CS550: Massive Data Mining and Learning
Homework 1

Due 11:59pm Thursday, Feb 19, 2021
Only one late period is allowed for this homework (11:59pm Friday 2/20)

Submission Instructions

Assignment Submission Include a signed agreement to the Honor Code with this assignment. Assignments are due at 11:59pm. All students must submit their homework via Canvas. Students can typeset or scan their homework. Students also need to include their code in the final submission zip file. Put all the code for a single question into a single file.

Late Day Policy Each student will have a total of *two* free late days, and for each homework only one late day can be used. If a late day is used, the due date is 11:59pm on the next day.

Honor Code Students may discuss and work on homework problems in groups. This is encouraged. 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 who were part of their discussion group. Using code or solutions obtained from the web is considered an honor code violation. We check all the submissions for plagiarism. We take the honor code seriously and expect students to do the same.

Discussion Group (People with whom you discussed ideas used in your answers): Rohan Shah (ras513)

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

I acknowledge and accept the Honor Code.

(Signed) RG

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

Answer to Question 1

Each line in the input file is parsed by the `map()` method of the `Q1Map` class. The key in the key-value pairs outputted by this method is always a user, and the value is always a *relationship*. A relationship is simply a tuple containing another user and this user's relationship ("already friends" or "has mutual friend") to the key user. For each input parsed by this method, the method outputs a "already friends" relationship between the specified user (key) and each of the user's friends. The method also then outputs "has mutual friend" relationships between each pairing of the user's friends. In the case of an empty friends list, this method simply returns the specified user and a constant that will be recognized by the reducer.

In the `reduce()` method of the `Q1Reduce` class, the input is always a key-value pair in which the value is the set of all values (relationships) produced by the `map()` method for the given key (user). For clarity, let us refer to this user as u. These relationships are stored in a hash-map in order to record how many times each relationship occurred; each key in the hash-map is another user, and the value associated with each key is the number of mutual friends between the key user and u. The `reduce()` method first checks for the constant representing an empty friends list, in which case it outputs an empty list to be associated with u. Then, this method iterates through all of the relationships outputted for u by `map()`. For each "has mutual friend" relationship, the corresponding entry in the hash-map will be incremented. However, if an "already friends" relationship is seen, then the corresponding entry in the hash-map is changed to a constant, and it will never be updated by future "has mutual friend" relationships that are seen. Then, the keys of the hash-map are sorted by their paired values as per the specified ordering. Finally, the top 10 recommendations for u are outputted from this sorted list. During this output process, any key that mapped to the "already friends" constant is ignored.

The recommendations for the specified users are as follow:

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)

To understand why the lack of $P(B)$ in the calculation of $conf(A \rightarrow B)$ is a drawback, let us look at an example. First, assume $conf(A \rightarrow B) = 1$, but $P(B)$ is much greater than $P(A)$. In this case, A occurred in only a few baskets, whereas B was a common item set throughout the basket list. However, this scenario tells us that B is guaranteed to occur if A occurs. This conclusion is relatively useless, since B is a common item set and would occur many times regardless, but it is also very misleading to claim that an occurrence of a rare item set will always result in the occurrence of a common item set. Ultimately, though, these disadvantages are unknown since $P(B)$ is not considered in the confidence calculation. Overall, because $conf(A \rightarrow B)$ only takes into consideration $P(A)$, the metric can be exaggerated and thus distort the association's significance.

This drawback does not exist for lift and conviction because both of these calculations take into consideration $P(B)$. $P(B)$ is defined as the probability that B occurred in the baskets; if there are N baskets, and B has a support of s, then $P(B) = s/N$. The lift calculation considers $P(B)$ in its denominator, whereas the conviction calculation considers $P(B)$ in the numerator.

Answer to Question 2(b)

In the response below, s_X represents the support of item set X in the given set of baskets.

