COMPSCI 514: Problem Set 2

Due: 10/14 by 11:59pm in Gradescope.

Instructions:

- You are allowed to, and highly encouraged 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.
- You may talk to members of other groups at a high level about the problems but **not work through the solutions in detail together**.
- You must show your work/derive any answers as part of the solutions to receive full credit.

1. Moment Bounds and Exponential Concentration (8 points)

Consider flipping n independent coins, each of which hits heads with probability 1/2 and tails otherwise. Let $\mathbf{X}$ be the number of heads that you see.

1. (1 point) For $n = 1000$, exactly compute $\Pr(\mathbf{X} \geq 600)$.
2. (1 point) For $n = 1000$, use Markov's inequality to upper bound $\Pr(\mathbf{X} \geq 600)$.
3. (1 point) For $n = 1000$, use Chebyshev's inequality to upper bound $\Pr(\mathbf{X} \geq 600)$.
4. (1 point) For $n = 1000$, use a Chernoff bound inequality to upper bound $\Pr(\mathbf{X} \geq 600)$.
5. (2 points) For any $z > 0$, give a formula for $\mathbb{E}[\exp(z\mathbf{X})]$. Use this to derive an upper bound on $\Pr(\mathbf{X} \geq t)$ as a function of n, t, and z. **Hint:** Use that for independent $\mathbf{X}, \mathbf{Y}$, $\mathbb{E}[\mathbf{XY}] = \mathbb{E}[\mathbf{X}] \cdot \mathbb{E}[\mathbf{Y}]$.
6. (2 points) Apply the formula above to give the best upper bound you can on $\Pr(\mathbf{X} \geq 600)$ when $n = 1000$. **Hint:** Optimize the bound over $z > 0$.

2. Streaming Averages (5 points)

1. (1 point) Consider a stream of numbers $x_1, \ldots, x_n$. Describe an algorithm that processes this stream using $O(1)$ space and exactly computes the average $\mu = \frac{1}{n} \sum_{i=1}^{n} x_i$. **Note:** Do not worry about bit complexity. It suffices to describe an algorithm that accomplishes the task by storing just $O(1)$ numbers.

2. (4 points) Again consider a stream of numbers $x_1, \ldots, x_n$ all lying in $[-M, M]$. Let μ_d be the average of the *distinct elements* in the data stream. Describe an algorithm that, given $\epsilon, \delta \in (0,1)$, uses $O(\log(1/\delta)/\epsilon^2)$ space and outputs, with probability at least $1-\delta$, an estimator $\tilde{\mu}_d$ with $|\tilde{\mu}_d - \mu_d| \leq \epsilon \cdot M$.

   Hint: First figure out how to take random samples from the stream which are equal to μ_d in expectation. Then apply a concentration inequality.

3. A Better Method for Similarity Estimation (8 points)

Consider estimating the Jaccard similarity $J(A,B) = \frac{|A \cap B|}{|A \cup B|}$ between two sets A and B via the following simple strategy based on repeated MinHashing:

- Choose k independent uniform random hash functions $\mathbf{h}_1, \ldots, \mathbf{h}_k : U \to [0,1]$.
- Let $\mathbf{s}^A = \{\mathbf{s}^A_1, \ldots, \mathbf{s}^A_k\}$ where $\mathbf{s}^A_i = \min_{a \in A} \mathbf{h}_i(a)$.
- Let $\mathbf{s}^B = \{\mathbf{s}^B_1, \ldots, \mathbf{s}^B_k\}$ where $\mathbf{s}^B_i = \min_{b \in B} \mathbf{h}_i(b)$.

Given $\mathbf{s}^A$ and $\mathbf{s}^B$, each a list of k numbers, estimate $J(A,B)$ as $\tilde{\mathbf{J}} = \frac{1}{k}\sum_{i=1}^{k} \mathbb{1}[\mathbf{s}^A_i = \mathbf{s}^B_i]$ (i.e., $\tilde{\mathbf{J}}$ is the fraction of colliding hashes in $\mathbf{s}^A$ and $\mathbf{s}^B$).

