CS246: Mining Massive Datasets
Homework 1

Answer to Question 1

Spark pipeline:

1) First I read in the data from the text file and map it twice to get rid of the tab that separates the user and its friends, and to remove the commas between the friends. After this step, user_friend_list_rdd holds a list of elements of the form (userID, [friends])

2) Next, I flatMap user_friend_list_rdd using my make_friend_pair function as the mapping function. make_friend_pair returns a list of pairs of the form ((user, friend), 0) if user and friend are friends else ((user, friend), 1) if they're mutual friends. This gives us all_friends_pairs_rdd, which is full of elements that are made of of pairs of friends, and whether they're actually friends, or if they're possibly mutual friends.

3) At this point, we groupByKey() to get a list of ((user1, user2), [(0 — 1)+]), which are key-value pairs where the key is a pair of users, and the value is a list of 1's and 0's, marking how many mutual friends they have in common. We filter out any pairs where there's a 0 in the value list, since that means the pair are already friends. Next, we map the resulting rdd to sum up the value list to transform the value from a list of 1's to a single int representing how many mutual friends the pair have. We end up with mutual_friends_rdd, whose elements are key-value pairs of the form ((user1, user2), number of mutual friends).

4) Lastly, we flatMap mutual_friends_rdd to get an element for each user (previously only had one element per pair, sorted by userID). After that, we groupByKey and map one more time to get a list of top recommendations.

The following are the requested users and their top recommended friends:
924 : ['11860', '43748', '15416', '6995', '2409', '439', '45881']
8941 : ['8944', '8943', '8940']
8942 : ['8939', '8943', '8940', '8944']
9019 : ['9022', '317', '9023']
9020 : ['9021', '9016', '9022', '9017', '317', '9023']
9021 : ['9020', '9016', '9022', '9017', '317', '9023']
9022 : ['9021', '9019', '9020', '9016', '317', '9017', '9023']
9990 : ['34485', '34642', '13478', '34299', '13877', '13134', '37941']
9992 : ['9989', '9987', '35667', '9991']
9993 : ['9991', '34642', '13877', '13478', '34485', '34299', '13134', '37941']

Answer to Question 2(a)

Confidence ignores $\Pr(B)$ because it only takes into account how often B occurs, given that A is already in the basket. This is a drawback because it doesn't take into account $\Pr(B)$. If $\Pr(B)$ is very high, then regardless of what bundle A you take, $\text{conf}(A \rightarrow B)$ will be high, since B just occurs in most baskets. This makes confidence not as informative of a measure as it could be if it included $\Pr(B)$. Lift and conviction both internalize $\Pr(B)$ because they contain *S(B)* terms, which are equal to $\Pr(B)$. Thus, they take into account the value of $\Pr(B)$, and use it to inform us more about the dependencies between A and B, using more information from the data set.

Answer to Question 2(b)

Confidence
Counterexample.

$$\text{conf}(A \to B) = \Pr(B \mid A) = \frac{\Pr(B \cap A)}{\Pr(A)} \tag{1}$$

$$\text{conf}(B \to A) = \Pr(A \mid B) = \frac{\Pr(A \cap B)}{\Pr(B)} \tag{2}$$

Expression 1 and 2 are equal only if $\Pr(A) = \Pr(B)$. Thus, confidence is not symmetrical. ■

Lift
Proof.
Assuming that $S(X) = \frac{\text{number of baskets containing } X}{\text{number of baskets}} = \Pr(X) = \Pr(\text{basket contains } X)$:

$$\text{lift}(A \to B) = \frac{\text{conf}(A \to B)}{S(B)} = \frac{\Pr(B \mid A)}{\Pr(B)} = \frac{\Pr(B \cap A)}{\Pr(A) * \Pr(B)} \tag{3}$$

$$\text{lift}(B \to A) = \frac{\text{conf}(B \to A)}{S(A)} = \frac{\Pr(A \mid B)}{\Pr(A)} = \frac{\Pr(A \cap B)}{\Pr(B) * \Pr(A)} \tag{4}$$

We also know that

$$\Pr(B \cap A) = \Pr(A \cap B) \tag{5}$$

Combining 3, 4 and 5, we know that $\text{lift}(A \to B) = \text{lift}(B \to A)$, so lift is symmetrical. ■

Conviction
Counterexample.

