CSE547: Machine Learning for Big Data
Homework 1

Answer to Question 1

First, I created an RDD with the user ID's as keys and the list of that ID's friends as value. Having in mind that two friends A,B of a user C are at least second-degree friends, I obtained all possible permutations over each user ID's friend list, flatMapped them and subtracted all those pairs that are already friends, leaving me with a large list of exactly second-degree friends. Making use of the fact that ID's like A and B could appear as second-degree friends several times (say if they are both friends with C and D), I used the counting method we used in HW0 to obtain how often exactly two ID's appear as second-degree friends, then re-sorted my key-value pairs such that I had one ID as key and then the pair (second degree friend ID, count) as value and then output the N second-degree friends with the highest count per ID - or all second-degree friends, if an ID had less than N. To account for existing ID's without any second-degree friends, I re-added all ID's as keys with empty lists as values, grouped by ID and then returned the first list in its corresponding value, returning a friend list for people with second-degree friends and the empty list for people with no second-degree friends. For the given user ID's, I obtained the following 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)

Ignoring the Pr(B) is insofar problematic, as that conf$(A \to B) = \frac{\mathrm{P}(B \cap A)}{\mathrm{P}(A)}$ depends increasingly less on B the more frequent B has been observed (and hence the more probable B is). In the worst case, B is so abundant that it is almost always observed in combination with any other product, leading to $\frac{\mathrm{P}(B \cap A)}{\mathrm{P}(A)} \approx \frac{\mathrm{P}(A)}{\mathrm{P}(A)} = 1$ for every A. But then the confidence is not able to distinguish the case where B is overall rare, but actually frequently bought alongside A (which is an interesting observation) and the case where B is just abundant and therefore also frequently bought alongside A (an uninteresting case) . This is the 'milk' example we had in lecture. Lift alleviates this issue by 'lifting' cases where the confidence is high, but B is also a rare item, making it stand out (cases where B is abundant will still linger around 1). Conviction goes a slightly different but similar route. The denominator pushes the value up, the higher the confidence is, whereas the numerator pushes the value down, the more probable B is overall. So just like in lift, cases where confidence is high and B is also rare will stand out.

Answer to Question 2(b)

Confidence: Let A='Dice throw value is below 4', B='Dice throw value is 2' for a single dice throw. Then $\text{conf}(A \to B) = P(B|A) = \frac{1}{3} \neq 1 = P(A|B) = \text{conf}(B \to A)$. Hence confidence is not symmetric.

Lift: Note that S(B) is the empirical probability of B appearing in a given basket. We will therefore assume S(B)=P(B). Then

$$\text{lift}(A \to B) = \frac{\text{conf}(A \to B)}{P(B)} = \frac{P(B|A)}{P(B)} = \frac{P(B \cap A)}{P(A)P(B)} = \frac{P(A|B)}{P(A)} = \frac{\text{conf}(B \to A)}{P(A)} = \text{lift}(B \to A)$$

Therefore, lift is symmetric.

Conviction: We use the same example from above. Note that $P(A) = \frac{1}{2}, P(B) = \frac{1}{6}$.Then

$$\begin{aligned} \text{conv}(A \to B) &= \frac{1 - \frac{1}{6}}{1 - \frac{1}{3}} = \frac{5}{4} \\ conv(B \to A) &= \lim_{x \to P(A|B),\ x \leq P(A|B)} = \frac{1 - \frac{1}{2}}{1 - x} = \infty \end{aligned}$$

using the fact that $P(A|B) = 1$. We could use the limit from below since $P(A|B) \leq 1$. Therefore, conviction is also not symmetric.

Answer to Question 2(c)

Confidence: It holds $\text{conf}(A \to B) = P(B|A) \leq 1$ and $\text{conf}(A \to B) = 1 \Leftrightarrow P(B|A) = 1$. Hence confidence is desirable.
Lift: Let $P(B|A) = 1, P(B) = 1$. Then $\text{lift}(A \to B) = 1$. But for P(B)=.5, we have $\text{lift}(A \to B) = 2$. Hence in the first case, the measure did not reach its maximum despite $A \to B$ being a perfect implication. Therefore, it is not desirable
Conviction: Assume a perfect implication, i.e. $P(B|A) = \text{conf}(A \to B) = 1$. Then $\text{conv}(A \to B) = \lim_{x \to P(B|A),\ x \leq P(B|A)} = \frac{1-P(B)}{1-x} = \infty$ for any arbitrary fixed $P(B) \in [0, 1)$. Hence conviction is also desirable.