1. (3 points) Show that if we set $k \geq \frac{1}{\epsilon^2 \delta}$, for $\epsilon, \delta \in (0,1)$, then with probability at least $1-\delta$, $\left|\tilde{\mathbf{J}} - J(A,B)\right| \leq \epsilon\sqrt{J(A,B)}$.

Now consider a different strategy:

- Choose a single uniform random hash functions $\mathbf{h} : U \to [0,1]$.
- Let $\mathbf{s}^A$ contain the k smallest values obtained when $\mathbf{h}$ is applied to all the items in A. Similarly, let $\mathbf{s}^B$ contain the k smallest values obtained when $\mathbf{h}$ is applied to all the items in B.
- Let $\mathbf{s}$ contain the k smallest hash values from $\mathbf{s}^A \cup \mathbf{s}^B$.

Estimate $J(A,B)$ as $\tilde{\mathbf{J}} = \frac{|\mathbf{s}^A \cap \mathbf{s}^B \cap \mathbf{s}|}{k}$. I.e., $\tilde{\mathbf{J}}$ is the fraction of values in $\mathbf{s}$ that are in both $\mathbf{s}^A$ and $\mathbf{s}^B$.

2. (2 points) Show that $\mathbb{E}[\tilde{\mathbf{J}}] = J(A,B)$.

3. (2 points) Show that if we set $k \geq \frac{1}{\epsilon^2 \delta}$, for $\epsilon, \delta \in (0,1)$, then with probability at least $1-\delta$, $\left|\tilde{\mathbf{J}} - J(A,B)\right| \leq \epsilon\sqrt{J(A,B)}$.

   Hint: You may use the following result on sampling *without replacement*: Let $\mathbf{X}_1, \ldots, \mathbf{X}_k$ be independent and identically distributed random variables, drawn independently and uniformly at random with replacement from a finite multi-set U. Let $\mathbf{Y}_1, \ldots, \mathbf{Y}_k$ be drawn uniformly at random *without replacement* from U. Then $\mathrm{Var}\left(\sum_{i=1}^{k} \mathbf{Y}_i\right) \leq \mathrm{Var}\left(\sum_{i=1}^{k} \mathbf{X}_i\right)$.

4. (1 point) Computationally, why might this above method be preferred over the simple repeated MinHashing approach in part 1?

4. Improved Bounds and Variants of Count-Min Sketch (10 points)

In class we showed that the Count-Min sketch algorithm implemented with $t = O(\log(1/\delta))$ tables of size m returns a frequency estimate $\tilde{f}(x)$ for any item x, satisfying with probability $\geq 1 - \delta$, $f(x) \leq \tilde{f}(x) \leq f(x) + \frac{cn}{m}$, where n is the total frequency of items in the data stream and c is a small constant ($c = 2$ in the analysis shown in class).

1. (4 points) Let $f_1, \ldots, f_k$ be the frequencies of the k most frequent items in our data stream and let $n_k = n - \sum_{i=1}^{k} f_i$. Prove that Count-Min sketch implemented with $t = O(\log(1/\delta))$ tables of size $m = O(k)$ returns a frequency estimate $\tilde{f}(x)$ for any item x, satisfying with probability $\geq 1 - \delta$, $f(x) \leq \tilde{f}(x) \leq f(x) + \frac{cn_k}{m}$ for some constant c.

2. (2 points) Describe a scenario in which you think that the error bound above will be much better than the error bound shown in class.

3. (4 points) Consider a variation on count-min sketch: instead of incrementing each counter $A_1[\mathbf{h}_1(x_i)], \ldots, A_t[\mathbf{h}_t(x_i)]$ when x_i comes in, we compute $M = \min_{j \in [t]} A_j[\mathbf{h}_j(x_i)]$. Then we only increment $A_j[\mathbf{h}_j(x_i)]$ if $A_j[\mathbf{h}_j(x_i)] = M$. Show that the estimate output by this variation can *only be better* than the estimate of the count-min sketch algorithm presented in class.