NYU CS-GY 6763: Algorithmic Machine Learning and Data Science Fall 2023

Homework 1

Name: Solution key

Problem 1

(a) Assume our hash table has cm^2 slots for large constant c. The probability that the i^{th} item does not causes a collision when inserted is greater than or equal to:

$$1 - \frac{i-1}{cm^2}.$$

To see that this is that case, note that the table only has *at most* $i-1$ filled slots out of cm^2 total, so the probablilty of a collision is $\leq \frac{i-1}{cm^2}$. Using the expression above, we have that the probability there are no collisions at all is greater than:

$$\left(1 - \frac{0}{cm^2}\right) \cdot \left(1 - \frac{1}{cm^2}\right) \cdot \left(1 - \frac{2}{cm^2}\right) \cdot \ldots \cdot \left(1 - \frac{m-1}{cm^2}\right) \geq \left(1 - \frac{1}{cm}\right)^m.$$

As hinted, for $cm \geq 2$, $(1 - \frac{1}{cm})^{cm} \geq 1/2e$. So, $(1 - \frac{1}{cm})^m = \left((1 - \frac{1}{cm})^{cm}\right)^{1/c} \geq (1/2e)^{1/c}$. Choosing $c \geq 17$ yields $(1/2e)^{1/c} \geq 9/10$. So, as long as the table has $\geq 17m^2$ slots, there is no collision with probability $\geq 9/10$.

(b) By linearity of expectation, we have the $\mathbb{E}[\|x\|_2^2] = \sum_{i=1}^d \mathbb{E}[x_i^2]$, so we focus on bounding $\mathbb{E}[x_i^2]$ for a single i. We have that $x_i = \sum_{j=1}^n s_j$ where:

$$s_j = \begin{cases} 0 & \text{with probability } 1 - 1/d \\ -1 & \text{with probability } 1/2d \\ +1 & \text{with probability } 1/2d \end{cases}$$

So we calculate the expectation:

$$\begin{aligned} \mathbb{E}[x_i^2] = \mathbb{E}\left[\left(\sum_{j=1}^n s_j\right)^2\right] &= \mathbb{E}\left[s_1^2 + \ldots + s_n^2 + s_1 s_2 + \ldots + s_j s_k + \ldots + s_{n-1} s_n\right] \\ &= \mathbb{E}[s_1^2] + \ldots + \mathbb{E}[s_n^2] + \mathbb{E}[s_1 s_2] + \ldots + \mathbb{E}[s_j s_k] + \ldots + \mathbb{E}[s_{n-1} s_n] \end{aligned}$$

We have that for each s_j, $\mathbb{E}[s_j] = 0$ and s_j and s_k are independent, so all of the terms of the form $\mathbb{E}[s_j s_k]$ equal 0. So we have that:

$$\mathbb{E}[x_i^2] = \mathbb{E}[s_1^2] + \ldots + \mathbb{E}[s_n^2].$$

It can be directly calculated that for each j, $\mathbb{E}[s_j^2] = 1 \cdot \frac{1}{2d} + 1 \cdot \frac{1}{2d} = \frac{1}{d}$. So $\mathbb{E}[x_i^2] = \frac{n}{d}$ and finally,

$$\mathbb{E}[\|x\|_2^2] = \sum_{i=1}^d \mathbb{E}[x_i^2] = d \cdot \frac{n}{d} = n.$$

Problem 2

We use the notation from the Lecture 1 notes. Consider a single table A and let $\tilde{f} = A(h(v))$ be the frequency estimate for just that table. It suffices to show that, with probability $\geq c$ for any constant c,

$$f(v) \leq \tilde{f} \leq f(v) + \frac{1}{m}C.$$

As long as this is the case, then we can improve the success probability to 9/10 by increasing the number of tables used to $\log_{1-c}(1/10) = O(1)$, exactly as was done in class for the standard Count-Min analysis.

To get the improved bound, we split up $A(h(v))$ in a slightly refined way. The intuition is that, with high probability, item v will not collide with *any* of the top m most frequent items. Call these items $v_1, \ldots, v_m$. Call the less frequent items $v_{m+1}, v_{m+2}, \ldots$. We have:

$$A(h(v)) = f(v) + \sum_{i=1,\ldots,m, v_i \neq v} \mathbb{1}[h(v) = h(v_i)] \cdot f(v_i) + \sum_{i>m, v_i \neq v} \mathbb{1}[h(v) = h(v_i)] f(y).$$