Answer to Question 2(d)

- 'DAI93865'⇒ 'FRO40251', Confidence: 1.0
- 'GRO85051' ⇒ 'FRO40251', Confidence: 0.999176276771005
- 'GRO38636' ⇒ 'FRO40251', Confidence: 0.9906542056074766
- 'ELE12951' ⇒ 'FRO40251', Confidence: 0.9905660377358491
- 'DAI88079' ⇒ 'FRO40251', Confidence: 0.9867256637168141

Answer to Question 2(e)

- ’DAI23334’, ’ELE92920’ ⇒ ’DAI62779’, Confidence: 1.0
- ’DAI31081’, ’GRO85051’ ⇒ ’FRO40251’, Confidence: 1.0
- ’DAI55911’, ’GRO85051’ ⇒ ’FRO40251’, Confidence: 1.0
- ’DAI62779’, ’DAI88079’ ⇒ ’FRO40251’, Confidence: 1.0
- ’DAI75645’, ’GRO85051’ ⇒ ’FRO40251’, Confidence: 1.0

Answer to Question 3(a)

We first prove the following. For $x \leq y,\ x, y \in \mathbb{N}^+ \setminus \{0\}$ it holds:

$$\frac{x}{y} \leq \frac{x+n}{y+n} \text{ for all } n \in \mathbb{N}^+ \tag{1}$$

Let x, y, n as above. Then

$$\begin{align}
\frac{x}{y} &\leq \frac{x+n}{y+n} \tag{2}\\
\Leftrightarrow x(y+n) &\leq y(x+n) \tag{3}\\
\Leftrightarrow xy + nx &\leq xy + ny \tag{4}\\
\Leftrightarrow nx &\leq ny \tag{5}
\end{align}$$

which holds true because we assumed $x \leq y$. We used the fact that $x, y, n \geq 0$ in Equation (3).

Now, all possible ways to draw k from the n-m zeros is $\binom{n-m}{k}$, whereas all possible draw possibilities is $\binom{n}{k}$. First, it is clear that if the number of ones $m = 0$, the probability of drawing k zeros is 1. Hence in the following, assume $m > 0$. Then the probability of drawing k zeros is

$$\begin{aligned}
\frac{\binom{n-m}{k}}{\binom{n}{k}} &= \frac{(n-m)!(n-k)!}{n!(n-m-k)!} = \frac{(n-k) \cdot \ldots \cdot (n-m-k+1)}{n \cdot \ldots \cdot (n-m+1)}\\
&= \underbrace{\frac{n-k}{n} \cdot \ldots \cdot \frac{n-m-k+1}{n-m+1}}_{\text{m terms}} \overset{(1)}{\leq} \underbrace{\frac{n-k}{n} \cdot \ldots \cdot \frac{n-k}{n}}_{\text{m terms}}\\
&= (\frac{n-k}{n})^m
\end{aligned}$$

We could use Equation (1), since it holds $m - 1 \geq 0$ and so even for the smallest term in the product we get $\frac{n-m-k+1}{n-m+1} \leq \frac{n-m-k+1+(m-1)}{n-m+1+(m-1)} = \frac{n-k}{n}$.

Answer to Question 3(b)

Using the upper bound from before, it is sufficient to find a k that satisfies

$$(\frac{n-k}{n})^m \leq e^{-10} \Leftrightarrow \frac{n-k}{n} \leq e^{-10/m}$$
$$\Leftrightarrow k \geq n - ne^{-10/m} = n(1-e^{-10/m}) = n(1-(e^{-1})^{10/m}) = n(1-(1-\frac{1}{m})^{10})$$

Hence k needs to be at least $(1-(1-\frac{1}{m})^{10})$.

