Assignment 3

CS 412: Introduction to Data Mining (Spring 2023)
Instructor: Hanghang Tong

Release date: Feb. 14th, 2023
Due date: Mar. 23rd, 2023

- This assignment will cover the content from Chapters #4 (Mining Frequent Patterns, Association and Correlations: Pattern Evaluation) and #5 (Advanced pattern Mining).
- Feel free to discuss with other members of the class when doing the homework. You should, however, write down your own solution **independently**. ***Very Important Notes*: (1) there is a fine line between collaboration and completing the assignment by yourself and (2) aiding others to cheat would have the same consequence as the cheating itself. Please try to keep the solution brief and clear.**
- Please use Piazza first if you have questions about the assignment. Also feel free to send us e-mails and come to office hours.
- The assignment is due at 11:59 PM on the due date. We will be using Canvas for collecting homework assignments. **Please do not hand in a scan of your hand-written solution, only the typed solution (e.g., Microsoft Word, Latex, etc) will be graded**. Contact the TAs if you are having technical difficulties in submitting the assignment. We do **NOT** accept late assignment!
- The assignment should be submitted as a **single** PDF file using the name convention yourNetID_HW3.pdf. If you use additional source code for solving problems, you are required to submit them and use the file names to identify the corresponding questions. For instance, yourNetID_HW3_problem1.py refers to the python source code for Problem 1, replace netid with your netid. Compress all the files (PDF and source code files) into one zip file. Submit the compressed file **ONLY**. (If you did not use any source code, submitting the PDF file without compression will be fine)
- For each question, **you will NOT get full credits if you only give out a final result.** Necessary calculation steps are required. If the result is not an integer, round your result to 4 decimal places.

NOTE: Unless otherwise specified, in this assignment, we use $sup(A)$ to denote the absolute support of an itemset A and $s(A)$ to denote the relative support of an itemset A.

Problem 1. True or False (24 points)

Please justify your answers with **at most** 3 sentences (1 point for true or false and 2 points for the justification).

(1) (3 points) Suppose two association rules are derived from the same frequent pattern, they (i.e., the two association rules) must share the same support and confidence.

False. Two association rules derived from the same frequent pattern can have different supports and confidences. For example, if the frequent pattern is {A, B, C} with a support of 10, one rule could be A → B with a confidence of 90% (support of {A, B, C} is 9), while another rule could be B → C with a confidence of 80% (support of {A, B, C} is 8).

(2) (3 points) Pattern A is a max-pattern if A is frequent and there has no super-pattern $B \supset A$ that is frequent. Besides, max-pattern is a lossless compression of frequent patterns.

False, max-pattern is not a lossless compression of frequent pat- terns.

(3) (3 points) Given two itemsets A and B, the range of $Kulczynski(A, B)$ measure of A and B is $(0, 1)$, i.e., $Kulczynski(A, B) \in (0, 1)$.

True. The Kulczynski measure is defined as the arithmetic mean of the Jaccard similarity coefficient and the cosine similarity coefficient, both of which range from 0 to 1. Therefore, Kulczynski measure is also bounded by 0 and 1.

(4) (3 points) In Apriori Algorithm, if there is any itemset which is infrequent, its superset can not be generated.

True. The Apriori algorithm generates candidate itemsets by joining frequent (k-1)-itemsets, so if an itemset is infrequent, none of its supersets can be frequent. This is due to the downward closure property of frequent itemsets.

(5) (3 points) When a given database to be mined is extremely large and FP-tree cannot fit into the memory, it is **impossible** to mine the database using FPGrowth algorithm. [Hint: you may refer to textbook for answer.]

False. The FPGrowth algorithm is designed specifically to handle large databases that do not fit into memory. It uses an FP-tree structure to compress the database and mine frequent itemsets efficiently without needing to load the entire database into memory at once.

(6) (3 points) Given two frequent itemsets A and B, we denote s(A), s(B) and s($A \cup B$) as the relative support of A, the relative support of B and the relative support of $A \cup B$, respectively. If $s(A \cup B) \ll s(A) \times s(B)$, it implies itemsets A and B rarely occur together.

True. The Kulczynski measure can be calculated as s(A ∩ B) / ((s(A) + s(B)) / 2). Therefore, if s(A ∪ B) is much smaller than s(A) × s(B), it implies that s(A ∩ B) is much smaller than (s(A) + s(B)) / 2, which indicates that A and B rarely occur together.

