Homework 1

Algorithms for Big Data: CS498 ABG/ABU, Fall 2022
Due: Friday at 10pm, 16th September 2022

Instructions and Policy:

- Each homework can be done in a group of size at most two. Only one homework needs to be submitted per group. However, we recommend that each of you think about the problems on your own first.
- Homework needs to be submitted in pdf format on Gradescope. See `https://courses.engr.illinois.edu/cs374/fa2018/hw-policies.html` for more detailed instructions on Gradescope submissions.
- Follow academic integrity policies as laid out in student code. You can consult sources but cite all of them including discussions with other classmates. Write in your own words. See the site mentioned in the preceding item for more detailed policies.

Problem 1. Combining samples for median. We saw a randomized sampling algorithm to estimate an ϵ-approximate median. The goal of this problem is to illustrate simple ways to combine samples from different streams. Suppose we use reservoir sampling on a stream σ_1 of length n and stored a k-sample (with replacement) S_1. We also used reservoir sampling on another stream σ_2 of length $3n$ and stored a k-sample S_2. You wish to compute an ϵ-approximate median of the concatenated stream $\sigma_1\sigma_2$ (assume the streams consist of numbers). Describe an algorithm to generate k uniform samples for the concatenated stream from S_1 and S_2. Say $\epsilon = 0.1$ and $k = 10000$. What is the probability that your algorithm will return an ϵ-approximate median? You can use the median analysis in the lecture.

Problem 2. Sampling, Chebyshev vs Chernoff. Suppose you want to estimate the average of n numbers via sampling, for example the average wealth of people in a town. The average can be very skewed by outliers — perhaps there are a few billionaires that will not make it to the sample but will clearly affect the average. However, we can obtain an accurate estimate if we assume that the numbers are within some limited range. Assume the input numbers $z_1, z_2, \ldots, z_n$ are from $[a, b]$ where $a, b \in \mathbb{R}$ with $a \leq b$. Suppose you sample k input numbers (with replacement) and output their average as the estimate for the true average $\alpha = (\sum_i z_i)/n$. Let X be the random variable denoting the output value.

- Using Chebyshev's inequality, show that for $k \geq \frac{(b-a)^2}{\delta\epsilon^2}$, we have
$$\mathrm{P}[|X - \alpha| \geq \epsilon] \leq \delta.$$
- Using the Chernoff inequality, show that there exists a constant $c > 0$ such that for $k \geq \frac{c(b-a)^2 \log(2/\delta)}{\epsilon^2}$, we have
$$\mathrm{P}[|X - \alpha| \geq \epsilon] \leq \delta.$$

Problem 3. Importance sampling. Importance sampling is a fundamental technique in statistics that is also used in several algorithms. Here we illustrate it in two simple scenarios.

- Recall the problem of estimating the mean via uniform sampling. The issue is the variance which can be high due to outliers. Suppose we want to estimate the mean of n non-negative numbers $a_1, a_2, \ldots, a_n$ and we also have some crude estimates $w_1, w_2, \ldots, w_n$ (we will not worry about where these estimates came from) such that for each i, $w_i/\alpha \leq a_i \leq \alpha w_i$ for some constant $\alpha > 1$ (say $\alpha = 5$). Consider estimating the mean via weighted sampling with $w_1, w_2, \ldots, w_n$: pick an $i \in [n]$ where the probability of picking i is w_i/W with $W = \sum_{i=1}^{n} w_i$. The estimator is $\frac{1}{n} a_i W/w_i$. You take k such samples and average them. Let Z be this average. Argue that Z is an exact estimator for the mean of $a_1, \ldots, a_n$. Upper bound the variance of Z as a function of α, k, and μ where $\mu = \frac{1}{n}\sum_{i=1}^{n} a_i$. Using Chebyshev's inequality, what is the number of samples k you need to guarantee that $\mathrm{P}[|Z - \mu| \geq \epsilon\mu] \leq \delta$ for given ϵ and δ in $[0, 1]$?

