COMS 4995 (Randomized Algorithms): Problem Set #2

Due by 11:59 PM on Wednesday, October 9, 2019

Instructions:

(1) Form a group of 1-3 students. You should turn in only one write-up for your entire group.

(2) Submission instructions: We are using Gradescope for the homework submissions. Go to www.gradescope.com to either login or create a new account. Use the course code 9D6V5E to register for this class. Only one group member needs to submit the assignment. When submitting, please remember to add all group member names in Gradescope. See the course Web site for detailed instructions.

(3) Please type your solutions if possible and we encourage you to use the LaTeX template provided on the course home page.

(4) Write convincingly but not excessively.

(5) Some of these problems are difficult, so your group may not solve them all to completion. In this case, you can write up what you've got (subject to (3), above): partial proofs, lemmas, high-level ideas, counterexamples, and so on.

(6) Except where otherwise noted, you may refer to your lecture notes and the specific supplementary readings listed on the course Web page *only*. You can also review any relevant materials from your undergraduate algorithms course. If you do use any approved sources, make you sure you cite them appropriately, and make sure that all your words are your own.

(7) You can discuss the problems verbally at a high level with other groups. And of course, you are encouraged to contact the course staff (via Piazza or office hours) for additional help.

(8) If you discuss solution approaches with anyone outside of your group, you must list their names on the front page of your write-up.

(9) Refer to the course Web page for the late day policy and the School of Engineering honor code.

Problem 7

(14 points) The goal of this problem is to devise a two-level hashing scheme that has excellent worst-case performance (rather than just good average-case performance over the choice of hash function). To make this doable, we will assume that the data set S to be hashed is *known in advance*.

(a) (4 points) Let $\mathcal{H}$ be a universal family of hash functions (see Exercise #11), mapping a large set U of items to the n buckets $\{0, 1, 2, \ldots, n-1\}$. Let $S \subseteq U$ contain m elements. Suppose that a hash function h is chosen uniformly at random from $\mathcal{H}$ and used to hash all of the elements of S. Prove that the expected number of collisions — the number of unordered pairs $x, y \in S$ of distinct elements with $h(x) = h(y)$ — is less than $m^2/2n$. (Note the expectation here is over the random choice of $h \in \mathcal{H}$.)

(b) (4 points) Explain how (a) and Markov's inequality (Exercise #3) lead to a good randomized algorithm for finding a hash function $h \in \mathcal{H}$ that hashes all of S while suffering less than m^2/n collisions. (Your algorithm should be Las Vegas (see Problem #5) and have expected running time polynomial in m and n. You can assume that a random hash function can be chosen in polynomial time, and that every hash function can be evaluated in polynomial time.) As a special case, what does this imply when we are willing to tolerate a quadratic blow-up in space (taking $n = m^2$)?

(c) (6 points) Here's how we retain constant-time worst-case search performance while using only $O(m)$ space. We first pick an "outer" hash table with m buckets. By part (b), we can compute efficiently a hash function h^* that hashes all of S with a total of at most m collisions (note that when $n = m$ we have $m^2/n = m$); fix such a hash function for the rest of the problem.

Now, for each bucket $i \in \{1, 2, \ldots, m\}$ of the outer hash table, let A_i denote the elements of S that h^* hashes to i. Denote $|A_i|$ by a_i. For each i, we allocate an "inner" hash table with a_i^2 buckets. Again, part (b) implies that we can compute efficiently a hash function $h_i : U \to \{0, 1, 2, \ldots, a_i^2 - 1\}$ so that no elements of A_i collide in this inner hash table. Note that this two-level scheme guarantees constant-time lookup: given an element x, we first compute the hash value $h^*(x)$, and then fetch the (unique) element in the bucket $h_{h^*(x)}(x)$ of the inner hash table corresponding to $h^*(x)$. (Unsuccessful searches are similarly easy.) It remains to prove that this scheme requires only linear space.

Prove that the total space

$$m + \sum_{i=1}^{m} a_i^2$$

required by the two-level scheme is $O(m)$. Here "space" refers to the sum of the array lengths of all of the hash tables used (e.g., we're ignoring the number of bits needed to actually describe each object of S).