The second two terms are our error terms. To bound the first, note that the event $\mathbb{1}[h(v) = h(v_i)]$ only happens with probability $1/m$ if our table has m slots in it. Moreover, $\mathbb{1}[h(v) = h(v_i)]$ is independent from $\mathbb{1}[h(v) = h(v_j)]$ for $i \neq j$. So, none of the events $\mathbb{1}[h(v) = h(v_i)]$ happen with probability:

$$(1 - 1/m)^m \geq 1/2e \approx .18.$$

Moreover, using the same Markov's inequality analysis from class, we have that the final term $\sum_{y \notin v \cup \{v_1,\ldots,v_m\}} \mathbb{1}[h(v) = h(v_i)] f(y)$ is upper bounded by $\frac{10}{m} \sum_{y \notin v \cup \{v_1,\ldots,v_m\}} f(y) = \frac{10}{m} \cdot C$ with probability 9/10. So, by a union bound, with probability at least .08, we have that:

$$\sum_{i=1,\ldots,m, v_i \neq v} \mathbb{1}[h(v) = h(v_i)] \cdot f(v_i) = 0$$

and

$$\sum_{y \notin v \cup \{v_1,\ldots,v_m\}} \mathbb{1}[h(v) = h(v_i)] f(y) \leq \frac{10}{m} \cdot C.$$

Adjusing constants on m gives the desired result.

Problem 3

1. Split the people being tested into C equal groups with n/C people each. At most k of the groups will test positive since, in the worst case, each infected person will be in a different group. For each positive group, we need to rerun n/C individual tests. So the number of tests run is at most $C + \frac{n}{C} \cdot k$. If we set $C = \sqrt{nk}$ then the total number of tests is:

$$C + \frac{n}{C} \cdot k = \sqrt{nk} + \frac{n}{\sqrt{n}\sqrt{k}} k = 2\sqrt{nk},$$

as desired.

2. Assume $q = c \log n$ for sufficiently large constant c and assume $C = 2k$. Consider any single individual who is negative for COVID. For this individual to be falsely reported positive, they would need to test positive in all q group tests they participates in.

What is the probability they test positive in any one group test? For this to happen, there would need to be a positive individual in that group. Let Z be the number of positive individuals in any given group. Each group has size $\frac{n}{2k}$, so by linearity of expectation, $\mathbb{E}[Z] = \frac{n}{2k} \cdot \frac{k}{n} = \frac{1}{2}$, since the probability that any group member is positive is k/n. Then by Markov's inequality, $\Pr[Z \geq 1] \leq \frac{1}{2}$. I.e., with probability $\geq \frac{1}{2}$ any given group contains *no infected individuals*.

So, the probability our negative individual has a positive person in all $q = c \log n$ of their groups is less than $\frac{1}{2}^q = \frac{1}{n^c}$, which is $\leq \frac{1}{10n}$ for sufficiently large c. Thus, the probability any given negative individual gets a false positive result is $\leq \frac{1}{10n}$. There are $n - k < n$ negative individuals and by a union bound, the probability *any* of them gets a false positive test is $\leq \frac{1}{10n} \cdot n \leq \frac{1}{10}$. In other words, we get no false positives with probability $\geq 9/10$.

3. **Informal**: The lower bound is via a counting argument. One way of framing our goal is that we need to return a bit string of length n which has a 1 in exactly k places and zeros everywhere else, with the 1's indicating the people we believe are infected, and the 0's indicating uninfected.

Before starting the testing scheme, we don't know what the correct bit string is, and there are $\binom{n}{k}$ different possibilities. When we run any testing scheme, we will receive a binary reponse to every test $t_1, \ldots, t_T$ and from that response will decide how to run the next test. In the end, we will decide on which bit string of the $\binom{n}{k}$ different possibilities is correct. The solution we return must be a function of $t_1, \ldots, t_T$: it could also depend on what subsets of people were tested, but at any time i, that must be a function of $t_1, \ldots, t_{i-1}$, via induction.

So, to obtain a correct answer after T tests, it must be that the number of possible test results $t_1, \ldots, t_T$ exceeds the number of possible solutions $\binom{n}{k}$. In other words that $2^t \geq \binom{n}{k}$. Taking logs on both sides, and noting that $\log \binom{n}{k} = O(k \log(n/k))$ gives the result.

