CSE547: Machine Learning for Big Data
Homework 1

Answer to Question 1

(2).
spark pipeline: There are mainly three steps:
first, we get the combinations of the usr and his/her friend, and use value 0 to denote that they are already friend. Then get all the combinations of usr's friends, and use value 1 to denote that they have 1 common friend, i.e. usr.
Second, we reduce the RDD: if two users are already friends, then their value is 0; else, if they have common friends, then their value is the count of common friends.
Third, for usr, we drop all the pairs containing usr with value = 0 and rank the top 10 pairs according to the value of count.
(3).
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)

The reason why that it ignores Pr(B) is a problem is that it ignores the situation that A and B might be independent, i.e. the occurrence of A is irrelevant to that of B. In this case, conf(A $\rightarrow$ B) = Pr(B|A) = P(B) (A independent of B). So if P(B) is large and thus conf(A $\rightarrow$ B) is also large. We will consider A to B as an effective rule according to Confidence. However, other measures, lift and conviction, consider P(B) and thus avoid this problem.

Answer to Question 2(b)

Lift is symmetrical.
pf: lift(A $\rightarrow$ B) = $\frac{conf(A \rightarrow B)}{S(B)} = \frac{Pr(B|A)}{S(B)} = \frac{Pr(A,B)}{Pr(A)Pr(B)}$ (easily see that this equation is symmetrical about A and B, and thus,) = lift(B $\rightarrow$ A).
Confidence is not symmetrical.
counterexample: assume A, B are independent and thus Pr(A) $\neq$ Pr(B). Then from the Question 2(a), we have: conf(A $\rightarrow$B)$\neq$conf(B$\rightarrow$A).
Conviction is not symmetrical.
counterexample: first we assume Pr(A)=Pr(B) but # {A appears without B} $\neq$ #{B appears without A}. Conviction compares the probability that A appears without B if they were independent with the actual frequency of the appearance of A without B. When 'A$\rightarrow$B' is changed to 'B$\rightarrow$A', probability that A appears without B if they were independent = probability that B appears without A if they were independent. But if actual frequency of the appearance of A without B $\neq$ actual frequency of the appearance of B without A, then the conv(A$\rightarrow$B) $\neq$ conv(B$\rightarrow$A)

Answer to Question 2(c)

Assume A$\rightarrow$B is a perfect implication, that is, every time A occurs B will also occur. Then, obviously, Conf(A$\rightarrow$ B) = Pr(B|A).
if S(B) $\neq$ 1, conv(A$\rightarrow$ B) = infinity, else conv(A$\rightarrow$ B) = $\frac{0}{0}$
So, confidence and conviction are desirable, but for lift, if depends the value of S(B). If B occurs when A does not occur, S(B) may be large, leading to small lift(A$\rightarrow$ B).

Answer to Question 2(d)

DAI93865⇒FRO40251 1.0
GRO85051⇒FRO40251 0.99917627677100
GRO38636⇒FRO40251 0.9906542056074766
ELE12951⇒FRO40251 0.9905660377358491
DAI88079⇒FRO40251 0.9867256637168141

Answer to Question 2(e)

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)

For selected k rows, the probability that a column only contains 0 is $\binom{n-k}{m}/\binom{n}{m} = \frac{(n-k)!(n-m)!}{n!(n-k-m)!} = \frac{(n-k)(n-k-1)...((n-k-m+1)(n-k-m)!(n-m)!}{n(n-1)...(n-m+1)(n-m)!(n-k-m)!} = \frac{(n-k)(n-k-1)...((n-k-m+1)}{n(n-1)...(n-m+1)}$. Both the denominator and numerator have m terms and each term are smaller than $\frac{n-k}{n}$. And thus we obtain the final result.

Answer to Question 3(b)

Since $(1-\frac{1}{x})^x \approx \frac{1}{e}$ for large x, $(\frac{n-k}{n})^m = (1-\frac{k}{n})^{\frac{n}{k}*\frac{k}{n}m} \approx (\frac{1}{e})^{\frac{km}{n}}$ for large n. Therefore, we should make $\frac{km}{n} \geq 10$, or equivalently, k $\geq \frac{10n}{m}$

Answer to Question 3(c)

Denote the document matrix as $\begin{pmatrix} 1 & 1 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}$. And the similarity is 1/2. However, the signature matrix M is $\begin{pmatrix} 1 & 1 \\ 3 & 3 \\ 2 & 3 \end{pmatrix}$. Thus we have the probability that a random cyclic permutation yields the same min-hash value for both S1 and S2 is 2/3, which is not equal to 1/2.

