CSE-4502/5717 Big Data Analytics
Written Assiggnment 1
Data Processing;Rule Mining; Clustering

1. (total 10 marks) It is important to define or select similarity measures in data analysis. However, there is no commonly-accepted subjective similarity measure. Results can vary depending on the similarity measures used. Nonetheless, seemingly different similarity measures may be equivalent after some transformation.

Suppose we have the following two-dimensional data set:

Data Points	A_1	A_2
x_1	1.5	1.8
x_2	2.1	1.9
x_3	1.6	1.9
x_4	1.3	1.6
x_5	1.5	1.1

(a) (5 marks) Consider the data as two-dimensional data points. Given a new data point, x = (1.5, 1.3) as a query, rank the database points based on similarity with the query using Euclidean distance, Manhattan distance, and cosine similarity. Please provide a table to list all the similarity values calculated.

(b) (5 marks) Normalize the data set to make the norm of each data point equal to 1 (normalized into a vector (a,b) such that $a^2 + b^2 = 1$). Use Euclidean distance on the transformed data to rank the data points.

Answer-1:-

Using formula:-

Euclidean distance:

$$d(i,j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{in} - x_{jn})^2}.$$

Manhattan distance:

$$d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \cdots + |x_{in} - x_{jn}|.$$

And cosine similarity :

$\cos(d_1, d_2) = (d_1 \bullet d_2) / ||d_1||\ ||d_2||$

(a)

Here's the detailed calculation:

Euclidian distance formula: $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

Distance between $x = (1.5, 1.3)$ & $x_1 = (1.5, 1.8)$

$$= \sqrt{(1.5 - 1.5)^2 + (1.8 - 1.3)^2}$$
$$= \sqrt{(0)^2 + (0.5)^2}$$
$$= 0.5$$

Distance between $x = (1.5, 1.3)$ & $x_2 = (2.1, 1.9)$

$$= \sqrt{(2.1 - 1.5)^2 + (1.9 - 1.3)^2}$$

$$= \sqrt{(.6)^2 + (.6)^2}$$

$$= \sqrt{.36 + .36}$$

$$= 0.8485$$

Distance between $x = (1.5, 1.3)$ & $x_3 = (1.6, 1.9)$

$$= \sqrt{(1.6 - 1.5)^2 + (1.9 - 1.3)^2}$$

$$= \sqrt{(.1)^2 + (.6)^2}$$

= 0.6082

Distance between $x = (1.5, 1.3)$ & $x_4 = (1.3, 1.6)$

$$= \sqrt{(1.3 - 1.5)^2 + (1.6 - 1.3)^2}$$

$$= \sqrt{(-.2)^2 + (.3)^2}$$

$$= 0.3605$$

Distance between $x = (1.5, 1.3)$ & $x_5 = (1.5, 1.1)$

$$\sqrt{(1.5 - 1.5)^2 + (1.1 - 1.3)^2}$$

$$= \sqrt{(0)^2 + (-0.2)^2}$$

$$= \sqrt{.4} = 0.2$$

Manhattan Distance = $|\boldsymbol{x_2} - \boldsymbol{x_1}| + |\boldsymbol{y_2} - \boldsymbol{y_1}|$

Distance between $x = (1.5, 1.3)$ & $x_1 = (1.5, 1.8)$

$$= |1.5 - 1.5| + |1.8 - 1.3|$$

$$= 0.5$$

Distance between $x = (1.5, 1.3)$ & $x_2 = (2.1, 1.9)$

$$= |2.1 - 1.5| + |1.9 - 1.3|$$

$$= 0.6 + 0.6 = 1.2$$

Distance between $x = (1.5, 1.3)$ & $x_3 = (1.6, 1.9)$

$$= |1.6 - 1.5| + |1.9 - 1.3|$$

$$= 0.1 + 0.6 = 0.7$$

Distance between $x = (1.5, 1.3)$ & $x_4 = (1.3, 1.6)$

$$= |1.3 - 1.5| + |1.6 - 1.3|$$

$$= 0.2 + 0.3 = 0.5$$

Distance between $x = (1.5, 1.3)$ & $x_5 = (1.5, 1.1)$

$$= |1.5 - 1.5| + |1.1 - 1.3|$$

$$= 0.2$$

Cosine Similarity = $\frac{(x_1 \times y_1) + (x_2 \times y_2)}{\sqrt{x_1^2 + x_2^2}\sqrt{y_1^2 + y_2^2}}$

Cosine Similarity between $x = (1.5, 1.3)$ & $x_1 = (1.5, 1.8)$

$$= \frac{1.5 \times 1.3 + 1.5 \times 1.8}{\sqrt{1.5^2 + 1.5^2}\sqrt{1.3^2 + 1.8^2}}$$

$$= 0.9874$$

Cosine Similarity between $x = (1.5, 1.3)$ & $x_2 = (2.1, 1.9)$

$$= \frac{1.5 \times 1.3 + 2.1 \times 1.9}{\sqrt{1.5^2 + 2.1^2}\sqrt{1.3^2 + 1.9^2}}$$

