COMP5331 Knowledge Discovery in Databases (Fall Semester 2019)
Homework 1
Deadline: 9 Oct, 2019 10:30am
(Please hand in during lecture.)
Full Mark: 100 Marks

Coupon Instructions:

1. You can use a coupon to waive any question you want and obtain full marks for this question.
2. You can waive at most one question in each assignment.
3. You can also answer the question you will waive. We will also mark it but will give full marks to this question.
4. The coupon is non-transferrable. That is, the coupon with a unique ID can be used only by the student who obtained it in class.
5. Please staple the coupon to the submitted assignment.
6. Please write down the question no. you want to waive on the coupon.

Q1 [20 Marks]

(a) In general, we have a number of customers. For illustration, we are given two customers, namely X and Y. The following shows 5 transactions for these two customers. Each transaction contains three kinds of information: (1) customer ID (e.g., X and Y), (2) the time that this transaction occurred, and (3) all the items involved in this transaction.

Customer X, time 1, items A, B, C
Customer Y, time 2, items A, F
Customer X, time 3, items D, E
Customer X, time 4, item G
Customer Y, time 5, items D, E, G

For example, the first transaction corresponds to that customer X bought item A, item B and item C at time 1, while the last transaction corresponds to that customer Y bought item D, item E and item G at time 5.

A sequence is defined to be a series of itemsets in form of $<S_1, S_2, S_3, \ldots, S_m>$ where S_i is an itemset for $i = 1, 2, \ldots, m$. The above transactions can be transformed into two sequences as follows.

X: <{A, B, C}, {D, E}, {G}>
Y: <{A, F}, {D, E, G}>

After this transformation, each customer is associated with a sequence.

Given a sequence S in form of $<S_1, S_2, S_3, \ldots, S_m>$ and another sequence S' in form of $<S_1', S_2', S_3', \ldots, S_n'>$, S is said to be a subsequence of S' if $m \leq n$ and there exist m integers, namely $i_1, i_2, \ldots, i_m$, such that (i) $1 \leq i_1 < i_2 < \ldots < i_m \leq n$, and (2) $S_j \subseteq S_{i_j}'$ for $j = 1, 2, \ldots, m$. If S is a subsequence of S', then S' is defined to be a super-sequence of S.
The support of a sequence S is defined to be the total number of customers which sequences are super-sequences of S.

Given a positive integer k, a sequence in form of $<S_1, S_2, S_3, \ldots, S_m>$ is said to be a k-sequence if $\sum_{i=1}^{m} |S_i|=k$.

Can the Apriori algorithm be adapted to mining all k-sequences with support at least 2 where k = 2, 3, 4, …. ? If yes, please write down the proposed method using the concept of the Apriori algorithm and illustrate your algorithm with the above example. If no, please explain the reason.

(b) We want to study the same problem setting described in (a). However, each customer is associated to one binary attribute called "Rich" to indicate whether this customer is rich or not. There are only 2 possible values in this attribute, namely "Yes" and "No". In our example, customer X could have "Yes" in attribute "Rich" and customer Y could have "No" in attribute "Rich".

Given a k-sequence s and a value v in attribute "Rich", the support of a sequence S with respect to value v is defined to be the total number of customers which sequences are super-sequences of S and are associated with value v in attribute "Rich". The important ratio of s is defined to be the support of s with respect to value "Yes" divided by the support of s with respect to value "No".

Can the Apriori algorithm be adapted to mining all k-sequences with important ratio at least 2 and the support at least 1 where k = 2, 3, 4, ….? If yes, please write down the proposed method using the concept of the Apriori algorithm and illustrate your algorithm with the above example. If no, please explain the reason.

Note that when we compute the important ratio, if we encounter a division of a non-zero number by zero, we could regard it as a positive infinity value.

Q2 [20 Marks]

Given a positive integer K, we denote S_K to be a set of K-itemsets with support at least 1.
Given a positive integer K and a positive integer *l*, we define a set $S_{K,l}$ which is a subset of S_K such that each K-itemset in $S_{K,l}$ has its support at least s_l where s_l is the *l*-th greatest value in the multi-set of the supports of all K-itemsets in S_K. For example, the second greatest value in a multi-set of {4, 4, 3, 2} is 4 while the second greatest value of another multi-set of {4, 3, 3, 2} is 3.

We are given six items, namely A, B, C, D, E and F.

Suppose *l* is fixed and is set to 2.
We want to find $S_{K,l}$ for K = 1, 2 and 3.

The following shows four transactions with six items. Each row corresponds to a transaction where 1 corresponds to a presence of an item and 0 corresponds to an absence.