Problem 8

(15 points) The point of this problem is to outline a simple construction of a small family of 4-wise independent hash functions, as required by the AMS F_2 streaming algorithm (Lecture #6). Let $n = |U|$ be the universe size, let $\mathbb{F}$ be a finite field with 2^r elements, with $r \in \mathcal{N}$ and $n < |\mathbb{F}| \leq 2n$.[1] Associate the elements of U with n distinct elements from $\mathbb{F}$ (arbitrarily). The key property we need of fields is *polynomial interpolation*: given target points $(x_1, y_1), \ldots, (x_d, y_d)$ with all x_i's and y_i's in $\mathbb{F}$ (and with the x_i's distinct), there is a unique degree-$(d-1)$ polynomial $p(x)$ (with coefficients in $\mathbb{F}$) that satisfies $p(x_i) = y_i$ for all $i = 1, 2, \ldots, d$.[2]

(a) (5 points) As a warm up, for a pair $(a, b) \in \mathbb{F}^2$ of coefficients, define

$$h_{ab}(x) = ax + b$$

for $x \in U$, where all operations are in the field $\mathbb{F}$. Prove that the family $\mathcal{H} = \{h_{ab} : a, b \in \mathbb{F}^2\}$ is pairwise independent, meaning that for every distinct pair $x, y \in U$ and every image $z, w \in \mathbb{F}$,

$$\mathbf{Pr}[h_{ab}(x) = z \text{ and } h_{ab}(y) = w] = \mathbf{Pr}[h_{ab}(x) = z] \cdot \mathbf{Pr}[h_{ab}(y) = w] = \frac{1}{|\mathbb{F}|^2}. \quad (1)$$

(b) (5 points) For a 4-tuple $(a, b, c, d) \in \mathbb{F}^4$ of coefficients, define

$$h_{abcd}(x) = ax^3 + bx^2 + cx + d,$$

where all operations take place in the field $\mathbb{F}$. Let $\mathcal{H}$ denote the set of all $|\mathbb{F}|^4$ such functions. Prove that $\mathcal{H}$ is a 4-wise independent family, meaning that the analog of (1) with four values (inputs x_1, x_2, x_3, x_4 and outputs y_1, y_2, y_3, y_4) holds with "$1/|\mathbb{F}|^2$" replaced by "$1/|\mathbb{F}|^4$."

[1] Recall that in a *field* you can add, subtract, multiply, and divide by non-zero numbers. The real and rational numbers are familiar examples. (The integers are not an example, because they are not closed under division by non-zero elements.) There are also fields with a finite number of elements. For example, if p is prime, then the set $\{0, 1, 2, \ldots, p-1\}$ forms a field with the operations defined by modular arithmetic; division is possible because each non-zero element has a multiplicative inverse. (E.g., if $p = 7$, $2^{-1} = 4$ (since $2 \cdot 4 \equiv 1$ modulo 7), $3^{-1} = 5$, $4^{-1} = 2$, $5^{-1} = 3$, and $6^{-1} = 6$.) It turns out that for every prime power p^r (and only for these), there exists a unique finite field (up to relabeling of the elements) with p^r elements.

[2] For example, the proof using Lagrange polynomials works for any field.

(c) (5 points) Define $g_{abcd}(x)$ as +1 if $h_{abcd}(x)$ is an even and -1 otherwise.[3] Prove that $\mathcal{G} = \{g_{abcd} : a, b, c, d \in \mathbb{F}\}$ is 4-wise independent, meaning that for all distinct $x_1, x_2, x_3, x_4 \in U$ and all $z_1, z_2, z_3, z_4 \in \{\pm 1\}$,

$$\mathbf{Pr}_{g \in \mathcal{G}}[g(x_i) = z_i \text{ for } i = 1, 2, 3, 4] = \frac{1}{16}.$$

[Note: describing a function of $\mathcal{H}$ or $\mathcal{G}$ requires only $O(\log n)$ bits, for the four coefficients $a, b, c, d \in \mathbb{F}$. The evaluation of such a hash function can also be carried out with a logarithmic amount of space.]

