Q. 1 (a) - i

PAM Algorithm :

PAM algorithm can be described as:

1. Select k arbitary objects as medoid
2. For each pair of medoid i and non-medoid object h, calculate the swapping cost.
3. Select the object with the lowest swapping cost
   - If $TC_{ih} < 0$

   then swap (update) the current medoid i with h.
   - Assign each non-selected objects w.r.t the new medoid.
4. Repeat steps until there is no change.

Q. 1 (a) – ii

The total cost of swapping an object is called swapping cost. It can also be defined as sum of cost

i-e

$$TC_{ih} = \sum_{j=1}^{n} C_{j,ih}$$

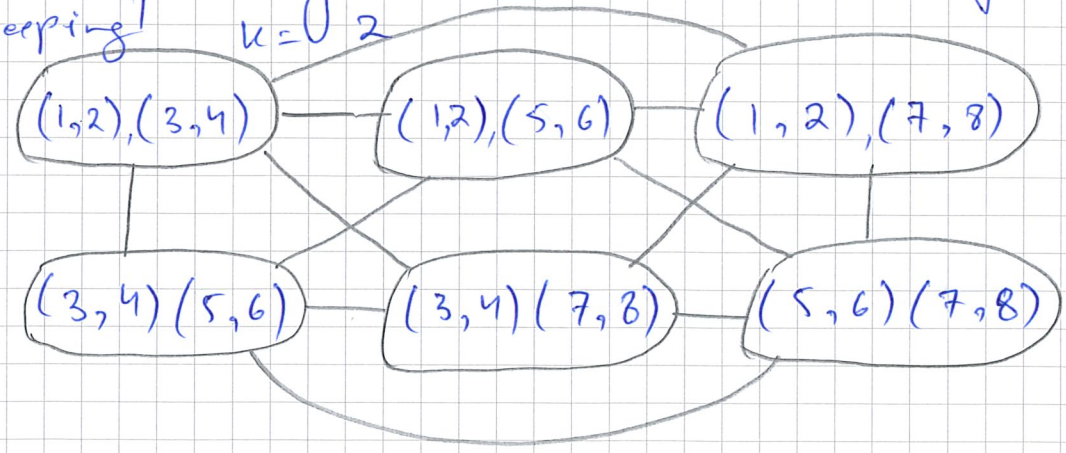
Q. 1 (a) – iii

Consider a dataset in 2 dimension

i-e

$$\{(1,2), (3,4), (5,6), (7,8)\}$$

The following nodes will be formed keeping $k = 2$



Now take initial k medoids

(3,4), (5,6)

cluster 1 : (3,4) → (3,4), (1,2)

cluster 2 : (5,6) → (5,6), (7,8)

Considering Manhattan dist (for reducing complexity of calculation

cost of it will be

$TC_{(3,4),(5,6)} = 0 + 4 + 0 + 4 = 8$

↳ old cost

Now take another neighboring point to compare its swapping cost with our initially selected medoids

(1,2) (5,6)

cluster 1 : (1,2) → (1,2), (3,4)

cluster 2 : (5,6) → (5,6), (7,8)

$TC_{(1,2)(5,6)} = 0 + 4 + 0 + 4 = 8$

↳ New_cost

Swapping Cost = New cost − Old cost

= 8 − 8

= 0

Thus the swapping is 0 as there is no change in cost

2nd Condition:

Now consider (1,2), (3,4) as initial medoids

cluster 1 : (1,2) → (1,2)

cluster 2 : (3,4) → (3,4), (5,6), (7,8)

$$TC_{(1,2),(3,4)} = 0 + 0 + 4 + 8 = 12$$

↳ old_cost

for its neighbour

(3,4), (5,6)

cluster 1 : (3,4) → (1,2), (3,4)

cluster 2 : (5,6) → (5,6), (7,8)

$$TC_{(3,4),(5,6)} = 4 + 0 + 0 + 4 = 8$$

↳ new_cost

Then,

$$\Rightarrow \text{new\_cost} - \text{old\_cost}$$

$$= 8 - 12$$

$$= -4$$

Since new medoid is better than the previously chosen, thus the result is strictly negative.

Q. 1 (b)

PAM and ~~CLARANS~~ ✓

Q. 1 (c)

none

Q. 2

Agglomerative Hierarchical Clustering:

Agglomerative hierarchical clustering is a hierarchical clustering approach in which a data set is divided into smaller clusters.

Agglomerative hierarchical clustering could be complete link or single link.

AGNES is an example of agglomerative hierarchical clustering.