Answer to Question 4(a)

Since $\mathcal{H}$ is $(\lambda, c\lambda, p_1, p_2)$-sensitive, thus for x_1, x_2 with $d(x_1, x_2) > c\lambda$, we have $\Pr\{h_i(x_1) = h_i(x_2)\} < p_2$. Therefore, for x $\in$ T$\cap W_j$, we have $h_{ij}(x_1) = h_ij(x_2)$ for i=1,...,k. Thus $\Pr(x \in T \cap W_j) = \Pr(h_{ij}(x_1) = h_{ij}(x_2)$ for i=1,...,k) $\leq P_2^k = 1/$n. Therefore, take expectation and we have E[$|T \cap W_j|$] $\leq$ n*1/n = 1. Then by Markov's inequality, Pr[$\sum_{j=1}^{L} |T \cap W_j| \geq 3L$] $\leq$ L*E[$|T \cap W_j|$]/(3L) = 1/3.

Answer to Question 4(b)

For x^* s.t. d(x^*, z)$\leq \lambda$, we have Pr($h_i(x^*) = h_i(z) \geq p_1$. Therefore, Pr($g_i(x^*) = g_i(z)) \geq p_1^k$ and hence Pr($g_i(x^*) \neq g_i(z)) \leq 1 - p_1^k$.
Since L $= n^\rho = n^{\frac{klog(1/p_1)}{klog(1/p_2)}} = n^{\frac{klog(1/p_1)}{log(n)}}$, which gives 1/L $= n^{\frac{klog(p_1)}{log(n)}} = n^{log_n(p_1^k)} = p_1^k$.
Hence, $Pr(g_i(x^*) \neq g_i(z)) \leq 1 - 1/L$, which indicates $Pr(\forall i, g_i(x^*) \neq g_i(z)) \leq (1 - 1/L)^L \leq 1/e$ (since it's an increasing function in [1,Inf] and its limit is 1/e.

Answer to Question 4(c)

If the reported data point, say, w, is not an actual (c,λ)-ANN. And there are two possibilities. First, all the points which hash to the same buckets as the query point are not (c,λ)-ANN, and therefore when we retrieve these points from L buckets to which the query point hashes, we cannot get (c,λ)-ANN points at all. The second possibility is that there are indeed (c,λ)-ANN points in the buckets where the query point hashes, but there are more than 3L data points which are not (c,λ)-ANN points in these L buckets. So when we retrieve 3L data points from the L buckets, we may miss the (c,λ)-ANN point but however pick 3L non-(c,λ)-ANN points.

Note that W_j j=1,...,L represents the bucket to which x and z hash. In the first situation, all the points in W_j except z for every j are not what we need. In part (b), x^* is a (c,λ)-ANN point. Then in the first situation, it does not belong to any W_j, which means $\forall i, g_i(x^*) \neq g_i(z)$), and this possibility is smaller than 1/e. So the possibility that the set of all (c,λ)-ANN points and W_j are disjoint is smaller than 1/e.

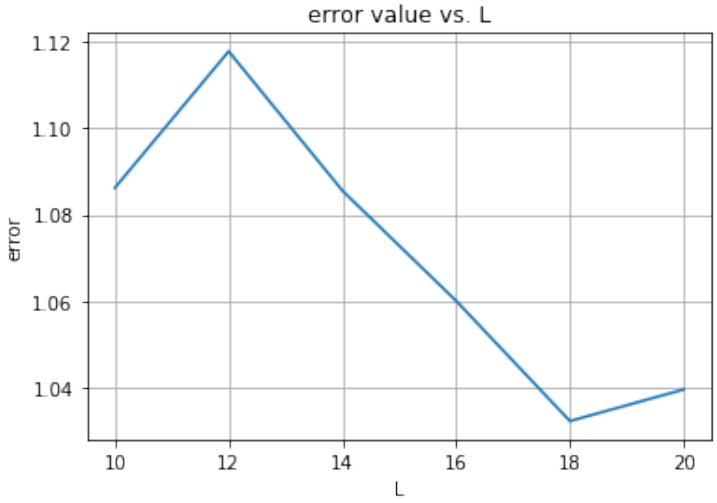
And the possibility of the second situation is smaller than 1/3 from part (a).