$$\text{conv}(A \to B) = \frac{1 - S(B)}{1 - \text{conf}(A \to B)} = \frac{1 - \Pr(B)}{1 - \Pr(B \mid A)} = \frac{1 - \Pr(B)}{1 - \frac{\Pr(A \cap B)}{\Pr(A)}} = \frac{1 - \Pr(B)}{\frac{\Pr(A) \ - \ \Pr(A \cap B)}{\Pr(A)}} \tag{6}$$

$$\text{conv}(B \to A) = \frac{1 - S(A)}{1 - \text{conf}(B \to A)} = \frac{1 - \Pr(A)}{1 - \Pr(A \mid B)} = \frac{1 - \Pr(A)}{1 - \frac{\Pr(A \cap B)}{\Pr(B)}} = \frac{1 - \Pr(A)}{\frac{\Pr(B) \ - \ \Pr(A \cap B)}{\Pr(B)}} \tag{7}$$

Expressions 6 and 7 aren't equivalent if $\Pr(A)\ !\ = \Pr(B)$. Thus, conviction is not symmetrical. ■

Answer to Question 2(c)

For the following answers, assume we have an arbitrary perfect implication, $r = A \rightarrow B$.

Confidence
Given the definition of confidence,

$$\text{conf}(r) = \Pr(B \mid A)$$

and our assumption that r is a perfect implication, meaning that the associated conditional probability, $\Pr(B \mid A) = 1$, we know that $\text{conf}(r) = 1$. This is the maximum value for confidence, and it will be achieved by all perfect implications, since perfect implications have conditional probability of 1. For $\Pr(A) = 0$, we would potentially have a case where $\text{conf}(r) > 1$, because $\text{conf}(r) = \Pr(B|A) = \frac{\Pr(A \cap B)}{\Pr(A)}$. But if $\Pr(A) = 0$, then $\Pr(A \cap B) = 0$, so we have $\text{conf}(r) = \frac{0}{0}$, which we can ignore. Thus, confidence is desirable. ■

Lift Given the definition of lift,

$$\text{lift}(r) = \frac{\text{conf}(r)}{S(B)}$$

we see that by setting $\text{conf}(r) = \Pr(B|A) = 1$ because r is a perfect implication, we get

$$\text{lift}(r) = \frac{1}{S(B)}$$

which means it only reaches its maximum value if $S(B) = 0$. We know that $S(B)$ cannot equal 0 in this case because that would prevent $\text{conf}(r)$ from being equal to 1. Thus, lift is not desirable. ■

Conviction
Given the definition of conviction,

$$\text{conv}(r) = \frac{1 - S(B)}{1 - \text{conf}(r)}$$

we see that by setting $\text{conf}(r) = \Pr(B \mid A) = 1$ because r is a perfect implication, we get

$$\text{conv}(r) = \frac{1 - S(B)}{1 - 1}$$

where the denominator evaluates to 0. Thus, $\text{conv}(r) = \infty$, which is conviction's maximum value. Thus, conviction is desirable. ■

Answer to Question 2(d)

1. conf(DAI93865 → FRO40251) = 1.0
2. conf(GRO85051 → FRO40251) = 0.999176276771
3. conf(GRO38636 → FRO40251) = 0.990654205607
4. conf(ELE12951 → FRO40251) = 0.990566037736
5. conf(DAI88079 → FRO40251) = 0.986725663717

Answer to Question 2(e)

1. conf((DAI31081, GRO85051) → FRO40251) = 1.0
2. conf((DAI55911, GRO85051) → FRO40251) = 1.0
3. conf((DAI75645, GRO85051) → FRO40251) = 1.0
4. conf((GRO21487, GRO85051) → FRO40251) = 1.0
5. conf((GRO85051, SNA80324) → FRO40251) = 1.0

Answer to Question 3(a)

We know that there are n rows, m of which are 1's. When we choose k of those n rows, the result of the minhashing is "don't know" if there are no 1's in that group of k rows.
$\frac{n-k}{n}$ represents the probability that a given row wasn't chosen to be part of the k rows. If all m rows containing 1's are not part of the k chosen rows, we have a result of "don't know." This corresponds to the following probability:

$$\text{P("don't know")} = \prod_{i=0}^{m-1} (\frac{n-k-i}{n-i})$$

which we can expand to be:

$$\text{P("don't know")} = \prod_{i=0}^{m-1} (1 - \frac{k}{n-i})$$

We know that for $i < j$, the following holds: $(1 - \frac{k}{n-i}) > (1 - \frac{k}{n-j})$.
Since we know there are m terms in the original product, we know that:

$$\text{P("don't know")} = \prod_{i=0}^{m-1} (1 - \frac{k}{n-i}) \leq (1 - \frac{k}{n})^m$$

