Lecture Notes CS:5360 Randomized Algorithms
Lectures 12 and 13: Sep 27 and Oct 2, 2018
Scribe: Cory Kromer-Edwards

1 Chernoff Bounds

The most commonly used tail bounds. They can be much more powerful than Markov and Chebyshev.

Setting: Let $X_1, X_2, \ldots, X_n$ be mutually independent binary random variables. Let $Pr(X_i = 1) = P_i$ for $i = 1, 2, \ldots, n$. Let $X = \sum_{i=1}^{n} X_i$. Let use denote $E[X] = \mu$.

Note: $\mu = E[X] = \sum_{i=1}^{n} E[X_i] = \sum_{i=1}^{n} P_i$

Chernoff Bounds

(a) For any $\delta \geq 0$,

$$Pr(X \geq (1+\delta)\mu) \leq (\frac{e^{\delta}}{(1+\delta)^{1+\delta}})^{\mu}$$

(b) For $0 \leq \delta \leq 1$,

$$Pr(X \geq (1+\delta)\mu) \leq e^{-\mu\delta^2\frac{1}{3}}$$

(c) For $R \geq 6\mu$,

$$Pr(X \geq R) \leq 2^{-R}$$

Note about a-c: All upper tail bounds

Example: Coin tossing Let X = number of heads we get when we toss n fair coins independently.

$X = X_1 + X_2 + \ldots + X_n$ where

$$X = \begin{cases} 1 & \text{if ith coin toss = Heads} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

$\mu = E[X] = \frac{n}{2}$
What is $Pr(X \geq \frac{3}{4}n) = Pr(X \geq (1+\frac{1}{2})\frac{n}{2})$?
$\delta = \frac{1}{2}$, so using form (b) we get:

$$Pr(X \geq \frac{3}{4}n) \leq e^{-\frac{n}{2}*\frac{1}{4}*\frac{1}{3}} = e^{-\frac{n}{24}}$$

Recall

- MI: $O(1)$
- Chebyshev: $O(\frac{1}{n})$
- Chernoff: $O(\frac{1}{exp(n)})$

What is $Pr(X \geq \frac{n}{2} + \frac{1}{2}\sqrt{6nlnn})$? Remember: $\mu = \frac{n}{2}$, so:

$$= Pr(X \geq (1 + \sqrt{\frac{6nlnn}{n}})\frac{n}{2})$$

In this case, $\delta = \sqrt{\frac{6nlnn}{n}}$ therefore by form (b) of Chernoff Bounds,

$$Pr(X \geq \frac{n}{2} + \frac{1}{2}\sqrt{\frac{6nlnn}{n}}) \leq e^{-\frac{n}{2} * \frac{6nlnn}{n} * \frac{1}{3}} = e^{-lnn} = \frac{1}{n}$$

This means that the number of heads stays really close to $\frac{n}{2}$, and gets closer as n increases.

Example: Probability Amplification for BPP algorithms

Recall: BPP = class of decision problemsX such that there is a Polynomial time Monte Carlo algorithm A for X:

- if x is a yes-instance of X then $Pr(A(x) = 1) \geq \frac{2}{3}$
- if x is a no-instance of X then $Pr(A(x) = 0) \geq \frac{2}{3}$

Theorem 1 *if X ϵ BPP then there exists a Polynomial time Monte Carlo algorithm A' for X:*

- *if x is a yes-instance of X then* $Pr(A'(x) = 1) \geq 1 - \frac{1}{e^{|x|}}$
- *if x is a no-instance of X then* $Pr(A'(x) = 0) \geq 1 - \frac{1}{e^{|x|}}$

Where $|x|$ is the input size.

Algorithm A' on input x: Repeat A(x) k times and output the majority answer.

Example: Let x be a yes-instance of X. Let Y be the number of "no" answers returned by A. For $i = 1, 2, \ldots, k$, let

$$Y_i = \begin{cases} 1 & \text{if A(x) = 0 when A is called the ith time} \\ 0 & \text{if A(x) = 1 when A is called the ith time} \end{cases} \quad (2)$$

Then, $Y = \sum_{i=1}^{k} Y_i$.
Since the k calls are independent of each other, the Y_i's are mutually independent.

