CSCI 1520: Algorithmic Aspects of Machine Learning (Spring 2025)

Written Assignment 1

Due at 11:59pm ET, Thursday, Feb 20

1. (5 points) The Jaccard similarity between two sets S and T is defined as $\text{sim}(S,T) = \frac{|S \cap T|}{|S \cup T|}$. Recall that MinHash provides a family $\mathcal{F}$ of functions such that, for any S and T,

$$\Pr\left[h(S) = h(T)\right] = \text{sim}(S,T) \ ,$$

where h is chosen uniformly at random from $\mathcal{F}$.

Suppose we want to find pairs of sets with Jaccard similarity at least 0.5. Let $\text{sim}(S_1,T_1) = 0.6$ and $\text{sim}(S_2,T_2) = 0.4$. For each algorithm below, compute the probability that the algorithm incorrectly outputs "no" for (S_1,T_1) (false negative) and the probability that the algorithm incorrectly outputs "yes" for (S_2,T_2) (false positive). In these algorithms, each function h_i is sampled independently and uniformly at random from $\mathcal{F}$.

(Round your answer to 2 decimal places. For this question, you may use Mathematica, Wolfram Alpha, or similar software/websites without explicitly mentioning them.)

(1) A single hash function: output "yes" iff $h_1(S) = h_1(T)$.

(2) A 5-way AND followed by a 30-way OR: output "yes" iff

$$\bigvee_{i=0}^{29} \left(\bigwedge_{j=1}^{5} \big(h_{5i+j}(S) = h_{5i+j}(T)\big) \right) \ .$$

(3) A 5-way OR followed by a 30-way AND: output "yes" iff

$$\bigwedge_{i=0}^{29} \left(\bigvee_{j=1}^{5} \big(h_{5i+j}(S) = h_{5i+j}(T)\big) \right) \ .$$

2. (5 points) We will construct a family of locality-sensitive functions for the *angular distance*. The angular distance between two points $x, y \in \mathbb{R}^n$ is the normalized angle between them:

$$d(x,y) = \frac{1}{\pi} \cos^{-1}\left(\frac{x^\top y}{\|x\|_2 \|y\|_2} \right) \ .$$

Let $v \in \mathbb{R}^n$ be a unit vector drawn uniformly at random.

(1) Let $n = 2$, $x = (1,0)$, and $y = (1,\sqrt{3})$.
Compute $d(x,y)$. Compute the probability that $v^\top x$ and $v^\top y$ have different signs.

(2) Prove that for any $n \geq 2$, $x \neq 0$, and $y \neq 0$, we have

$$\Pr[v^\top x \text{ and } v^\top y \text{ have the same sign}] = 1 - d(x, y) \ .$$

(3) Given Part (2), construct a family $\mathcal{F}$ of functions for angular distance such that $\mathcal{F}$ is $(d_1, d_2, 1 - d_1, 1 - d_2)$-sensitive for any $0 \leq d_1 < d_2 \leq 1$.

Definition (Locality-Sensitive Functions)**.** Let d be a metric and let $d_1 < d_2$. A family $\mathcal{F}$ of functions is (d_1, d_2, p_1, p_2)-sensitive if, for f drawn uniformly at random from $\mathcal{F}$:

- If $d(x, y) \leq d_1$, then $\Pr[f(x) = f(y)] \geq p_1$.
- If $d(x, y) \geq d_2$, then $\Pr[f(x) = f(y)] \leq p_2$.

3. (4 points) Bloom filters provide a space-efficient way to test set membership: Let S be a set of m elements and let B be an n-bit array. Let $(h_i)_{i=1}^k$ be k hash functions, each of which maps an element to one of the n positions independently and uniformly at random.

   Initially, all bits in B are set to 0. Then for each $a \in S$, we set $B[h_i(a)]$ to 1 for all $1 \leq i \leq k$. To test if an element x is in the set S, we check the bits $B[h_i(x)]$ for all $1 \leq i \leq k$, and return "yes" iff these bits are all 1.

   (1) Consider a variant of Bloom filter where B is divided into k disjoint parts, one for each hash function. Suppose n is a multiple of k, and each hash function maps an element to one of $\frac{n}{k}$ positions, i.e., $0 \leq h_i(x) \leq \frac{n}{k} - 1$ for all i and x.
   For each $a \in S$, we set $B[\frac{n}{k}(i-1) + h_i(a)]$ to 1 for all $1 \leq i \leq k$. To test if an element x is in the set S, we return "yes" iff the bits $B[\frac{n}{k}(i-1) + h_i(x)]$ are all 1 for $1 \leq i \leq k$.
   What is the false positive rate of this approach?
   What is the limit of the false positive rate when $\frac{n}{m}$ and k are fixed and $m \to \infty$?

   (2) Suppose we want to allow the set S to change over time. Can we generalize Bloom filters to allow adding and removing elements from S?
   (Hint: What if each position of B is an integer instead of a single bit?)

