CSCI 527 Week 11 Assignment

Due Date: Friday, March 27, 2020

Name: Vijash Chandra Thunga

CWID: 50240342

1. Use the basic K-means algorithm to cluster the following 2-dimensional data points into three clusters. Use Euclidean distance as the distance function, and (2, 0), (2, 1), and (2, 2) as the initial centroids. Show detailed steps. (10%)

When the iteration stops, compute the Within cluster Sum of Squared error (WSS) and the Between cluster Sum of Squared error (BSS). (10%)

|  |  |
| --- | --- |
| X | Y |
| 0 | 0 |
| 0 | 1 |
| 1 | 0 |
| 1 | 1 |
| 2 | 0 |
| 2 | 1 |
| 3 | 0 |
| 3 | 1 |
| 2 | 2 |
| 2 | 3 |
| 3 | 2 |
| 3 | 3 |

Initial Centroids,

Given points

|  |  |  |  |
| --- | --- | --- | --- |
|  | C1(2,0) | C2(2,1) | C3(2,2) |
| P1(0,0) | **2** | √5 | √8 |
| (0,1) | √5 | **2** | √5 |
| (1,0) | **1** | √2 | √5 |
| (1,1) | √2 | **1** | √2 |
| (3,0) | **1** | √2 | √5 |
| (3,1) | √2 | **1** | √2 |
| (2,3) | 3 | 2 | **1** |
| (3,2) | √5 | √2 | **1** |
| (3,3) | √10 | √5 | **√2** |

Cluster 1= (2,0), (0,0), (1,0), (3,0)

Cluster 2= (2,1), (0,1), (1,1), (3,1)

Cluster 3= (2,2), (2,3), (3,2), (3,3)

After calculation, New centroids are,

C1= (0+1+2+3/4, 0+0+0+0/4) = (1.5,0)

C2= (0+1+2+3/4, 1+1+1+1/4) = (1.5,1)

C3= (2+2+3+3/4, 2+3+2+3/4) = (2.5,2.5)

Iteration 2:

Given points

|  |  |  |  |
| --- | --- | --- | --- |
|  | C1(1.5,0) | C2(1.5,1) | C3(2.5,2.5) |
| (0,0) | **√2.25** | √3.25 | √12.5 |
| (0,1) | √3.25 | **√2.25** | √8.5 |
| (1,0) | **√0.25** | √1.25 | √8.5 |
| (1,1) | √1.25 | **√0.25** | √4.5 |
| (2,0) | **√0.25** | √1.25 | √6.5 |
| (2,1) | √1.25 | **√0.25** | √2.5 |
| (3,0) | **√2.25** | √3.25 | √6.5 |
| (3,1) | √3.25 | **√2.25** | √2.5 |
| (2,2) | √4.25 | √1.25 | **√0.5** |
| (2,3) | √9.25 | √4.25 | **√0.5** |
| (3,2) | √6.25 | √3.25 | **√0.5** |
| (3,3) | √11.25 | √6.25 | **√0.5** |

Cluster 1= (0,0), (1,0), (2,0), (3,0)

Cluster 2= (0,1), (1,1), (2,1), (3,1)

Cluster 3= (2,3), (2,2), (3,2), (3,3)

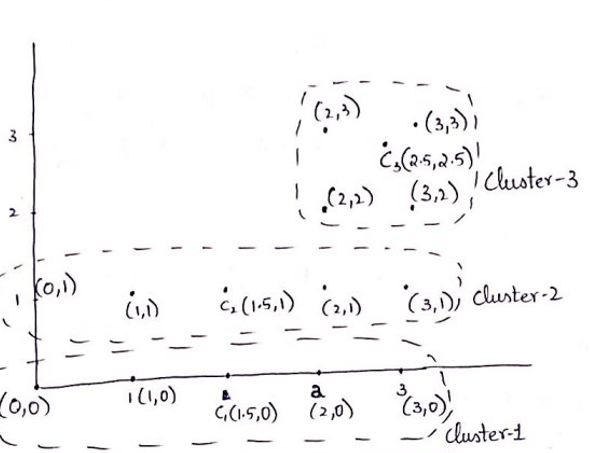
Since no points change to other cluster, Iteration stops with centroids,

C1= (1.5,0)

C2= (1.5,1)

C3= (2.5,2.5)

