
CS6170: RANDOMIZED ALGORITHMS
PROBLEM SET #2

NAME: Akash Roy
ROLL No: CS22M007

DUE: Sep 12, 23:59

Problem 1

?? marks

Suppose that n jobs are distributed to m servers (you can assume that m divides n). A job takes 1 step with probability p and $k > 1$ steps with probability $1 - p$. Use Chernoff bounds to calculate the upper and lower bounds (that hold with high probability) on when all jobs will be completed if we randomly assign exactly n/m jobs to each server.

Solution: Discussed with Krishna (Electrical PhD.)

If all the servers are running in parallel. All the server nodes have n/m jobs. We define one random variable $X_i = 1$ if job i runs for 1 step (with probability p), 0 otherwise. Time to complete all jobs = \hat{X} .

$$\begin{aligned} X &= \sum_{i=1}^n [1 * I[X_i = 1] + k * I[X_i = 0]] \\ &= \sum_{i=1}^n [1(1 - X_i) + k(X_i)] \\ &= (k - 1) \sum_{i=1}^n X_i + n \\ \therefore \mathbb{E}[X] &= n + \sum_{i=1}^n (k - 1)\mathbb{E}[X_i] \\ \therefore \mathbb{E}[X] &= n + (k - 1)(1 - p)n \end{aligned}$$

where I is an indicator function that takes value 1 if $X_i = 1$ and 0 otherwise and X is time to run jobs on a single machine.

Trivially $\hat{X} \leq X$. Therefore we can bound the time taken to run all the jobs ($= \hat{X}$) as following

$$\frac{X}{m} \leq \hat{X} \leq X$$

$$\Pr[X \geq (1 + \delta)(n + (k - 1)(1 - p)n)] \leq \exp\left(-\frac{(n + (k - 1)(1 - p)n)\delta^2}{3}\right)$$

Given

$$\begin{aligned}
\frac{X}{m} &\leq \hat{X} \leq X \\
\Pr[X \geq (1 + \delta)(n + (k - 1)(1 - p)n)] &= \Pr[\hat{X} \geq (1 + \delta) \frac{(n + (k - 1)(1 - p)n)}{m}] \\
&= \Pr\left[\hat{X} \geq \left(1 + \left[\frac{\delta}{m} + \frac{1}{m} - 1\right]\right) (n + (k - 1)(1 - p)n)\right] \\
&\leq \exp\left[\frac{\mu(\hat{\delta})^2}{3}\right] \\
&\leq \exp\left[\frac{(n + (k - 1)(1 - p)n) \left(\left[\frac{\delta}{m} + \frac{1}{m} - 1\right]\right)^2}{3}\right]
\end{aligned}$$

Problem 2

?? marks

Consider throwing m balls into n bins, and let the bins be numbered from 0 to $n - 1$. We say there is a k -gap at bin i if bins $i, i + 1, \dots, i + k - 1$ are all empty.

(a) (2 marks) Determine the expected number of k -gaps.

Solution: Let's define indicator random variable $X_i = 1$ if there is a gap from index $i \rightarrow i + k - 1$.

Then number of k gaps is $X = \sum_{i=0}^{n-k-1} X_i$

Now we can say the following via linearity of expectations

$$E[X] = \sum_{i=0}^{n-k-1} E[X_i]$$

For each of these random variable X_i the value of $E[X_i] = 1 * \Pr[X_i = 1]$. Now we need to find the value for $\Pr[X_i = 1]$.

$\Pr[X_i = 1]$ means the probability of throwing m balls into n bins and none of those falling into either of those k bins.

$$\Pr[X_i = 1] = \left(\frac{n - k}{n}\right)^m$$

Finally

$$\begin{aligned}
E[X] &= \sum_{i=0}^{n-k-1} E[X_i] \\
&= (n - k + 1) * \left(\frac{n - k}{n}\right)^m
\end{aligned}$$

(b) (5 marks) Prove a Chernoff-like bound for the number of k -gaps.

Hint: If you let $X_i = 1$ if there is a k -gap starting at i , then X_i and X_{i+k} are dependent. Consider X_i and X_{i+k} .

Solution: Let's define one random variable $X_i = 1$ if there is a k gap at position i and 0 otherwise.

$$X = \sum_{i=0}^{n-k-1} X_i$$

defines the number of k gaps if m balls are thrown into n bins.

Let's consider one poisson process and y_i are the balls choosen for bin i according to the poisson distribution, x_i is the number of balls in bin i . Also let's define one indicator function f such that $X_i = f(x_1, x_2, \dots, x_{i+k-1}) = 1$ if there is a k gap starting from i and 0 otherwise.