4. (1 bonus point) Consider the same setting as in Question 1. Compute the false negative and false positive probabilities on input (S_1, T_1) and (S_2, T_2) for the following algorithm:

   A majority rule over 99 hash functions: output "yes" iff

   $$|\{1 \le i \le 99 : h_i(S) = h_i(T)\}| \ge 50 .$$

5. (1 bonus point) Prove that any deterministic algorithm for computing a (1 ± 0.1)-approximation of the number of distinct elements in an n-element data stream must use $\Omega(n)$ bits of space.

   (Hint: Suppose Alice has the first $\frac{n}{2}$ numbers and Bob has the last $\frac{n}{2}$ numbers. The following claim might be useful.)

   Claim 1. *For all even $m \ge 2$, there exists a set of bit strings $S \subseteq \{0,1\}^m$ such that:*

   - *Every bit string in S has exactly $b = \frac{m}{2}$ ones.*
   - *Any two strings in S have at most $\frac{b}{2}$ overlapping ones.*
   - *$|S| \ge 2^{cm}$ for some universal constant $c > 0$.*

6. (1 bonus point) Consider the problem of counting the number of distinct elements in a data stream. Let $1 \le a_1, \ldots, a_n \le m$ denote the first n elements in the data stream. Let d denote the number of distinct elements in $(a_1, \ldots, a_n)$.

   Consider the following algorithm. Algorithm **??** hashes every element in the stream, maintains the t smallest distinct hash values, and then uses the t-th smallest hash value to estimate d.

   We write $[n]$ for $\{1, \ldots, n\}$.

Algorithm 1: Estimating the number of distinct elements.

Input : $m \ge 10, n \ge 1, 0 < \epsilon < 1$, and a stream of n numbers $1 \le a_1, \ldots, a_n \le m$.
Output: an estimation of the number of distinct elements in $(a_1, \ldots, a_n)$.
Let $M = m^3$ and $t = \frac{40}{\epsilon^2}$.
Suppose h is a hash function that maps $[m]$ to $[M]$ uniformly at random.
Initialize $S \leftarrow \varnothing$.
for $i = 1$ **to** n **do**
    if $h(a_i) \notin S$ **then**
        if $|S| < t$ **then**
            $S \leftarrow S \cup \{h(a_i)\}$.
        else
            If $h(a_i)$ is smaller than the largest number in S, replace that number with $h(a_i)$.
if $|S| < t$ **then**
    return $|S|$.
else
    Let v be the largest number in S.
    return $\widetilde{d} = \frac{tM}{v}$.

(1) Assume that h can be stored for free and $h(x)$ can be evaluated in $O(1)$ time. Show that implementing Algorithm **??** using (balanced) binary search trees requires $O(\frac{\log m}{\epsilon^2})$ bits of space, and each iteration of the for loop runs in time $O(\log m \log(1/\epsilon))$.

Next, we will prove one side of the correctness of Algorithm **??**: $\Pr[\widetilde{d} > (1+\epsilon)d] \leq \frac{1}{10}$ (over the randomness in h).

The event $\frac{tM}{v} = \widetilde{d} > (1+\epsilon)d$ happens iff $v < \frac{tM}{(1+\epsilon)d}$. In other words, for $\widetilde{d} > (1+\epsilon)d$ to happen, there must be at least t hash values that are less than $\frac{tM}{(1+\epsilon)d} \leq (1-\frac{\epsilon}{2})\frac{tM}{d}$.

Because the output of Algorithm **??** does not depend on the order of the elements, we can assume w.l.o.g. that $a_1, \ldots, a_d$ are the d distinct elements. Let $X_i \in \{0,1\}$ be the indicator random variable for the event $h(a_i) \leq (1-\frac{\epsilon}{2})\frac{tM}{d}$. Let $Y = \sum_{i=1}^{d} X_i$.

That is, Y is the total number of hash values that are below the threshold, and we want to upper bound $\Pr[Y \geq t]$.

(2) Prove that $\mathbb{E}[X_i] \leq (1-\frac{\epsilon}{2})\frac{t}{d}$ and $\mathbb{E}[Y] \leq (1-\frac{\epsilon}{2})t$.

(3) Prove that $\mathrm{var}[X_i] \leq (1-\frac{\epsilon}{2})\frac{t}{d} \leq \frac{t}{d}$ and $\mathrm{var}[Y] \leq t$.

(Hint: You can use the following facts without proving them.)

- For two random variables X_1 and X_2, we have $\mathbb{E}[X_1 + X_2] = \mathbb{E}[X_1] + \mathbb{E}[X_2]$.
- For two independent random variables X_1 and X_2, $\mathrm{var}[X_1 + X_2] = \mathrm{var}[X_1] + \mathrm{var}[X_2]$.

Consequently, by Chebyshev's inequality, Parts (3) and (4), and the choice of $t = \frac{40}{\epsilon^2}$,

$$\Pr[\widetilde{d} > (1+\epsilon)d] \;\leq\; \Pr[Y \geq t] \;\leq\; \Pr\Big[|Y - \mathbb{E}[Y]\,| > \frac{\epsilon t}{2}\Big] \leq \frac{\mathrm{var}[Y]}{(\frac{\epsilon t}{2})^2} \leq \frac{4}{\epsilon^2 t} \leq \frac{1}{10}\,.$$