Final graph is



WSS for last iteration=∑ ∑(x-mi)2

i x ∊Ci

(√2.252 + √0.252 +√0.252 +√2.252) + (√2.252 +√0.252 +√0.252 +√2.252 )+( √0.52 +√0.52 +√0.52 +√0.52 )

=2.25\*4+0.25\*4+0.5\*4

=12

WSS=12

BSS for Last iteration= =∑ |Ci|(m-mi)2

i

centroids for 3 clusters= C1= (1.5,0), C2= (1.5,1), C3= (2.5,2.5)

centroid for all the points = C (1.83,1.16)

Calculate (m-mi)2 for C to C1, C to C2, C to C3

CC1= √(1.83-1.5)2 +( 1.16-0)2 = √1.45

CC2= √(1.83-1.5)2 +( 1.16-1)2 = √0.13

CC3= √0.45 +1.79= √2.24

BSS= 4(1.45) + 4(0.13) + 4(2.24) =15.28

1. Given the same data above, use (0, 0), (3, 1), and (2,3) as initial centroids, and cluster the 2-dimensional data points into three clusters by K-means. Show detailed steps. (10%) When the iteration stops, show WSS and BSS. (10%) Comparing to the result of Question 1, what have you learnt? (5%)

Given points C1(0,0) C2(3,1) C3(2,3)

(0,1) **1** 3 √8

(1,0) **1** √5 √10

(1,1) **√2** 2 √5

(2,0) 2 **√2** 3

(2,1) √5 **1**  2

(3,0) 3 **1** √10

(2,2) √8 √2 **1**

(3,2) √13 **1** √2

(3,3) √18 2 **1**

Cluster 1= (0,0), (0,1), (1,0), (1,1)

Cluster 2= (3,1), (2,0), (3,2), (2,1), (3,0)

Cluster 3= (2,2), (2,3), (3,3)

New Centroids are

C1= (0+0+1+1/4, 0+1+0+1/4) = (0.5,0.5)

C2= (3+2+3+2+3/5, 1+0+2+1+0/5) = (2.6,0.8)

C3= (2+2+3/3, 2+3+3/3) = (2.3,2.6)

Iteration 2:

Given points C1(0.5,0.5) C2(2.6,0.8) C3(2.3,2.6)

(0,0) **√0.5** √0.5 √0.5

(0,1) **√0.5** √0.5 √0.5

(1,0) **√0.5** √0.5 √0.5

(1,1) **√0.5** √0.5 √0.5

(2,0) √2.5  **1** 1

(2,1) √2.5 **√0.4** √0.4

(3,0) √6.5 **√0.80** √0.8

(3,1) √6.5 **√0.2** √0.2

(2,2) √4.5 √1.80 **√0.15**

(2,3) √8.5 √5.20 **√0.25**

(3,2) √8.5 √1.60 **√0.85**

(3,3) √12.5 √4.90 **√0.65**

After 2nd Iteration

Cluster 1= (0,0), (0,1), (1,0), (1,1)

Cluster 2= (2,0), (2,1), (3,0), (3,1)

Cluster 3= (2,2), (2,3), (3,2), (3,3)

New Centroids

C1= (0+0+1+1/4, 0+1+0+1/4) = (0.5,0.5)

C2= (2+2+3+3/4, 0+1+0+1/4) = (2.5,0.5)

C3= (2+2+3+3/4, 2+3+2+3/4) = (2.5,2.5)

Iteration 3:

Given points C1(0.5,0.5) C2(2.5,0.5) C3(2.5,2.5)

(0,0) **√0.5** √6.5 0.5

(0,1) **√0.5** √6.5 √8.5

(1,0) **√0.5** √2.5 √8.5

(1,1) **√0.5** √2.5 √4.5

(2,0) √2.5 **√0.5** √6.5

(2,1) √2.5 **√0.5** √2.5

(3,0) √6.5 **√0.5** √0.5

(3,1) √6.5 **√0.5** √2.5

(2,2) √4.5 √2.5 **√0.5**

(2,3) √8.5 √6.5 **√0.5**

(3,2) √8.5 √2.5 **√0.5**

(3,3) √12.5 √6.5 **√0.5**

After 3rd Iteration

Cluster 1= (0,0), (0,1), (1,0), (1,1)

