COL 758: Advanced Algorithms

Lecture 07

Chernoff Bound (More usable forms)

Let $X_1, X_2, \ldots, X_n$ be n independent <u>Binary</u> random variables with parameters $p_1, p_2, \ldots, p_n$, that is

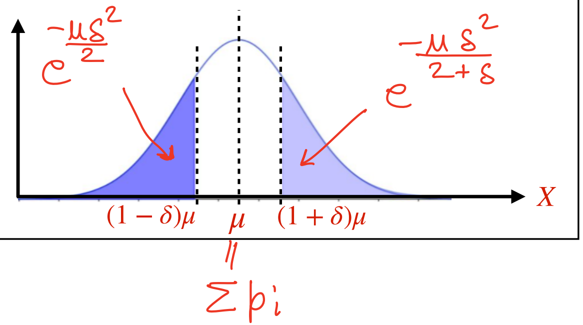
$$X_i = \begin{cases} 1 & \text{with probability } p_i \\ 0 & \text{with probability } 1 - p_i \end{cases}$$

Let X be the sum of random variables $X_1, X_2, \ldots, X_n$, and let $\mu = E(X) = \sum_{i=1}^{n} p_i$.

Then, for any $\delta > 0$,

$$P(X \geq (1+\delta)\mu) \;\leq\; e^{-\mu\delta^2/(2+\delta)}$$

$$P(X \leq (1-\delta)\mu) \;\leq\; e^{-\mu\delta^2/2}$$



Median Estimation

Given: A list L with n **elements.**

Find: Approximate median of elements of L, more precisely,

an element $x \in L$ satisfying: $\text{rank}(x) \in \left[\frac{n}{4}, \frac{3n}{4}\right]$.

How efficiently can you find such an element?

What is time complexity of best deterministic algorithm?

$\Omega(n)$

Algo 1 : Sort all the element — $O(n \log n)$

Algo 2 : (Median of median) — $O(n)$

Median Estimation

Given: A list L with n **elements.**

Find: Approximate median of elements of L, more precisely,

an element $x \in L$ satisfying: rank$(x) \in \left[\frac{n}{4}, \frac{3n}{4}\right]$.

How efficiently can you find such an element?

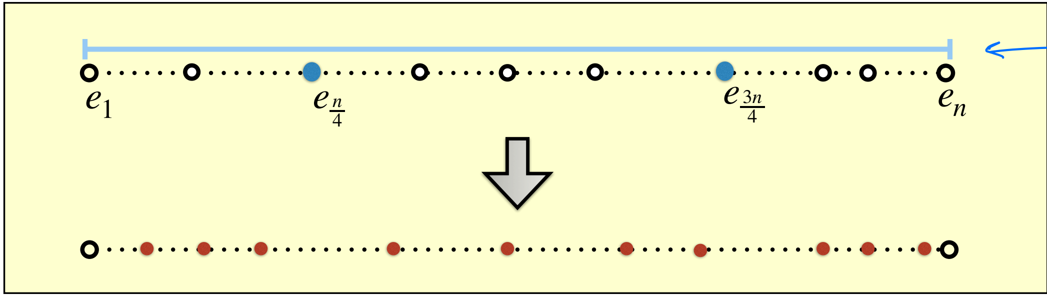
Randomized Algorithm

1. Take J to be a uniformly random subset of L of size $k = 16\log n$ (chosen with replacement).
2. Compute x = Median of J.
3. Output x.

Median Estimation

1. Take J to be a uniformly random subset of L of size $k = 16 \log n$ (chosen with replacement).
2. Compute x = Median of J.
3. Output x.

Random set J is well-distributed



elements of L in sorted order

$$\left\{\begin{array}{l} |J \cap \{e_1, \ldots, e_{\frac{n}{4}}\}| < 8 \log n \\ |J \cap \{e_{\frac{3n}{4}+1}, \ldots, e_n\}| < 8 \log n \end{array}\right\} \Rightarrow \text{Median of } J \in \left[e_{\frac{n}{4}}, e_{\frac{3n}{4}}\right]$$

Given: Set $A = \{e_1, e_2, \ldots, e_n\}$ with n elements,

Random set $J \subseteq A$ of size $k = 16 \log n$, elements picked uniformly randomly with replacement.

Find:

a. Expected size of $|J \cap \{e_1, \ldots, e_{\frac{n}{4}}\}|$.

b. Probability that $|J \cap \{e_1, \ldots, e_{\frac{n}{4}}\}| \geq 8 \log n$.

Solution (a):

For $i \in [1, k]$, define random variable

$$Y_i = \begin{cases} 1 & \text{if } i^{th} \text{element of } J \text{ lies in } [e_1, \ldots, e_{\frac{n}{4}}] \\ 0 & \text{otherwise} \end{cases}$$

So, $E[Y_i] = \cancel{1}\ 1/4$. Next define $Y = (Y_1 + \cdots + Y_k)$.

Then, $\cancel{|J \cap \{e_1, \ldots, e_{\frac{n}{4}}\}|} = E[Y] = \sum_{i=1}^{k} E[Y_i] = 4 \log n.$

$Y = |J \cap \{e_1 .. e_{\frac{n}{4}}\}|$

Solution (b):

Observe $Y_1, \ldots, Y_k$ are independent.

