COMPSCI 514: Problem Set 2

Due: 10/11 by 11:59pm in Gradescope.

Instructions:

- You are allowed to work on this problem set in a group of up to three members.
- You should choose your group from within your own class (either online or in-person).
- You may talk to members of other groups at a high level about the problems but **not work through the solutions in detail together**.
- Each group should **submit a single solution set**: one member should upload a pdf to Gradescope, marking the other members as part of their group in Gradescope.
- You must show your work/derive any answers as part of the solutions to receive full credit.

Core Competency Problems

1. Bloom Filters with Efficient Hash Functions (10 points)

In the Bloom filter analysis in class, we assume use of a fully independent hash function. Here we will analyze a variant of Bloom filters that just uses 2-universal hashing.

Consider a Bloom filter variant consisting of k bit arrays: $A_1, \ldots, A_k$, each of length m, along with k 2-universal hash functions $h_1, \ldots, h_k : U \to [m]$. Assume the hash functions are chosen independently of each other. To insert an item x, we mark $A_i[h_i(x)] = 1$ for all $i \in [k]$. To query if an item x is in the dataset, we check if $A_i[h_i(x)] = 1$ for all $i \in [k]$ and return 'YES' if this condition is true.

1. (2 points) Let x be some item that has not been inserted into the filter. Give an upper bound on $\Pr[A_i[h_i(x)] = 1]$ as a function of the number of inserted items n and the number of bits in the array m.

2. (2 points) Use the above to give an upper bound on the false positive rate of the filter, as a function of n, m, and k.

3. (2 points) The total space complexity used by the filter is $s = m \cdot k$. Given a fixed space budget $s > 0$, prove that the optimal setting of k (which minimizes the false positive rate upper bound from part (2)) is $k = \frac{1}{e} \cdot \frac{s}{n}$. **Note:** As with standard Bloom filters, this optimal setting may not be an integer.

4. (2 points) Using the above optimal setting of k, to store n items with false positive rate δ in this data structure, how many bits of space do you need? Give your answer without using big-O notation, i.e., explicitly calculate the leading constant. **Note:** Do your computations using the exactly optimal setting of k, even if it is not an integer.

5. (2 points) Compare the above bound to what you would get using the standard Bloom filter analysis in class assuming a false positive rate of $\left(1 - e^{\frac{-kn}{m}}\right)^k$. Is the leading constant on the space usage better or worse? **Note:** Be careful about the bases of your logarithms, as which base you use will affect the leading constants.

2. Approximating the Median in a Data Stream (8 points)

Given a set $S \subset [n]$ of m distinct values and a value x, we define

$$rank_S(x) := |\{y \in S : y \leq x\}|$$

i.e., the number of values in S that are less or equal to x. We say x is an ϵ-approximate median if

$$(1/2 - \epsilon)m \leq rank_S(x) \leq (1/2 + \epsilon)m \ .$$

1. (2 points) Consider the following algorithm for sampling an element from a stream $x_1, x_2, \ldots, x_m$ where you may assume throughout this question that all values in the stream are distinct:

   (a) Initialize $s \leftarrow x_1$

   (b) For $i = 1, 2, \ldots, m$: with probability $1/i$ update $s \leftarrow x_i$.

   (c) Return s

   Prove that at the end of the stream, s is equally likely to be any of the elements in the stream, i.e., s is chosen uniformly at random from the set of elements in the stream. Note that this method doesn't need to know the value of m in advance.

2. (2 points) Consider sampling r elements uniformly and independently at random (with replacement) from the stream and let Z_t be the random variable corresponding to the number of samples that are less or equal to z_t where z_t is the t-th smallest element in the stream. Compute the expectation and variance of Z_t.

3. (2 points) Consider an algorithm that samples r elements uniformly and independently at random (with replacement) from the data stream and returns the median of the sampled elements. How large must r be such that the output of this algorithm is an ϵ-approximate median with probability at least 99/100? You may assume that $\epsilon < 1/4$ and give your answer in big-O notation. **Hint:** Consider the random variables $Z_{(1/2-\epsilon)m}$ and $Z_{(1/2+\epsilon)m}$.