Cluster 2= (2,0), (2,1), (3,0), (3,1)

Cluster 3= (2,2), (2,3), (3,2), (3,3)

New Centroids

C1= (0+0+1+1/4, 0+1+0+1/4) = (0.5,0.5)

C2= (2+2+3+3/4, 0+1+0+1/4) = (2.5,0.5)

C3= (2+2+3+3/4, 2+3+2+3/4) = (2.5,2.5)

Final centroids are

C1= (0.5,0.5)

C2= (2.5,0.5)

C3= (2.5,2.5)

As no point changes in cluster iteration stops

WSS for last iteration=∑ ∑(x-mi)2

i x ∊Ci

(√0.52 + √0.52 +√0.52 +√0.52) +(√0.52 + √0.52 +√0.52 +√0.52) +( √0.52 +√0.52 +√0.52 +√0.52)

=12\*0.5

=6

WSS=6

BSS for Last iteration= =∑ |Ci|(m-mi)2

i

centroids for 3 clusters= C1= (1.5,0), C2= (1.5,1), C3= (2.5,2.5)

centroid for all the points = C (1.83,1.16)

Calculate (m-mi)2 for C to C1, C to C2, C to C3

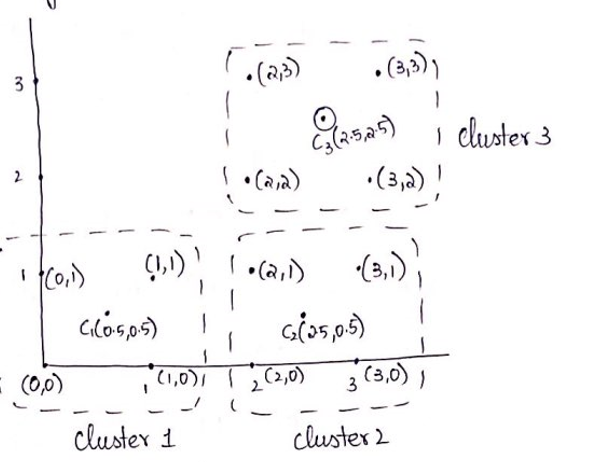
CC1=d1= √(1.83-0.5)2 +( 1.16-0.5)2 = √2.2

CC2=d2= √(1.83-2.5)2 +( 1.16-0.5)2 = √0.88

CC3=d3= √(1.83-2.5) 2 +(1.16-2.5) 2 = √2.24

BSS= 4(2.2) + 4(0.88) + 4(2.24) =21.28

Graph for last Iteration:



Comparing problem 1 and 2 :

* Clusters are dependent on initial centroids.
* If we select initial centroids correctly, we can reduce the number of iterations. As we can see in problem 1 there are 2 iterations but in problem 2 there are 3 iterations.
* Therefore, we can reduce iterations by choosing better initial centroids.

1. Use the distance matrix in the following table to perform agglomerative hierarchical clustering on the five data points (P1, P2,…, P5) using Complete Link (MAX). Show your results by drawing a dendrogram. The dendrogram should clearly show the order in which clusters are merged and the distance between two clusters merged. (20%)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | P1 | P2 | P3 | P4 | P5 |
| P1 | 0.0 | 0.2 | 0.4 | 0.7 | 0.3 |
| P2 | 0.2 | 0.0 | 0.8 | 0.6 | 1.1 |
| P3 | 0.4 | 0.8 | 0.0 | 0.5 | 1.0 |
| P4 | 0.7 | 0.6 | 0.5 | 0.0 | 0.9 |
| P5 | 0.3 | 1.1 | 1.0 | 0.9 | 0.0 |

Given points [P1….P5], Min of distance is 0.2. So, cluster of both P1&P2

We need to choose Max distance for every point, 1st cluster P1 and P2 is

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | P1P2 | P3 | P4 | P5 |
| P1P2 | 0.0 | 0.8 | 0.7 | 1.1 |
| P3 | 0.8 | 0 | 0.5 | 1.0 |
| P4 | 0.7 | 0.5 | 0 | 0.9 |
| P5 | 1.1 | 1.0 | 0.9 | 0 |

