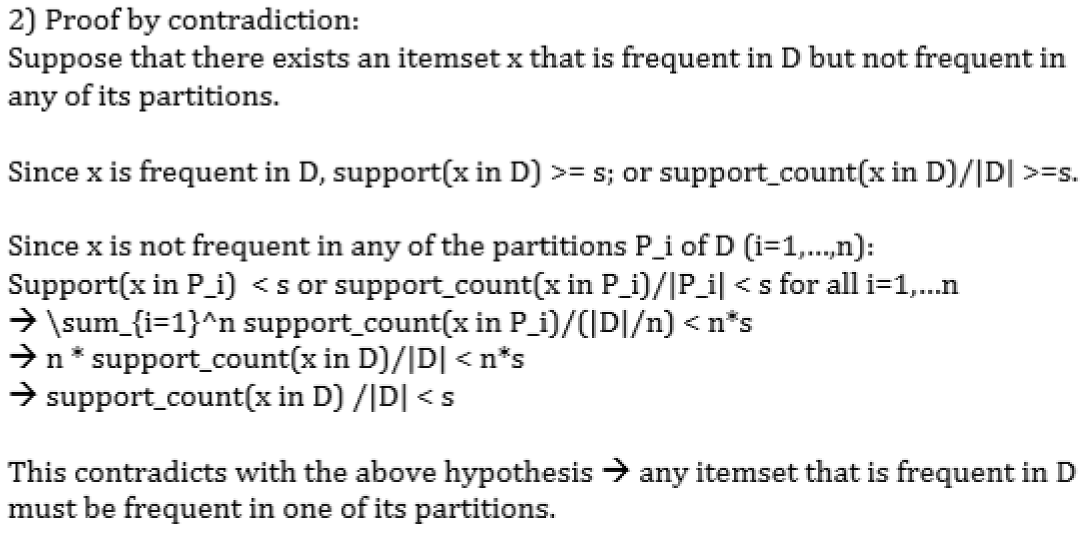
1) Given two objects represented by tuples (22, 1, 42, 10) and (20, 0, 36, 8)

a) Compute the Euclidean distance between two objects

b) Compute the Manhattan distance between two objects

2) A *partitioning* variation of Apriori subdivides the transactions of a database *D* into *n* nonoverlapping partitions. Prove that any itemset that is frequent in *D* must be frequent in at least one partition of *D*.



2) Proof by contradiction:
Suppose that there exists an itemset x that is frequent in D but not frequent in any of its partitions.

Since x is frequent in D, support(x in D) >= s; or support_count(x in D)/|D| >=s.

Since x is not frequent in any of the partitions P_i of D (i=1,...,n):
Support(x in P_i) < s or support_count(x in P_i)/|P_i| < s for all i=1,...n
→ \sum_{i=1}^n support_count(x in P_i)/(|D|/n) < n*s
→ n * support_count(x in D)/|D| < n*s
→ support_count(x in D) /|D| < s

This contradicts with the above hypothesis → any itemset that is frequent in D must be frequent in one of its partitions.

3) A database has five transactions. Let *min sup* = 60% and *min conf* = 80%.