We also define $K_{\text{GAP } X} = X_i + X_{i+k} + X_{i+2k} \dots$ and $K_{\text{GAP } \text{Poi}} = Y_i + Y_{i+k} + Y_{i+2k} \dots$

Number of k gaps can now be defined as follows

$$X = \sum_{i=0}^{k-1} K_{\text{GAP } X}$$

Let's define one more function \mathbb{G} . We say $\mathbb{G} = 1$ if and only if $K_{\text{GAP } X} \geq (1 + \delta)\mathbb{E}[X]$ and 0 otherwise.

We also say $\mathbb{G} = 1$ if and only if $K_{\text{GAP } Y} \geq (1 + \delta)\mathbb{E}[Y]$ and 0 otherwise.

From class we can say that

$$\mathbb{E}[\mathbb{G}(x_1, x_2, \dots)] \leq e\sqrt{m}\mathbb{E}[\mathbb{G}(y_1, y_2, \dots)]$$

$$\Pr[K_{\text{GAP } X} \geq (1 + \delta)\mathbb{E}[X]] \leq e\sqrt{m} \Pr[K_{\text{GAP } Y} \geq (1 + \delta)\mathbb{E}[Y]]$$

Problem 3

?? marks

Suppose you have a set X and I have a set Y , each with m elements (for example, this could be our playlist of songs). We both create a Bloom filter for our sets using the same number of bits n and the same set of k hash functions. Determine the expected number of bits where our Bloom filters will differ, in terms of m, n, k , and $|X \cap Y|$. Explain how this could be used as a tool to find people with the same tastes (say, in music here) easily.

Solution: Let's say that bloom filter associated with X is B_X and Y is B_Y . Now for every entry in $X \cap Y$ B_X and B_Y will not differ.

The only difference will be when B_X is set 1 for all elements in $X - X \cap Y$ and 0 for elements in Y and vice versa.

Probability a particular bit is 1 in B_X is (from class)

$$\left[1 - \left(1 - \frac{1}{n} \right)^{k(|X| - |X \cap Y|)} \right]$$

Probability a particular bit is 0 in B_Y is (from class notes)

$$\left(1 - \frac{1}{n} \right)^{k|Y|}$$

Probability that they differ in this case for one entry in B_X and B_Y is (such that particular bit is 1 in B_X and 0 in B_Y)

$$\left(1 - \frac{1}{n} \right)^{k|Y|} * \left[1 - \left(1 - \frac{1}{n} \right)^{k(|X| - |X \cap Y|)} \right]$$

By symmetry probability that they differ in case such that particular bit is 0 in B_X and 1 in B_Y for one entry in B_X and B_Y is also

$$\left(1 - \frac{1}{n} \right)^{k|Y|} * \left[1 - \left(1 - \frac{1}{n} \right)^{k(|X| - |X \cap Y|)} \right]$$

Say an indicator random variable $\mathbb{J}_i = 1$ if i^{th} location in B_X and B_Y differs.

$$\Pr[\mathbb{J}_i = 1] = 2 \left(1 - \frac{1}{n} \right)^{k|Y|} * \left[1 - \left(1 - \frac{1}{n} \right)^{k(|X| - |X \cap Y|)} \right]$$

and say \mathbb{J} is total number of differences accross bloom filter B_X and B_Y . Then

$$\begin{aligned}
\mathbb{J} &= \sum_{i=0}^n \mathbb{J}_i \\
E[\mathbb{J}] &= E\left[\sum_{i=0}^n \mathbb{J}_i\right] \\
&= \sum_{i=0}^n E[\mathbb{J}_i] \\
&= \sum_{i=0}^n \Pr[\mathbb{J}_i = 1] \\
&= \sum_{i=0}^n 2 \left(1 - \frac{1}{n}\right)^{k|Y|} * \left[1 - \left(1 - \frac{1}{n}\right)^{k(|X| - |X \cap Y|)}\right] \\
&= 2n \left(1 - \frac{1}{n}\right)^{k|Y|} * \left[1 - \left(1 - \frac{1}{n}\right)^{k(|X| - |X \cap Y|)}\right]
\end{aligned}$$

Given $|X| = |Y| = m$ expected value of the number of differences in those 2 bloom filter is $2n \left(1 - \frac{1}{n}\right)^{km} * \left[1 - \left(1 - \frac{1}{n}\right)^{k(m - |X \cap Y|)}\right]$

Problem 4

?? marks

Consider the scenario of n autonomous agents in a distributed setting vying for a resource. Assume that there are n copies of the resource available, but an agent will be served by a copy of the resource if it is the only agent that has chosen that instance of the resource. If there are multiple agents that choose the same copy, then that copy gets blocked and the agents will have to wait for the next round. Our goal is to understand the number of rounds before all n agents get served.

