9318 ass1

Q1

1) List the tuples in the complete data cube of R in a tabular form with 4 attributes, i.e., Location, Time, Item, SUM(Quantity)?

Answer:

cuboid	Location	Time	Item	SUM(Quantity)
LTI	Sydney	2005	PS2	1400
LTI	Sydney	2006	PS2	1500
LTI	Sydney	2006	Wii	500
LTI	Melbourne	2005	XBox 360	1700
LT	Sydney	2005	ALL	1400
LT	Sydney	2006	ALL	2000
LT	Melbourne	2005	ALL	1700
LI	Sydney	ALL	PS2	2900
LI	Sydney	ALL	Wii	500
LI	Melbourne	ALL	XBox 360	1700
TI	ALL	2005	PS2	1400
TI	ALL	2006	PS2	1500
TI	ALL	2006	Wii	500
TI	ALL	2005	XBox 360	1700
L	Sydney	ALL	ALL	3400
L	Melbourne	ALL	ALL	1700
T	ALL	2005	ALL	3100
T	ALL	2006	ALL	2000
I	ALL	ALL	PS2	2900
I	ALL	ALL	Wii	500
I	ALL	ALL	XBox 360	1700
	ALL	ALL	ALL	5100

2) Write down an equivalent SQL statement that computes the same result (i.e., the cube). You can only use standard SQL constructs, i.e., no CUBE BY clause.

Answer:

SELECT L, T, I, SUM(Quantity)

```
FROM Sales
GROUP BY L, T, I
UNION ALL
SELECT L, T, ALL, SUM(Quantity)
FROM Sales
GROUP BY L, T
UNION ALL
SELECT L, ALL, I, SUM(Quantity)
FROM Sales
GROUP BY L, I
UNION ALL
SELECT ALL, T, I, SUM(Quantity)
FROM Sales
GROUP BY T, I
UNION ALL
SELECT L, ALL, ALL, SUM(Quantity)
FROM Sales
GROUP BY L
UNION ALL
SELECT ALL, T, ALL, SUM(Quantity)
FROM Sales
GROUP BY T
UNION ALL
SELECT ALL, ALL, I, SUM(Quantity)
FROM Sales
GROUP BY I
UNION ALL
SELECT ALL, ALL, ALL, SUM(Quantity)
FROM Sales
```

3) Consider the following ice-berg cube query

Answer:

cuboid	Location	Time	Item	Quantity
LT	Sydney	2006	ALL	2000
LI	Sydney	ALL	PS2	2900
L	Sydney	ALL	ALL	3400
T	ALL	2005	ALL	3100
T	ALL	2006	ALL	2000
I	ALL	ALL	PS2	2900
	ALL	ALL	ALL	5100

4)Assume that we adopt a MOLAP architecture to store the full data cube of R, with the following mapping functions

Answer:

We can get the map 3 * 4 * L + 4 * T + I = 12L + 4T + I = f(Location,Time,Item)

Location	Time	Item	SUM(Quantity)	f(L,T,I)
1	1	1	1400	17
1	2	1	1500	21
1	2	3	500	23
2	1	2	1700	30
1	1	0	1400	16
1	2	0	2000	20
2	1	0	1700	28
1	0	1	2900	13
1	0	3	500	15
2	0	2	1700	26
0	1	1	1400	5
0	2	1	1500	9
0	2	3	500	11
0	1	2	1700	6
1	0	0	3400	12
2	0	0	1700	24
0	1	0	3100	4
0	2	0	2000	8
0	0	1	2900	1
0	0	3	500	3
0	0	2	1700	2
0	0	0	5100	0

So we have the final result.

offset	quantity

17	1400
21	1500
23	500
30	1700
16	1400
20	2000
28	1700
13	2900
15	500
26	1700
5	1400
9	1500
11	500
6	1700
12	3400
24	1700
4	3100
8	2000
1	2900
3	500
2	1700
0	5100

Q2

Consider the given similarity matrix. You are asked to perform group average hierarchical clustering on this dataset. You need to show the steps and final result of the clustering algorithm. You will show the final results by drawing a dendrogram. The dendrogram should clearly show the order in which the points are merged.

Answer:

Using the formula to calculate the similarity of two clusters (group average), which is the average of pair-wise similarity between points in the two clusters.