T1: {L, I, O, N}

T2: {T, I, G, E, R}

T3: {M, A, K, E}

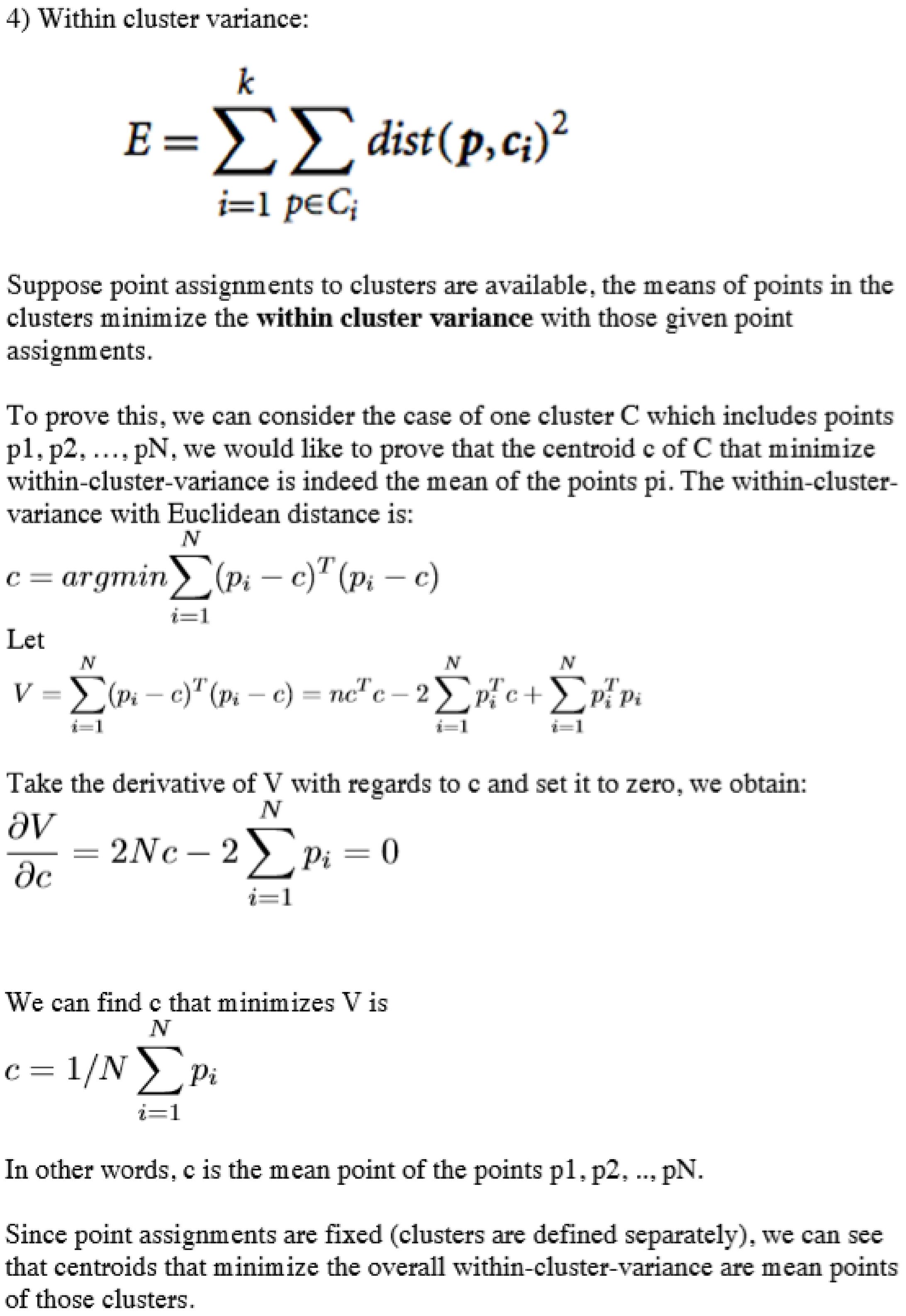
T4: {M, U, C, K, Y}

T5: {C, O, O, K, I, E}

Find all frequent itemsets using Apriori and FP-growth, respectively.

4) (Contributed by Hang Yu Deng)

K-means algorithm: prove that given point assignments for a cluster, the mean of the points is the desired centroid that minimizes the inter-cluster variance.



4) Within cluster variance:

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} dist(\mathbf{p}, \mathbf{c_i})^2$$

Suppose point assignments to clusters are available, the means of points in the clusters minimize the **within cluster variance** with those given point assignments.

To prove this, we can consider the case of one cluster C which includes points p1, p2, …, pN, we would like to prove that the centroid c of C that minimize within-cluster-variance is indeed the mean of the points pi. The within-cluster-variance with Euclidean distance is:

$$c = argmin \sum_{i=1}^{N} (p_i - c)^T (p_i - c)$$

Let

$$V = \sum_{i=1}^{N} (p_i - c)^T (p_i - c) = nc^T c - 2 \sum_{i=1}^{N} p_i^T c + \sum_{i=1}^{N} p_i^T p_i$$

Take the derivative of V with regards to c and set it to zero, we obtain:

