CS550: Massive Data Mining
Homework 1

Due 11:59pm Monday, February 23, 2026
Please see the homework file for late policy

Submission Instructions

Honor Code Students may have discussions about the homework with peers. However, each student must write down their solutions independently to show they understand the solution well enough in order to reconstruct it by themselves. Students should clearly mention the names of all the other students with whom they have discussions about the homework. Directly using the code or solutions obtained from the web or from others is considered an honor code violation. We check all the submissions for plagiarism and take the honor code seriously, and we hope students to do the same.

Discussions (People with whom you discussed ideas used in your answers):

On-line or hardcopy documents used as part of your answers:

I acknowledge and accept the Honor Code.

*(Signed)*____

If you are not printing this document out, please type your initials above.

Answer to Question 1

Algorithm Description

This problem can be addressed using a single MapReduce job, described as follows.
Map phase: Given a user U whose friend list is $F = \{f_1, f_2, \ldots, f_n\}$, the mapper produces two types of key-value pairs:

1. For each pair of friends (f_i, f_j) within F, it outputs $(f_i, (f_j, 1))$ and $(f_j, (f_i, 1))$. This captures the fact that f_i and f_j have U as a common friend, making them candidate recommendations for one another.
2. For each $f_i \in F$, it outputs $(U, (f_i, -1))$, which serves as a flag indicating that U and f_i already have a direct friendship.

Reduce phase: The reducer collects all values associated with a given user U. It accumulates the mutual-friend counts per candidate and removes any candidate already in U's friend list (identified by the -1 flag). The surviving candidates are then ranked by their mutual-friend count in descending order; ties are resolved by choosing the smaller user ID first. Finally, the top 10 candidates are returned as recommendations.

Recommendations for Specified Users

User	Recommendations
924	439, 2409, 6995, 11860, 15416, 43748, 45881
8941	8943, 8944, 8940
8942	8939, 8940, 8943, 8944
9019	9022, 317, 9023
9020	9021, 9016, 9017, 9022, 317, 9023
9021	9020, 9016, 9017, 9022, 317, 9023
9022	9019, 9020, 9021, 317, 9016, 9017, 9023
9990	13134, 13478, 13877, 34299, 34485, 34642, 37941
9992	9987, 9989, 35667, 9991
9993	9991, 13134, 13478, 13877, 34299, 34485, 34642, 37941

Answer to Question 2(a)

One key limitation of confidence is that it does not take into account the prior probability of item B. Recall that confidence is given by

$$\mathrm{conf}(A \to B) = P(B \mid A).$$

When B appears very frequently in transactions (e.g., a staple product like bread), $P(B)$ is inherently high. Under statistical independence between A and B, we still get

$$P(B \mid A) = P(B),$$

which means $\mathrm{conf}(A \to B)$ can be elevated solely due to B's popularity rather than any meaningful relationship between A and B. As a result, confidence can be misleading by flagging spurious rules driven by a common consequent.

Lift overcomes this issue by normalizing against $P(B)$:

$$\mathrm{lift}(A \to B) = \frac{P(B \mid A)}{P(B)} = \frac{P(A \cap B)}{P(A)\,P(B)}.$$

When A and B are independent, $P(A \cap B) = P(A)\,P(B)$, which yields

$$\mathrm{lift}(A \to B) = 1.$$

This means lift returns a neutral value of 1 whenever there is no genuine association, regardless of how frequent B might be.

Likewise, **conviction** accounts for $P(B)$ through its definition:

$$\mathrm{conv}(A \to B) = \frac{1 - P(B)}{1 - P(B \mid A)}.$$

Under independence, $P(B \mid A) = P(B)$, so

$$\mathrm{conv}(A \to B) = 1.$$

Hence, conviction also defaults to a baseline value of 1 when no real dependency exists, and is therefore not susceptible to the same flaw as confidence.

Answer to Question 2(b)

We call a measure symmetrical when measure$(A \to B)$ = measure$(B \to A)$ holds for all A and B.

Confidence. We have

$$\text{conf}(A \to B) = \frac{P(A \cap B)}{P(A)}.$$

To show this is not symmetrical, consider a concrete example. Let

$$P(A) = 0.2, \quad P(B) = 0.5, \quad P(A \cap B) = 0.1.$$

Computing both directions:

$$\text{conf}(A \to B) = \frac{0.1}{0.2} = 0.5, \qquad \text{conf}(B \to A) = \frac{0.1}{0.5} = 0.2.$$

Because $0.5 \neq 0.2$, we conclude that **confidence is not symmetrical**.