= 0.9998

Cosine Similarity between $x = (1.5, 1.3)$ & $x_3 = (1.6, 1.9)$

$$= \frac{1.5 \times 1.3 + 1.6 \times 1.9}{\sqrt{1.5^2 + 1.6^2}\sqrt{1.3^2 + 1.9^2}}$$

$$= 0.9883$$

Cosine Similarity between $x = (1.5, 1.3)$ & $x_4 = (1.3, 1.6)$

$$= \frac{1.5 \times 1.3 + 1.3 \times 1.6}{\sqrt{1.5^2 + 1.3^2}\sqrt{1.3^2 + 1.6^2}}$$

$$= 0.9848$$

Cosine Similarity between $x = (1.5, 1.3)$ & $x_5 = (1.5, 1.1)$

$$= \frac{1.5 \times 1.3 + 1.5 \times 1.1}{\sqrt{1.5^2 + 1.5^2}\sqrt{1.3^2 + 1.1^2}}$$

$$= 0.9965$$

I computed the Euclidian, Manhattan distance & cosine similarity between the input data point and each of the data points in the data set. Doing so yields the following table. I have used excel to calculate the formulas and here's the screenshot of the same.

A1	A2	Euclidian	Manhattan	Cosine similarity
1.5	1.8	0.5	0.5	0.9874
2.1	1.9	0.8485	1.2	0.9998
1.6	1.9	0.6082	0.7	0.9883
1.3	1.6	0.3605	0.5	0.9848
1.5	1.1	0.2	0.2	0.9965

These values produce the following rankings of the data points based on similarity:

Euclidean distance: x5,x4,x1,x3,x2

Manhattan distance: x5,x4,x1,x3,x2

Cosine similarity: x4,x1,x3,x5,x2

(b)

The normalized query is (0.7556891, 0.654930538). The normalized data set is given by the following table. I used excel for calculation:

For x = (1.5, 1.3) the normalized value is : $\frac{1.5}{\sqrt{1.5^2+1\cdot 3^2}}, \frac{1.3}{\sqrt{1.5^2+1\cdot 3^2}}$ = (0.7557,0.6549)

Applying same formula as above:

For x1 = (1.5, 1.8) the normalized value is : = (0.6402,0.7682)
For x2 = (1.5, 1.3) the normalized value is : = (0.7415,0.6709)
For x3 = (1.5, 1.3) the normalized value is : = (0.6441,0.7649)
For x4 = (1.5, 1.3) the normalized value is : = (0.6305,0.7761)
For x5 = (1.5, 1.3) the normalized value is : = (0.8064,0.5914)

Now calculating distances:

Distance between $x = (.7557, .6549)$ & $x_1 = (.6402, .7682)$

$$= \sqrt{(.6402 - .7557)^2 + (.7682 - .6549)^2}$$
$$= 0.1616$$

The Euclidean distance is 0.1616

Distance between $x = (.7557, .6549)$ & $x_2 = (.7415, .6709)$

$$= \sqrt{(.7415 - .7557)^2 + (.6709 - .6549)^2}$$

= 0.0214

Distance between $x = (.7557, .6549)$ & $x_3 = (.6441, .7649)$

$$= \sqrt{(.6441 - .7557)^2 + (.7649 - .6549)^2}$$

$$= 0.1568$$

Distance between $x = (.7557, .6549)$ & $x_4 = (.6305, .7761)$

$$= \sqrt{(.6305 - .7557)^2 + (.7761 - .6549)^2}$$

$$= 0.1741$$

Distance between $x = (.7557, .6549)$ & $x_5 = (.8064, .5914)$

$$\sqrt{(.8064 - .7557)^2 + (.5914 - .6549)^2}$$

$$= 0.0812$$

Using Similarity ranking on Euclidian distance:

Similarity Ranking	Euclidean Distance
1st	$x_2 = .0214$
2nd	$x_5 = .0812$
3rd	$x_3 = .1568$
4th	$x_1 = .1616$
5th	$x_4 = .1741$

Above table results in the final ranking of the transformed data points: x2,x5,x3,x1,x4.

2. (total 15 marks) Consider the following set of frequent 2-itemsets:
{p, q}, {p, r}, {p, s}, {p, t}, {q, r}, {q, t}, {r, s}, {s, t}.
(a) (5 marks) List all the candidate 3-itemsets produced during the candidate generation step of the Apriori algorithm.

(b) (5 marks) List all the candidate 3-itemsets that survive the pruning step of the Apriori algorithm.

