

Prajakta Deokule

3330

A1

ASSIGNMENT 3

Part 2- K Means for 2 dimensional Array

```
import java.util.*;
public class TwoDArray{

    public static void main(String args[]) {

        int dataset[][] = {
            {5,2},
            {2,3},
            {7,4},
            {1,5},
            {8,0},
            {7,1},
            {0,4},
            {6,6},
            {5,1},
            {3,6}
        };

        int i,j,k=2;
        int part1[][] = new int[10][2];
        int part2[][] = new int[10][2];

        float mean1[][] = new float[1][2];
        float mean2[][] = new float[1][2];

        float temp1[][] = new float[1][2];
        float temp2[][] = new float[1][2];

        int sum11 = 0, sum12 = 0, sum21 = 0, sum22 = 0;

        double dist1, dist2;
        int i1 = 0, i2 = 0, itr = 0;
        double a,b,c,d;

        System.out.println("Dataset: ");

        for(i=0;i<10;i++)
            System.out.println(dataset[i][0]+" "+dataset[i][1]);

        System.out.println("\nNumber of partitions: "+k);

        // Taking (2,2) and (5,7) as random means
        mean1[0][0]=2;
        mean1[0][1]=2;
```

```

    mean2[0][0]=5;
    mean2[0][1]=7;

// Loop till the new mean and previous mean are same
while(!Arrays.deepEquals(mean1, temp1) || !Arrays.deepEquals(mean2, temp2)) {

    //Emptying the partitions

    for(i=0;i<10;i++) {
        part1[i][0] = 0;
        part1[i][1] = 0;
        part2[i][0] = 0;
        part2[i][1] = 0;
    }

    i1 = 0; i2 = 0;

//Finding distance between mean and data point and store the data point in the
corresponding partition
//Euclidean distance  $d = \sqrt{(x1 - x2)^2 + (y1 - y2)^2}$  )

    for(i=0;i<10;i++) {

        a=Math.pow(dataset[i][0] - mean1[0][0],2);
        b=Math.pow(dataset[i][1] - mean1[0][1],2);

        c=Math.pow(dataset[i][0] - mean2[0][0],2);
        d=Math.pow(dataset[i][1] - mean2[0][1],2);

        dist1 = Math.sqrt(a+b);
        dist2 = Math.sqrt(c+d);

        if(dist1 < dist2) {
            part1[i1][0] = dataset[i][0];
            part1[i1][1] = dataset[i][1];
            i1++;
        }

        else {
            part2[i2][0] = dataset[i][0];
            part2[i2][1] = dataset[i][1];
            i2++;
        }
    }

    //Storing the previous mean
    temp1[0][0] = mean1[0][0];
    temp1[0][1] = mean1[0][1];

    temp2[0][0] = mean2[0][0];
    temp2[0][1] = mean2[0][1];

//Finding new mean for new partitions
    sum11 = 0; sum12 = 0; sum21 = 0; sum22 = 0;

    for(i=0;i<i1;i++) {
        sum11 += part1[i][0];
        sum12 += part1[i][1];
    }

```

```

    }

    for(i=0;i<i2;i++) {
        sum21 += part2[i][0];
        sum22 += part2[i][1];
    }

    mean1[0][0] = (float)sum11/i1;
    mean1[0][1] = (float)sum12/i1;

    mean2[0][0] = (float)sum21/i2;
    mean2[0][1] = (float)sum22/i2;
    itr++;
}

System.out.println("\nFinal Partition: ");
System.out.println("Part1:");

for(i=0;i<i1;i++)
    System.out.println(part1[i][0]+" "+part1[i][1]);

System.out.println("\nPart2:");
for(i=0;i<i2;i++)
    System.out.println(part2[i][0]+" "+part2[i][1]);

System.out.println("\nFinal Means: ");
    System.out.println("Mean1 : "+mean1[0][0]+" "+mean1[0][1]);
    System.out.println("Mean2 : "+mean2[0][0]+" "+mean2[0][1]);
    System.out.println("\nTotal Iteration: "+itr);
}
}

```

Output

Dataset:

5 2

2 3

7 4

1 5

8 0

7 1

0 4

6 6

5 1

3 6

Number of partitions: 2

Final Partition:

Part1:

5 2

2 3

1 5

8 0

7 1

0 4

5 1

Part2:

7 4

6 6

3 6

Final Means:

Mean1 : 4.0 2.2857144

Mean2 : 5.3333335 5.3333335

Total Iteration: 2