CS550: Massive Data Mining and Learning
Homework 1

Twisha Gaurang Naik (tn268)

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.

I acknowledge and accept the Honor Code.

(Signed) Twisha Gaurang Naik (tn268)

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

Answer to Question 1: Map-Reduce

(i) Code: "question_1.java"

(ii) Description of algorithm:
MutualFriendWritable:
A new writable is written to handle the specific requirement of storing pairs of user and mutual friend. This writable has two properties, ***user*** and ***mutualFriend***.
Map function:

- Map function takes as input one line of the input file at a time in form of key-value pair. Key is the offset from the start of file and value is the text of that particular line. Thus <LongWritable, Text> is the format of input pair.
- After reading the input it outputs key value pairs which of type <LongWritable, MutualFriendWritable>. For the user and each of his/her friends an entry with mutual friend -1 is made. -1 indicates that two people are already friends. Key will be the user, value will be an object of MutualFriendWritable with user as friend and mutualFriend as -1 or other. For all the possible pairs of the friend list, an entry is added with the current user as the mutual friend.
- Example: Input line [a: b, c]
  Map output: <a: (b, -1)>, <a: (c, -1)>, <b: (c, a)>, <c: (b, a)>

Reduce Function:

- Reduce function takes the output generated by map function as input. The format is <LongWritable, MutualFriendWritable>.
- The reduce function creates a separate hashmap for each user with key as a recommended friend and value as the list of mutual friends. If two people are already friends, a null value is added so that it does not appear in final result.
- For recommending 10 people with maximum number of mutual friends, the keys of the hashmap are sorted based on the length of values corresponding to them. The output format is <LongWritable, Text>.

(iii) 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

Question 2: Association Rules

Answer to Question 2(a)

- **Confidence:**

$$conf(A \rightarrow B) = Pr(B|A) = \frac{Pr(A \cap B)}{Pr(A)}$$

Drawback: Incorrect rules
Explanation: This completely ignores the probability of B, Pr(B). Suppose, item B is too frequent. There is a chance that product B appears with every other product A giving a rule $(A \rightarrow B)$ with high confidence. Thus, all high confidence rules might not be significant.
In other words, it is possible that occurrence of B is independent of A. Int such a case, $conf(A \rightarrow B) = Pr(B|A) = Pr(B)$. If Pr(B) is high, i.e. if support(B) is high, it will lead to a rule $(A \rightarrow B)$ which is not correct.

- **Lift:**

$$lift(A \rightarrow B) = \frac{conf(A \rightarrow B)}{support(B)}$$

In terms of probability:

$$lift(A \rightarrow B) = \frac{Pr(A \cap B)}{Pr(A).Pr(B)}$$

- **Conviction** compares the "probability that A appears without B if they were independent" with the "actual frequency of the appearance of A without B".

$$conv(A \rightarrow B) = \frac{1 - support(B)}{1 - conf(A \rightarrow B)}$$

In terms of probability:

$$conv(A \rightarrow B) = \frac{Pr(A)(\neg Pr(B))}{Pr(A \cap \neg B)}$$

Explanation: As it can be observed from the the equations of lift and conviction, Pr(B) is clearly taken into consideration. Hence, it will not face the suffer from the similar drawback.

Answer to Question 2(b)

- **Confidence:** *Not symmetric*

$$conf(A \rightarrow B) = Pr(B|A) = \frac{Pr(A \cap B)}{Pr(A)}$$

$$conf(B \rightarrow A) = Pr(A|B) = \frac{Pr(A \cap B)}{Pr(B)}$$

  Thus, when Pr(A) $\neq$ Pr(B), confidence is not symmetric.

- **Lift:** *Symmetric*

$$lift(A \rightarrow B) = lift(B \rightarrow A) = \frac{Pr(A \cap B)}{Pr(A).Pr(B)}$$

- **Conviction:** *Not symmetric*

$$conv(A \rightarrow B) = \frac{1 - support(B)}{1 - conf(A \rightarrow B)} = \frac{Pr(A)(\neg Pr(B))}{Pr(A \cap \neg B)}$$

$$conv(B \rightarrow A) = \frac{1 - support(A)}{1 - conf(B \rightarrow A)} = \frac{Pr(B)(\neg Pr(A))}{Pr(B \cap \neg A)}$$

  In other words, conviction is dependent on confidence which is not symmetric. Thus, it is not symmetric.

Example: Baskets = {A, B}, {A, E}, {C, D}

Pr(A) = 2/3, Pr(B) = 1/3, $Pr(A \cap B) = 1/3$

Confidence
$conf(A \rightarrow B) = \frac{1/3}{2/3} = \mathbf{0.50}$
$conf(B \rightarrow A) = \frac{1/3}{1/3} = \mathbf{1}$
$\therefore conf(A \rightarrow B) \neq conf(B \rightarrow A)$

Conviction
$conv(A \rightarrow B) = \frac{1-Pr(B)}{1-conf(A \rightarrow B)} = \frac{1-1/3}{1-1/2} = \frac{2/3}{1/2} = \frac{4}{3}$
$conv(B \rightarrow A) = \frac{1-Pr(A)}{1-conf(B \rightarrow A)} = \frac{1-2/3}{1-1} = \infty$
$\therefore conv(A \rightarrow B) \neq conv(B \rightarrow A)$

Answer to Question 2(c)

Suppose, the rule $A \rightarrow B$ holds 100% of times (perfect implication).