(c) (5 marks) Based on the list of candidate 3-itemsets given above, is it possible to generate at least one frequent 4-itemset? State your reason clearly.

Answer-2:-

Frequent 2-itemsets
{p,q}
{p,r}
{p,s}
{p,t}
{q,r}
{q,t}
{r,s}
{s,t}

(a) All the candidate 3-itemsets produced during the candidate generation step of Apriori algorithm:

{p,q,r}, {p,q,s},{p,q,t}, {p,r,s}, {p,r,t}, {p,s,t}, {q,r,s}, {q,r,t}, {q,s,t}, {r,s,t}

(b) All candidate 3 -itemsets that survive pruning step of Apriori: (As {q,s} and {r,t} combinations aren't included in itemset so, any superset of them will be pruned as well.)

{p,q,r},{p,q,t},{p,s,t},{p,r,s}

(c) Based on above candidate 3- item sets , it isn't possible to generate at least 1-frequent itemset.

Based on above candidate 3 item sets, all 4-itemsets will be {p,q,r,t},{p,q,s,t},{p,q,r,s},{p,q,s,t}. But none of them will be formed because {q,s} and {r,t} are part of all the itemsets.

3. (total 20 marks) Here we need to solve two problems related to association rule mining, one related to Apriori and one related to FPTree.

(1) (10 marks) A database has 7 transactions (TID: Transaction Index). Let the minimum support threshold *min_sup* is 0.5.

TID	Item Bought
T1	a, c, d, f, g
T2	a, b, d, e, g
T3	a, d, f, g
T4	b, d, f
T5	e, f, g
T6	a, b, c, d, g
T7	a, b, e, g

Use the Apriori algorithm to generate the frequent item sets. Please explain the process of the generation in details, including all the candidate item sets and frequent item sets. Please use the tables/diagrams shown in Association Rule Mining Part I (Slide 25) for the result demonstration.

Answer:

The Minimum Support Count would be count of transactions, so it would be 50% of the total number of transactions. If the number of transactions is 7, the minimum support count would be 7*50/100 = 3.5.

Minimum support: 3.5

TID	Items Bought
T1	a,c,d,f,g
T2	a,b,d,e,g
T3	a,d,f,g
T4	b,d,f
T5	e,f,g
T6	a,b,c,d,g
T7	a,b,e,g

1st scan

C1

Items	Support
{a}	5
{b}	4
{c}	2
{d}	5
{e}	2
{f}	4
{g}	6

L1

Items	Support
{a}	5
{b}	4
{d}	5
{f}	4
{g}	6

2nd scan

C2

Items	Support
{a,b}	3
{a,d}	4
{a,f}	2
{a,g}	5
{b,d}	2
{b,f}	0
{b,g}	3
{d,f}	3
{f,g}	3

L2

Items	Support
{a,d}	4
{a,g}	5

C3

Items	Support
{a,d,g}	4

3rd scan

L3

TID	Items
{a,d,g}	4

(2) (10 marks) We are given the transaction database as:

TID	Item Bought
T1	A, C, D
T2	A, B, C
T3	A, B, E
T4	B, E

Please build the FPTree for the transaction database with the minimum support count 2. Please provide clear and readable figure or screenshot of the constructed FPTree (refer to Slide 9, FPTree, "Association Rule Mining Part II"). We assume alphabetical order for items with the same frequency.

Answer:-

First deducing the ordered frequent items. For items with the same frequency, the order is given by alphabetical order.

Minimum support =2

Item	Frequency
A	3
B	3
C	2
D	1
E	2

Keeping items having support

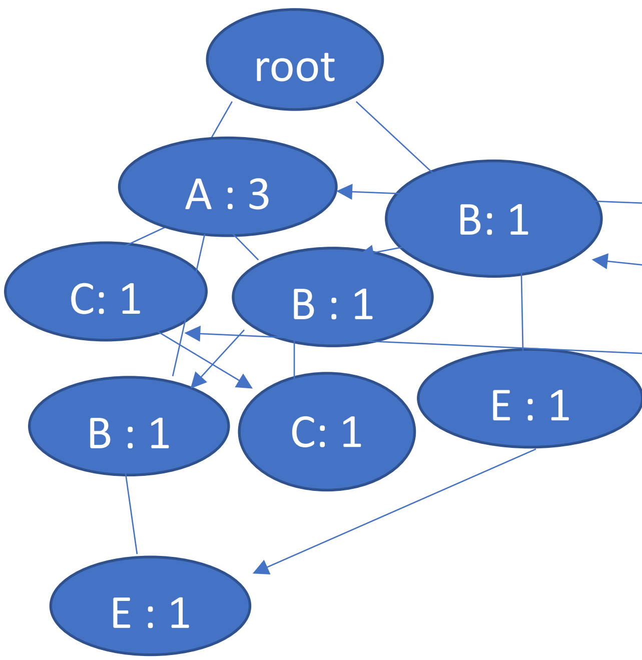
Item	Frequency
A	3
B	3
C	2
E	2

TID	Item Bought
T1	A, C, D
T2	A, B, C
T3	A, B, E
T4	B, E

Ordered frequent items

Item
A,C
A,B,C
A,B,E
B,E

Constructing FP-Tree from above data-----→



Item	Head of node-link
A	
B	
C	
E	

Conditional FP-Tree on "E" :--- (Minimum support =2)