	1	2	3	4	5
1	0				
2	8	0			
3	3	4	0		
4	1	7	9	0	
5	10	2	6	5	0

Step 1: we start with the smallest value then we combine the nodes

Step 2: determine the distance of combined nodes / points to every other point (for complete link we take max dist)

$$dist((1,4),2) = \max(d(1,2), d(4,2)) = \max(8,7) = 8$$

$$dist((1,4),3) = \max(d(1,3), d(3,4)) = \max(3,9) = 9$$

$$dist((1,4),5) = \max(d(1,5), d(4,5)) = \max(10,5) = 10$$

	(1,4)	2	3	5
(1,4)	0			
2	8	0		
3	9	4	0	
5	10	2	6	0

Now we repeat step 1 and 2 until all nodes are ~~comp~~ combined (or threshold 'u reached if defined)

$$d((2,5),(1,4)) = \max(d(2,(1,4)), d(5,(1,4))) = \max(8,10)$$
$$= 10$$

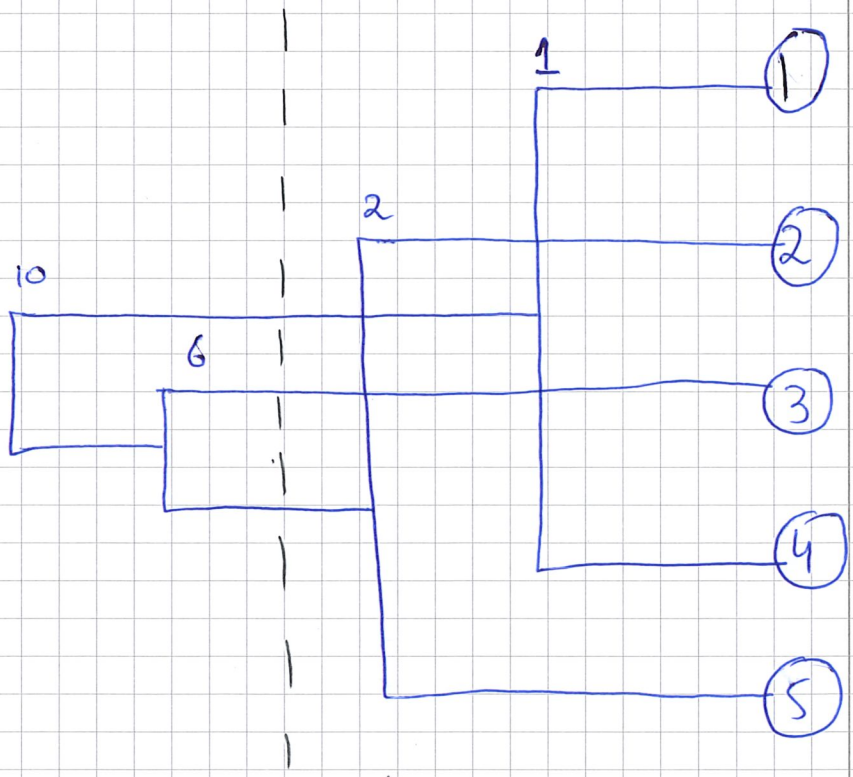
$$d((2,5),(3)) = \max(d(2,3), d(5,3)) = \max(4,6)$$
$$= 6$$

	(1,4)	(2,5)	3
(1,4)	0		
(2,5)	10	0	
3	9	6	0

combine 3 and (2,5)

$$d((3,(2,5)),(1,4)) = \max(d(3,(1,4)), d((2,5),(1,4)))$$
$$= \max(9,10) = 10$$

	(1,4)	(3,(2,5))
(1,4)	0	
(3,(2,5))	10	0



Threshold = 4 ⟹ If defined

(Then process would have stopped here

Q. 3

ROCK Algorithm:

ROCK is a hierarchical clustering approach in which clustering is done by checking similarity/proximity between the points.

- It maximizes the similarity between points i-e number of links within a cluster.
- Minimizes the link between points in different cluster.

Algorithm:

→ Take an arbitary point p. after scanning it, finds neighbour of the seleted point and determine the links.

→ Label data on disk: marks data having the highest goodness of measure (or above threshold)

Neighbour :

An object is said to be neighbour if it has at most difference of 1 item i.e

$$sim \geq t$$ (where t = threshold)

it can be explained by the following example :

abc abd abe acd ace

ade bcd bce cde

abf abg afg bfg

Consider an object abc its similarity with itself can be defined by

$$d(abc, abc) = \frac{a}{a+b+c} = \frac{3}{3} = 1$$

a	b	c
1	1	1
1	1	1

Now consider another object abd the similarity b/w abc and abd is

$$d(abc, abd) = \frac{a}{a+b+c} = \frac{2}{2+1+1} = \frac{2}{4} = \frac{1}{2}$$

$$= 0.5$$

a	b	c	d
1	1	1	0
1	1	0	1

Thus in the given scenario, an object will be neighbour if

$$sim \geq 0.5$$

Common Neighbour :

Two objects is said to have common neighbours if neighbour of "A" is also a neighbour of "B" i-e the neighbour must have at most 1 item different from node "A" and node "B".

For example :

If we consider the same data set as in previous example,

abd is a neighbour of abc

If we take another node bcd and determine its similarity with abd then

$$d(bcd, abd) = \frac{2}{2+1+1} = 0.5$$

a	b	c	d
0	1	1	1
1	1	1	0

which implies that abd is a common neighbour of abc and bcd

Link for Objects:

Link for objects can be defined as number of common neighbours between point P_i and P_j.