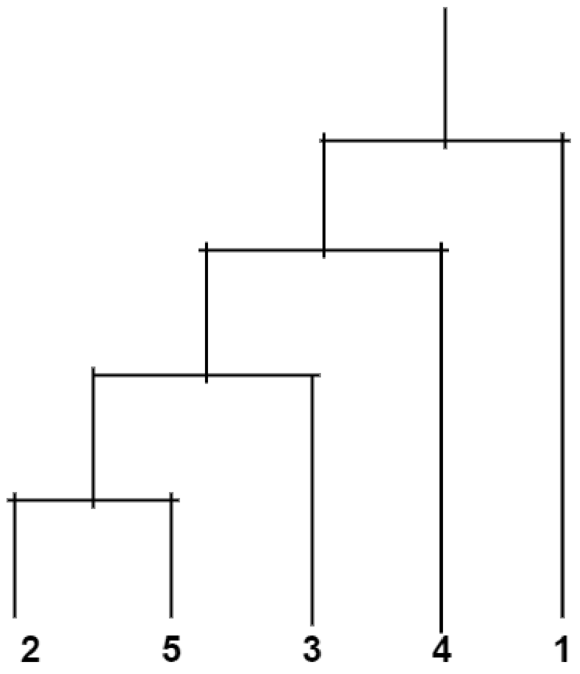
So first we get the larger one, which is p2 and p5.
Sim(1,25) = 2 * (0.10 + 0.35 + 0.98) / 6 = 0.477
Sim(3,25) = 2 * (0.64 + 0.85 + 0.98) / 6 = 0.823
Sim(4,25) = 2 * (0.76 + 0.47 + 0.98) / 6 = 0.737

	P1	P3	P4	P25
P1	1.00	0.41	0.55	0.477
P3		1.00	0.44	0.823
P4			1.00	0.737
P25				1.00

Sim(1,235) = 2 * (0.10 + 0.41 + 0.35 + 0.64 + 0.98 + 0.85) /12 = 0.555
Sim(4,235) = 2 * (0.47 + 0.44 + 0.76 + 0.64 + 0.98 + 0.85) /12 = 0.69
Then it become

	P1	P4	P235
P1	1.00	0.55	0.555
P4		1.00	0.69
P235			1.00

So the final result is



Q3

1) Assume that the stopping criterion is till the algorithm converges to the final k clusters. Can you insert several lines of pseudo-code to the algorithm to implement this logic? You

are not allowed to change the first 7 lines though.

Answer:

When there is no change of the centers then it can stop.

Red one means what I add
logic: using lastC to record the previous centers' condition, if after the update then there is no change in C, it can stop.

```
8    lastC <- C
9    For each group g
10        ci <- ComputeCenter(g);
11   if lastC = C then
12        canStop = True
11 return G
```

2)

Answer:

The first for loop, every points can find the nearest center, as the location of each point does not change and the location of each center does not change, we can get the best cost(gi), which is the smallest. So the cost must smaller than previous iteration.
In the second for loop we need to find the new center which is nearest to its neighbors. As the center will change, so every point is nearest to its center, then the cost can decrease as well. So the cost must smaller than previous iteration. So the cost of every cluster can decrease.
As a result, the cost of k clusters as evaluated at the end of each iteration never increases.

3) Prove that the cost of clusters obtained by k-means algorithm always converges to a local minima. You can make use of the previous conclusion even if you have not proved it.

Answer:

As we all know EM algorithm can converge to a local minima. If we use the squared loss as the loss function

$$J(\mu_1, \mu_2, \ldots, \mu_k) = \frac{1}{2} \sum_{j=1}^{K} \sum_{i=1}^{N} (x_i - \mu_j)^2$$

Here is the E-step, we need to assign each point to its nearest cluster.

$$\gamma_{nk} = \begin{cases} 1, & \text{if } k = argmin_j ||x_n - \mu_j||^2 \\ 0, & \text{otherwise} \end{cases}$$

Here is the M-step, we need to update the center according to its points.

$$\mu_k = \frac{\sum_n \gamma_{nk} x_n}{\sum_n \gamma_{nk}}$$

So we want to minimize the loss function. In the E-step, we need to find the most nearest γ and in the M-step we fix the γ then update μ. So it can converge to a local minima.