COMPSCI 514: Problem Set 2

Due: 3/8 by 11:59pm in Gradescope.

Instructions:

- You are allowed to work on this problem set in a group of up to three members.
- 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.
- The problem set is meant for your own practice. We strongly discourage the use of LLMs to directly solve problems.
- Each problem will be graded on the following scale:
  - ✓+: (2 points) Submitted work demonstrates a full understanding of the problem. There may be some errors, omissions, or unclear steps, but overall, a reader would be able to understand how to solve the problem by looking at the submitted work.
  - ✓-: (1 point) Submitted work demonstrates partial understanding of the concepts, but contains significant omissions or errors.
  - X: (0 points) Not completed or submitted work doesn't not provide enough information to determine whether there is understanding of the problem.
- The 'Challenge Problems' are **completely optional** and not worth any extra credit. You may want to complete them e.g., if you are interested in further advanced study or research in the area, or if you just find the problem interesting.

1. Exponential Concentration Bound Practice

1. Let $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n$ be independent indicator random variables, let $\mu = \mathbb{E}\left[\sum_{i=1}^n \mathbf{X}_i\right]$, and let $\sigma^2 = \mathrm{Var}\left(\sum_{i=1}^n \mathbf{X}_i\right)$. Prove that $\sigma^2 \leq \mu$.
2. Use part (1) and Bernstein's inequality to prove the following, which is a (very) slightly weaker version of the Chernoff bound that we saw in class.

$$\Pr\left[\left|\sum_{i=1}^n \mathbf{X}_i - \mu\right| \geq \delta\mu\right] \leq 2\exp\left(-\frac{\delta^2\mu}{2 + 4/3 \cdot \delta}\right)$$

3. Improve the upper bound from part (2) to $2\exp\left(-\frac{\delta^2\mu}{2+2/3\cdot\delta}\right)$, which is slightly better than the Chernoff bound of $2\exp\left(-\frac{\delta^2\mu}{2+\delta}\right)$ stated in class. **Hint:** Can you improve M in your application of Bernstein's inequality?

4. We plan to conduct an opinion poll to find out the percentage of people in a community who support a certain candidate for president. Assume that every person answers either yes or no. Let p denote the actual fraction of people who support the candidate. We want an estimate $\tilde{\mathbf{p}}$ of p such that:

$$\Pr(|\tilde{\mathbf{p}} - p| \geq \epsilon p) \leq \delta,$$

for some given error parameter $\epsilon \in (0, 1)$ and failure probability $\delta \in (0, 1)$. Assume that we sample N people independently and uniformly at random from the community and let $\tilde{\mathbf{p}}$ be the fraction of this sample supporting the candidate. How large does N need to be, as a function of p, ϵ, and δ, for the above bound to hold? **Hint:** Apply a Chernoff bound.

5. Calculate how large the sample size N needs to be from your bound if $\epsilon = 0.1$, $\delta = 0.05$, and we know that $p \geq .3$.

Note: Parts 4-5 taken in part from *Probability and Computing* textbook.

2. Random Group Testing

Suppose that we wish to control the spread of some contagious illness by testing as many individuals as possible. Unfortunately testing is expensive. A common method to address is issue is to test individuals in *groups*. I.e., the biological samples from multiple patients are combined into a single sample and tested for the disease all at once. If the test returns negative, it means that all individuals in the group are negative. If the test comes back positive, it means that at least one individual in the group has the disease. We'll see how this strategy can be used to significantly save on the number of tests required to identify a small subset of positive individuals.

1. Show that there is a deterministic algorithm that uses $O(\sqrt{nk})$ tests and, if at most k individuals in the populations are positive, correctly identifies all these individuals, without identifying any individuals that are negative. You may assume that k is known in advance (often it can be estimated from the positive rate of prior tests). **Hint:** Consider a two-stage approach that first tests groups and then tests individuals within the groups.

2. Say we are testing the UMass Amherst student body. $n = 30,000$ and there is a 1% positivity rate, so $k = 300$. How many tests does the strategy of part (1) save over simply individually testing each member of the student body? You may assume that the constant in the big-Oh notation is 2 – i.e. that the algorithms uses $\leq 2\sqrt{nk}$ tests. There is an algorithm achieving that constant.

3. Consider the following randomized scheme: collect r samples from each individual. Then, repeat the following r times: randomly partition the population into G groups (i.e., each individual is assigned independently to group i with probability $1/G$ for $i = 1, 2, ..., G$), and test each group in aggregate. Once this process is complete, report that an individual is positive if *every group they were part of tested positive.* Report that an individual is negative if *any of the groups they were part of tested negative.* Show that for $G \geq 2k$ this scheme finds all truly positive patients, and that each negative patient is marked positive with probability $\leq \frac{1}{2^r}$.