Problem 9

(25 points) Recall the heavy hitters setup from Lecture #5. There is a known universe U of size n, and a data stream of m elements $x_1, \ldots, x_m$ that arrive one-by-one. Let f_j denote the number of occurrences of an element $j \in U$ in the stream. The goal in this problem is to compute estimates of all the f_j's while using a small amount of space.[4] Consider the following variation of the Count-Min-Sketch algorithm from Lecture #5, where instead of always incrementing a counter on an insert, we increment *or decrement* a counter, with the sign determined by a second hash function:[5]

1. Initialize an array A of b counters to 0 (where b is a parameter to be set later).
2. Let h be a random oracle from U to $\{1, 2, \ldots, b\}$.
3. Let g be a random oracle from U to $\{-1, +1\}$.
4. For $i = 1, 2, \ldots, m$:
   (a) $A[h(x_i)] := A[h(x_i)] + g(x_i)$.

After the stream has been processed, the data structure's estimate X_j of the frequency of an element $j \in U$ is defined as $g(j)$ times the final value of $A[h(j)]$.

(a) (5 points) For every $j \in U$, prove that X_j is an unbiased estimator (no matter what b is):

$$\mathbf{E}[X_j] = f_j.$$

(The expectation is over the choice of the random oracles h and g.)

(b) (5 points) For every $j \in U$, prove that

$$\text{Var}[X_j] \leq \frac{F_2}{b},$$

where $F_2 = \sum_{j \in U} f_j^2$ denotes the second frequency moment.

(c) (5 points) Choose b (as a function of the user-specified δ and ϵ) so that the following guarantee holds: for every $j \in U$, with probability at least $1 - \delta$,

$$|X_j - f_j| \leq \epsilon \cdot \|\mathbf{f}\|_2,$$

where $\|\mathbf{x}\|_2 = \sqrt{\sum_i x_i^2}$ denotes the ℓ_2-norm of a vector and $\mathbf{f}$ denotes the vector of frequencies (i.e., the f_j's).

(d) (3 points) The analysis in parts (a)–(c) assumed that the hash functions were random oracles. Prove that the exact same analysis holds with much weaker assumptions on the hash functions.

(e) (3 points) What is the space usage of your algorithm? (Assume that each hash function can be stored and evaluated in $O(\log n)$ space.)

[3] Don't worry about the exact definition of "even" and "odd" elements of $\mathbb{F}$; you can assume that there is the same number of each.

[4] One can keep track of candidate heavy hitters along the way, if desired; see the notes for Lecture #5 for details.

[5] The hope is that the random signs cause helpful cancellations when different elements collide.

(f) (4 points) Compare the space and correctness guarantees of the Count-Min-Sketch from Lecture #5 and the variant in this problem. For each of the two solutions, propose a situation (e.g. properties of the data stream, or requirements of the motivating heavy hitters application, etc.) where that solution would be preferable to the other.

Problem 10

(16 points) As mentioned in Lecture #6, it is trivial to compute F_1 (i.e., to count) using $\approx \log_2 m$ space, where m is the number of objects being counted. But what if we only care about about counting approximately and probabilistically, up to a $(1 \pm \epsilon)$ factor with probability at least $1 - \delta$? Here's a way to reduce the space to $O(\epsilon^{-2}\delta^{-1} \log\log m)$.[6] The basic idea is to count (probabilistically) $\log m$ rather than m itself, and then aggregate many independent estimates (as in our analysis of the AMS algorithm for F_2 estimation).

(a) (4 points) The basic estimator is the following. Initialize $Z = 0$. When a new object arrives, increment Z with probability 2^{-Z} (else leave it unchanged). At the end, output $X = 2^Z - 1$.

Prove that the estimator is unbiased, that $\mathbf{E}[X] = m$.

[Hint: prove by induction on i that, after seeing i objects, $\mathbf{E}\left[2^Z\right] = i + 1$.]

(b) (4 points) Prove that $\mathbf{E}\left[2^{2Z}\right] = \frac{3}{2}m^2 + \frac{3}{2}m + 1$.

[Hint: again, induction on i.]

(c) (4 points) Conclude that $\mathrm{Var}[X] = \frac{m(m-1)}{2}$.

(d) (4 points) Use the average of several independent estimators and Chebyshev's inequality (Exercise #14) to prove that a probabilistic $(1 \pm \epsilon)$-approximate counter requires only $O(\epsilon^{-2}\delta^{-1} \log\log m)$ space (where δ upper bounds the probability of failing to compute a $(1 \pm \epsilon)$-approximation).