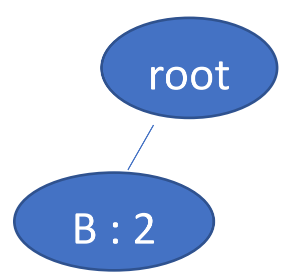
{A:1,C:1,B:1,E:1}
{B:1,E:1}

Item	Frequency
A	1
C	1
B	2
E	2

→

Keeping items greater than or equal to minimum support = 2

Item	Frequency
B	2
E	2



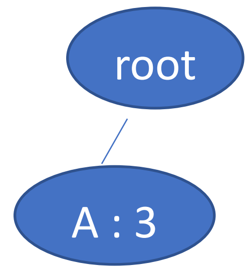
Conditional FP-Tree on “C” :--- (Minimum support =2)

{A:1,C:1}
{A:2,B:1,C:1}

Item	Frequency
A	3
B	1
C	2

Keeping items greater than or equal to minimum support = 2

Item	Frequency
A	3
C	2



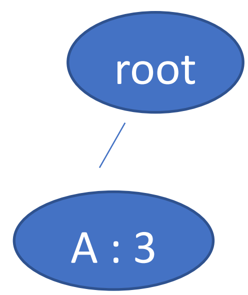
Conditional FP-Tree on “B” :--- (Minimum support =2)

{A:1,C:1,B:1}
{A:2,B:1}
{B:1}

Item	Frequency
A	3
B	3
C	1

Keeping items greater than or equal to minimum support = 2

Item	Frequency
A	3
B	3



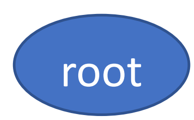
Conditional FP-Tree on “A” :--- (Minimum support =2)

{A:3}

Item	Frequency
A	3

Keeping items greater than or equal to minimum support = 2

Item	Frequency
A	3



The generated frequent patterns are:

- {E: 2}, {B:2, E:2}
- {C: 2}, {A:3, C:2}
- {B: 3}, {A:3, B:3}
- {A: 3}

4. (total 20 marks) Consider the closing prices for five stocks (A, B, C, D, and E) listed in the following table. Suppose you are interested in applying association rule mining to the data.

Figure 1 Example of Stock Market Data

Day	A	B	C	D	E
1	10.50	11.00	20.00	80.00	95.00
2	12.30	10.40	26.40	76.50	90.20
3	12.00	10.80	26.50	75.50	91.00
4	11.20	10.00	25.50	72.00	87.10
5	11.30	10.20	25.30	73.20	88.90
6	12.50	10.70	27.50	70.00	88.50
7	13.00	10.80	28.80	72.00	90.20
8	13.80	11.00	29.80	71.80	91.00
9	12.95	10.80	27.90	71.00	91.20
10	12.05	10.10	26.10	72.60	92.80
11	11.40	10.05	24.95	70.40	90.10

We first convert the stock market prices into transaction data. For each stock X on a trading day, compute the change in its closing price,

$$\Delta_X(t) = \frac{p_t(X) - p_{t-1}(X)}{p_{t-1}(X)}$$

which is the percentage of increase/decrease compared with the previous stock price. $p_t(X)$ is the price of stock X on day t. Next, create an "item" X-UP for a trading day if the increase is at least 5% ($\Delta_X(t)$ is greater than 0.05; if the closing price is up by at least 5%), or X-DOWN if decrease is at least 5% ($\Delta_X(t)$ is lower than -0.05; if the closing price is down by at least 5%). Assume each transaction corresponds to a trading day (starting from Day 2). Note that there are 10 possible items: A-UP, A-DOWN, B-UP, B-DOWN, …, E-UP, E-DOWN. Based on the original transactions, we can generate 10 transactions from above table as:

Transaction1: {A-UP, B-DOWN, C-UP, E-DOWN};
Transaction2: {};
Transaction3: {A-DOWN, B-DOWN};
Transaction4: {};
Transaction5: {A-UP, C-UP};
Transaction6: {};
Transaction7: {A-UP};
Transaction8: {A-DOWN, C-DOWN};
Transaction9: {A-DOWN, B-DOWN, C-DOWN};
Transaction10: {A-DOWN};

(a) (10 marks) Assuming the minimum support threshold is 20%, i.e., an itemset has to appear at least twice in the transaction data to be considered *frequent*, list all the frequent 1-itemsets, 2-itemsets, and so on (including their *support values*), that can be extracted from the data.