Answer to Question 3(c)

Consider the following columns:

0	1
0	0
1	1

Then the Jaccard similarity is $\frac{1}{2}$. But when using the minhash variant over cyclic premuations only, we get the following permutations

1	1
0	1
0	0

0	0
1	1
0	1

0	1
0	0
1	1

and the corresponding signature matrix:

1	1
2	2
3	1

Hence, when allowing only cyclic permutations, the probability that their minhash values agree is $\frac{2}{3} \neq \frac{1}{2}$.

Answer to Question 4(a)

Since $\sum_{j=1}^{L} |T \cap W_j|$ is a sum over magnitudes and hence a non-negative random variable and $L \geq 0$, we can use Markov's inequality. Furthermore, each $h_i, i \in \{1, \ldots, k\}$ is i.i.d. sampled from $\mathcal{H}$. Also, each $g_j, j \in \{1, \ldots, L\}$ is randomly and independently drawn from G and hence i.i.d. This means that the $W_j, j \in \{1, \ldots, L\}$ are identically distributed, which will we use in (*):

$$P(\sum_{j=1}^{L} |T \cap W_j| \geq 3L) \leq \frac{\mathbb{E}[\sum_{j=1}^{L} |T \cap W_j|]}{3L} = \frac{\sum_{j=1}^{L} \mathbb{E}[|T \cap W_j|]}{3L} \overset{(*)}{=} \frac{L\mathbb{E}[|T \cap W_1|]}{3L}$$
$$= \frac{\mathbb{E}[|T \cap W_1|]}{3}$$

Further, with our assumptions of the hash functions being i.i.d., we get

$$x \in T \Leftrightarrow d(x,z) > c\lambda \overset{(\lambda, c\lambda, p_1, p_2)-s.}{\Rightarrow} P(g_1(x) = g_1(z)) = P(\bigcap_{i \in \{1,\ldots,k\}} h_i(x) = h_i(z))$$
$$= \prod_{i=1}^{k} P(h_i(x) = h_i(z)) = \prod_{i=1}^{k} P(h_1(x) = h_1(z)) \leq (p_2)^k$$

So each of the n points in $\mathcal{A}$ has a probability of p_2^k to be in $|T \cap W_1|$. Note that the following holds:

$$k = \log_{1/p_2}(n) \Leftrightarrow (\frac{1}{p_2})^k = n \Leftrightarrow p_2^k = \frac{1}{n}$$

Hence

$$P(\sum_{j=1}^{L} |T \cap W_j| \geq 3L) \leq \frac{\mathbb{E}[|T \cap W_1|]}{3} = \frac{np_2^k}{3} = \frac{n\frac{1}{n}}{3} = \frac{1}{3}$$

Answer to Question 4(b)

Each $h_i, i \in \{1, \ldots, k\}$ is drawn i.i.d. from $\mathcal{H}$. Also, each $g_j, j \in \{1, \ldots, L\}$ is randomly and independently drawn from G and hence i.i.d. Therefore,

$$\begin{aligned}
&P(g_1(x^*) \neq g_1(z), \ldots, g_L(x^*) \neq g_L(z)) = P(g_1(x^*) \neq g_1(z)) \cdot \ldots \cdot P(g_L(x^*) \neq g_L(z)) \\
=&P(g_1(x^*) \neq g_1(z))^L = (1 - P(g_1(x^*) = g_1(z)))^L \\
=&(1 - P(h_1(x^*) = h_1(z), \ldots, h_k(x^*) = h_k(z)))^L \\
=&(1 - (P(h_1(x^*) = h_1(z)) \cdot \ldots \cdot P(h_k(x^*) = h_k(z))))^L \\
=&(1 - P(h_1(x^*) = h_1(z))^k)^L
\end{aligned}$$