A	B	C	D	E	F
0	0	1	1	0	0
0	1	0	0	1	1
1	0	1	1	0	0
1	0	1	1	0	0

(a) (i) What is $S_{1,2}$?
(ii) What is $S_{2,2}$?
(iii) What is $S_{3,2}$?

(b) Can algorithm FP-growth be adapted to finding $S_{1,2}$, $S_{2,2}$ and $S_{3,2}$. If yes, please write down how to adapt algorithm FP-growth and illustrate the adapted algorithm with the above example. If no, please explain the reason.

(c) There are two parameters of finding $S_{K,l}$. They are K and *l*. In the traditional problem of finding frequent itemsets, we need to provide only one parameter, a support threshold.

It seems that it is troublesome to set one more parameter in the problem of finding $S_{K,l}$ (compared with the traditional frequent itemset mining you learnt). What are the advantages of the problem of finding $S_{K,l}$ compared with the traditional problem?

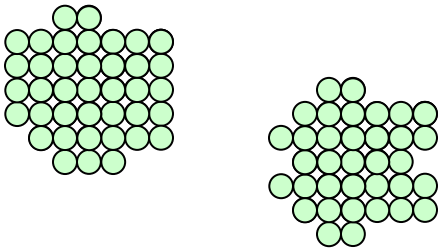
Q3 [20 Marks]

(a) One disadvantage of method k-means is that k (i.e., the number of clusters) should be pre-determined. One may suggest the following method to determine parameter k.

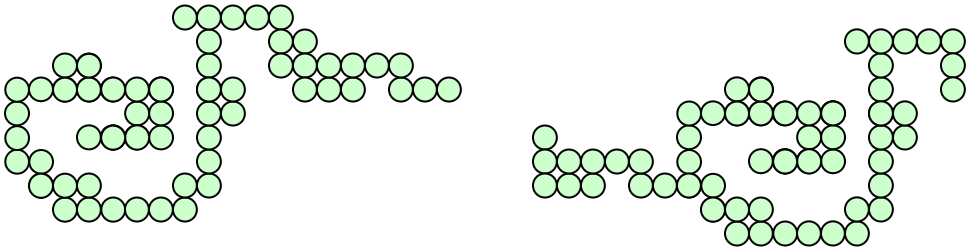
- **Step 1:** Set variable e_0 to ∞
- **Step 2:** Set k to 1 initially
- **Step 3:** Run the original k-means method and obtain k cluster centers (or means)
- **Step 4:** Set variable e_k to the sum of the distances between points and their closest cluster centers (according to the k cluster centers found).
- **Step 5:** If e_k converges (i.e., $(e_{k-1} - e_k)$ is equal to 0 or an extremely small number), then return k. Otherwise, increment k by 1 and repeat Step 3 to Step 4.

Can the above method determine a good value for k (i.e., the number of clusters)? Please explain.
If your answer is no, please also give an algorithm to determine a good value for k and explain why it is better than the above method.

(b) In a typical setting for k-means, we calculate the distance between a point and a cluster by computing the Euclidean distance between a point and the center (or mean) of the cluster.
However, this computation method can only capture "circular" clusters as shown below.



In some cases, clusters are not "circular". Instead, they may appear as lines as shown below.



Please give a method or a distance function to capture both the "circular" clusters and the "appear-as-line" clusters.

Q4 [20 Marks]

(a) Consider the following eight two-dimensional data points:

x_1:(20, 15), x_2: (8, 15), x_3: (20, 17), x_4: (8, 19), x_5: (23, 18), x_6: (6, 12), x_7: (15, 6), x_8: (15, 35)

(i) Assume that we adopt the Euclidean distance metric as the distance between any two points. Please write a matrix where its entries correspond to the pairwise distances between any two points.
(ii) Please use the agglomerative approach to group these points with centroid linkage. Draw the corresponding dendrogram for the clustering. You are required to specify the distance metric in the dendrogram.

(b) According to (a), we have eight data points. Suppose that we have one more data point. Totally, we have nine data points. Is it possible for you to design an algorithm so that the dendrogram obtained in (a) could be updated efficiently? If yes, please give a step-by-step algorithm. Otherwise, please elaborate in detail.

Q5 [20 Marks]

In the class, we learnt algorithm DBSCAN with 4 principles on a given dataset.

(a) Please write down a pseudo-code in order to perform clustering according to these 4 principles.
(b) Suppose that the dataset is updated from time to time.
(i) Please give a pseudo-code in order to perform clustering according to these 4 principles on this updating dataset. It is expected that the algorithm should return clustering results based on *both* the information stored based on the past data together with the new data.
(ii) Please analyze the time complexity of the algorithm.