Answer:

The Minimum Support Count would be count of transactions, so it would be 20% of the total number of transactions. If the number of transactions is 10, the minimum support count would be 10*20/100 = 2.

TID	item sets
1	{a-up,b-down,c-up,e-down}
2	{}
3	{a-down,b-down}
4	{}
5	{a-up,c-up}
6	{}
7	{a-up}
8	{a-down,c-down}
9	{a-down,b-down,c-down}
10	{a-down}

Finding support counts of items

1-itemsets

Itemset	Support
{a-up}	3
{b-down}	3
{c-up}	2
{e-down}	1
{a-down}	4
{c-down}	2

As minimum support is 2, pruning items in 1-item

Itemset	Support
{a-up}	3
{b-down}	3
{c-up}	2
{e-down}	1
{a-down}	4
{c-down}	2

Frequent 1-itemsets

Itemset	Support
{a-up}	3
{b-down}	3
{c-up}	2
{a-down}	4
{c-down}	2

Creating 2-itemsets and then pruning all item sets whose support count is less than 2:-

Itemset	Support
{a-up,b-down}	1
{a-up, c-up}	2
{a-up, a-down}	0
{a-up, c-down}	0
{b-down, c-up}	1
{b-down, a-down}	2
{b-down,c-down}	1
{c-up, a-down}	0
{c-up, c-down}	0
{a-down,c-down}	2

Frequent 2-itemsets

Itemset	Support
{a-up, c-up}	2
{b-down, a-down}	2
{a-down,c-down}	2

So finally, we get 2-itemsets {a-up,c-up} , {b-down, a-down}, {a-down,c-down} whose support count is greater than or equal to 2. No further 3 – item sets are to be found out.

(b) (10 marks) Based on the frequent itemsets found in part (a), generate all the association rules with minsup = 20% and minconf = 60%. In your answer, for every rule generated you should list the support and confidence calculated. Please ignore the rules in which their left or right hand side correspond to an empty set.

Answer:

The Minimum Support Count would be count of transactions, so it would be 20% of the total number of transactions. If the number of transactions is 10, the minimum support count would be 10*20/100 = 2.

1-itemsets

TID	item sets
1	{a-up,b-down,c-up,e-down}
2	{}
3	{a-down,b-down}
4	{}
5	{a-up,c-up}
6	{}
7	{a-up}
8	{a-down,c-down}
9	{a-down,b-down,c-down}
10	{a-down}

Finding support counts of items

Itemset	Support
{a-up}	3
{b-down}	3
{c-up}	2
{e-down}	1
{a-down}	4
{c-down}	2

As minimum support is 2, pruning items in 1-item

Itemset	Support
{a-up}	3
{b-down}	3
{c-up}	2
{e-down}	1
{a-down}	4
{c-down}	2

Frequent 1-itemsets

Itemset	Support
{a-up}	3
{b-down}	3
{c-up}	2
{a-down}	4
{c-down}	2

Creating 2-itemsets and then pruning all item sets whose support count is less than 2:-

Itemset	Support
{a-up,b-down}	1
{a-up, c-up}	2
{a-up, a-down}	0
{a-up, c-down}	0
{b-down, c-up}	1
{b-down, a-down}	2
{b-down,c-down}	1
{c-up, a-down}	0
{c-up, c-down}	0
{a-down,c-down}	2

Frequent 2-itemsets

Itemset	Support
{a-up, c-up}	2
{b-down, a-down}	2
{a-down,c-down}	2

So finally, we get 2-itemsets {a-up,c-up} , {b-down, a-down}, {a-down,c-down} whose support count is greater than or equal to 2. No further 3 – item sets are to be found out.

Association rules to be formed:-

{a-up} -> {c-up} (S= 2/10 = 0.2 , C = 2/3 = 0.67)
{c-up} -> {a-up} (S= 2/10 = 0.2 , C = 2/2 = 1)
{b-down} -> {a-down} (S= 2/10 = 0.2 , C = 2/3 = 0.67)
{a-down} -> {b-down} (S= 2/10 = 0.2 , C = 2/4 = 0.5)
{a-down} -> {c-down} (S= 2/10 = 0.2 , C = 2/4 = 0.5)
{c-down} -> {a-down} (S= 2/10 = 0.2 , C = 2/2 = 1)

{a-up} -> {c-up} , {c-up} -> {a-up} , {b-down} -> {a-down} and {c-down} -> {a-down}.

5. (15 marks) Consider the following eight two-dimensional data points:

x_1: (23, 12), x_2: (6, 6), x_3: (15, 0), x_4: (15, 28), x_5:(20, 9), x_6: (8, 9), x_7: (20, 11), x_8: (8, 13),

Consider k-means algorithm to answer the following questions. You are required to show the information about each final cluster (including the mean of the cluster and all data points in this cluster). You can consider writing a program for this part but you are not required to submit the program.