1. Confidence is not symmetrical. The following is a counter-example: In a given set of $N = 40$ baskets, let $s_A = 10$ and $s_B = 20$. Furthermore, let $s_{[AB]} = 5$. The calculation of confidence can be rewritten as follows:

$$conf(X \rightarrow Y) = P(Y|X) = \frac{P(X,Y)}{P(X)} = \frac{\frac{s_{[XY]}}{N}}{\frac{s_Y}{N}} = \frac{s_{[XY]}}{s_Y}$$

   Using this formula, $conf(A \rightarrow B) = 0.5$, but $conf(B \rightarrow A) = 0.25$. Therefore, confidence is not symmetrical.

2. Lift is symmetrical. First, let us rewrite the formula for confidence:

$$conf(X \rightarrow Y) = P(Y|X) = \frac{P(X,Y)}{P(X)}$$

   Given a set of N baskets, $P(Y) = \frac{s_Y}{N}$. The formula for lift then becomes:

$$lift(X \rightarrow Y) = \frac{\frac{P(X,Y)}{P(X)}}{P(Y)} = \frac{P(X,Y)}{P(X)P(Y)}$$

   This final expression is commutative, so the order of the inputs X and Y does not change the final value of their corresponding lift calculation. Therefore, lift is symmetrical.

3. Conviction is not symmetrical. The following is a counter-example: In a given set of $N = 40$ baskets, let $s_A = 10$ and $s_B = 20$. Furthermore, let $s_{[AB]} = 10$. Consequently, using the given formula for confidence, $conf(A \rightarrow B) = 1$ and $conf(B \rightarrow A) = 0.5$. Additionally, $P(A) = \frac{s_A}{N} = 0.25$ and $P(B) = \frac{s_B}{N} = 0.5$. Thus, using the given formula for conviction, $conv(A \rightarrow B) = \infty$, whereas $conv(B \rightarrow A) = 1.5$. Therefore, conviction is not symmetrical.

Answer to Question 2(c)

If a rule $A \rightarrow B$ holds 100% of the time, then that means that an occurrence of item set A is always paired with an occurrence of item set B. By definition, this implies that confidence is a desirable measure. $conf(A \rightarrow B)$ has a maximum of 1, and this can only occur if all baskets that contain A also contain B, i.e. $A \rightarrow B$ holds 100% of the time. (It is important to note that confidence may not be totally "desirable" because it does not consider $P(B)$, but in the context of this question, that is an auxiliary consideration.)

Consequently, conviction is also a desirable measure. If a rule $A \rightarrow B$ holds 100% of the time, then it has confidence 1. As a result, its conviction would be ∞, since the denominator of the conviction formula would be $1 - conf(A \rightarrow B) = 1 - 1 = 0$, and clearly this is the maximum "value" for the conviction measure. Since this maximum can only occur when a rule's confidence is 1, and since a rule can only have confidence 1 if it holds 100% of the time, then the conviction measure is maximal only when the rule holds 100% of the time. Therefore, conviction is a desirable measure.

Answer to Question 2(d)

The top 5 rules for item pairs in terms of confidence are as follow:

Rule	**Confidence**
DAI93865 → FRO40251	100.0%
GRO85051 → FRO40251	99.92%
GRO38636 → FRO40251	99.07%
ELE12951 → FRO40251	99.06%
DAI88079 → FRO40251	98.67%

Answer to Question 2(e)

The top 5 rules for item triples in terms of confidence are as follow:

Rule	**Confidence**
DAI23334,ELE92920 → DAI62779	100.0%
DAI31081,GRO85051 → FRO40251	100.0%
DAI55911,GRO85051 → FRO40251	100.0%
DAI62779,DAI88079 → FRO40251	100.0%
DAI75645,GRO85051 → FRO40251	100.0%

Answer to Question 3(a)