~~The above example~~

For example:

abc abd abe acd ace

ade bcd bce cde bde

abf abg afg bfg

in above example

abc has the following neighbours

abc, abd, abe, acd, ace

bcd, bce, abf, abg

and

cde has following neighbours

acd, ace, ade, bcd, bce, cde

Here it can be seen that acd, ace, bcd and bce are ~~acc~~ common neighbours of abc & cde so link between abc and cde will be given as

$$link(abc, cde) = 4$$

Link for clusters :

Link for clusters is defined as number of cross links between points in C_i and points in C_j where C_i and C_j are two clusters.

$$Link(C_i, C_j) = \sum_{\substack{P_i \in C_i \\ P_j \in C_j}} link(P_i, P_j)$$

Goodness Measure :

Goodness measure can be defined as ratio of cross links between points in cluster C_i with points in cluster C_j to the expected number of ~~clusters~~ cross links ~~b/w~~ ~~clus~~ points in cluster C_i & points in C_j.

$$G_1 = \frac{\text{Link}(C_i, C_j)}{\text{Expected \# of cross links between } C_i \text{ \& } C_j}$$

Expected number of Cross Links Between C_i & C_j is given by :

↳ Expected number of links in $C_i \cup C_j$

↳ expected number of links in C_i

↳ expected number of links in C_j

$$G_1 = \frac{\text{Link}(C_i, C_j)}{(n_i + n_j)^{1+2f(t)} - n_i^{1+2f(t)} - n_j^{1+2f(t)}}$$

where

expected # of links in $C_i \cup C_j$ is given by $(n_i + n_j)^{1+2f(t)}$

expected # of links in $C_i = n_i^{1+2f(t)}$

expected # of links in $C_j = n_j^{1+2f(t)}$

Q.4 (a)

The answer is both YES and NO.

PROVE:

If point 'p' is density connected to 'q' w.r.t Eps and Minpts, then it implies that there exist a point 'o' from which 'p' and 'q' are density reachable.

If q = 'o', then p will become density reachable from 'q'

General case

Counter Argument:

It might be possible that 'q' and 'p' are density reachable from ~~connected to~~ 'o' but 'q' is not a core object i-e 'q' does not have minpts in its ~~eps~~ eps neighborhood. In this case p is not density reachable from 'q' i-e q ≠ 'o' in any possible way (since it's not a core object)

Q. 4 (b)

OPTICS provides ordering to clustering structure.
OPTICS uses the principles of DBSCAN but with some extension.

OPTICS uses two extra parameters (as compared to DBSCAN)

1) Core distance : The minimum eps value to make an object a core point
If not found then undefined.
2) Reachability distance :

Max (core distance (o), d(o,p)) i.e maximum of core distance for a point 'o' and the distance of object 'o' to 'p'.

These parameters make optics perform better and consider area of high density while clustering.

OPTICS is often used for determining the Eps parameter for DBSCAN (or other density based algorithm)

Q. 5

	A	B	C	D	E	F	G
K	gold	(0,0)	Y	N	Y	N	silver
L	bronze	(1,1)	N	N	N	N	not value

for dist(K, L) for variable A

ranks

gold = 3

silver = 2

bronze = 1

$$z_f = \frac{r_f - 1}{m_f - 1}$$ where m_f = max rank

$$z_f(\text{Gold}) = \frac{3-1}{3-1} = \frac{2}{2}$$

$$z_f(\text{Gold}) = 1$$

$$z_f(\text{Silver}) = \frac{2-1}{3-1} = \frac{1}{2} = 0.5$$

$$z_f(\text{bronze}) = \frac{1-1}{3-1} = 0$$

dist(Gold, bronze) = d(1, 0)

considering Euclidean distance

$$= \sqrt{(1-0)^2} = \sqrt{(1)^2}$$

d(Gold, bronze) = 1

dist for attribute B:

$= (|0-1|) + (|0-1|)$

$= |-1| + |-1| = |+1|$

$d(K,L) = 2$

For binary symmetric variables if value is same (matching), $d(K,L) = 0$ else 1

For binary assymetric 0,0 case, delta = 0 else 1

so

$$d(K,L) = \frac{\sum_{i=1}^{N} \delta_{i(s)}\, d_{i(f)}}{\sum_{i=1}^{N} \delta_{i(s)}}$$

$$= \frac{\overset{A}{1}\times 1 + \overset{B}{2}\times 1 + \overset{C}{1}\times 1 + \overset{D}{0}\times 1 + \overset{E}{1}\times 1 + \overset{F}{0}\times 0 + \overset{G}{0}}{1+1+1+1+1}$$

$$= \frac{1+2+1+1}{5}$$

$$= 5/5$$

$d(K,L) = 1$

Q.6 (a)

According to apriori property, the infrequent itemsets are pruned in pruning step.

Q.6 (b)

Candidates are generated by performing self-join operation on itemset produced in k step.

Data is first scanned, then candidates are generated using self-join that is if I_k and J represents itemset then

- join is applied such that

$$I_1, \ldots\ldots I_{k-1} > J_{k-1}$$

In each k-step, the candidate size is increased by one if constraints are satisfied.

Ö: subset checking!