Using the fact that $\mathcal{H}$ is $(\lambda, c\lambda, p_1, p_2)$-sensitive, we know that since $d(x^*, z) \leq \lambda$ then $P(h_i(x^*) = h_i(z)) \geq p_1$ for all $i \in \{1, \ldots, k\}$. Hence

$$(1 - P(h_1(x^*) = h_1(z))^k)^L \leq (1 - p_1^k)^L$$

Further, note

$$\begin{aligned}
k = \log_{1/p_2}(n) \Leftrightarrow (\frac{1}{p_2})^k = n \Leftrightarrow k \log(\frac{1}{p_2}) = \log(n) \Leftrightarrow \log(\frac{1}{p_2}) = \frac{\log(n)}{k} \\
\text{and } \rho = \frac{\log(1/p_1)}{\log(1/p_2)} \Leftrightarrow \log(\frac{1}{p_2}) = \frac{\log(1/p_1)}{\rho} \\
\Rightarrow \frac{\log(1/p_1)}{\rho} = \frac{\log(n)}{k} \Leftrightarrow k \log(1/p_1) = \rho \log(n) \Leftrightarrow (\frac{1}{p_1})^k = n^\rho \Leftrightarrow p_1^k = \frac{1}{n^\rho} = \frac{1}{L}
\end{aligned}$$

Putting all together, we get:

$$P(g_1(x^*) \neq g_1(z), \ldots, g_L(x^*) \neq g_L(z)) \leq (1 - p_1^k)^L = (1 - \frac{1}{L})^L \overset{(L \text{ finite})}{<} \frac{1}{e}$$

which was to show.

Answer to Question 4(c)

We know that

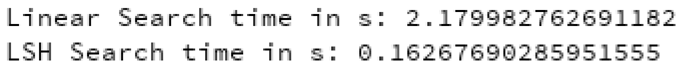
$$P(\sum_{j=1}^{L} |T \cap W_j| \geq 3L) \leq \frac{1}{3} \Leftrightarrow P(\sum_{j=1}^{L} |T \cap W_j| \leq 3L) \geq \frac{2}{3}$$

Meaning the probability of having less than 3L undesirable points (points that are more than $c\lambda$ away) across the L buckets z mapped to is bigger than $\frac{2}{3}$. If we can pick $3L$ points across the L buckets, the chance of getting a (c, λ)-ANN is therefore at least $\frac{2}{3}$. If not (i.e. if there are less than 3L points across the L buckets), then b) tells us that with probability larger than $1 - \frac{1}{e}$, the optimal nearest neighbor is in one of the buckets of our query point. Since there are less than 3L points across the buckets, we retrieve all points in the buckets, including the closest one. Hence the probability of getting a (c,λ)-ANN is larger than $1 - \frac{1}{e}$ in that case.

Answer to Question 4(d)

(I) The average search times found are depicted in Figure 1. No numpy was used for the linear search.

(II) As can be seen in Figure 2, for a fixed L, a higher k implies a higher error rate.



```
Linear Search time in s: 2.179982762691182
LSH Search time in s: 0.16267690285951555
```

Figure 1: Times for linear and lsh search.

Conversely, for a fixed k, a higher L tends to result in a lower error rate. Note that the hash results are not deterministic, hence a little noise is to be expected.

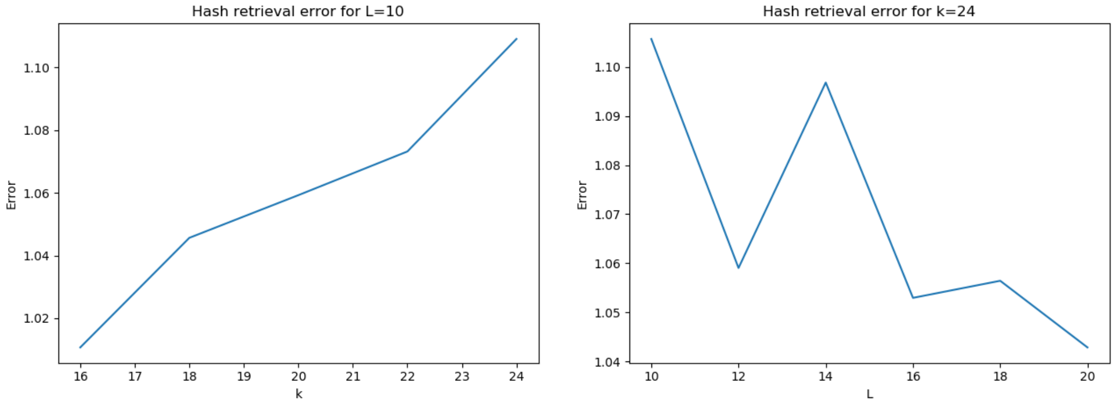


Figure 2: Error rates for constant L, constant k, respectively.