Thus, we know that P("don't know") $\leq (\frac{n-k}{n})^m$. ■

Answer to Question 3(b)

Let $(\frac{n-k}{n})^m$ = the exact probability of "don't know."
We know that for very large x, $(1-\frac{1}{x})^x \approx \frac{1}{e}$, so we have $(1-\frac{1}{x})^{10x} \approx \frac{1}{e^{10}}$.
We want Pr("don't know") to be at most e^{-10}, so we set $(\frac{n-k}{n})^m \leq e^{-10}$.
We substitute $m = 10x$, then let $\frac{k}{n} = \frac{1}{x}$
Rearranging terms, we get:

$$\begin{aligned}
&(\frac{n-k}{n})^m \\
&= (1-\frac{k}{n})^m \\
&= (1-\frac{1}{x})^{10x} \\
&= \frac{1}{e^{10}}
\end{aligned}$$

Thus, we have $k = \frac{n}{x} = \frac{n}{\frac{m}{10}} = \frac{10n}{m}$. ■

Answer to Question 3(c)

a)

$S1$	$S2$
1	0
1	0
1	0
0	0
0	0
0	0
0	0
1	1

b) Jaccard Similarity$(S1, S2) = |S1 \cap S2| \ / \ |S1 \cup S2| = \frac{1}{4}$

c) If the cyclic permutation method chooses any of rows 4-8 as the starting row, then it finds row 8 as the first 1 in both $S1$ and $S2$. Picking rows 1-3 as the starting point results in different minhash values for $S1$ and $S2$. Thus, Pr(min(S1) = min(S2), using cyclic permutations) $= \frac{5}{8}$, which is not the same value we got for Jaccard Similarity.

Answer to Question 4(a)

Let $W_j = \{x \in A | g_j(x) = g_j(z)\}$ $(1 \leq j \leq L)$
Let $T = \{x \in A | d(x, z) > c\lambda\}$
Prove $\Pr[\sum_{j=1}^{L} |T \cap W_j| \geq 3L] \leq \frac{1}{3}$
Using Markov's Inequality, we have $\Pr(X \geq 3L) \leq \frac{E[X]}{3L}$, where $X = \sum_{j=1}^{L} |T \cap W_j|$
We need to prove, therefore, that $\frac{E[X]}{3L} \leq \frac{1}{3}$, or equivalently that $E[X] \leq L$
$E[X]$ is equal to the expected number of x's such that $x \in T \cap W_j$ for some $1 \leq j \leq L$. We know that $\Pr(x \in T \cap W_j) = p_2^k$
By linearity of expectation, we have $E[X] = L * \sum_{i=1}^{|T|} p_2^k$
We know that $k = log_{1/p_2}(n)$, so:

$$\begin{aligned} p_2^k &= p_2^{log_{1/p_2}(n)} \\ &= p_2^{log_{p2}(n)/log_{p2}(1/p_2)} \\ &= p_2^{log_{p2}(n)/(-1)} \\ &= p_2^{log_{p2}(1/n)} \\ &= \frac{1}{n} \end{aligned}$$

Substituting into our original equation, we have $E[X] = L * \sum_{i=1}^{|T|} \frac{1}{n} = L\frac{|T|}{n}$, where we know that $|T| \leq n$, so we have $E[X] = L\frac{|T|}{n} \leq L$.
Thus, $E[X] \leq L$, so we've proven that $\Pr[\sum_{j=1}^{L} |T \cap W_j| \geq 3L] \leq \frac{1}{3}$. ■

Answer to Question 4(b)

Let $x^* \in A$ be a point such that $d(x^*, z) \leq \lambda$. Prove:

$$\Pr[\forall 1 \leq j \leq L, g_j(x*) \neq g_j(z)] \leq \frac{1}{e}$$

We're trying to prove that the $\Pr(x^*$ misses all L sets of g hash sets$) < \frac{1}{e}$
We know that $\forall x^*,\ h \in H$, $\Pr(h(x^*) = h(z)) \geq p_1$.
Thus:

$$\begin{aligned}
\Pr(h(x^*) = h(z)) &\geq p_1 \\
\Pr(g_j(x^*) = g_j(z)) &\geq p_1^k \\
1 - \Pr(g_j(x^*) = g_j(z)) &< 1 - p_1^k \\
\Pr[\forall 1 \leq j \leq L, g_j(x*) \neq g_j(z)] &< (1 - p_1^k)^L
\end{aligned}$$

