**Experiment 7a**

Nidhi Parab

Subject: DWM

Class: TE4

Roll no: 35

Batch: C

**Aim:Implementation of Clustering algorithm (K-means/K-medoids) 1 D**

**1d K-means Clustering**

**Program:**

import java.util.\*;

public class D {

static int count1, count2, count3;

static int d[];

static int k[][];

static int tempk[][];

static double m[];

static double diff[];

static int n, p;

static int cal\_diff (int a)

{

for (int i = 0; i < p; ++i)

{

if (a > m[i])

diff[i] = a - m[i];

else

diff[i] = m[i] - a;

}

int val = 0;

double temp = diff[0];

for (int i = 0; i < p; ++i)

{

if (diff[i] < temp)

{

temp = diff[i];

val = i;

}

}

return val;

}

static void cal\_mean ()

{

for (int i = 0; i < p; ++i)

m[i] = 0;

int cnt = 0;

for (int i = 0; i < p; ++i)

{

cnt = 0;

for (int j = 0; j < n - 1; ++j)

{

if (k[i][j] != -1)

{

m[i] += k[i][j];

++cnt;

}

}

m[i] = m[i] / cnt;

}

}

static int check1 ()

{

for (int i = 0; i < p; ++i)

for (int j = 0; j < n; ++j)

if (tempk[i][j] != k[i][j])

{

return 0;

}

return 1;

}

public static void main (String args[])

{

Scanner scr = new Scanner (System.in);

System.out.print ("\nEnter the number of elements: ");

n = scr.nextInt ();//Accepting no. of elements

d = new int[n];//Creation of Object for an integer array

System.out.println ("Enter " + n + " elements: ");

for (int i = 0; i < n; ++i)

d[i] = scr.nextInt ();

System.out.print ("\nEnter the number of clusters: ");

p = scr.nextInt ();//Accepting the number of clusters

k = new int[p][n];//Creation of an object for integer matrix

tempk = new int[p][n];//Creation of object for integer matrix

m = new double[p];//Creation of object for double array

diff = new double[p];//Creation of object for double array

for (int i = 0; i < p; ++i)

m[i] = d[i];

int temp = 0;

int flag = 0;

do

{

for (int i = 0; i < p; ++i)

for (int j = 0; j < n; ++j)

{

k[i][j] = -1;

}

for (int i = 0; i < n; ++i)

{

temp = cal\_diff (d[i]);

if (temp == 0)

k[temp][count1++] = d[i];

else if (temp == 1)

k[temp][count2++] = d[i];

else if (temp == 2)

k[temp][count3++] = d[i];

}

cal\_mean ();//Calling cal\_mean ()

flag = check1 ();

if (flag != 1)

for (int i = 0; i < p; ++i)

for (int j = 0; j < n; ++j)

tempk[i][j] = k[i][j];

System.out.println ("\n\nAt this step");

System.out.println ("\nValue of clusters");

for (int i = 0; i < p; ++i)

{

System.out.print ("K" + (i + 1) + "{ ");

for (int j = 0; k[i][j] != -1 && j < n - 1; ++j)

System.out.print (k[i][j] + " ");

System.out.println ("}");

}

System.out.println ("\nValue of m ");

for (int i = 0; i < p; ++i)

System.out.print ("m" + (i + 1) + "=" + m[i] + " ");

count1 = 0;

count2 = 0;

count3 = 0;

}

while (flag == 0);

System.out.println ("\n\n\nThe Final Clusters By Kmeans are as follows: ");

for (int i = 0; i < p; ++i)

{

System.out.print ("K" + (i + 1) + "{ ");

for (int j = 0; k[i][j] != -1 && j < n - 1; ++j)

System.out.print (k[i][j] + " ");

System.out.println ("}");

}

}

}

**Output:**

Enter the number of elements: 9

Enter 9 elements:

1

2

6

7

8

10

15

17

20

Enter the number of clusters: 3

At this step

Value of clusters

K1{ 1 }

K2{ 2 }

K3{ 6 7 8 10 15 17 20 }

Value of m

m1=1.0 m2=2.0 m3=11.857142857142858

At this step

Value of clusters

K1{ 1 }

K2{ 2 6 }

K3{ 7 8 10 15 17 20 }

Value of m

m1=1.0 m2=4.0 m3=12.833333333333334

At this step

Value of clusters

K1{ 1 2 }

K2{ 6 7 8 }

K3{ 10 15 17 20 }

Value of m

m1=1.5 m2=7.0 m3=15.5

At this step

Value of clusters

K1{ 1 2 }

K2{ 6 7 8 10 }

K3{ 15 17 20 }

Value of m

m1=1.5 m2=7.75 m3=17.333333333333332

At this step

Value of clusters

K1{ 1 2 }

K2{ 6 7 8 10 }

K3{ 15 17 20 }

Value of m

m1=1.5 m2=7.75 m3=17.333333333333332

The Final Clusters By Kmeans are as follows:

K1{ 1 2 }

K2{ 6 7 8 10 }

K3{ 15 17 20 }