$$\frac{\partial V}{\partial c} = 2Nc - 2 \sum_{i=1}^{N} p_i = 0$$

We can find c that minimizes V is

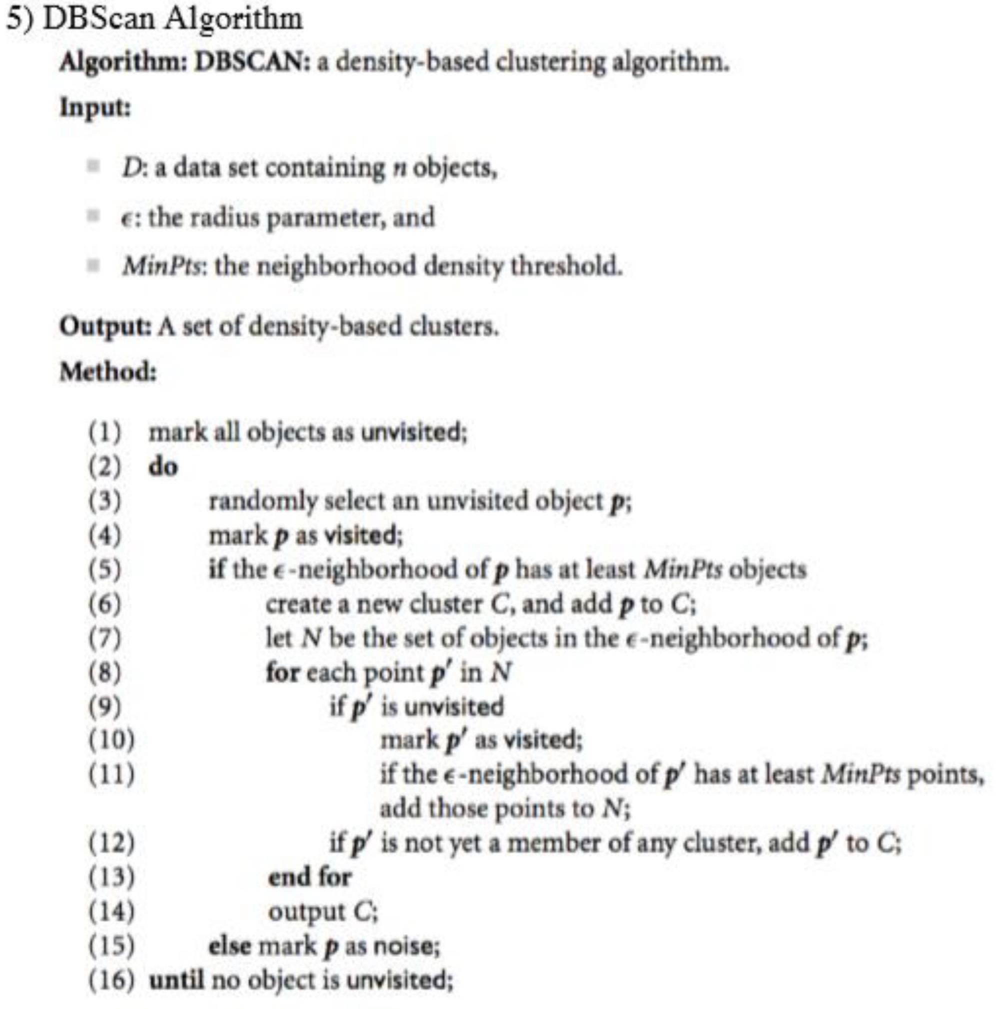
$$c = 1/N \sum_{i=1}^{N} p_i$$

In other words, c is the mean point of the points p1, p2, .., pN.

Since point assignments are fixed (clusters are defined separately), we can see that centroids that minimize the overall within-cluster-variance are mean points of those clusters.

5) (Contributed by Hang Yu Deng)

In DBSCAN algorithm, if p doesn't have enough neighborhoods, then it is marked as noise. But what if p's neiborhood contains a core object q? Naturally, p should be added to the cluster of the core object q. Explain whether DBSCAN can obtain this property.



5) DBScan Algorithm

Algorithm: DBSCAN: a density-based clustering algorithm.

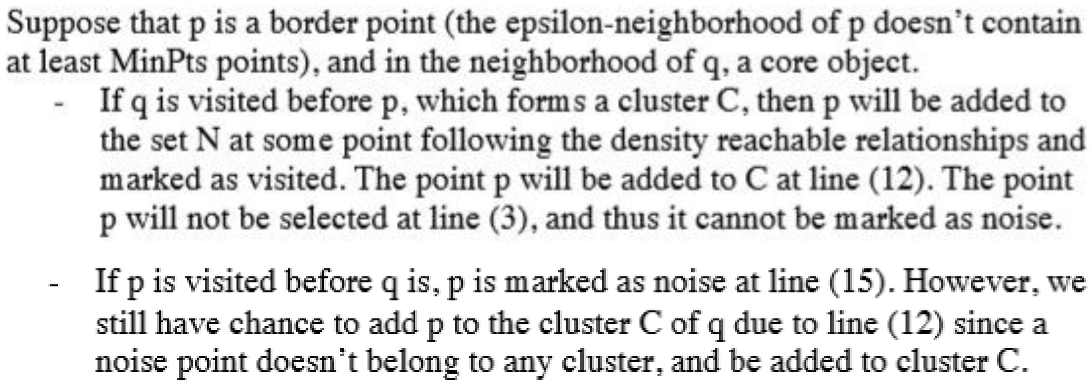
Input:

- *D*: a data set containing *n* objects,
- ϵ: the radius parameter, and
- *MinPts*: the neighborhood density threshold.