4. (2 points) Another way to achieve uniform sampling is, for each $i \in [m]$, to randomly pick a value y_i is uniformly from $[0, 1]$. Then the stream element x_i where $i = \arg\min_j y_j$ is drawn uniformly at random from the set $\{x_1, x_2, \ldots, x_m\}$. However, suppose at the end of the stream we are given a value $s \in [m]$ and now need to return a random value in the set $\{x_s, x_{s+1}, \ldots, x_m\}$. It suffices to return x_i where $i = \arg\min_{s \leq j \leq m} y_j$. Describe an algorithm that uses $O(\log m)$ space in expectation to output $\arg\min_{s \leq j \leq m} y_j$. The algorithm does not know s while processing the stream.

3. Designing Locality Sensitive Hash Functions (10 points)

1. (2 points) The Hamming distance $H(x, y)$ between two bit strings $x, y \in \{0, 1\}^n$ is the number of positions in which they differ. Describe a locality sensitive hash function for the Hamming distance with collision probability $\Pr[h(x) = h(y)] = 1 - H(x, y)/n$.

2. (2 points) Consider interpreting the bit strings as the sets of locations in which they contain ones. I.e., x corresponds to the set $\{i : x(i) = 1\}$. How different can the MinHash collision probability be from the collision probability obtained in part (1)? I.e., what is the maximum possible value of $|\Pr[MH(x) = MH(y)] - (1 - H(x, y)/n)|$?

3. (2 points) The weighted Jaccard similarity between two sets $A, B \subset U$ is defined as

$$J_w(A, B) = \frac{\sum_{x \in A \cap B} w(x)}{\sum_{x \in A \cup B} w(x)},$$

where $w : U \to [W]$ is some weight function that assigns each element to an integer weight in $1, \ldots, W$, indicating its importance. Describe a locality sensitive hash function with collision probability $Pr[h(A) = h(B)] = J_w(A, B)$.

4. (2 points) For two vectors $x, y \in \mathbb{R}^n$, design a locality sensitive hash function with collision probability $\Pr[h(x) = h(y)] = \frac{\sum_{i=1}^{n} \min(x_i, y_i)}{\sum_{i=1}^{n} \max(x_i, y_i)}$. You may assume that x and y have non-negative, finite precision, and bounded entries. I.e., each entry has value $i/2^{16}$ for some $i \in \{0, 1, \ldots, 2^{16}\}$.

5. (2 points) Let $d : U \times U \to [0, 1]$ be any distance function mapping pairs of elements to a distance in the range $[0, 1]$. Prove that, for there to exist a locality sensitive hash function h with collision probability $\Pr[h(x) = h(y)] = 1 - d(x, y)$, then d must satisfy the triangle inequality. I.e., for all x, y, z we must have $d(x, z) \leq d(x, y) + d(y, z)$. **Hint:** Rewrite $d(x, y) = \Pr[h(x) \neq h(y)]$.

Challenge Problems (Complete 1 of 2)

C1. A Different Approach for Distinct Elements (10 points) 🌶🌶

Let D denote the number of distinct elements in a stream of m elements $x_1, \ldots, x_m$ where each $x_\ell \in [n]$. Let $h_1, \ldots, h_k : [n] \to [n]$ be k independent hash functions where each hash function is fully independent. For $i \in [k]$ and $g \in \{1, 2, 4, 8, 16 \ldots, 2^{\lceil \log_2 n \rceil - 1}\}$, compute:

$$c_{i,g} = |\{j \in [m] : h_i(x_j) \leq g\}|$$

Let $\alpha_g = |\{i \in [k] : c_{i,g} = 0\}|/k$.

1. (2 points) Compute the expected value of α_g as a function of D, n, and g.