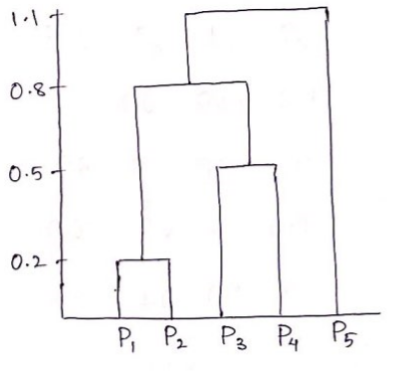
Minimum of the distance between P3 all is 0.5. Distance between P3 and P4,

|  |  |  |  |
| --- | --- | --- | --- |
|  | P1P2 | P3P4 | P5 |
| P1P2 | 0 | 0.8 | 1.1 |
| P3P4 | 0.8 | 0 | 1.0 |
| P5 | 1.1 | 1.0 | 0 |

Min of the distance all is P1P2 and P3P4 = 0.8.

|  |  |  |
| --- | --- | --- |
|  | P1P2P3P4 | P5 |
| P1P2P3P4 | 0 | 1.1 |
| P5 | 1.1 | 0 |

Dendrogram:



1. Given the same distance matrix above, perform agglomerative hierarchical clustering on the same set of data points, using Group Average (Average Link) approach to decide the distance between two clusters. Show your result by drawing a dendrogram. The dendrogram should clearly show the order in which clusters are merged and the distance between two clusters merged. (20%)

Hierarchical Clustering:

Distance matrix for given problem:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | P1 | P2 | P3 | P4 | P5 |
| P1 | 0.0 | 0.2 | 0.4 | 0.7 | 0.3 |
| P2 | 0.2 | 0.0 | 0.8 | 0.6 | 1.1 |
| P3 | 0.4 | 0.8 | 0.0 | 0.5 | 1.0 |
| P4 | 0.7 | 0.6 | 0.5 | 0.0 | 0.9 |
| P5 | 0.3 | 1.1 | 1.0 | 0.9 | 0.0 |

By selecting avg of the given problem min of distance [P1…..P5] is 0.2

i.e., P1 and P2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | P1P2 | P3 | P4 | P5 |
| P1P2 | 0 | 0.6 | 0.65 | 0.7 |
| P3 | 0.6 | 0 | 0.5 | 1.0 |
| P4 | 0.65 | 0.5 | 0 | 0.9 |
| P5 | 0.7 | 1.0 | 0.9 | 0 |

Min of Distance between all is 0.5

i.e., P3 and P4

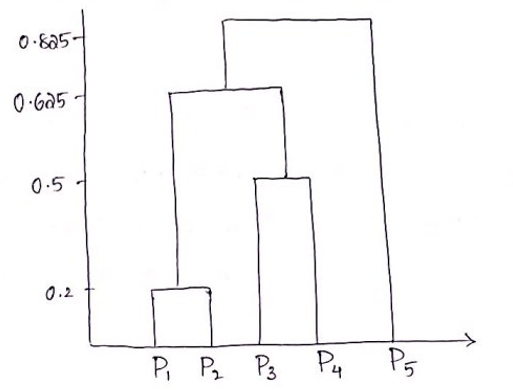
|  |  |  |  |
| --- | --- | --- | --- |
|  | P1P2 | P3P4 | P5 |
| P1P2 | 0 | 0.625 | 0.7 |
| P3P4 | 0.625 | 0 | 0.95 |
| P5 | 0.7 | 0.95 | 0 |

Min distance between all is 0.625

i.e., P1P2 & P3P4

|  |  |  |
| --- | --- | --- |
|  | P1P2P3P4 | P5 |
| P1P2P3P4 | 0 | 0.825 |
| P5 | 0.825 | 0 |

Dendrogram:



1. Briefly describe DBSCAN, including its advantages and disadvantages. (15%)

Density based spatial clustering of applications with noise (DBSCAN) is a data clustering algorithm. It is a density-based clustering non parametric algorithm. It does a great job in seeking areas in the data that have ahigh density of observations. DBSCAN can sort data into clusters of varying shapes.

Advantages:

* DBSCAN can find arbitrarily shaped clusters. It can even find a cluster completely surrounded by a different cluster.
* Can discover arbitrarily shaped clusters.
* Robust towards outlier detection(noise)

Disadvantages:

* Not partitionable for multiprocessor systems.
* Sensitive to clustering parameters.
* Fails to identify cluster if density varies and is the dataset is too sparse.