CS550: Massive Data Mining and Learning
Homework 1

Due 11:59pm Thursday, March 5, 2020
Only one late period is allowed for this homework (11:59pm Friday 3/6)

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 Sakai. 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):

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

I acknowledge and accept the Honor Code.

*KD*_____

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

Answer to Question 1

Algorithm:
Define a "degree" between 2 users such that degree = 1 if the users are directly connected and degree = 2 if the users have a mutual friend. Consider an example of the input:
A B,C,D
Mapper

1. Computing degree 1 friends of A: every user in the list is a direct friend of A and hence will be a degree 1 friend of A.
   Output:
   (Key = A, Value = (B,1))
   (Key = A, Value = (C,1))
   (Key = A, Value = (D,1))

2. Computing degree 2 friends: Every combination of user in the friends list of A will have a degree 2 since the users have a mutual friend A. The pair of users might be direct friends too but we do not have that information yet. This will be taken into account in the reducer.
   Output:
   (Key = B, Value = (C,2)), (Key = C, Value = (B,2))
   (Key = C, Value = (D,2)), (Key = D, Value = (C,2))
   (Key = B, Value = (D,2)), (Key = D, Value = (B,2))

Reducer

1. The reducer receives as input user as key and iterable pair of (user,degree) as values. A hashmap is used to keep track of the count of degree 2 connections between the key user and user in the value pair.

2. If a degree 1 friend shows up, indicate in hashmap as -1 and donot use it for counting. Delete it later.

3. Use a priority queue to sort the hashmap according to number of mutual friends.

Recommendations for users:
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)

Confidence takes into account the probability of B being purchased along with A, but ignores the popularity of item set B. This may give incorrect results. For e.g., if B is popular, its occurrence might be independent of A but rule will be valid because of high support of B. Lift and Conviction both take Pr(B) into consideration and hence will not suffer the same problem.

Answer to Question 2(b)

Lift is symmetrical. Confidence and Conviction are not symmetrical.

- **Confidence:** Confidence of rule is given as:
  $Conf(A \to B) = \frac{Pr(A \cap B)}{Pr(A)}$
  $Conf(B \to A) = \frac{Pr(A \cap B)}{Pr(B)}$
  When Pr(A) $\neq$ Pr(B) $\Rightarrow$Conf(A $\to$ B) $\neq Conf(B \to A)$.
- **Lift**
  $Lift(A \to B) = Lift(B \to A) = \frac{S(A \cap B)}{S(A)S(B)}$
- **Conviction**
  Conviction is dependent on Confidence and hence it will not be symmetrical as well.

Example: Consider the basket B = AB, CD, AD.
S(A) = 2/3, S(B) = 1/3, $Pr(A \cap B) = 1/3$

Confidence
$Conf(A \to B) = \frac{1/3}{2/3} = \mathbf{0.50}$
$Conf(B \to A) = \frac{1/3}{1/3} = \mathbf{1}$
$Conf(A \to B) \neq Conf(B \to A)$

Conviction
$Conv(A \to B) = \frac{1-S(B)}{1-conf(A \to B)} = \frac{1-1/3}{1-1/2} = \frac{2/3}{1/2} = \frac{4}{3}$
$Conv(B \to A) = \frac{1-S(A)}{1-conf(B \to A)} = \frac{1-2/3}{1-1} = \infty$
$\Rightarrow$Conv(A $\to$ B) $\neq Conv(B \to A)$

Answer to Question 2(c)

Consider the rule $B \rightarrow A$. For rules that hold 100% of the time, $Pr(B|A) = 1$.
$\implies$ Conf(B $\rightarrow A$) = 1
$\implies$ Conv(B $\rightarrow A$) = $\frac{1-S(A)}{1-conf(B \rightarrow A)} = \frac{1-S(A)}{1-1} = \infty$
which are the maximal values possible for the respective measures. Hence, *Confidence* and *Conviction* are desirable.

But, the value of *lift* will depend on the Pr(B) and hence can vary for different rules. As a result, the value of *lift* may or may not be maximal making it not desirable.