(a) (5 marks) If k = 2 and the initial means are (20, 9) and (8, 9), what is the output of the algorithm? In the output, you are required to show the information about each final cluster (including the mean of the cluster and all data points in this cluster).

(b) (5 marks) If k = 2 and the initial means are (15, 0) and (15, 29), what is the output of the algorithm? In the output, you are required to show the information about each final cluster (including the mean of the cluster and all data points in this cluster).

(c) (5 marks) What are the advantages and the disadvantages of the k-means algorithm? For each disadvantage, please also give a suggestion to enhance the k-means algorithm.

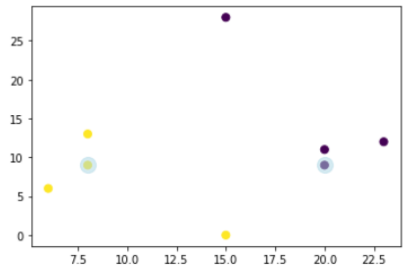
Answer:

(a) If K=2 and the initial means are (20,9) and (8,9), then :

Distances between given initial mean and all other data points is found as –

	Object	X_value	Y_value	C1_Distance	C2_Distance
0	Object 1	23	12	4.242641	15.297059
1	Object 2	6	6	14.317821	3.605551
2	Object 3	15	0	10.295630	11.401754
3	Object 4	15	28	19.646883	20.248457
4	Object 5	20	9	0.000000	12.000000
5	Object 6	8	9	12.000000	0.000000
6	Object 7	20	11	2.000000	12.165525
7	Object 8	8	13	12.649111	4.000000

Two clusters are found out to be : (yellow and blue dots)



Cluster-1 and Cluster-2 mean are as follows -

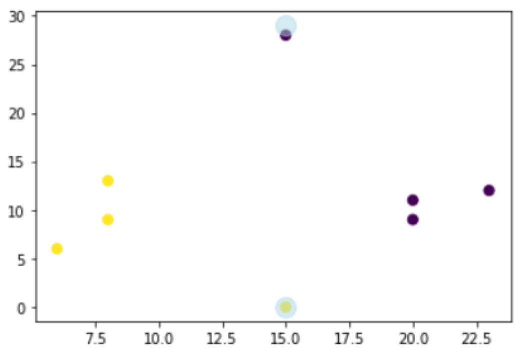
Data Points - X & Y		ED from centroid c1 = (20,9)	ED from centroid c2 = (8,9)	Cluster	Mean of c1		Mean of c2	
23	12	4.24264069	15.2970585	c1	x	y	x	y
6	6	14.3178211	3.60555128	c2	18.6	12	7.333333	9.333333
15	0	10.2956301	11.4017543	c1				
15	28	19.6468827	20.2484567	c1				
20	9	0	12	c1				
8	9	12	0	c2				
20	11	2	12.1655251	c1				
8	13	12.6491106	4	c2				

(b) If K=2 and the initial means are (15,0) and (15,29), then :

Distances between given initial mean and all other data points is found as –

	Object	X_value	Y_value	C1_Distance	C2_Distance
0	Object 1	23	12	14.422205	18.788294
1	Object 2	6	6	10.816654	24.698178
2	Object 3	15	0	0.000000	29.000000
3	Object 4	15	28	28.000000	1.000000
4	Object 5	20	9	10.295630	20.615528
5	Object 6	8	9	11.401754	21.189620
6	Object 7	20	11	12.083046	18.681542
7	Object 8	8	13	14.764823	17.464249

Two clusters are found out to be : (yellow and blue dots)



Cluster-1 and Cluster-2 mean are as follows -

Data Points		ED from centroid c1 = (15,0)	ED from centroid c2 = (15,29)	Cluster	Mean of c1		Mean of c2	
					x	y	x	y
23	12	14.4222051	18.7882942	c1	14.28571	15	8.571429	28
6	6	10.8166538	24.6981781	c1				
15	0	0	29	c1				
15	28	28	1	c2				
20	9	10.2956301	20.6155281	c1				
8	9	11.4017543	21.1896201	c1				
20	11	12.083046	18.6815417	c1				
8	13	14.7648231	17.4642492	c1				

(c) Advantages of K- means algorithm :

- Comparatively simple to implement
- Can be implemented effectively on large datasets
- Generalizes clusters of different shapes and sizes
- K-means converges for common similarity measures.
- Most of the convergences happens in the first few iterations.
- Position of centroids can be assigned easily

Disadvantages of K-Means algorithm :

- Always must choose K value manually. (For finding optimal value of K, we may use silhouette score approach. we can start by randomly choosing k value. We may generate many clusters and then perform a hierarchical clustering. K-means++ is a robust way of selecting the K initial centroids.)
- Clustering data of varying sizes and density.(To overcome this problem we may generate many clusters. Then by finding parts of multiple clusters we aggregate them to form defined clusters).
- Centroids can drag by outliers. Even outliers might get their own cluster instead of being ignored. We may remove the outliers before clustering.