4. Show that if we set $r = O(\log n)$ and $G = 2k$, then the method of part (3) yields no false negatives, requires just $O(k \log n)$ tests, and has no false positives with probability $\geq 99/100$.

3. A Bloom Filter Alternative

Consider the following approximate set data structure as an alternative to a bloom filter: we have two fully random hash functions, $\mathbf{h} : U \to [m]$ which maps items to positions in an m-bucket hash table, and $\mathbf{s} : U \to \{0,1\}^k$, which maps items to random bit "signatures" of length k. Assume that $\mathbf{h}$ and $\mathbf{s}$ are both fully random and independent of each other. Our data structure works as follows: insert(x) checks if $\mathbf{s}(x)$ is already stored at slot $\mathbf{h}(x)$ of the hash table and it not, stores $\mathbf{s}(x)$ there. Collisions are resolved using a simple linked-list to store all signatures hashed to a given bucket. query(x) checks if $\mathbf{s}(x)$ is stored in bucket $\mathbf{h}(x)$. If it is it returns YES. If not, it returns NO.

1. Does the above data structure have false negatives? Briefly explain why or why not.
2. Assume that n items have been stored in the data structure. Let w be some item that has not been stored. What is $\Pr[\text{query}(w) = \text{YES}]$, as a function of m, n, and k. I.e., what is the false positive rate?
3. Assuming that m is large, show that the false positive rate is approximately $1 - \exp\left(-\frac{n}{m \cdot 2^k}\right)$. **Hint:** Use that for large x, $(1 - 1/x)^x \approx 1/e$.
4. Using the approximate false positive rate from part (3), show that for $m = n$ and $k = \log_2(1/\delta)$, the false positive rate is at most δ. **Hint:** Use that for any x, $e^x \geq 1 + x$.
5. How does the above data structure compare to a bloom filter in terms of the amount of storage space needed to obtain false positive rate δ when storing n items? What about the insertion/query running times?

4. Improving Count-Min Sketch

In class we learned about the Count-Min sketch algorithm for counting frequent items in a stream. The algorithm maintains t length-m arrays $\mathbf{A}_1, \ldots, \mathbf{A}_t$ along with t random hash functions $\mathbf{h}_1, \ldots, \mathbf{h}_t$ mapping any item to an index in $1, \ldots, m$. When a new item x is presented, we increment $\mathbf{A}_i[\mathbf{h}_i(x)]$ for each i. At the end of the stream, we estimate the frequency $f(x)$ as $\tilde{f}(x) = \min_{i \in [t]} \mathbf{A}_i[\mathbf{h}_i(x)]$. We showed in class that, for $t = O(\log(1/\delta))$, with probability at least $1 - \delta$:

$$f(x) \leq \tilde{f}(x) \leq f(x) + \frac{3 \sum_{y \neq x} f(y)}{m}. \tag{1}$$

Consider a modification of Count-Min sketch that may increment or decrement the count in each step. In addition to $\mathbf{h}_1, \ldots, \mathbf{h}_t$, chose random hash functions $\mathbf{s}_1, \ldots, \mathbf{s}_t$ mapping each item to $\{-1, 1\}$. When a new item x is presented, we perform the update $\mathbf{A}_i[\mathbf{h}_i(x)] := \mathbf{A}_i[\mathbf{h}_i(x)] + \mathbf{s}_i(x)$ for each i. At the end of the stream, we estimate the frequency $f(x)$ as $\tilde{f}(x) = \text{median}_{i \in [t]} [\mathbf{s}_i(x) \cdot \mathbf{A}_i[\mathbf{h}_i(x)]]$.

1. Intuitively, why might you expect this method to decrease error as compared to classic Count-Min sketch?
2. Show that for any i, $\mathbb{E}[\mathbf{s}_i(x) \cdot \mathbf{A}_i[\mathbf{h}_i(x)]] = f(x)$.
3. Show that for any i, with probability $\geq \frac{3}{4}$, $|\mathbf{s}_i(x) \cdot \mathbf{A}_i[\mathbf{h}_i(x)] - f(x)| \leq \frac{2 \cdot \sqrt{\sum_{y \neq x} f(y)^2}}{\sqrt{m}}$.
4. Show that for $t = O(\log(1/\delta))$, with probability $\geq 1 - \delta$, $|\tilde{f}(x) - f(x)| \leq \frac{2 \cdot \sqrt{\sum_{y \neq x} f(y)^2}}{\sqrt{m}}$.

5. Given a fixed setting of m and t, when do you expect the error bound of part (4) to be better than the Count-Min sketch bound in (1)? Again, fixing m and t, which algorithm do you generally expect to be more accurate on real world streams?

Challenge Problems

C1. Exponential Tail Bounds from Scratch 🌶🌶

Here we will see how to prove exponential tail bounds from scratch, using the moment generating function approach discussed in class. We will prove a generalization of the Chernoff bound to random variables lying in $[0,1]$, rather than just $\{0,1\}$. Throughout, let $\exp(x)$ denote e^x.