$E[Y] = \sum_{i=1}^{k} E[Y_i] \leq \frac{k}{3}$
A' will output an incorrect answer if the number of "no"s returned by k calls to a is $\geq \frac{k}{2}$

$$\rightarrow \text{A' output incorrect answer} = Y \geq \frac{k}{2} \rightarrow \text{our goal is to upper bound } Pr(Y \geq \frac{k}{2})$$

$$Pr(Y \geq \frac{k}{2}) = Pr(Y \geq (1+\frac{1}{2})\frac{k}{3}) \leq Pr(Y \geq (1+\frac{1}{2})\mu) \leq e^{-\mu * \frac{1}{4} * \frac{1}{3}}$$

We cannot plug $\frac{k}{3}$ in for μ because $Y \leq \frac{k}{2}$ instead of $Y = \frac{k}{2}$. So, let Z_i be a binomial random variable such that $Pr(Z_i = 1) = \frac{1}{3}$. Let $Z = \sum_{i=1}^{k} Z_i$ therefor, $E[Z] = \frac{k}{3}$

Claim: $Pr(Y \geq \frac{k}{2}) \leq Pr(Z \geq \frac{k}{2})$ Z stochastically dominates Y.

$$Pr(Z \geq \frac{k}{2}) = Pr(Z \geq (1+\frac{1}{2})\frac{k}{3}) \leq Pr(Z \geq (1+\frac{1}{2})\mu) \leq e^{-\mu * \frac{1}{4} * \frac{1}{3}} \leq e^{-\frac{k}{36}}$$

Set $k = 36|x|$. Then, $Pr(Z \geq \frac{k}{2}) \leq e^{-|x|}$

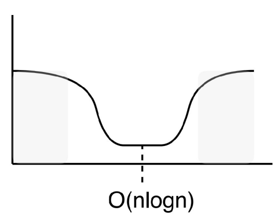
Notes:

- Even though $k = 36|x|$, the running time of A' is polynomial in $|x|$
- x being a no-instance is symmetric

Example: High Probability Analysis of Randomized QuickSort

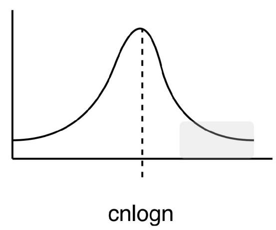
Recall: We showed a Las Vegas algorithm (randomized QuickSort) with expected running time O(nlogn).

The time could look like:



We will show that for some constant c,

$$Pr(\text{running time of QuickSort} \geq cnlogn) \leq \frac{1}{n}$$



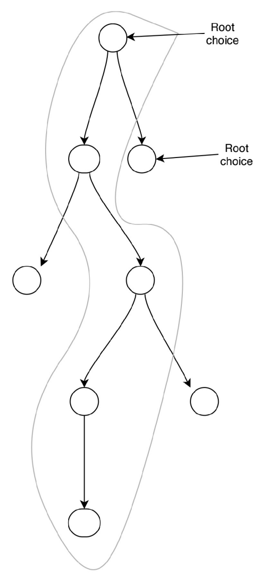
Recall: Before we showed

$$X_ij = \begin{cases} 1 & \text{if } y_i \text{ and } y_j \text{ are compared} \\ 0 & \text{Otherwise} \end{cases} \quad (3)$$

let $X = \sum_{i,j} x_{ij}$ then, $E[x] = O(nlogn)$

We cannot use Chernoff bounds with this because X_{ij}'s are not mutually independent.

Setup: Consider the recurison tree of randomized QuickSort



node is good = pivot is in middle third of input

Questions:

1. How many good nodes can there be in a root-leaf path?
   ANS: $\leq c - log_2 n$

2. Can you show by Chernoff Bounds that a root-leaf path cannot be too long, $e_i \geq 10c * log_2 n$
   $Prob \leq \frac{1}{n^2}$