Introduction to Data Science Assignment 3

Due time: 11:59 pm, November 9, 2023
5 questions and 100 points in total

This assignment covers the section of "mining patterns."

Every student is expected to complete the assignment independently. Discussion between students is encouraged, but no plagiarism will be tolerated.

Please submit your answers in a PDF file. If you want to also submit your code separately, you can pack all files, including a readme file, into a compressed file.

Question 1 (20 points)

Consider a transaction database $TDB = \{t_1, \dots, t_m\}$, where t_i is a transaction. Denote by $\tau > 0$ the absolute support threshold, that is, τ is about the number of transactions instead of percentage. We randomly select n $(n \geq \tau)$ transactions $t_{i_1}, \dots, t_{i_n}$ and let $X = t_{i_1} \cap \cdots \cap t_{i_n}$.

1. (10 points) If $X \neq \emptyset$, is X a frequent itemset? If your answer is yes, give a proof. Otherwise, give an example to show that X may not be frequent.
2. (10 points) Is X a closed frequent itemset? If your answer is yes, give a proof. Otherwise, give an example to show that X may not be a closed itemset.

Question 2 (10 points)

Given a transaction database T, let $x_1, x_2, \dots, x_k$ be the k $(k > 0)$ most frequent items in T. Prove, for any length-k itemset Y, $sup(Y) \leq \{ sup(x_1), sup(x_2), \dots, sup(x_k)\}$.

Question 3 (15 points)

Given a transaction database T, let X and Y be two itemsets such that $sup(X) = sup(Y)$ and $X \cap Y \neq \emptyset$. For example, $X = abc$ and $Y = cde$. If $sup(X) = sup(Y) = sup(X \cap Y)$, what is the relation between $sup(X \cap Y)$ and $sup(X \cup Y)$? Use one or multiple examples to observe and explain the relation and then present a concrete proof.

Question 4 (20 points)

Suppose that a large transaction database TDB of 10 million transactions is distributed in 4 databases, denoted by $TDB1$ (1 million transactions), $TDB2$ (4 million transactions), $TDB3$ (2 million transactions), and $TDB4$ (3 million transactions). Each transaction is stored in only one database. That is, TDB is the union of the four databases. We want to find frequent itemsets with respect to the minimum support threshold 1% (equivalently, 1 million) in TDB. You can assume a central server to manage the mining process and collect the frequent itemsets. Design a solution so that we do not need to move any transactions crossing the databases or moving to the central server.

Question 5 (35 points)

In this question, we use the Pima Indians Diabetes Database that was used in Assignment 1 and only consider the attributes Pregnancies, Glucose (discretized into groups according to the highest digit, i.e., [0, 9], [10, 19], …, [190, 199]), BloodPressure (discretized into groups according to the highest digit), age (discretized into groups according to the highest digit), and Outcome. Treat each record in the database as a transaction. Please note that two attributes may take the same value, e.g., 0, but the same value on different attributes should be treated as different items.

Find the 100 most frequent itemsets such that each itemset found should contain an item in attribute Outcome. You can write your own program or use any existing program available on the web or open-source suites. Please give reference to the programs that are not developed by you. You can use either FP-growth, Apriori, or any variants. Show the code.

1. (10 points) Describe your approach, particularly, how do you modify the original FP-growth or Apriori algorithms to ensure each frequent pattern contains an item in attribute Outcome.
2. (10 points) Report the 100 most frequent itemsets found.
3. (15 points) Using the 100 most frequent itemsets found, can you form 5 association rules with the highest confidence? Describe your method and list the rules.