Let us model this as a balls into bins process. Here the agents are the balls and the copies of the resource are the bins. In the first round, n balls are thrown independently and uniformly at random into n bins. After round i , we discard all balls that fell into a bin by itself in round i . We continue with the remaining balls in a similar fashion for round $i + 1$, where they are thrown independently and uniformly at random into n bins.

- (a) (2 marks) If there are b agents waiting to be served at the start of a round, what is the expected number of agents remaining at the start of the next round?

Solution: There is n resources and n agents. One resource is valid to serve if and only if one agent lands there.

Reducing the problem to balls and bins, we here see if we throw n balls in to n bin what is the expected number of bins that'll have 1 ball in it.

There are total b agents in the starting. We fix a bin i , the probability that one lands in bin i is $\frac{1}{n}$ and the probability that all the other balls land in other bins is $\left(1 - \frac{1}{n}\right)^{b-1}$.

$$\Pr[\text{One ball lands in bin } i] = \left(1 - \frac{1}{n}\right)^{b-1}$$

Say a indicator random variable x_i is such that if one ball lands in bin i then $x_i = 1$. Then we can say (X is the number of agents in this round)

$$\begin{aligned} X &= \sum_{i=0}^b x_i \\ \mathbb{E}[X] &= \sum_{i=0}^b \mathbb{E}(x_i) \\ &= \sum_{i=0}^b \Pr[x_i = 1] \\ &= \sum_{i=0}^b \left(1 - \frac{1}{n}\right)^{b-1} \\ &= b \left(1 - \frac{1}{n}\right)^{b-1} \end{aligned}$$

Agents to be served the next round is $b - b \left(1 - \frac{1}{n}\right)^{b-1}$.

- (b) (4 marks) Suppose that in every round the number of agents that are served is exactly the expected number. Show that all the balls would be served in $O(\log \log n)$ rounds.

Solution: I looked at a hint given in the [MU] book exercise for this. Say X_j is the number of agents served in the last round. We look at X_{j+1}

$$\begin{aligned} X_{j+1} &= X_j - X_j \left(1 - \frac{1}{n}\right)^{X_j-1} \\ &\leq X_j - X_j \left(1 - \frac{X_j-1}{n}\right) \end{aligned}$$

Through the approximation $(1 - x)^t \geq 1 - tx$

$$\begin{aligned}
X_{j+1} &\leq X_j - X_j \left(1 - \frac{X_j - 1}{n}\right) \\
&\leq \left(\frac{X_j^2 - X_j}{n}\right) \\
&\leq \frac{X_j^2}{n}
\end{aligned}$$

From the above equation we can see that in order to go from n to $\frac{n}{2}$ jobs we need to spend a constant r rounds.

$$\begin{aligned}
x_{j+1} &\leq \frac{X_j^2}{n} \\
x_{j+2} &\leq \frac{X_{j+1}^2}{n} \leq \frac{X_j^4}{n^3} \\
x_{j+3} &\leq \frac{X_{j+2}^2}{n} \leq \frac{X_j^8}{n^7}
\end{aligned}$$

So we can say for any general r , $x_{j+r} \leq \frac{\left(\frac{n}{2}\right)^{2^r}}{n^{2^r-1}}$

$$\begin{aligned}
x_{j+r} &\leq \frac{\left(\frac{n}{2}\right)^{2^r}}{n^{2^r-1}} \\
&\leq \frac{n}{2^{2^r}}
\end{aligned}$$

So the value x_{j+r} becomes 1 when r goes $O(\log \log n)$. This statement tells the fact that within $\log \log n$ + some constant j steps all the agents will be served $\equiv O(\log \log n)$.

- (c) (6 marks) Show that there is a constant c such that all the agents will be served within $c \log \log n$ rounds with probability at least $1 - o(1)$.

Solution: NO IDEA

Problem 5

?? marks

In this problem, our goal is to devise an algorithm for a packet routing problem on a connected undirected graph G . We want to route N packets whose source, destination, and the exact route through the graph G is given. In each time-step, the packet can either traverse an edge or wait at a node. Furthermore, at most one packet can traverse an edge at a given time-step.

A *schedule* for a set of packets specifies the timings for those packets, i.e. it specifies which packet should stay at a node and which should move for every time step. Our

goal is to design a schedule that minimizes the total time and the maximum queue size to route all packets to their destinations. We will denote by c , the congestion in the network, which is the maximum number of packets that must traverse a single edge in the network throughout the entire course of routing. By d , we denote the maximum distance travelled by any packet.

- (a) (4 marks) First consider the following *unconstrained schedule* where multiple packets are allowed to pass through an edge during one time-step: For a constant α , choose an integral delay independently and uniformly at random from the interval $[1, \lceil \alpha c / \log(Nd) \rceil]$ for each packet. If the delay is x , then the packet stays at the source for x time steps, and then gets routed on its path without any delay at any of the intermediate nodes.