(7) (3 points) Let A be an itemset and V be another bigger itemset. Is $A \subseteq V$ anti-monotone? If true, please explain the reason. Otherwise, please provide an counterexample.

True. A ⊆ V is anti-monotone because if an itemset A is a subset of a bigger itemset V, and we remove an item from V, A becomes a superset of the new itemset. Therefore, if the support of A decreases, the support of the new itemset must also decrease, which implies that A ⊆ V is anti-monotone.

(8) (3 points) Let $V = \{a, b, c, d, e\}$ be the set of all items, $A = \{a, b, c\}$ and $B = \{a, b, c, e\}$ be two itemsets. Is $A \subseteq B$ monotone? If true, please explain the reason. Otherwise, please provide an counterexample.

True. A ⊆ B is monotone because if an itemset A is a subset of another itemset B, and we add an item to B, A remains a subset of the new itemset. Therefore, if the support of A increases, the support of the new itemset must also increase, which implies that A ⊆ B is monotone.

Problem 2. Basics of Patterns (22 points)

(a) (4 points) Given a transaction database *TDB*, we partition it into two parts, TDB_1 and TDB_2. If an itemset X is infrequent in both TDB_1 and TDB_2 with respect to a minimum (relative) support threshold s, is it possible for X to be frequent in original TDB? Why?

In TDB1, assume that $\frac{a}{c} < s$, in TDB2, assume that $\frac{b}{d} < s$. Therefore, $a < cs, b < ds$, $a + b < (c + d)s, \frac{a+b}{c+d} < s$. Therefore, it is impossible for X to be frequent.

(b) (9 points) Suppose we have a TDB_1 with the following transactions:

$T_1 = \{a_{10}, a_{11}, ..., a_{20}\}$, $T_2 = \{a_1, a_2, ..., a_{20}\}$, $T_3 = \{a_1, a_2, ..., a_{25}\}$

i. (3 points) For TDB_1, how many max pattern(s) do we have and what is(are) it(they) if minimum (absolute) support is 1?

$\{a_1, a_2, ..., a_{25}\}:1$

ii. (3 points) For TDB_1, how many max pattern(s) do we have and what is(are) it(they) if minimum (absolute) support is 2?

$\{a_1, a_2, ..., a_{20}\}:2$

iii. (3 points) For TDB_1, how many max pattern(s) do we have and what is(are) it(they) if minimum (absolute) support is 3?

$\{a_{10}, a_{11}, ..., a_{20}\}:3$

(c) (9 points) Suppose we have a TDB_2 with the following transactions:

$T_4 = \{a_1, a_2, ..., a_{10}\}$, $T_5 = \{a_{20}, a_{21}, ..., a_{25}\}$, $T_6 = \{a_1, a_2, ..., a_{25}\}$

i. (3 points) For TDB_2, how many max pattern(s) do we have and what is(are) it(they) if minimum (absolute) support is 2?

$\{a_1, a_2, \ldots, a_{10}\}$:2

$\{a_{20}, a_{21}, \ldots, a_{25}\}$:2

ii. (3 points) For TDB_2, how many closed pattern(s) do we have and what is(are) it(they) if minimum (absolute) support is 1?

$\{a_1, a_2, \ldots, a_{10}\}$:2

$\{a_{20}, a_{21}, \ldots, a_{25}\}$:2

$\{a_1, a_2, \ldots, a_{25}\}$:1

iii. (3 points) For TDB_2, how many closed pattern(s) do we have and what is(are) it(they) if minimum (absolute) support is 2?

$\{a_1, a_2, \ldots, a_{10}\}$:2

$\{a_{20}, a_{21}, \ldots, a_{25}\}$:2

Problem 3. Null Invariance (16 points)
Giving two itemsets A and B and the following contigency table (Table 1).

	A	$\neg A$	*row*
B	a	b	$a+b$
$\neg B$	c	d	$c+d$
col	$a+c$	$b+d$	$a+b+c+d$

Table 1: Contigency Table

(a) (4 points) What is the lift $Lift(A, B)$ of A and B? Under what condition would A and B be independent?

$$Lift(A,B) = \frac{s(A \cup B)}{s(A) * s(B)} = \frac{\frac{a}{a+b+c+d}}{\frac{a+c}{a+b+c+d} * \frac{a+b}{a+b+c+d}} = \frac{(a)(a+b+c+d)}{(a+c)(a+b)}$$

A and B be independent when $Lift(A,B) = 1$.