Lift. Recall that

$$\text{lift}(A \to B) = \frac{P(A \cap B)}{P(A)\, P(B)}.$$

Swapping the roles of A and B:

$$\text{lift}(B \to A) = \frac{P(B \cap A)}{P(B)\, P(A)}.$$

Since $P(A \cap B) = P(B \cap A)$ and scalar multiplication is commutative, the two expressions are identical:

$$\text{lift}(A \to B) = \text{lift}(B \to A).$$

Therefore, **lift is symmetrical**. □

Conviction. Recall that

$$\text{conv}(A \to B) = \frac{1 - P(B)}{1 - P(B \mid A)}.$$

We disprove symmetry with an example. Let

$$P(A) = 0.1, \quad P(B) = 0.6, \quad P(A \cap B) = 0.09.$$

First, $P(B \mid A) = 0.09/0.1 = 0.9$, giving

$$\text{conv}(A \to B) = \frac{1 - 0.6}{1 - 0.9} = \frac{0.4}{0.1} = 4.$$

In the opposite direction, $P(A \mid B) = 0.09/0.6 = 0.15$, so

$$\text{conv}(B \to A) = \frac{1 - 0.1}{1 - 0.15} = \frac{0.9}{0.85} \approx 1.06.$$

Since $4 \neq 1.06$, **conviction is not symmetrical**.

Answer to Question 2(c)

We say a rule $A \to B$ is a *perfect implication* when $P(B \mid A) = 1$, meaning B is always present whenever A is. A measure is called desirable if it attains its maximum value precisely for such perfect rules, allowing them to be easily distinguished.

Confidence. Since

$$\mathrm{conf}(A \to B) = P(B \mid A) \in [0, 1],$$

a perfect implication gives $\mathrm{conf}(A \to B) = 1$, which is the largest value confidence can take. Thus, **confidence achieves its maximum for perfect implications**.

Lift. We have

$$\mathrm{lift}(A \to B) = \frac{P(B \mid A)}{P(B)}.$$

When $A \to B$ is a perfect implication, this becomes $1/P(B)$. However, this quantity varies with $P(B)$ and is not bounded above by any fixed constant. In particular, a non-perfect rule involving a rare consequent can exceed the lift of a perfect rule. For instance, suppose we have a perfect rule where $P(B) = 0.9$:

$$\mathrm{lift} = \frac{1}{0.9} \approx 1.11.$$

Compare this with a non-perfect rule $A' \to B'$ where $P(B') = 0.01$ and $P(B' \mid A') = 0.5$:

$$\mathrm{lift}(A' \to B') = \frac{0.5}{0.01} = 50 \gg 1.11.$$

Since a non-perfect rule can produce a higher lift, **lift lacks the desired property**.

Conviction. Recall

$$\mathrm{conv}(A \to B) = \frac{1 - P(B)}{1 - P(B \mid A)}.$$

For a perfect implication, the denominator equals $1 - 1 = 0$, causing

$$\mathrm{conv}(A \to B) \to +\infty.$$

Any rule that is not perfect has $P(B \mid A) < 1$, yielding a finite conviction value. Hence, **conviction reaches its maximum (diverges to infinity) exclusively for perfect implications**.

To conclude, among the three measures, only **confidence** and **conviction** possess this desirable property.

Answer to Question 2(d)

We apply the Apriori algorithm to discover frequent itemsets in a level-wise fashion. The support of an itemset X counts how many transactions contain X:

$$\text{support}(X) = |\{t \in T : X \subseteq t\}|$$

A pair $\{X, Y\}$ qualifies as frequent when its support meets the threshold of 100.
From each such frequent pair $\{X, Y\}$, two association rules are derived: $X \Rightarrow Y$ and $Y \Rightarrow X$. Their confidence is computed as the ratio of the pair's support to the antecedent's support:

$$\text{conf}(X \Rightarrow Y) = \frac{\text{support}(\{X, Y\})}{\text{support}(X)}$$

The resulting rules are ranked by confidence in descending order, with ties broken lexicographically by the left-hand-side item. The top 5 rules are listed below:

1. DAI93865 ⇒ FRO40251 (confidence = 1.000000)
2. GRO85051 ⇒ FRO40251 (confidence = 0.999176)
3. GRO38636 ⇒ FRO40251 (confidence = 0.990654)
4. ELE12951 ⇒ FRO40251 (confidence = 0.990566)
5. DAI88079 ⇒ FRO40251 (confidence = 0.986726)

Answer to Question 2(e)

Extending the Apriori procedure to size-3 itemsets, a triple $\{X, Y, Z\}$ is deemed frequent whenever

$$\text{support}(\{X, Y, Z\}) \geq 100.$$

Each frequent triple yields three association rules:

$$(X, Y) \Rightarrow Z, \quad (X, Z) \Rightarrow Y, \quad (Y, Z) \Rightarrow X.$$

The confidence of a rule with a pair on the left-hand side is:

$$\text{conf}((X, Y) \Rightarrow Z) = \frac{\text{support}(\{X, Y, Z\})}{\text{support}(\{X, Y\})}$$