The following proof assumes that $n > m$ and $n - m > k$. The probability that the result of min-hashing is "don't know" for a given column is equivalent in meaning and in value to the probability that the chosen k rows contain no 1s in this column – let us refer to this probability as P. There are $\binom{n}{k}$ possible combinations of k rows chosen, and since there are $n - m$ rows that contain no 1s in the given column, there are $\binom{n-m}{k}$ possible combinations of k rows that satisfy the condition of P. Therefore:

$$P = \frac{\binom{n-m}{k}}{\binom{n}{k}} = \frac{\frac{(n-m)!}{(n-m-k)!k!}}{\frac{n!}{(n-k)!k!}} = \frac{(n-m)!}{n!}\frac{(n-k)!}{(n-k-m)!} = \frac{(n-k)!}{n!}\frac{(n-m)!}{(n-k-m)!}$$

Let $X = \frac{(n-k)!}{n!}$ and $Y = \frac{(n-m)!}{(n-k-m)!}$. X can then be expanded as follows:

$$X = \frac{(n-k)}{n}\frac{(n-k-1)}{(n-1)}\cdots\frac{(n-k-m+1)}{(n-m+1)}\frac{(n-k-m)!}{(n-m)!}$$

Therefore, since P is the product of X and Y, and the last factor of X as shown above cancels out Y, P can be rewritten as follows:

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

When written in this form, P has m factors, and since all of these factors are less than or equal to $\frac{n-k}{n}$, the following conclusion can be made:

$$P \leq \left(\frac{n-k}{n}\right)^m$$

Answer to Question 3(b)

Let us denote P as the probability that the result of min-hashing is "don't know" for a given column. The focus, then, is to rewrite $\left(\frac{n-k}{n}\right)^m$ to fit the approximation $(1-\frac{1}{x})^x \approx \frac{1}{e}$.

$$P = \left(\frac{n-k}{n}\right)^m = \left(1-\frac{k}{n}\right)^m = \left(1-\frac{1}{\frac{n}{k}}\right)^m = \left(1-\frac{1}{\frac{n}{k}}\right)^{\frac{n}{k}\frac{km}{n}}$$

$$\left(1-\frac{1}{\frac{n}{k}}\right)^{\frac{n}{k}} \approx \frac{1}{e} \Longrightarrow \left(1-\frac{1}{\frac{n}{k}}\right)^{\frac{n}{k}\frac{km}{n}} \approx \left(\frac{1}{e}\right)^{\frac{km}{n}}$$

$$e^{-\frac{km}{n}} \leq e^{-10}$$

Solving for k in this last equation, we find that the smallest value of k that will ensure that $P \leq e^{-10}$ is $\frac{10n}{m}$.

Answer to Question 3(c)

$$M = \begin{bmatrix} S_1 & S_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 0 \\ 1 & 1 \\ 1 & 0 \end{bmatrix}$$

$S_1 \cap S_2 = 1$, and $S_1 \cup S_2 = 4$, so the Jaccard similarity is $\frac{S_1 \cap S_2}{S_1 \cup S_2} = 0.25$.

Since there are 5 rows, there are 5 cyclic permutations:

$$P_1 = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix} \quad P_2 = \begin{bmatrix} 2 \\ 3 \\ 4 \\ 5 \\ 1 \end{bmatrix} \quad P_3 = \begin{bmatrix} 3 \\ 4 \\ 5 \\ 1 \\ 2 \end{bmatrix} \quad P_4 = \begin{bmatrix} 4 \\ 5 \\ 1 \\ 2 \\ 3 \end{bmatrix} \quad P_5 = \begin{bmatrix} 5 \\ 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

Using these permutations (starting from P_1 and ending with P_5), the min-hash values are as follow:

$$\begin{bmatrix} 2 & 1 \\ 1 & 2 \\ 1 & 1 \\ 2 & 2 \\ 1 & 3 \end{bmatrix}$$

The min-hash values are equivalent for S_1 and S_2 in 2 of the 5 permutations, so the probability of this condition is 0.4. Therefore, the probability that the min-hash values of S_1 and S_2 agree is not the same as their Jaccard similarity.