We're done once we prove that $(1 - p_1^k)^L \leq \frac{1}{e}$, because that shows then we have:

$$\begin{aligned}
\Pr[\forall 1 \leq j \leq L, g_j(x*) \neq g_j(z)] &< (1 - p_1^k)^L \leq \frac{1}{e} \\
\Pr[\forall 1 \leq j \leq L, g_j(x*) \neq g_j(z)] &< \frac{1}{e}
\end{aligned}$$

We know that $(1 - \frac{1}{x})^x \leq \frac{1}{e}$. If we set $x = L$ and substitute, we get $(1 - p_1^k)^x$, so if we show that $p_1^k = \frac{1}{L}$, we'll have $(1 - p_1^k)^L = (1 - \frac{1}{x})^x \leq \frac{1}{e}$, which is what we set out to prove.
We're given:

$$p = \frac{log(1/p_1)}{log(1/p_2)} \tag{8}$$

$$k = log_{1/p_2}(n) \tag{9}$$

$$L = n^p \rightarrow \frac{log_{1/p_2} L}{p} = log_{1/p_2}(n) = k \tag{10}$$

Now, we'll take p_1^k and substitute to find that $p_1^k = \frac{1}{L}$

$$\begin{aligned}
p_1^k &= p_1^{log_{1/p_2}(n)} \\
&= p_1^{\frac{log_{1/p_2} L}{p}} \\
&= p_1^{\frac{log_{1/p_2} L}{log(1/p_1)/log(1/p_2)}} \\
&= p_1^{[log_{p1} L/log_{p1}(1/p_2)]/[log_{p1}(1/p_1)/log_{p1}(1/p_2)]} \\
&= p_1^{[log_{p1} L]/[log_{p1}(1/p_1)]} \\
&= p_1^{log_{p1} L/(-1)} \\
&= p_1^{-log_{p1} L} \\
&= \frac{1}{L} \quad \blacksquare
\end{aligned}$$

Answer to Question 4(c)

Answer to Question 4(d)

LSH Average Search Time: 1.04381232262
L1 Average Search Time: 19.2510334253

In comparing the attached images, it seems that the algorithm is able to recognize that they, for the most part, have a light section on the right half of the image, and a dark section on the left half of the image.

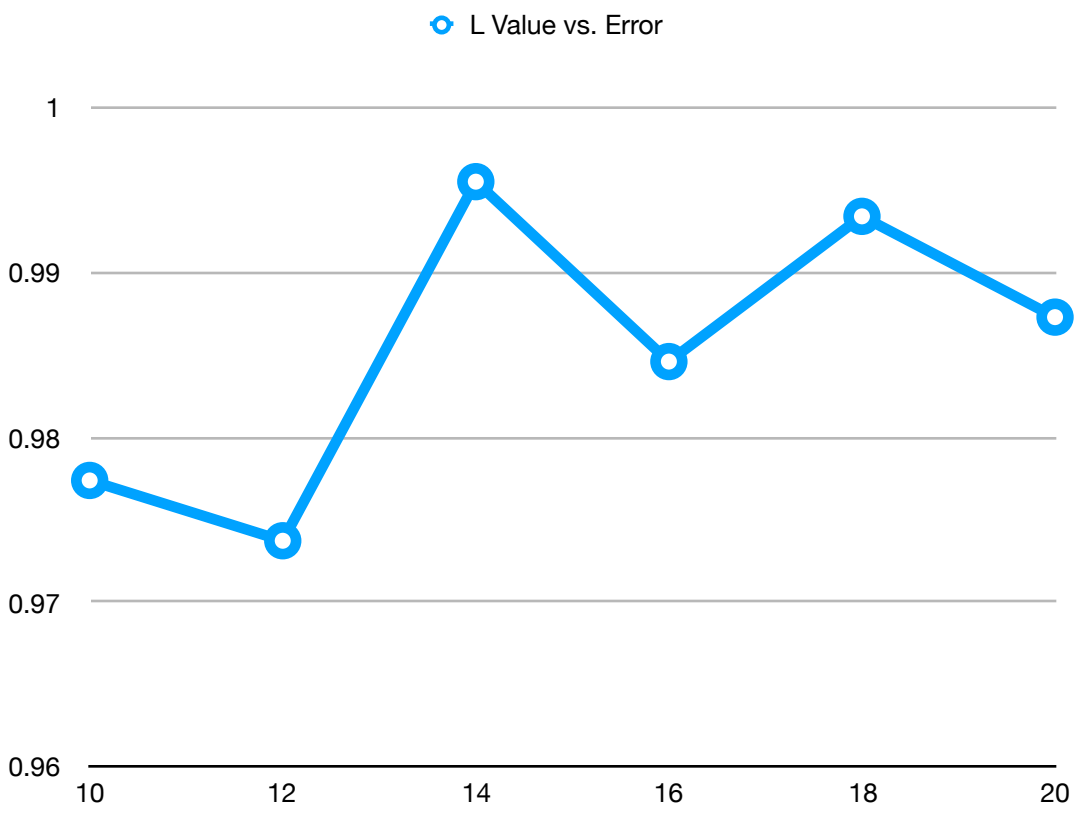

Fig. 1. This graph shows the relationship between L value and resulting error. As you can see, there seems to be an upward trend; as L value gets closer to 20, the error value increases.