1. Let $\mathbf{Y}$ be a random variable that takes values in the interval $[0,1]$. Prove that for $t > 0$, $\mathbb{E}[\exp(t\mathbf{Y})] \leq 1 + \mathbb{E}[\mathbf{Y}](e^t - 1)$. **Hint:** First show that $\exp(ty) \leq 1 + y(e^t - 1)$ for all $y \in [0,1]$. It might be helpful to plot these two functions to compare them before trying to give a formal proof.

2. Let $\mathbf{X}_1, \ldots, \mathbf{X}_n$ be independent random variables that take values in the interval $[0,1]$. Let $\mathbf{X} = \sum_{i=1}^{n} \mathbf{X}_i$ and $\mu = \mathbb{E}[\mathbf{X}]$. Prove that for $0 < \delta < 1$,

$$\Pr[\mathbf{X} \geq (1+\delta)\mu] = \Pr[\exp(t\mathbf{X}) \geq \exp(t(1+\delta)\mu)] \leq \frac{\mathbb{E}[\exp(t\mathbf{X})]}{\exp(t(1+\delta)\mu)} .$$

   Hint: Use Markov's inequality.

3. Prove that $\Pr[\mathbf{X} \geq (1+\delta)\mu] \leq \exp(-\delta^2\mu/3)$.

   Hint: Consider setting $t = \ln(1+\delta)$ and using part one of the question. You may use the fact that $(1+\delta)^{1+\delta} \geq e^{\delta+\delta^2/3}$ and that for any $x \geq 0$, $(1+x) \leq e^x$. You may also want to recall that for independent random variables $\mathbf{Y}, \mathbf{Z}$, $\mathbb{E}[\mathbf{YZ}] = \mathbb{E}[\mathbf{Y}] \cdot \mathbb{E}[\mathbf{Z}]$.

   Hint: For simplicity, you may assume here that each $\mathbf{X}_i$ has the same mean. I.e., that $\mathbb{E}[\mathbf{X}_i] = \mu/n$. The stated bound holds even without this assumption, but making this assumption may make the proof easier, and will suffice for achieving full credit.

C2. A Faster Algorithm for Distinct Elements 🌶🌶

Let $\mathbf{x}_1, \ldots, \mathbf{x}_d$ be chosen independent and uniformly at random in $[0,1]$. For any $k \in \{1, 2, \ldots, d-1\}$, let $\mathbf{s}$ be the $(k+1)^{st}$ smallest value of $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$. Let $\mathbf{s}_1$ be the k^{th} smallest of $\{\mathbf{x}_2, \ldots, \mathbf{x}_d\}$. I.e., $\mathbf{s}_1$ is the k^{th} smallest of all the values except the first one. Finally, let $\mathcal{S} \subset \{1, \ldots, d\}$ be the set $\mathcal{S} = \{i \in \{1, \ldots, d\} | \mathbf{x}_i < \mathbf{s}\}$. That is, $\mathcal{S}$ contains the indices of the k values lying below $\mathbf{s}$.

1. Let $\mathbf{Y} = \begin{cases} 1/\mathbf{s} \text{ if } 1 \in \mathcal{S} \\ 0 \text{ otherwise} \end{cases}$. Let $\mathbf{Y}_1 = \begin{cases} 1/\mathbf{s}_1 \text{ if } 1 \in \mathcal{S} \\ 0 \text{ otherwise} \end{cases}$. Argue that $\mathbf{Y} = \mathbf{Y}_1$.

2. Prove that $\mathbb{E}[\mathbf{Y}] = 1$. **Hint:** First consider $\mathbb{E}[\mathbf{Y}_1 | \mathbf{s}_1 = t]$ for any $t \geq 0$.

3. Prove that $\mathbb{E}[\mathbf{Y}] = \mathbb{E}\left[\frac{1}{\mathbf{s}}\right] \cdot \frac{k}{d}$. Conclude that $\mathbb{E}\left[\frac{1}{\mathbf{s}}\right] = \frac{d}{k}$.

4. Use a similar approach to show that $\text{Var}[\frac{1}{s}] = \frac{d(d-k)}{k^2(k-1)}$.

5. Argue that, for any $\epsilon, \delta \in (0,1)$, if we set $k \geq \frac{1}{\epsilon^2 \delta} + 1$ and $\tilde{d} = \frac{k}{\mathbf{s}}$, then with probability at least $1 - \delta$, we will have $|\tilde{d} - d| \leq \epsilon d$.

6. Describe how the above analysis can be used to obtain an algorithm for distinct items estimation using $O(1/(\epsilon^2 \delta))$ units of space (i.e., matching the algorithm discussed in class) but just one hash function. Why might this algorithm be preferable to the one discussed in class?