6. (total 20 marks) Consider the following set of one-dimensional data points: {0.1, 0.2, 0.4, 0.5, 0.6, 0.8, 0.9}.

Index of Iteration	**Cluster assignment of data points** (put the label of cluster, either A, B or C for each data point; iteration 0 means initialization and no label is assigned)							**Centroid Locations** (calculate the updated coordinate for the centroid)		
	0.1	0.2	0.4	0.5	0.6	0.8	0.9	A	B	C
0	-	-	-	-	-	-	-	0.00	0.25	0.60
1										
2										
3										

(a) (15 marks) Suppose we apply kmeans clustering to obtain three clusters, A, B, and C. If the initial centroids are located at {0, 0.25, 0.6}, respectively, show the cluster assignments and locations of the centroids after the first **three** iterations (you can use a table as above and fill in the missing values).

(b) (5 marks) Compute the SSE (sum of squared errors) of the k-means solution (after **3** iterations; SSE is calculated after the centroid locations are updated). You can show the calculation process.

Answer:

(a)

For each iteration the Euclidian distance between individual data points and centroids are found out. Then mean of centroid is calculated . Based on smallest distance the cluster locations are assigned as shown in table. I have used Excel to apply formulas for calculating distances and the mean or centroid locations in each iteration.

Euclidian distance between data points and centroid is calculated by formula:-

d(p, c) = $\sqrt{\sum_{i=1}^{n}(c_i - P_i)^2}$ Where p designates data points and c designates centroid.

For each cluster new mean is calculated based on data points in the cluster.

Here is the step by step calculation:-

1st Iteration:							
Data Points - X & Y	ED from centroid c1/A = (0)	ED from centroid c2/B = (0.25)	ED from centroid c3/C = (0.6)	Cluster	Mean of A	Mean of B	Mean of C
0.1	0.1	0.15	0.5	A	0.1	0.3	0.7
0.2	0.2	0.05	0.4	B			
0.4	0.4	0.15	0.2	B			
0.5	0.5	0.25	0.1	C			
0.6	0.6	0.35	0	C			
0.8	0.8	0.55	0.2	C			
0.9	0.9	0.65	0.3	C			

Centroids
0
0.25
0.6

In first iteration, for each data point distance between datapoint and cluster centroid is determined and shown in columns 2,3,4. Here I used Euclidian distance formula. Then in column 5 datapoint is assigned to closest centroid based on smallest distance value to each centroid. And cluster names are shown in column 5. Then for each cluster the new mean is calculated based on the datapoints to the cluster.

For cluster A mean is equal to (0.1) as it is the only point in cluster.
For cluster B mean is equal to ((0.2+0.4)/2 = 0.3)).

For cluster C mean is equal to ((0.5+0.6+0.8+0.9)/4 = 0.7)).

2nd Iteration:							
Data Points - X & Y	ED from centroid c1/A = (0.1)	ED from centroid c2/B = (0.3)	ED from centroid c3/C = (0.7)	Cluster	Mean of A	Mean of B	Mean of C
0.1	0.05	0.35	0.66	A	0.15	0.45	0.766667
0.2	0.05	0.25	0.56	A			
0.4	0.25	0.05	0.36	B			
0.5	0.35	0.05	0.26	B			
0.6	0.45	0.15	0.16	C			
0.8	0.65	0.35	0.04	C			
0.9	0.75	0.45	0.14	C			

Centroids	
0.1	
0.3	
0.7	

The previous means are now considered as centroids in second iteration.
In second iteration, for each data point distance between datapoint and cluster centroid is determined and shown in columns 2,3,4. Here I used Euclidian distance formula. Then in column 5 datapoint is assigned to closest centroid based on smallest distance value to each centroid. And cluster names are shown in column 5. Then for each cluster the new mean is calculated based on the datapoints to the cluster.

For cluster A mean is equal to ((0.1+0.2)/2 = 0.15))
For cluster B mean is equal to ((0.4+0.5)/2 = 0.45))
For cluster C mean is equal to ((0.6+0.8+0.9)/3 = 0.76))

3rd Iteration:							
Data Points - X & Y	ED from centroid c1/A = (0.15)	ED from centroid c2/B = (0.45)	ED from centroid c3/C = (0.76)	Cluster	Mean of A	Mean of B	Mean of C
0.1	0.05	0.35	0.66	A	0.15	0.5	0.85
0.2	0.05	0.25	0.56	A			
0.4	0.25	0.05	0.36	B			
0.5	0.35	0.05	0.26	B			
0.6	0.45	0.15	0.16	B			
0.8	0.65	0.35	0.04	C			
0.9	0.75	0.45	0.14	C			

Centroids
0.15
0.45
0.76

The previous means are now considered as centroids in second iteration.