Continued Answer to Question 4(d)

(III) From a visual comparison, the results from the linear search seem to be closer to the original image: Many of them have the (vaguely) horizontal strides and a change from bright in the middle to bottom right to darker towards the other three corners. For LSH, some images seem to miss the strides (especially the first two as seen in Figure 3), but some of them actually have the same resemblances as in linear search, which is almost horizontal strides, with a change from brighter to dark on the top and bottom left. Interestingly, a lot of them seem to have a vertical or horizontal darker section vs. a generally lighter area in the rest of the picture.

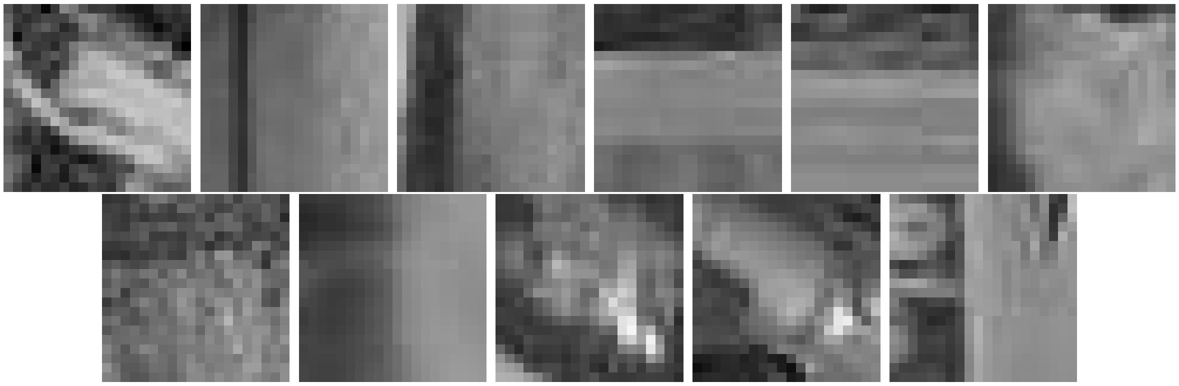


Figure 3: Top Left: Original Image. Rest: Nearest Neighbors as found by LSH

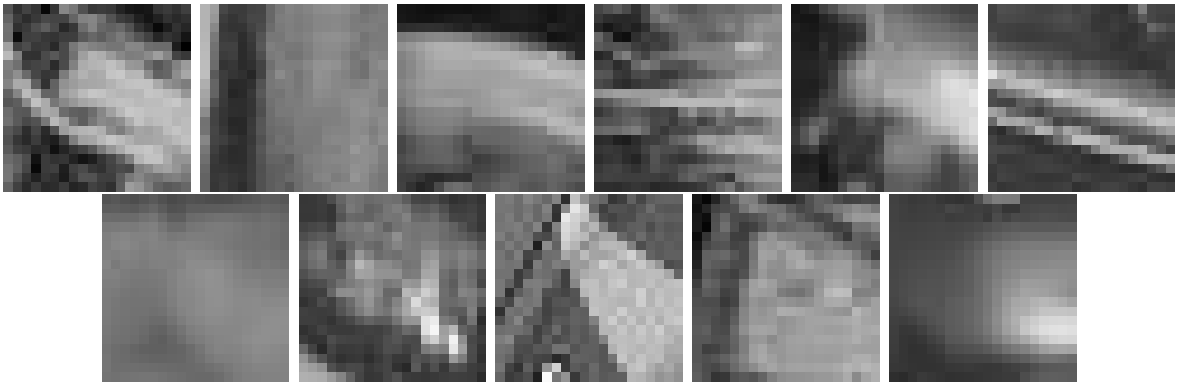


Figure 4: Top Left: Original Image. Rest: Nearest Neighbors as found by Linear Search.

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):

Romain Camilleri, Andrew Wagenmaker, Nick Nuechterlein, Erin Wilson, Galen Weld.

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

None

I acknowledge and accept the Honor Code.

Emil Azadian