Answer to Question 2(d)

Top 5 pairs with support = 100:

Rules	**Confidence**
DAI93865 ->FRO40251	1.0
GRO85051 ->FRO40251	0.999176276771005
GRO38636 ->FRO40251	0.9906542056074766
ELE12951 ->FRO40251	0.9905660377358491
DAI88079 ->FRO40251	0.9867256637168141

Answer to Question 2(e)

Top 5 triples with support = 100:

Rules	**Confidence**
DAI23334, ELE92920 ->DAI62779	1.0
DAI31081, GRO85051 ->FRO40251	1.0
DAI55911, GRO85051 ->FRO40251	1.0
DAI62779, DAI88079 ->FRO40251	1.0
DAI75645, GRO85051 ->FRO40251	1.0

Answer to Question 3(a)

k rows are randomly chosen out of n rows to hash. If none of the k rows contains a 1 then the result of min-hashing will be "don't know". Given that in n rows of a column, there are m 1's and (n-m) 0's. We need to prove that the probability of getting "don't know" for such a column is:

$$(\frac{n-k}{n})^m \tag{1}$$

This column will have min-hash value "don't know" if all the k rows for hashing have a value of 0. Number of ways to select k rows from (n-m) rows containing 0's is:

$$\binom{n-m}{k} \tag{2}$$

Number of ways to choose k rows from all the available rows n is:

$$\binom{n}{k} \tag{3}$$

Hence, P(getting "don't know" as the min hash value)

$$= \frac{\binom{n-m}{k}}{\binom{n}{k}} \tag{4}$$

$$= \frac{(n-m)!k!(n-k)!}{(n-m-k)!k!n!} \tag{5}$$

$$= \frac{(n-m)!}{n!}.\frac{(n-k)!}{(n-m-k)!} \tag{6}$$

$$= \frac{(n-k)(n-k-1)\dots(n-(m-k)+1)}{n(n-1)\dots(n-m+1)} \tag{7}$$

Number of terms in both numerator and denominator are m.
Maximum value of each term = $\frac{n-k}{n}$.
Hence, probability we get "don't know" as the min-hash value = $(\frac{n-k}{n})^m$

Answer to Question 3(b)

From 3(a), we know that the probability of "don't know" as the min-hash value is at most $(\frac{n-k}{n})^m$. Now, we want the probability of "don't know" to be at most e^{-10}, given that $n \gg m, k$.

$$\Rightarrow (\frac{n-k}{n})^m \leq e^{-10} \tag{8}$$

$$\Rightarrow (1-\frac{k}{n})^m \leq e^{-10} \tag{9}$$

Dividing and multiplying the exponent by n/k, we get:

$$\Rightarrow ((1-\frac{k}{n})^{\frac{n}{k}})^{\frac{mk}{n}} \leq e^{-10} \tag{10}$$

For large x, $(1-\frac{1}{x})^x \approx \frac{1}{e}$. Since n $\gg$ k, $\frac{n}{k}$ is a large value. Using this,

$$\Rightarrow (\frac{1}{e})^{\frac{mk}{n}} \leq e^{-10} \tag{11}$$

$$\Rightarrow e^{-\frac{mk}{n}} \leq e^{-10} \tag{12}$$

$$\Rightarrow -\frac{mk}{n} \leq -10 \tag{13}$$

$$\Rightarrow \frac{mk}{n} \geq 10 \tag{14}$$

$$\Rightarrow k \geq \frac{10n}{m} \tag{15}$$

From Eq.15, the lowest value of k = $\frac{10n}{m}$

Answer to Question 3(c)

Consider thetwo columns:

$$C1 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}, C2 = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

For cyclic permutation, starting at row 1, the min-hash values of both the columns are given as follows:

Permutation	**Min Hash Values**	
	C1	**C2**
[1 2 3 4]	2	2
[4 1 2 3]	1	1
[3 4 1 2]	2	1
[2 3 4 1]	1	3

Jaccard similarity of C1 and C2 = $\frac{1}{3}$ = **0.67**

From the table, probability that a random cyclic permutation yields the same min-hash value for both C1 and C2 = $\frac{2}{4}$ = **0.50**

Hence, the probability (over cyclic permutations only) the min-hash values agree is not the same as the Jaccard similarity.