(b) (4 points) What is the imbalanced ratio $IR(A, B)$ of A and B?

$$IR(A,B) = \frac{|s(A) - s(B)|}{s(A) + s(B) - s(A \cup B)} = \frac{|c-b|}{a+b+c}$$

(c) (4 points) What is Cosine measure $Cosine(A, B)$ of A and B? And explain why $Cosine(A, B)$ is null-invariant.

$$Consine(A,B) = \frac{\frac{a}{a+b+c+d}}{\sqrt{\frac{a+c}{a+b+c+d} * \frac{a+b}{a+b+c+d}}} = \frac{a}{\sqrt{(a+c)*(a+b)}}$$

(d) (4 points) What is the difference between Lift $Lift(A, B)$ and Jaccard measure $Jaccard(A, B)$? Why would this difference make $Jaccard(A, B)$ null-invariant?

Lift and Jaccard measure are two different metrics used in data analysis to measure association or similarity between two variables or sets. Lift measures the degree of dependence between two events A and B, while Jaccard measures the similarity between two sets A and B. Jaccard measure can be modified to make it null-invariant for binary data by adding 1 to the numerator and denominator of the formula.

Problem 4. Pattern Mining Concepts & Algorithms (20 points)
Given you have transaction database from a grocery store shown in Table 2 and the corresponding product table in Table 3, please answer the following questions.

Tid	Items
1	B, C, E, F, G
2	A, B, C, F
3	A, B, C
4	E, F
5	A, B, G
6	B, C, D, E
7	A, B, C, D
8	A, C
9	A, B, G
10	A, D, E, F, G

Table 2: Transaction Database

Item	Price	Profit
A	10	4
B	16	8
C	46	20
D	40	0
E	37	12
F	30	-10
G	45	-5

Table 3: Product Table

(1) (4 points) Given an association rule $B \rightarrow C(s, c)$, what are its relative support s and confidence c?

s{B} = 5 / 10 = 0.5

c = sup{B,C} / sup(B) = 5/7 = 0.7143

(2) Given a collection of constraints as follows, identify their types (monotone, anti-monotone, data anti-monotone, succinct, convertible). If multiple types coexist in the following constraints, please list them all.

(i) (2 points) Total price of all purchased items is less than \$250.
Count(A) = 7
Count(B) = 7

Count(C) = 6
Count(D) = 3
Count(E) = 4
Count(F) = 4
Count(G) = 4
7*(10+16) + 6*(46) + 3*(40) + 4*(37+30+45) = 1026 > 250.
Anti-monotone, succinet.

(ii) (2 points) Total price of all purchased items is at least $200.

Monotone, data anti - monotone

(iii) (2 points) The minimal price of the purchased item is less than $50.

Monotone, data anti-Monotone, succinct

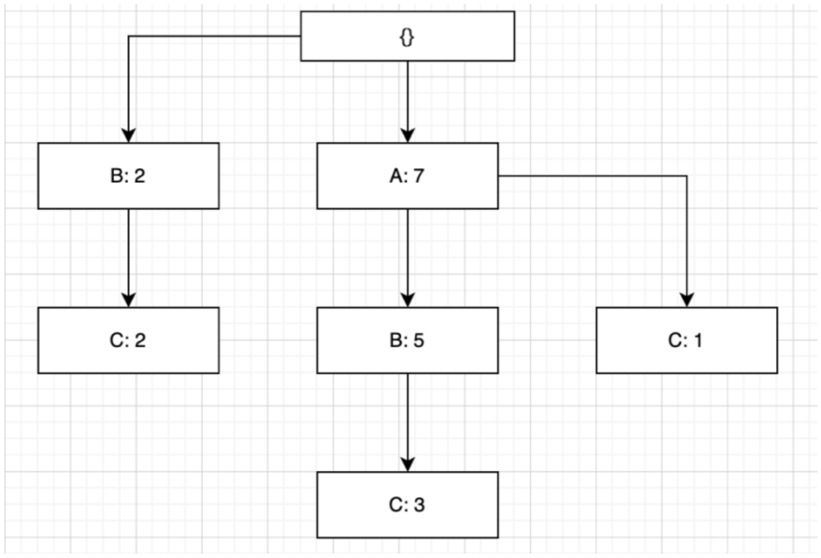
(3) (8 points) Construct the FP-tree from the transaction database above, when the minimum relative support is 0.5. Please show intermediate steps to get all credits. Note that you can use any software (hand drawing is also allowed) to draw the FP-tree and insert a screenshot of this FP-tree into the submitted file.