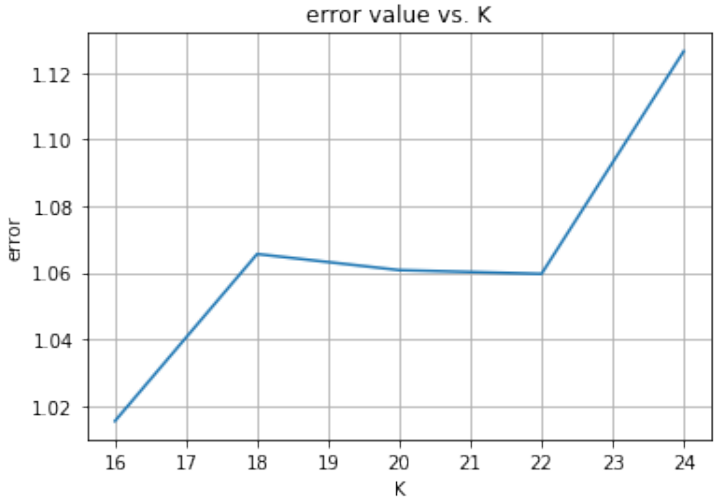
Then the possibility that the reported data point is not what we need is smaller than 1/3+1/e. So with probability greater than 1-1/3-1/e, the reported data point is an actual(c,λ)-ANN.

Answer to Question 4(d)

(1).The average time for LSH and for linear search is 2.2012181282043457 and 8.691458940505981, respectively. We do not take the time for LSH setup into account when computing the running time for LSH.

(2). Below are two plots for error value vs. L and error value vs. K, respectively.

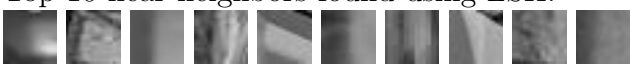




Generally speaking, the error values increase when k get larger, and decrease when L gets larger. But however, the error values are quite close to 1 (actually no larger than 1.2) for all considered k and L. In other words, the data points we get from LSH are quite close to the true nearest neighbour points.

(3). Image patch in row 100: 

Top 10 near neighbors found using LSH:



Top 10 near neighbors found using linear search:

Conclusion: the two methods will provide similar visual effect. And under this randomness, it find the true nearest point.

Continued Answer to Question 4(d)

Submission Instructions

Assignment Submission All students should submit their assignments electronically via GradeScope. Students may typeset or scan their **neatly written** homeworks (points **will** be deducted for illegible submissions). Simply sign up on the Gradescope website and use the course code 97EWEW. Please use your UW NetID if possible.

For the non-coding component of the homework, you should upload a PDF rather than submitting as images. We will use Gradescope for the submission of code as well. Please make sure to tag each part correctly on Gradescope so it is easier for us to grade. There will be a small point deduction for each mistagged page and for each question that includes code. Put all the code for a single question into a single file and upload it. Only files in text format (e.g. .txt, .py, .java) will be accepted. **There will be no credit for coding questions without submitted code on Gradescope, or for submitting it after the deadline**, so please remember to submit your code.

Late Day Policy All students will be given two no-questions-asked late periods, but only one late period can be used per homework and cannot be used for project deliverables. A late-period lasts 48 hours from the original deadline (so if an assignment is due on Thursday at 11:59 pm, the late period goes to the Saturday at 11:59pm Pacific Time).

Honor Code We take honor code extremely seriously:
(`https://www.cs.washington.edu/academics/misconduct`).
We strongly encourage students to form study groups. Students may discuss and work on homework problems in groups. However, each student must write down the solutions and the code independently. In addition, each student should write down the set of people whom they interacted with.

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.

*(Ruojin HE)*_____