1. **Confidence**: *Desirable*

$$conf(A \rightarrow B) = Pr(B|A) = \frac{Pr(AandB)}{Pr(A)}$$
$$conf(A \rightarrow B) = 1$$

   Thus, maximum value of 1 is achieved for perfect implications.

2. **Lift**: *Not Desirable*

$$\begin{aligned} lift(A \rightarrow B) &= \frac{Pr(AandB)}{Pr(A).Pr(B)} \\ &= \frac{conf(A \rightarrow B)}{Pr(B)} \\ &= \frac{1}{Pr(B)} \end{aligned}$$

   Thus, the value of lift will depend on the Pr(B). It does not maximize for perfect implications.

   Example:
   Baskets = {A, B, C}, {A, B, D}, {X, Y}, {X, Y, Z}, {W, X, Y}
   Perfect Implications: $A \rightarrow B$, $X \rightarrow Y$
   $lift(A \rightarrow B) = \frac{1}{2/5} = 2.5$
   $lift(X \rightarrow Y) = \frac{1}{3/5} = 1.67$
   As observed, the value of lift differs for different perfect implications based on the value of Pr(B).

3. **Conviction**: *Desirable*

$$\begin{aligned} conv(A \rightarrow B) &= \frac{1 - support(B)}{1 - conf(A \rightarrow B)} \\ &= \frac{1 - support(B)}{1 - 1} \\ &= \infty \end{aligned}$$

   Thus, the value of conviction reaches the maximum (Infinity!)

Answer to Question 2(d)

X→Y	**Confidence**
DAI93865 → FRO40251	1.0
GRO85051 → FRO40251	0.999176276771
GRO38636 → FRO40251	0.990654205607
ELE12951 → FRO40251	0.990566037736
DAI88079 → FRO40251	0.986725663717

Answer to Question 2(e)

(X, Y)→Z	**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

Question 3: Locality-Sensitive Hashing

Answer to Question 3(a)

Knowledge:

- Column has n rows with m 1's and therefore (n-m) 0's.
- Number of rows to be selected at random = k
- If none of the selected k rows have 1, result of min hashing is "don't know".

Compute: Probability we get "don't know" as the min-hash value for this column
Approach: This probability is nothing but the ratio of number of ways we can select k rows with only 0's and total ways to choose k rows from n rows.

$$
\begin{aligned}
P(\text{"}don't\ know\text{"}) &= \frac{\binom{n-m}{k}}{\binom{n}{k}} && (1)\\
&= \frac{\frac{(n-m)!}{k!\cdot(n-m-k)!}}{\frac{n!}{k!\cdot(n-k)!}} \\
&= \frac{(n-k)!}{n!} \cdot \frac{(n-m)!}{(n-k-m)!} && (2)\\
&= \frac{(n-k)}{n} \cdot \frac{(n-k-1)}{(n-1)} \cdot \frac{(n-k-2)}{(n-2)} \ldots (m\ terms) && (3)\\
&\leq \left(\frac{n-k}{n}\right)^m && (4)
\end{aligned}
$$

Explanation:

- In Equation 2, (n - k - m)! cancels out with the ending terms of (n - k)! and only the first m terms remain. Similarly, only the first m terms of n! remains.
- In Equation 3, (n - k)/n is the largest ratio. All the subsequent ratios keep decreasing. To find an upper bound, we replace all the subsequent terms with (n - k)/n leading to Equation 4.
- Thus, we get an upper bound as $\left(\frac{n-k}{n}\right)^m$.

Answer to Question 3(b)

We want the probability of "don't know" to be at most e^{-10}. From the previous question we know that, upper bound on probability of don't know is given by $\left(\frac{n-k}{n}\right)^m$.

Some facts:

1. Here, n is much larger than m or k. Thus, the fraction $\frac{n}{k}$ is very large.
2. For large x, $(1 - \frac{1}{x}) \approx 1/e$

$$\begin{aligned}
\left(\frac{n-k}{n}\right)^m &= \left(1 - \frac{1}{\frac{n}{k}}\right)^m \\
&= \left(\left(1 - \frac{1}{\frac{n}{k}}\right)^{n/k}\right)^{\frac{mk}{n}} \\
&= \left(\frac{1}{e}\right)^{\frac{mk}{n}} \quad (\because \text{Using facts 1 and 2}) \\
&= e^{-\frac{mk}{n}} \qquad (5)
\end{aligned}$$

To the probability of "don't know" to be at most e^{-10}, we just need to compare the answer in Equation 5 with e^{-10}.

$$\begin{aligned}
e^{-\frac{mk}{n}} &\leq e^{-10} \\
-\frac{mk}{n} &\leq -10 \\
\frac{mk}{n} &\geq 10
\end{aligned}$$

$$\boxed{k \geq \frac{10n}{m}}$$

Thus, **smallest value of k** that will assure the probability of "don't know" to at most e^{-10} is $\boldsymbol{\frac{10n}{m}}$.

Answer to Question 3(c)

Suppose, the two columns are:

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

For cyclic permutations, the min-hash values of both the columns are:

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

Jaccard similarity of C1 and C2 = $\frac{\mathbf{1}}{\mathbf{2}} = \mathbf{0.5}$
NOTE: If we consider all the possible 3! permutations for computing the min-hash value of columns, then we get the exact value as Jaccard similarity. 3 out of 6 permutations will have matching min-hash values.

From the table, probability that a random cyclic permutation yields the same min-hash value for both C1 and C2 = $\frac{\mathbf{2}}{\mathbf{3}} = \mathbf{0.67}$
This proves that just the cyclic permutations are not enough to estimate the original Jaccard similarity.