MMTE-007: Soft Computing and its Applications

Session 1 and 2

Implement the FCM algorithm, using C/C++ language and test it to find the final fuzzy pseudo partition and cluster centers for the two-dimensional data sets given in Table 1 and 2, assuming that convergence is achieved when the difference between the two values is less than or equal to 0.05. Plot the data to determine how many clusters you should assume. Starting with different initial membership matrices check whether the final membership functions are always same? After how many iterations did you achieve the final value in each case? What can you say about the clusters?

Table 1 (Dataset-I)

K	1	2	3	4	5	6	7	8	9	10
X_{k1}	0.958	1.043	1.907	0.780	0.579	0.003	0.001	0.014	0.007	0.004
X_{k2}	0.003	0.001	0.003	0.002	0.001	0.105	1.748	1.839	1.021	0.214

Table 2 (Dataset-2)

K	K	1	2	3	4	5	6	7	8
X_{k1}	X_{k1}	0.958	1.043	1.907	0.780	0.579	0.003	0.001	0.014
X_{k2}	X_{k2}	0.003	0.001	0.003	0.002	0.556	0.105	1.748	1.839

Session 3

Write a program to construct a function *C = knn(trainclass, traindata, data, k)* that performs k-nearest-neighbour classification. Matrix *traindata* contains training examples so that each column is a single example. Row vector *trainclass* contains the classes of the examples, so that element i of *trainclass* is the class of the example in column i of traindata. Matrix data contains simples to be classified, one in each column, and k is a parameter which tells number of nearest neighbours used in classification. Return value C is a row vector of classes and it should include one value for each column in data. Use Euclidean distance as distance measure.

Algorithm for k-nearest-neighbour classification can be described as follows:

- Find k nearest neighbours from the training set of a sample to be classified.
- Classify the sample to the class which has most training samples among the k nearest neighbours.

Session 4

Write a program to implement a function c = cmeans (data, k) that performs c-means clustering for given data. Matrix data contains training examples so that each column is a single example. Scalar k is a parameter which tells the desired number of clusters. Return value c is matrix which contains means of clusters, one in each column.

One of the simplest, most well-known, and most used clustering techniques is c-means which is also known as ISODATA and k-means. The c-means algorithms is as follows:

1. Choose the number of clusters.
2. Initialize the seed vectors for clusters.
3. Cluster the data according to the shortest distance to cluster means.
4. Calculate a mean vector for each cluster and set them as new mean vectors.
5. If there are changes in the mean vectors between iterations i and i-1, then goto 3. Means of clusters $\mu_1, \mu_2, \ldots\ldots, \mu_c$ must have some initial values. These can be obtained by randomly choosing c samples from the data, and assigning $\mu_1, \mu_2, \ldots\ldots, \mu_c$ to the values of these samples.

Session 5

Write a program to modify your cmeans function created in Session 4 so that progress of clustering process can be seen. This means that your function should plot original data and original guesses for means of clusters. After each iteration your function should show samples belonging to each cluster with different symbols and/or colors, and your function should also show trajectories of means of clusters starting from initial guesses and ending to final values.

Session 6

1) Implement the perception learning algorithm in the computer. Find the weights for an edge detection operator using this program. The input-output examples can be taken from a digitized picture of an object and another one in which only the edges of the object have been kept.
2) Implement using C, the fixed increment perceptron learning algorithm.

Session 7

Write a program to implement the back propagation algorithm. Show step by step output to input, hidden and output neurons as well as errors. How the weights W and V modified?

Session 8

Write a program to find the average correlation matrix for given input patterns M and N to design and train the Hopfield Network, the weight matrix and output of the Hopfield network.

Session 9

Write a program to find the updated/modified weights for Kohonen Networks.

Session 10

Write a program to $\max f(x) = \sqrt{x}$, subject to $1 \leq x \leq 100$ by considering the string length 8, using GA.

Session 11

Write a program to write the children solution of TSP consisting of n cities using crossover order $(\#1)$ and crossover order $(\#2)$.

Write a program language to write the children solution of a TSP consisting of n cities using cyclic crossover and position-based crossover.