Problem 11

(15 points) This problem concerns the same data stream model studied in Problem #9, but with the goal of computing the *number of distinct elements* in the stream. This quantity can be thought of as the 0th frequency moment F_0, with each element $j \in U$ contributing either 1 or 0 to it (depending on whether $f_j \geq 1$ or $f_j = 0$).

Like in Lecture #5, the plan is to aggregate a number of independent estimators. To keep things simple, we'll assume we have a random oracle h mapping each element $j \in U$ independently and uniformly to the real unit interval $[0, 1]$. (The same idea can be translated to reasonably practical hash functions with discrete range, but the details get a bit annoying.) Given such an h, here is the basic estimator, which tracks the minimum hash value seen so far:[7]

1. Initialize $Z := 1$.
2. Let h be a random oracle from U to $[0, 1]$.
3. For $i = 1, 2, \ldots, m$:
   (a) $Z := \min\{Z, h(x_i)\}$.
4. Return $X := \frac{1}{Z} - 1$.

[6] This subroutine can be used to reduce the dependence on m in the AMS F_2 estimation algorithm from $\log m$ to $\log\log m$.

[7] Intuition: if there are k distinct elements in the stream, then k distinct hash function values are observed. Under the random oracle assumption, these hash values are i.i.d. draws from the uniform distribution $[0, 1]$. The k expected order statistics split the interval $[0, 1]$ into $k + 1$ equal-length subintervals. (E.g., the expected minimum and maximum of two i.i.d. draws from the uniform distribution on $[0, 1]$ are $\frac{1}{3}$ and $\frac{2}{3}$, respectively.) Thus, $F_0 = k$ implies that the expected minimum is $1/(k + 1)$; the estimator inverts this relationship.

(a) (5 points) Prove that

$$\mathbf{E}[Z] = \frac{1}{F_0 + 1}.$$

[Hint: you might find the identity $\mathbf{E}[Z] = \int_0^\infty \mathbf{Pr}[Z \geq t]\, dt$ useful.]

(b) (5 points) Prove that

$$\mathbf{E}\left[Z^2\right] = \frac{2}{(F_0 + 1)(F_0 + 2)}$$

and hence $\mathrm{Var}[Z] \leq \frac{1}{(F_0+1)^2}$.

(c) (5 points) Extend the basic estimator above using averages of repeated trials (performed in parallel in a single pass over the data stream), to obtain an estimator Y with the following guarantee (for user-specified $\epsilon, \delta \in (0, 1)$): with probability at least $1 - \delta$,

$$|Y - F_0| \leq \epsilon \cdot F_0.$$

Be sure to specify precisely how many trials you need as a function of ϵ and δ.

Problem 12

(15 points) Recall from Lecture #7 that we proved the following (the "Leftover Hash Lemma"). Suppose X is a random variable with collision probability $cp(X)$ at most $1/K$. Suppose $\mathcal{H}$ is a universal family of hash functions (from the range of X to the range $\{0, 1, 2, \ldots, M-1\}$), and h is chosen uniformly at random from $\mathcal{H}$. Then the statistical distance between the joint distribution of $(h, h(X))$ and of the uniform distribution (on $\mathcal{H} \times \{0, 1, 2, \ldots, M-1\}$) is at most $\frac{1}{2}\sqrt{M/K}$.

For this problem, assume that you have a sequence $X_1, \ldots, X_T$ of random variables, with the property that for every i and fixed values of $X_1, \ldots, X_{i-1}$, the (conditional) collision probability of X_i is at most $1/K$ (i.e., a "block source with entropy $\log_2 K$"). Prove that the statistical distance between the joint distribution of $(h, h(X_1), \ldots, h(X_T))$ and of the uniform distribution on $\mathcal{H} \times [M]^T$ is at most $\frac{T}{2}\sqrt{M/K}$.

[Hint: One high-level approach is to prove, by downward induction on i, a bound of $\frac{(T-i)}{2}\sqrt{M/K}$ on the statistical distance between $(h, h(X_{i+1}), \ldots, h(X_T))$ and the uniform distribution for every fixed value of $X_1, \ldots, X_i$. The increase in statistical distance in the inductive step should come from the Triangle Inequality.]