For $\delta = 1$:

$$P(Y \geq (1+\delta)E[Y]) \;\leq\; e^{-E[Y]\delta^2/(2+\delta)} = \; e^{-E[Y]/3}$$

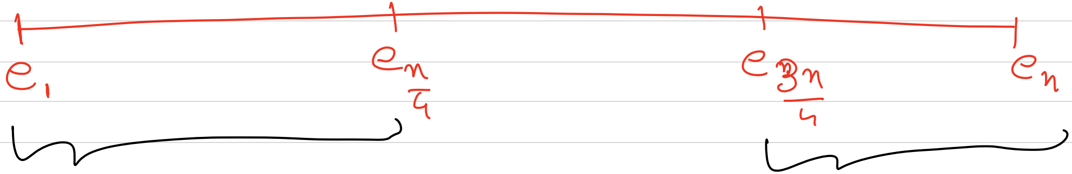
So,

$$P\big(|J \cap \{e_1, \ldots, e_{\frac{n}{4}}\}| \geq 8 \log n\big) \leq \; = \; e^{-\frac{4}{3}\log n} = \frac{1}{n^{4/3}}$$

Claim: Our algo succeeds with $\geq \left(1 - \frac{1}{n}\right)$ probability.

Proof:

$$|J| = k = 16 \log n$$



More than 50% of elements of J lie in this range

Prob is $\leq \frac{1}{n^{4/3}}$

$$\underbrace{|J \cap \{e_1, \ldots e_{\frac{n}{4}}\}| > 8 \log n}_{\text{Bad evet } E_1}$$

$$\underbrace{\text{More than } 50\% \text{ of } J \text{ lie here}}_{\text{Bad event } E_2} \leq \frac{1}{n^{4/3}}$$

$$\text{Prob}(E_1 \text{ or } E_2 \text{ happens}) \leq P(E_1 \text{ happens}) + P(E_2 \text{ happens}) \leq \frac{2}{n^{4/3}} \leq \frac{1}{n}$$

Homework I - Estimating Median with ϵ-error

Given: A list L with n **elements.**

Find: Approximate median of elements of L satisfying:

$$\text{rank}(x) \text{ lies in range } \left[(1-\epsilon)\frac{n}{2},\ (1+\epsilon)\frac{n}{2}\right]$$

How efficiently can you find such an element?

Algorithm

1. Take J to be a uniformly random subset of L of size $k =$ __________ (chosen with replacement).
2. Compute x = Median of J.
3. Output x.

Homework II - Estimating r^{th}-rank element with ϵ-error

Given: A list L with n **elements.**

Find: Approximate r^{th}-rank elements of L satisfying:

rank(x) lies in range $\left[(1-\epsilon)r,\ (1+\epsilon)r\right]$

How efficiently can you find such an element?

Algorithm

1. Take J to be a uniformly random subset of L of size $k =$ _________ (chosen with replacement).
2. Compute x = r^{th}-rank element of J.
3. Output x.

Puzzle: Estimating Set Size

There are p players playing the following game:

Game: Each player spins a wheel to generate a random number in $[0,1]$, and winner is the one with least score.

Rounds: k

Available information: Winning scores in all k rounds.

How to efficiently predict "p"?

Winning Scores

0.2

0.1

0.3 0.2 0.25

0.7 0.8 0.1

Puzzle: Estimating Set Size

There are p players playing the following game:

Game: Each player spins a wheel to generate a random number in $[0,1]$, and winner is the one with least score.

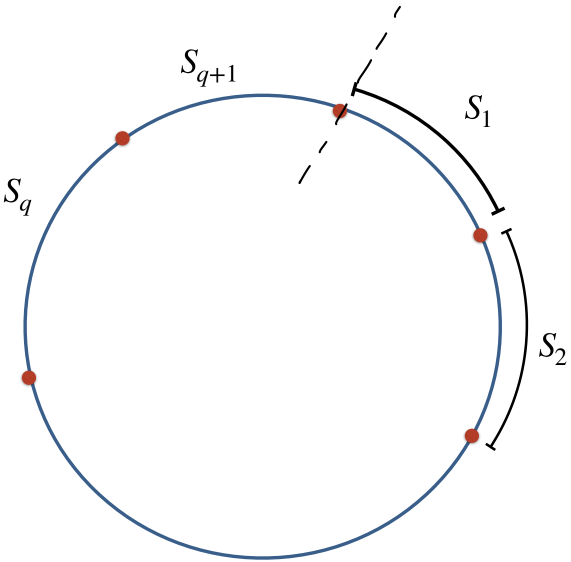
Rounds: k

Available information: Winning scores in all k rounds.

How to efficiently predict "p"?

Ques: What is expected value of winning score in each round?

Sampling $(q+1)$ points on a Circle of circumference 1:



$E[|S_1|] = E[|S_2|] = \dots = E[|S_{q+1}|]$

and $|S_1| + \dots + |S_{q+1}| = 1$

Thus, $E[|S_i|] = \frac{1}{q+1}$

Sampling q points on interval $[0,1]$

Since $(S_1 \dots S_{q+1})$ in both cases have identical distribution, we get that

$$E[|S_1|] = \frac{1}{q+1}$$