Show that in this unconstrained schedule, the probability that more than $O(\log Nd)$ packets pass through any edge at any given time-step is at most $1/(Nd)$ for a sufficiently large α .

Solution: Informations:

- Schedule is unconstrained, more than one message/packets can pass through one edge.
- We are choosing one delay $x \in_r \left(1, \frac{\alpha c}{\log(Nd)}\right)$

I choose one edge $e \in G.\text{edges}$ and one time frame t . I'll look at number of packets passing through e at time frame t . If I find the expected number of packets through e at time t then I can set a tail bound on the expected number of packets. Using chernoff bound I can find the probability of finding $O(\log Nd)$ packets in the edge at a given time frame (which is the definition for congestion in the network).

Say X is a random variable indicating the number packets passing through a particular edge at a given time t . We have to find $\mathbb{E}(X)$

Say X_i is a indicator random variable taking value 1 if i^{th} packet passes through edge e at time t .

$$X = \sum_{i=0}^N X_i$$

$$\mathbb{E}(X) = \sum_{i=0}^N \mathbb{E}(X_i)$$

Now we analyse $\mathbb{E}(X_i)$

$$\mathbb{E}[X_i] = \Pr[X_i = 1]$$

$\Pr[X_i = 1]$ is when i^{th} packet is passing through edge e at time t , to do this the wait time of i^{th} packet must finish before t otherwise this packet will not be present at edge e at time t . More precisely Wait time W_i and delay $\hat{d} \in_r \left[1, \frac{\alpha c}{\log Nd}\right]$ should be such that $W_i + \hat{d} = t$

$$\begin{aligned}\mathbb{E}[X_i] &= \Pr[X_i = 1] \\ &= \Pr[W_i = t - \hat{d}] \text{ for some delay } \hat{d}\end{aligned}$$

This probability is $= \frac{1}{\frac{\alpha c}{\log Nd}}$ because there are $\frac{\alpha c}{\log Nd}$ many choices for W_i .

At worst case through the network atmost c packets passes in one edge. So $\mathbb{E}[X]$ should be summed till c .

$$\begin{aligned}\mathbb{E}[X_i] &= \Pr[X_i = 1] \\ &= \frac{\log Nd}{\alpha c}\end{aligned}$$

$$\begin{aligned}\therefore \mathbb{E}[X] &= \sum_{i=0}^c \mathbb{E}(X_i) \\ &\leq \sum_{i=0}^c \frac{\log Nd}{\alpha c} \\ &\leq \frac{\log Nd}{\alpha}\end{aligned}$$

From the question we are required to find the probability that random variable $X \geq c_1 \log Nd$. X is binomial random variable with mean $\mu = \mathbb{E}(X) = \frac{\log Nd}{\alpha}$. So applying Chernoff bound on $\mathbb{E}(X)$

$$\Pr[X \geq c_1 \log Nd] = \Pr \left[X \geq (1 + (c_1 \alpha - 1)) \frac{\log Nd}{\alpha} \right]$$

Here $\delta = (c_1 \alpha - 1)$ and $\mu = \frac{\log Nd}{\alpha}$. c_1 is the constant from $O(\log Nd)$.

$$\begin{aligned}\Pr[X \geq c_1 \log Nd] &= \exp \left(-\frac{\frac{\log Nd}{\alpha} (c_1 \alpha - 1)^2}{2 + (c_1 \alpha - 1)} \right) \\ &= \exp \left(-\frac{\frac{\log Nd}{\alpha} (c_1 \alpha - 1)^2}{(1 + c_1 \alpha)} \right)\end{aligned}$$

Choosing values $c_1 = 2$ and $\alpha = 2$ we get

$$\begin{aligned}\Pr[X \geq c_1 \log Nd] &= \exp\left(-\frac{\frac{\log Nd}{\alpha}(c_1\alpha - 1)^2}{(1 + c_1\alpha)}\right) \\ &= \exp\left(-\frac{\frac{\log Nd}{2}(4 - 1)^2}{(1 + 4)}\right) \\ &= \exp\left(-\frac{9}{10} \log Nd\right) \\ &\leq \frac{1}{Nd}\end{aligned}$$

Probability that $X \geq O(\log Nd)$ is at most $\frac{1}{Nd}$.

Now we need to do union bound over all the edges for all the time t . We need to get the upper bound on the number of paths in the network and the total max time taken in the network.

- (b) (4 marks) Use the unconstrained schedule from Part (a) to devise a randomized algorithm that, with high probability, produces a schedule of length $O(c+d \log(Nd))$ using queues of size at most $O(\log(Nd))$ such that at most one packet crosses an edge at every time-step.

Solution: Type your solution here.