2. (2 points) Prove $k = O(\gamma^{-2} \log n)$ suffices to ensure that with probability at least 0.99,

$$\forall g \ , \qquad |\alpha_g - \mathbb{E}[\alpha_g]| \leq \gamma$$

3. (2 points) Assuming that $n \geq cD$ for some sufficiently large constant c, prove that there exists $g \in \{1, 2, 4, 8, \ldots, \}$ such that $0.8 \leq \mathbb{E}[\alpha_g] \leq 0.905$. **Hint:** You way want to use the inequality $1 - xy \leq (1 - x)^y \leq e^{-xy}$ for $x, y > 0$.

4. (2 points) Prove that if

$$0.8 - \gamma \leq \alpha_g \leq 0.905 + \gamma$$

then $\ln(\alpha_g)/\ln(1-g/n)$ is a $1+O(\gamma)$ approximation of D. You may assume $\gamma < 0.05$. **Hint:** You may want to use the fact that for a function f,

$$f(y) - |x-y|\tau \leq f(x) \leq f(y) + |x-y|\tau$$

where τ is any upper bound the absolute value of the derivative of f between x and y.

5. (2 points) Explain how it is possible to compute α_g for all g in $O(k \log\log n)$ space. You need not account for the space used to store and evaluate the hash functions.

C2. Testing Stream Properties (10 points)

Consider a stream of the form $x_1, \ldots, x_m$ where each $x_j = (b_j, a_j) \in \{-1, 1\} \times [n]$, i.e., each element in the stream is a pair of values where the first value is either 1 or -1 and the second value is an integer between 1 and n. Let f_i be the number of pairs of the form $(1, i)$ minus the number of terms of the form $(-1, i)$. E.g., for a stream

$$(1,2), (1,4), (-1,2), (1,4), (1,3), (1,1)$$

$f_1 = 1, f_2 = 0, f_3 = 1$, and $f_4 = 2$. Let $h_1, \ldots, h_k : [n] \to \{-1, 1\}$ be k independent hash functions where each hash function is fully independent. For $i \in [k]$, compute:

$$c_i = \sum_{j \in [m]} b_j h_i(a_j)$$

i.e., c_i is initialized to 0 and then when processing $x_j = (b_j, a_j)$ we update $c_i \leftarrow c_i + b_j h_i(a_j)$.

1. (2 points) Write an expression for c_i in terms of $f_1, f_2, \ldots, f_n$ and $h_i(1), \ldots, h_i(n)$. Prove that if there exists some ℓ such that $f_\ell \neq 0$ then $\Pr[c_i \neq 0] \geq 1/2$.

2. (2 points) We say a stream is *self-cancelling* if $f_1 = f_2 = \ldots = f_n = 0$. Design a data stream algorithm using $O(\log 1/\delta)$ space that determines if a stream is self-cancelling with probability at least $1 - \delta$. **Hint:** Consider computing $\max_{i \in [k]} |c_i|$ for some value of k.

3. (2 points) We say a stream is *uniform* if $f_1 = f_2 = \ldots = f_n$ (but they don't need to equal 0). Design a data stream algorithm using $O(\log 1/\delta)$ space that determines if a stream is uniform with probability at least $1 - \delta$. **Hint:** Consider computing c_i and $d_i = \sum_{\ell \in [n]} f h_i(\ell)$ where $f = (f_1 + f_2 + \ldots + f_n)/n$.

4. (2 points) We say a stream if *t-sparse* if at most t of the values $f_1, f_2, \ldots, f_n$ are non-zero. Design an algorithm using $O(t \cdot \log(mn) + \log(1/\delta))$ space that computes all the values $f_1, f_2, \ldots, f_n$ if the stream is t-sparse, with probability at least $1 - \delta$.

5. (2 points) If a stream is t-sparse and all f_i are non-negative, prove that the Count-Min sketch can be used to compute all f_i exactly in $O(t \log(n/\delta))$ space with probability at least $1-\delta$. You may assume fully random hash functions (although it actually suffices to still use 2-universal hash functions).