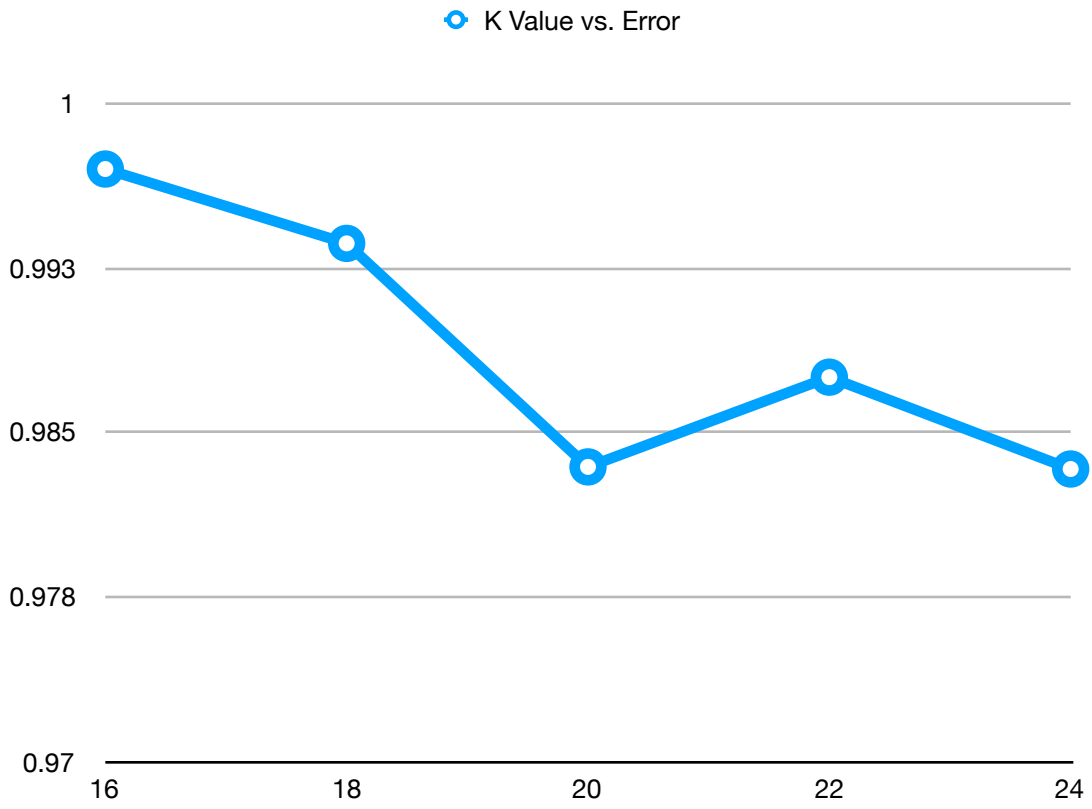

Fig. 2. This graph shows the relationship between K value and error. As you can see, as the K value increases, the error value tends to decrease.

Plot 100

Plot 7464

Plot 8196

Plot 7551

Plot 21780

Plot 12444

Plot 25289

Plot 25549

Plot 28251

Plot 22509

Plot 28351

Submission Instructions

Assignment Submission Include a signed agreement to the Honor Code with this assignment. Assignments are due at 11:59pm. All students (SCPD and non-SCPD) must submit their homeworks via GradeScope (`http://www.gradescope.com`). Students can typeset or scan their homeworks. Make sure that you answer each question on a separate page. Students also need to upload their code at `http://snap.stanford.edu/submit`. Put all the code for a single question into a single file and upload it. Please do not put any code in your GradeScope submissions unless explicitly asked to do so.

Late Day Policy Each student will have a total of *two* free late periods. *One late period extends deadlines until the day of the next lecture.* (Homeworks are usually due on Thursdays, which means the first late periods expires on the following Tuesday.) However, no assignment will be accepted more than *one* late period after its due date.

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 very 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.

(Signed)