Output: A set of density-based clusters.

Method:

```
(1)  mark all objects as unvisited;
(2)  do
(3)      randomly select an unvisited object p;
(4)      mark p as visited;
(5)      if the $\epsilon$-neighborhood of p has at least MinPts objects
(6)          create a new cluster C, and add p to C;
(7)          let N be the set of objects in the $\epsilon$-neighborhood of p;
(8)          for each point p' in N
(9)              if p' is unvisited
(10)                 mark p' as visited;
(11)                 if the $\epsilon$-neighborhood of p' has at least MinPts points,
                     add those points to N;
(12)             if p' is not yet a member of any cluster, add p' to C;
(13)         end for
(14)         output C;
(15)     else mark p as noise;
(16) until no object is unvisited;
```



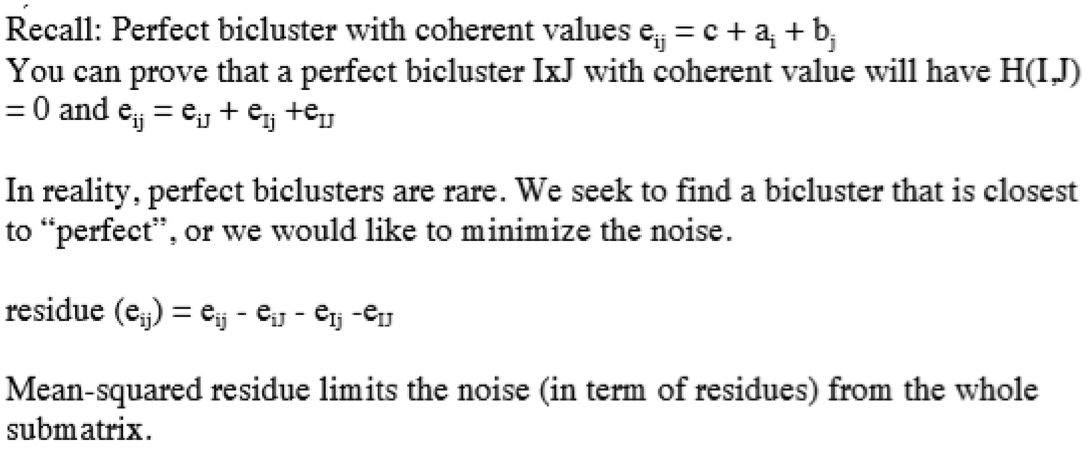
Suppose that p is a border point (the epsilon-neighborhood of p doesn't contain at least MinPts points), and in the neighborhood of q, a core object.

- If q is visited before p, which forms a cluster C, then p will be added to the set N at some point following the density reachable relationships and marked as visited. The point p will be added to C at line (12). The point p will not be selected at line (3), and thus it cannot be marked as noise.

- If p is visited before q is, p is marked as noise at line (15). However, we still have chance to add p to the cluster C of q due to line (12) since a noise point doesn't belong to any cluster, and be added to cluster C.

6) (Contributed by Hang Yu Deng)

In Biclustering, please explain why do we choose to minimize the mean-squared residual values?



Recall: Perfect bicluster with coherent values $e_{ij} = c + a_i + b_j$
You can prove that a perfect bicluster IxJ with coherent value will have H(I,J) = 0 and $e_{ij} = e_{iJ} + e_{Ij} + e_{IJ}$

In reality, perfect biclusters are rare. We seek to find a bicluster that is closest to "perfect", or we would like to minimize the noise.

residue (e_{ij}) = $e_{ij} - e_{iJ} - e_{Ij} - e_{IJ}$

Mean-squared residue limits the noise (in term of residues) from the whole submatrix.

7) Suppose that the data mining task is to cluster the following eight points (with (x, y) representing location) into three clusters.

A_1(2, 10), A_2(2, 5), A_3(8, 4), B_1(5, 8), B_2(7, 5), B_3(6, 4), C_1(1, 2), C_2(4, 9).

The distance function is Euclidean distance. Suppose initially we assign A_1, B_1, and C_1 as the center of each cluster, respectively. Use the k-means algorithm to show only the three cluster centers after the first round of execution