In third iteration, for each data point distance between datapoint and cluster centroid is determined and shown in columns 2,3,4. Here I used Euclidian distance formula. Then in column 5 datapoint is assigned to closest centroid based on smallest distance value to each centroid. And cluster names are shown in column 5. Then for each cluster the new mean is calculated based on the datapoints to the cluster.

For cluster A mean is equal to ((0.1+0.2)/2 = 0.15))
For cluster B mean is equal to ((0.4+0.5+0.6)/3 = 0.5))
For cluster C mean is equal to ((0.8+0.9)/2 = 0.85))

Final table with updated cluster assignments and locations of the centroids are shown below:

Index of Iteration	Cluster assignment of data points							Centroid Locations		
	0.1	0.2	0.4	0.5	0.6	0.8	0.9	A	B	C
0	_	_	_	_	_	_	_	0	0.25	0.6
1	A	B	B	C	C	C	C	0.1	0.3	0.7
2	A	A	B	B	C	C	C	0.15	0.45	0.76
3	A	A	B	B	B	C	C	0.15	0.5	0.85

(b)

For computation of SSE, we will first take the difference between each data point with it’s centroid and then add up all the squares of the differences calculated.

1st Iteration:										
Data Points - X & Y	Distance from centroid c1/A = (0)	Distance from centroid c2/B = (0.25)	Distance from centroid c3/C = (0.6)	Cluster	Mean of A	Mean of B	Mean of C	SSE	SSE_Final
0.1	0.05	0.35	0.66	A	0.1	0.3	0.7	0.0025	0.1819
0.2	0.05	0.25	0.56	B				0.0625	
0.4	0.25	0.05	0.36	B				0.0025	
0.5	0.35	0.05	0.26	C				0.0676	
0.6	0.45	0.15	0.16	C				0.0256	
0.8	0.65	0.35	0.04	C				0.0016	
0.9	0.75	0.45	0.14	C				0.0196	

Centroids
0
0.25
0.6

In first iteration , the column(SSE) is calculated by considering the data points involved in cluster formation. Square of distance between each data point and it's centroid is found and stored in respective SSE column of a certain datapoint.

For cluster A , only data point involved is o.1 and we consider distance between it and it's centroid A that is 0.05 and take square of it. Likewise, all such SSE values are determined for all rest of the data points. Then SSE_Final column shows the SSE value by adding up all the values calculated from previous step.

2nd Iteration:									
Data Points - X & Y	ED from centroid c1/A = (0.1)	ED from centroid c2/B = (0.3)	ED from centroid c3/C = (0.7)	Cluster	Mean of A	Mean of B	Mean of C	SSE	SSE_Final
0.1	0.05	0.35	0.66	A	0.15	0.45	0.766667	0.0025	0.0568
0.2	0.05	0.25	0.56	A				0.0025	
0.4	0.25	0.05	0.36	B				0.0025	
0.5	0.35	0.05	0.26	B				0.0025	
0.6	0.45	0.15	0.16	C				0.0256	
0.8	0.65	0.35	0.04	C				0.0016	
0.9	0.75	0.45	0.14	C				0.0196	

Centroids
0.1
0.3
0.7

In second iteration , the column(SSE) is calculated by considering the data points involved in cluster formation.

Square of distance between each data point and its centroid is found and stored in respective SSE column of a certain datapoint.

After finding the values in SSE column, all values are summed up to find final SSE value as shown in SSE_Final column.

3rd Iteration:

Data Points - X & Y	ED from centroid c1/A = (0.15)	ED from centroid c2/B = (0.45)	ED from centroid c3/C = (0.76)	Cluster	Mean of A	Mean of B	Mean of C	SSE	SSE_Final
0.1	0.05	0.35	0.66	A	0.15	0.5	0.85	0.0025	0.0537
0.2	0.05	0.25	0.56	A				0.0025	
0.4	0.25	0.05	0.36	B				0.0025	
0.5	0.35	0.05	0.26	B				0.0025	
0.6	0.45	0.15	0.16	B				0.0225	
0.8	0.65	0.35	0.04	C				0.0016	
0.9	0.75	0.45	0.14	C				0.0196	

Centroids	
0.15	
0.45	
0.76	

In Third iteration , the column(SSE) is calculated by considering the data points involved in cluster formation. Square of distance between each data point and its centroid is found and stored in respective SSE column of a certain datapoint.

After finding the values in SSE column, all values are summed up to find final SSE value as shown in SSE_Final column.