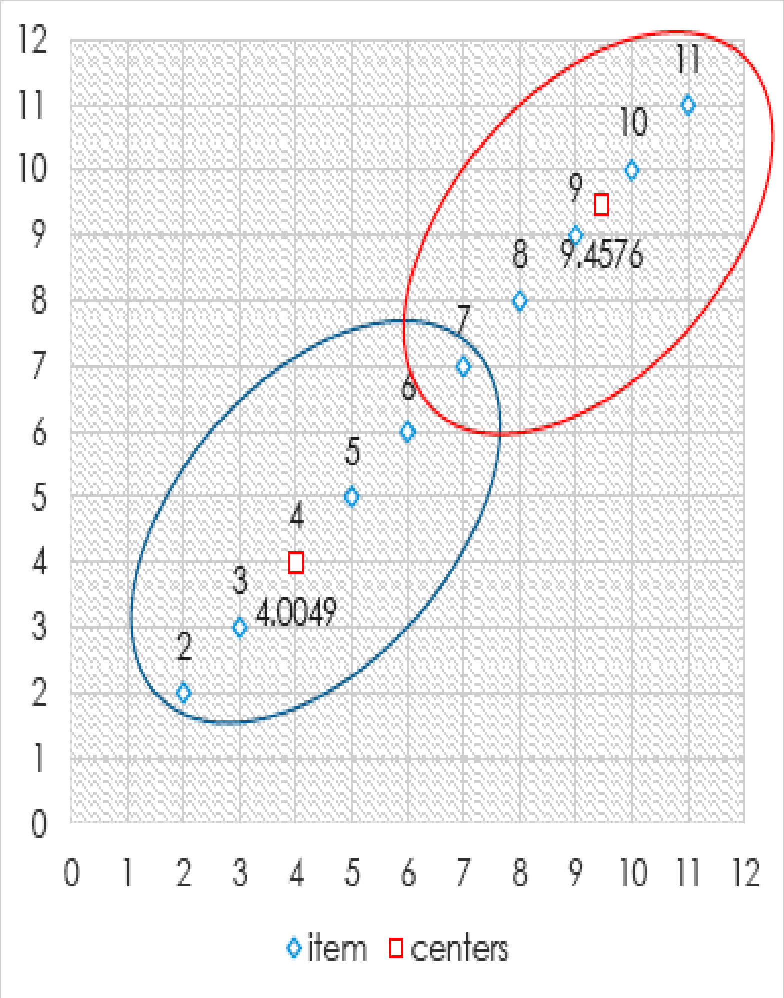
$$U32 = \frac{1}{\left(\frac{4-11}{4-3}\right)^{\frac{2}{2-1}} + \left(\frac{4-11}{4-11}\right)^{\frac{2}{2-1}}} = \frac{1}{49+1} = \frac{1}{50} = 2\%$$

The membership of first node to second cluster

Fuzzy C-Means Example

- And so on until we complete the set and get U matrix

X	cluster1	cluster2
2	0.9878	0.0122
3	1.0000	0
4	0.9800	0.0200
5	0.9000	0.1000
6	0.7353	0.2647
7	0.5000	0.5000
8	0.2647	0.7353
9	0.1000	0.9000
10	0.0200	0.9800
11	0	1.0000



Fuzzy C-Means Example

- Step2: now we compute new centers

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} \cdot x_i}{\sum_{i=1}^{N} u_{ij}^{m}}$$

$$c1 = \frac{(98.78\%)^2 * 2 + (100\%)^2 * 3 + (98\%)^2 * 4 + (50\%)^2 * 7 + \ldots}{(98.78\%)^2 + (100\%)^2 + (98\%)^2 + (50\%)^2 + \ldots} = 4.0049$$

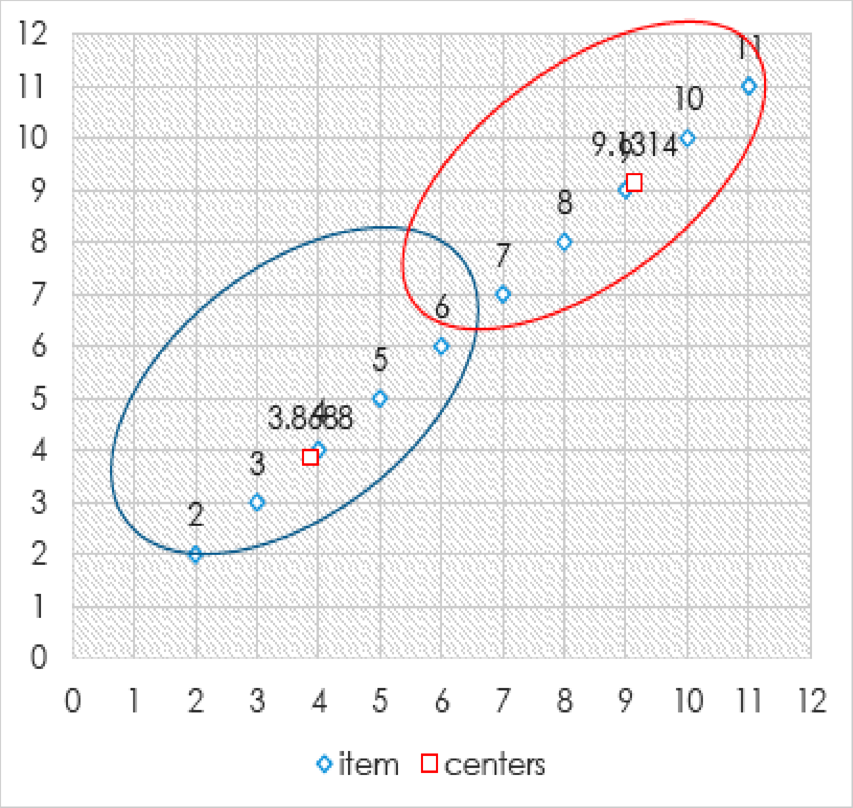
And c2=9.4576

Fuzzy C-Means Example

- **Repeat step until there is visible change.**
- Final iteration :
- U =

X	cluster1	cluster2
2	0.9357	0.0643
3	0.9803	0.0197
4	0.9993	0.0007
5	0.9303	0.0697
6	0.6835	0.3165
7	0.3167	0.6833
8	0.0698	0.9302
9	0.0007	0.9993
10	0.0197	0.9803
11	0.0642	0.9358

- c1 = 3.8688
- c2 = 9.1314



Fuzzy C Mean Pros and Cons

- **Pros**
  - Gives best result for overlapped data set and comparatively better then k-means algorithm.
  - Unlike k-means where data point must exclusively belong to one cluster center here data point is assigned membership to each cluster center as a result of which data point may belong to more then one cluster center.
- **Cons**
  - Apriori specification of the number of clusters.
  - more number of iteration required.