Problem 4

The probability a coin flipped 100 times comes up ≥ 60 times heads is the sum of the probabilities it comes up exactly 60 times heads, 61 times heads, ... 100 times heads. The probability a coin comes up x time heads equals $\binom{100}{x} \cdot (\frac{1}{2})^{100}$. So we have:

$$\Pr[\geq 60 \text{ heads out of } 100] = \sum_{i=60}^{100} \binom{100}{i} \cdot \left(\frac{1}{2}\right)^{100} = \mathbf{.028}$$

$$\Pr[\geq 600 \text{ heads out of } 1000] = \sum_{i=600}^{1000} \binom{1000}{i} \cdot \left(\frac{1}{2}\right)^{1000} = \mathbf{1.36 \cdot 10^{-10}}$$

They need to take some care when computing these numbers! For most languages the $(\frac{1}{2})^{1000}$ won't be a problem: 1/2 will be treated a floating point double, which can represent numbers as small

as $10^{-}308$. $(\frac{1}{2})^{1000}$ is roughly 10^{-302} so all good :). On the other hand, there is a chance that the language tries to represent $\binom{100}{i}$ as an integer, which will overflow... I check both Matlab and Python, which will work for different reasons. Matlab will represent the result as a floating point number. Python will detect the overflow and use an integer type with more bits to prevent it.

To compare to the Chernoff bounds, note that $\mu = 50$ and $\mu = 500$ for the two problems, and for both $\epsilon = .2$. Using the different bounds we get:

$$\Pr[\geq 60 \text{ heads out of } 100] \leq .4029$$
$$\Pr[\geq 60 \text{ heads out of } 100] \leq .3909$$

and

$$\Pr[\geq 600 \text{ heads out of } 1000] \leq 1.12 \cdot 10^{-4}$$
$$\Pr[\geq 600 \text{ heads out of } 1000] \leq 8.33 \cdot 10^{-5}$$

The take away is that, even though they improve on e.g. Chebyshev's inequality, Chernoff bounds are actually quite loose – off by orders of magnitude in comparison the precise answer. There isn't a huge difference between the looser bound we typically use and the tighter bound you can find online though, which is good news.

Problem 5

1. Recall that $\tilde{n} = \frac{m(m-1)}{2D}$, where m is the number of samples collected, and D is the number of duplicates observed. Also recall that $n = \frac{m(m-1)}{2\,\mathbb{E}[D]}$. So, using an argument identical to the distinct elements analysis in Lecture 2, it suffices to show that:

$$(1 - \epsilon/2)\,\mathbb{E}[D] \leq D \leq (1 + \epsilon/2)\,\mathbb{E}[D]. \tag{1}$$

In particular, if this is the case,

$$\frac{1}{1+\epsilon/2}\frac{1}{\mathbb{E}[D]} \leq \frac{1}{D} \leq \frac{1}{1-\epsilon/2}\frac{1}{\mathbb{E}[D]},$$

which implies that, for any $\epsilon \leq 1$,

$$(1 - \epsilon/2)\frac{1}{\mathbb{E}[D]} \leq \frac{1}{D} \leq (1 + \epsilon)\frac{1}{\mathbb{E}[D]},$$

and thus $(1 - \epsilon)n \leq \tilde{n} \leq (1 + \epsilon)n$.

So, we focus on proving (1). To do so, we will bound the variance of D and use Chebyshev's inequality. Use $s_1, \ldots, s_m$ to denote our m samples. We have that:

$$D = \sum_{i<j} \mathbb{1}[s_i = s_j].$$

Note that the terms in the sum are pairwise independent. In particular, the event that $s_i = s_j$ does not effect the probability that $s_k = s_j$ for some other value of k. Interesting the terms are not

three-wise independent, since the event that $s_i = s_j$ and $s_j = s_k$ implies that $s_i = s_k$. Regardless, pairwise independence is all we need to apply linearity of variance:

$$\mathrm{Var}[D] = \sum_{i<j} \mathrm{Var}[\mathbb{1}[s_i = s_j]].$$

$\mathbb{1}[s_i = s_j]$ is a binary random variable equal to 1 with probability $1/n$, so its variance is equal to $\frac{1}{n} - \frac{1}{n^2}$. We conclude that

$$\mathrm{Var}[D] \leq \sum_{i<j} \frac{1}{n} = \binom{m}{2} \cdot \frac{1}{n} = \frac{m(m-1)}{2n}.$$

Note that this is exactly equal to $\mathbb{E}[D]$. So, we have via Chebyshev's inequality that:

$$\Pr\left[|D - \mathbb{E}[D] \leq \sqrt{10}\sqrt{\mathbb{E}[D]}\right] \leq \frac{1}{10}.$$

To prove (1), we need to choose m large enough so that $\sqrt{10}\sqrt{\mathbb{E}[D]} \leq \epsilon\,\mathbb{E}[D]$. I.e., we need

$$\begin{aligned}
\sqrt{\frac{m(m-1)}{2n}} &\geq \frac{\sqrt{10}}{\epsilon} \\
m(m-1) &\geq \frac{20n}{\epsilon^2} \\
m &\geq \sqrt{\frac{40n}{\epsilon^2}}.
\end{aligned}$$

This proves the claim.