The two items in each left-hand-side pair are arranged in lexicographic order before ranking. Rules are then sorted by decreasing confidence; when confidence values coincide, ties are resolved by lexicographic comparison of the left-hand-side pair (first by the first item, then by the second). Below are the top 5 rules:

1. (DAI23334, ELE92920) ⇒ DAI62779 (confidence = 1.000000)
2. (DAI31081, GRO85051) ⇒ FRO40251 (confidence = 1.000000)
3. (DAI55911, GRO85051) ⇒ FRO40251 (confidence = 1.000000)
4. (DAI62779, DAI88079) ⇒ FRO40251 (confidence = 1.000000)
5. (DAI75645, GRO85051) ⇒ FRO40251 (confidence = 1.000000)

Answer to Question 3(a)

Consider a column with m entries equal to 1 and $n - m$ entries equal to 0. When we sample k rows uniformly at random without replacement, the outcome is "don't know" precisely when every selected row has a 0 — that is, all k rows are drawn from the $n - m$ zero-rows. The probability of this event equals

$$\Pr(\text{don't know}) = \frac{\binom{n-m}{k}}{\binom{n}{k}}.$$

Expanding the binomial coefficients gives

$$\frac{\binom{n-m}{k}}{\binom{n}{k}} = \frac{(n-m)(n-m-1)\cdots(n-m-k+1)}{n(n-1)\cdots(n-k+1)}.$$

An equivalent way to express this product is

$$\frac{\binom{n-m}{k}}{\binom{n}{k}} = \prod_{i=0}^{m-1} \frac{n-k-i}{n-i}.$$

Observe that for every $0 \le i \le m-1$,

$$\frac{n-k-i}{n-i} \le \frac{n-k}{n},$$

because subtracting i from both the numerator and denominator of $\frac{n-k}{n}$ can only decrease (or preserve) the ratio.
Applying this bound to each of the m factors yields

$$\Pr(\text{don't know}) = \prod_{i=0}^{m-1} \frac{n-k-i}{n-i} \le \left(\frac{n-k}{n}\right)^m.$$

□

Answer to Question 3(b)

By the result of part (a), we can write

$$\Pr(\text{don't know}) \le \left(\frac{n-k}{n}\right)^m = \left(1 - \frac{k}{n}\right)^m.$$

Our goal is to choose k so that this probability does not exceed e^{-10}:

$$\left(1 - \frac{k}{n}\right)^m \le e^{-10}.$$

Since n is much larger than both m and k, the ratio k/n is small. We can therefore invoke the standard approximation $(1-x)^m \approx e^{-mx}$ for small x. Setting $x = k/n$:

$$\left(1 - \frac{k}{n}\right)^m \approx e^{-mk/n}.$$

The requirement becomes

$$e^{-mk/n} \le e^{-10}.$$

Because the exponential function is monotonically increasing, this simplifies to

$$\frac{mk}{n} \ge 10 \quad \Longrightarrow \quad k \ge \frac{10n}{m}.$$

Hence, the minimum value of k satisfying the constraint is

$$\boxed{k_{\min} = \frac{10n}{m}}$$

Answer to Question 3(c)

Consider a matrix with $n = 4$ rows and two columns representing sets S_1 and S_2.
Construction: Define $S_1 = \{1, 2\}$ and $S_2 = \{2, 3\}$ over a universe of 4 elements. The characteristic matrix is:

Row	S_1	S_2
1	1	0
2	1	1
3	0	1
4	0	0

Jaccard Similarity: We have $S_1 \cap S_2 = \{2\}$ and $S_1 \cup S_2 = \{1, 2, 3\}$, so

$$J(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|} = \frac{1}{3}.$$

Min-hash agreement under cyclic permutations: With $n = 4$ rows, there are exactly 4 cyclic permutations, one for each starting row r. Denote the min-hash of a set S under a permutation by $h(S)$:

1. $r = 1$: row order $(1, 2, 3, 4)$. $h(S_1) = 1$, $h(S_2) = 2$. No match.
2. $r = 2$: row order $(2, 3, 4, 1)$. $h(S_1) = 2$, $h(S_2) = 2$. Match.
3. $r = 3$: row order $(3, 4, 1, 2)$. $h(S_1) = 1$, $h(S_2) = 3$. No match.
4. $r = 4$: row order $(4, 1, 2, 3)$. $h(S_1) = 1$, $h(S_2) = 2$. No match.

Out of the 4 cyclic permutations, only one produces matching min-hash values, giving a probability of $\frac{1}{4}$. The restricted set of n cyclic permutations does not sample the space of all $n!$ permutations uniformly, which is why the min-hash agreement probability deviates from the Jaccard similarity.
Since $\frac{1}{4} \neq \frac{1}{3}$, this demonstrates that cyclic permutations alone cannot correctly estimate the Jaccard similarity.