- Suppose you are using uniform sampling to estimate the mean of n non-negative numbers $a_1, \ldots, a_n$. This requires generating a random integer between 1 and n but what if we had only access to random bits? Let $N = 2^k$ where $2^{k-1} < n \leq 2^k$. If $n = N$ then we can pick k bits at random and generate perfectly uniform numbers between 1 and n. Otherwise, a standard technique is rejection sampling; pick a random number r in $[N]$ and reject it if $r > n$, otherwise we have a uniformly random number in $[n]$. However this has the problem that we can lose almost half the samples in expectation (if $n = 2^{k-1} + 1$). Further, it also has the disadvantage that there is no apriori bound on the number of useful samples from a given set of h samples (all of them could be rejected). One way to use all the generated samples is via importance sampling. Suppose we set up an onto mapping $\phi : [N] \to [n]$. We generate $i \in [N]$ and would like to use $j = \phi(i)$ where $j \in [n]$. Clearly this can create a non-uniform distribution over $[n]$. Show how you can adjust for this non-uniformity (knowing ϕ) so that you can still get an unbiased estimator for the mean. Suppose you use k such samples and take their average. Let Z be this random variable. Upper bound variance of Z when compared to the variance of the estimator based on uniform samples from $[n]$.

Problem 4. Quick Sort. Given an array A of n numbers (which we assume are distinct for simplicity), the algorithm picks a pivot x uniformly at random from A and computes the rank of x. If the rank of x is between $n/4$ and $3n/4$ (call such a pivot a good pivot), it behaves like the normal QuickSort in partitioning the array A and recursing on both sides. If the rank of x does not satisfy the desired property (the pivot picked is not good), the algorithm simply repeats the process of picking a pivot until it finds a good one. Note that in principle the algorithm may never terminate!

- Write a formal description of the algorithm.
- Prove that the expected run time of this algorithm is $O(n \log n)$ on an array on n numbers.
- Prove that the algorithm terminates in $O(n \log n)$ time with high probability.

Problem 5. Probabilistic counter. In lecture we analyzed probabilistic counting: initialize a counter X to 1, and for every increment instruction, increment X with probability $1/2^X$. By

averaging many such estimators, we obtained a $(1+\epsilon)$-approximation to n with good probability and space usage was $O(\log \log n)$. In this problem you will investigate a minor modification. Imagine we still initialize X to 1, but we increment it with probability $1/(1+a)^X$ for some fixed $a > 0$. (Note that your estimator for n would have to change from $2^X - 1$ to something else.)

How small must a be so that our estimate $\tilde{n}$ of n satisfies $|\tilde{n} - n| \leq \epsilon n$ with at least $9/10$ probability when we return the output of a single estimator instead of averaging many estimators we did in the lecture? Also derive a bound $S = S(n)$ on the space (in bits) so that this algorithm uses at most S space with at least $9/10$ probability by the end of the n increments.

Additional exercises (not to be submitted)

Problem 6. In class, we proved a powerful tail inequality called the "(multiplicative) Chernoff bound" that we will use time and time again. In this exercise, we rewrite the Chernoff inequality in a convenient form that is a little more interpretable and easier to apply.

Recall the Chernoff inequality, as follows. Let $X_1, X_2, \ldots, X_n \in [0,1]$ be n independent, non-negative, and uniformly bounded random variables. Let

$$\mu = \mathrm{E}\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} \mathrm{E}[X_i]$$

be the expected value of the sum. The Chernoff inequality states that for any $\delta > 0$, we have

$$\mathrm{P}[\textstyle\sum_{i=1}^{n} X_i \geq (1+\delta)\mu] \leq \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu} \text{ and } \mathrm{P}[\textstyle\sum_{i=1}^{n} X_i \leq (1-\delta)\mu] \leq \left(\frac{e^{-\delta}}{(1-\delta)^{1-\delta}}\right)^{\mu},$$

where for the second inequality we also further assume $\delta < 1$.

Now let $X_1, \ldots, X_n$ and μ be as above.

- Show that for $x \geq 0$ sufficiently small, we have

$$x - (1+x)\ln(1+x) \leq -\frac{x^2}{3}.$$

  Hint: Consider the Taylor expansion $\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \frac{x^5}{5} - \cdots$ *for* $x \in (-1, 1]$.

- Show that for $\epsilon \in [0,1]$,

$$\mathrm{P}\left[\sum_{i=1}^{n} X_i \geq (1+\epsilon)\mu\right] \leq e^{-\epsilon^2 \mu/3}.$$

- Show that for $x \in [0,1]$, we have

$$x + (1-x)\ln(1-x) \geq \frac{x^2}{2}.$$

- Show that for $\epsilon \in [0, 1]$,

$$\mathrm{P}\left[\sum_{i=1}^{n} X_i \leq (1-\epsilon)\mu\right] \leq e^{-\epsilon^2 \mu/2}.$$

Problem 7. Exercises 2 and 3 from HW 4 of the 2016 algorithms course (`https://courses.engr.illinois.edu/cs473/fa2016/Homework/hw4.pdf`)