s{A} = s{B} = 7 / 10 = 0.7

s{C} = 6 / 10 = 0.6

F-list = A-B-C

TID	Items in the Transaction	Ordered, frequent itemlist
1	B, C, E, F, G	B, C
2	A, B, C, F	A, B, C
3	A, B, C	A, B, C
4	E, F	
5	A, B, G	A, B
6	B, C, D, E	B, C
7	A, B, C, D	A, B, C
8	A, C	A, C
9	A, B, G	A, B
10	A, D, E, F, G	A



(4) (6 points) Why is FP-growth more efficient than Apriori algorithm? Is FP-growth **always** faster than Apriori? Why?

In general, FP-growth surpasses the Apriori algorithm in terms of efficiency due to its ability to diminish the number of dataset scans required and obviate the generation of candidate itemsets. Consequently, this reduces the computational time and memory usage.

However, FP-growth may not always be faster than Apriori, especially when the dataset has a low minimum support threshold or a large number of infrequent items.

Problem 5. Programming Problem for Frequent Pattern Mining (18 points)
Given a transaction dataset, implement the following algorithms.

- A frequent pattern mining algorithm (e.g., the Apriori algorithm or FP-Growth) to extract the frequent itemsets.
- A closed pattern mining algorithm to extract the closed itemsets. Hint: Use the frequent itemsets extracted in step 1 for identifying closed itemsets.
- A maximal pattern mining algorithm to extract the maximal itemsets. Hint: Use the frequent itemsets extracted in step 1 for identifying maximal itemsets.

You will not get credit if your code does not work.

Please download the "example_input.txt" and the "example_output.txt" for the detail of the input format and the output format. **Input Format.** The input describes a transaction dataset.

The first line of the input corresponds to the minimum support. Each following line of the input corresponds to one transaction. Items in each transaction are seperated by a space. Please refer to the sample input for illustration. In sample input 0, the minimum support is 2. The dataset contains 3 transactions and 5 item types (A, B, C, D and E).

Output Format. The output must satisfy the following formatting requirements to pass the test cases.

- The output consists of three successive parts: the frequent itemsets (part 1), the closed itemsets (part 2), and the maximal itemsets (part 3). Each part must be separated by an empty line.
- For each part, the corresponding itemsets must appear along with their support, one per line. The format of each such itemsets must be 'support: itemset'; where itemset elements must appear in alphabetical order, separated by space.
- The itemsets must be ordered as per their support (from largest to smallest). Ties should be resolved by ordering the itemsets as per their alphabetical order.

Please refer to the sample output ("example_output.txt"). In sample output, the first 9 itemsets are the frequent itemsets (part 1), the following 3 itemsets are the closed itemsets (part 2), and the last 2 itemsets are the maximal itemsets (part 3).

What you have to submit. Your code file (e.g., homework3.py) with a function freq_pattern_mining() which takes the input file ("test_input.txt"), and a clear README file as the instruction to run your code:

(a) (4 points) What is the support value for the pattern "D" and "D F" in the given "test_input.txt"?

The support value for pattern "D" is 455, for pattern "D F" is 247.

(b) (4 points) Is the pattern "B D E F" a closed itemset? Is the pattern "A E G H" a closed itemset? If Yes, what is the support value of this pattern?)

"B D E F" is a closed itemset, the support value of this pattern is 83.

"A E G H" is a closed itemset, the support value of this pattern is 59.

(c) (4 points) How many maximal itemsets in total? Is "A B C D E F G H I" a maximal itemset in your results?

There are 26 maximal itemset in total. "A B C D E F G H I" is not a maximal itemset.

(d) (6 points) We will check the correctness of your submitted programming file for the last 6 points with new test cases. So, you do not need to answer this question in the submitted PDF of Assignment 3.