Algorithms for Big Data Fall 2020

Homework 2

Due: Friday 10/16/20, 5pm PT

- **Solving 2 of the following 3 problems** will lead to full credit. You may attempt all three problems, but the grading will be based on the 2 problems with the highest scores.
- You may work in groups of size 1-3. If you do, please hand-in a single assignment with everyone's names on it. It is strongly encouraged to type up the solutions in Latex.
- If the question asks to prove something, you must write out a formal mathematical proof.
- If the question involves analyzing an algorithm, you must formally explain the time and/or space usage, along with the approximation guarantees (when applicable).
- When you are asked to prove a bound, it suffices to prove it up to multiplicative constants, i.e., using $O(\cdot)$, $\Theta(\cdot)$, or $\Omega(\cdot)$ notation. No need to optimize (multiplicative) constants!
- You may use other resources, but you must cite them. If you use any external sources, you still must provide a complete and self-contained proof/result for the homework solution.

Problem 1: Approximate Counting, Remix

In Lectures 4 and 5, we analyzed Morris' algorithm, which approximated the number of updates n by using the following estimator. Initialize a counter X to 0, and for each update, increment X with probability $1/2^X$. Then, the algorithm output $\tilde{n} = 2^X - 1$. To obtain good bounds on the probability that $|\tilde{n} - n| < \varepsilon n$, we considered Morris+ and Morris++ that eventually took the median of many means, where each mean averaged many estimates. Consider a different algorithm, where we still initialize X to 0, but we increment it with probability $1/(1+a)^X$ for some parameter a.

(a) Determine an estimator $\tilde{n}$ as a function of X and a such that it is an unbiased estimator, that is, $\mathbb{E}[\tilde{n}] = n$ after n updates.

(b) How small must a be so that our estimate $\tilde{n}$ of n satisfies $|\tilde{n} - n| < \varepsilon n$ with at least 9/10 probability when we return the output of a single estimator (instead of averaging many estimators as in class)?

(c) Derive a bound S on the space (in bits) as a function of n, ε, a so that this algorithm uses at most S space with at least 9/10 probability after n increments (in addition to satisfying $|\tilde{n} - n| < \varepsilon n$ with probability at least 9/10).

Problem 2: Pairwise Independence and Hashing

(a) Let q be a prime number. For integers $c, d \in \{0, 1, \ldots, q-1\}$, define the hash function $h_{c,d}$ as

$$h_{c,d}(x) = cx + d \mod q.$$

Let $\mathcal{H}$ be the set of all such hash functions, defined as

$$\mathcal{H} = \{h_{c,d} \mid c, d \in \{0, 1, \ldots, q-1\}\}.$$

Prove that $\mathcal{H}$ is a pairwise independent hash family. That is, prove that for any distinct $i \neq i'$ and any j, j', we have that

$$\Pr_{h_{c,d} \in \mathcal{H}}[\ h_{c,d}(i) = j \text{ and } h_{c,d}(i') = j'\ ] = \frac{1}{q^2},$$

where $h_{c,d} \in \mathcal{H}$ is chosen uniformly by choosing c, d at random in $\{0, 1, \ldots, q-1\}$.

Hint: Start with $q = 2$ and $\{0, 1\}$ values; then, generalize to all prime $q \geq 2$. For the general case, you can use that $cx + d = j$ has a unique solution in terms of x if $c \neq 0$.

(b) Let $Y_1, \ldots, Y_n$ be pairwise independent random variables. Prove that

$$\mathrm{Var}\left[\sum_{i=1}^{n} Y_i\right] = \sum_{i=1}^{n} \mathrm{Var}[Y_i].$$

(c) **Extra Credit.** Let q be a prime, and let k be an integer with $q \geq k$. Consider the set $\mathcal{H}$ of degree $k-1$ polynomials over $\mathbb{F}_q$. More precisely, let $\mathcal{H}$ be the set of polynomials $h_{\vec{c}}$ defined by a vector $\vec{c}$ of k coefficients $c_0, c_1, \ldots, c_{k-1} \in \{0, 1, \ldots, q-1\}$ such that

$$h_{\vec{c}}(x) = c_{k-1}x^{k-1} + c_{k-2}x^{k-2} + c_1 x + c_0 \mod q.$$

Prove that $\mathcal{H}$ is a k-wise independent hash family. That is, prove that for all distinct $i_1, i_2, \ldots, i_k$ and all $j_1, j_2, \ldots, j_k$, we have

$$\Pr_{\vec{c}}[\ h_{\vec{c}}(i_1) = j_1 \text{ and } h_{\vec{c}}(i_2) = j_2 \text{ and } \cdots \text{ and } h_{\vec{c}}(i_k) = j_k\ ] = \frac{1}{q^k},$$

where the probability is over uniformly random $c_0, c_1, \ldots, c_{k-1} \in \{0, 1, \ldots, q-1\}$.

Hint: Consider the $k \times k$ Vandermonde matrix, which is invertible.

Problem 3: Streaming Sampling

Consider the following algorithm for sampling a random element in a stream. You see $x_1, x_2, \ldots, x_m$ one at a time. For the first element x_1, store it as $s = x_1$, and initialize a counter $i = 1$. Every time you see an new element x_{i+1}, increment the counter, and flip a biased coin that comes up heads with probability $1/(i+1)$. If you get heads, then replace the stored element s with x_{i+1}.

(a) Prove that if you have seen m elements in the stream so far, then the probability that you have stored any given element is exactly $1/m$. That is, show that $\Pr[s = x_i] = 1/m$ for all $i = 1, 2, \ldots, m$ after you have seen all m elements.

(b) For a parameter $k > 1$, generalize the algorithm to sample k elements *without* replacement from the stream. As a hint, you can store the first k elements, and then replace one of the stored elements with a new element using a random process. Prove that for any subset $S \subseteq \{x_1, x_2, \ldots, x_m\}$ of size $|S| = k$, the algorithm outputs S with probability $1/\binom{m}{k}$.