EECS 495: Randomized Algorithms **Lecture 6**
Moments and Deviations

Reading: Text: Chapter 3

Take-Aways

Concepts/Techniques:

- Tail bounds: Markov, Chebyshev, Chernoff

Applications:

- Selecting k'th smallest elt
- Routing in hypercube

Tail Bounds

Claim: Markov Inequality: Let Y be a non-neg. random var. Then for all $t \in \Re^+$,

$$\Pr[Y \geq t] \leq E[Y]/t,$$

or, equivalently,

$$\Pr[Y \geq kE[Y]] \leq 1/k.$$

Proof: Define

$$f(y) = \begin{cases} 1 & : y \geq t \\ 0 & : y < t \end{cases}$$

Then since $f(y) \leq y/t$ for all y,

$$\begin{aligned} \Pr[Y \geq t] &= E[f(Y)] \\ &\leq E[\frac{Y}{t}] \\ &= \frac{E[Y]}{t}. \end{aligned}$$

Claim: Let X be a random var. with mean μ and variance $\sigma^2 = E[(X-\mu)^2]$. Then for any $t \in \Re^+$,

$$\Pr[|X-\mu| \geq t\sigma] \leq \frac{1}{t^2}.$$

Proof: Note that

$$\Pr[|X-\mu| \geq t\sigma] = \Pr[(X-\mu)^2 \geq t^2\sigma^2].$$

The random var. $Y = (X-\mu)^2$ has expectation σ^2 so result follows by Markov.

Randomized Selection

Given:

- An unordered set S of n elts
- A total order on elts of S

Output:

- k'th smallest elt of S

Question: Algorithms?

- Sort plus binary search: $O(n \log n)$
- Random pivot: $O(n)$

Idea: Take sample mean to narrow search.

Algorithm: LazySelect:

1. Pick $n^{3/4}$ elts R from S with replacement
2. Sort R
3. Find fences:
   (a) Let $x = \frac{k}{n}^{3/4} = kn^{-1/4}$
   (b) Pick left post $l = x - \sqrt{n}$, right post $h = x + \sqrt{n}$
   (c) Let $a = R_{(l)}$, $b = R_{(h)}$
   (d) Find $r_S(a)$ and $r_S(b)$ by comparing S to a, b
4. Project S to smaller space and solve:
   (a) Let $P = \{y \in S | a \leq y \leq b\}$
   (b) If $S_{(k)} \in P$ and $|P| \leq 4n^{3/4} + 2$, sort P and find $S_{(k)}$
   (c) Else start over

[[*Sample with replacement to make analysis easier.*]]

Claim: In each pass, LazySelect performs $2n + o(n)$ comparisions.

Proof: # of comparisons:

- sort sample: $O(n^{3/4} \log n) = o(n)$
- project: $2n$
- sort projection: $O(n^{3/4} \log n) = o(n)$

Claim: With probability $1 - O(n^{-1/4})$, LazySelect performs just one pass.

Note: Two failure modes:

- fences don't suround $S_{(k)}$
- $|P|$ is too big

Proof: Bound prob. bad fences because $a > S_{(k)}$:

Let $X_i = 1$ if i'th sample at most $S_{(k)}$, $X = \sum_{i=1}^{n^{3/4}} X_i$ be # samples less than $S_{(k)}$.

- $\mu(X_i) = \Pr[X_i = 1] = k/n$
- $\mu(X) = \sum_{i=1}^{n^{3/4}} \mu(X_i) = kn^{-1/4}$
- $\sigma^2(X_i) = \left(\frac{k}{n}\right)\left(1 - \frac{k}{n}\right) \leq \frac{1}{4}$
- $\sigma^2(X) = \sum_{i=1}^{n^{3/4}} \sigma^2(X_i) \leq \frac{n^{3/4}}{4}$

So $\sigma(x) \leq n^{3/8}/2$.

Apply Chebyshev:

$$\begin{aligned} \Pr[a > S_{(k)}] &= \Pr[X \leq kn^{-1/4} - \sqrt{n}] \\ &\leq \Pr[|X - \mu| \geq \sqrt{n}] \\ &\leq O(n^{-1/4}) \end{aligned}$$

since $t\sigma = \sqrt{n}$ implies $t = n^{1/8}/2$.

Similarly, $\Pr[b < S_{(k)}] \leq O(n^{-1/4})$ so by union bound, fences surround whp.

Other failure mode, similar analysis.

Tail Bounds

Claim: Chernoff Bound: Let $X_1, \ldots, X_n$ be independent Poisson trials with $\Pr[X_i = 1] = p_i$, where $0 < p_i < 1$, $X = \sum_{i=1}^{n} X_i$, $\mu = E[X]$, $0 < \delta \leq 1$,

$$\Pr[X < (1 - \delta)\mu] < \exp(-\mu\delta^2/2).$$

Alternatively, for $\delta > 2e - 1$,

$$\Pr[X > (1 + \delta)\mu] < \exp(-(1 + \delta)\mu).$$

Proof:

- exponentiate to study moment generating function $\exp(tX)$

- apply Markov to moment generating function and study $E[\exp(t\sum X_i)] = E[\prod \exp(tX_i)]$
- use independence to turn $E[\prod ...]$ into $\prod E[...]$
- pick t to get tightest possible bound

$\left[\left[\textit{Can only apply when variables are independent!}\right]\right]$

Routing in a Parallel Computer

Given:

- network of parallel processors represented by directed graph
  - nodes – processors
  - edges – communication links
- destination $d(i)$ for packet originating at processor i

Output:

- set of routes with min steps to get all packets to destinations

Note: Restrictions/assumptions:

- packets sent in sequence of synchronous steps
- at most one packet per link in each step
- permutation routing: $\{d(i)\}$ are a permutation of $\{1, \ldots, N\}$
- oblivious routing: route for packet of node i depends only on $d(i)$ and not $d(j)$ for $j \neq i$

Claim: Any deterministic alg. takes $\Omega(\sqrt{N/d})$ steps where d is out-degree of nodes.

Hypercube

[[*Popular network for parallel processing*]]

- N nodes represented as bit strings, $n = \log N$ dimensions
- Nn directed edges, from node i to node j iff $(i_0, \ldots, i_{n-1})$ and $(j_0, \ldots, j_{n-1})$ differ in exactly one position

Question: How many steps to route?

- paths of length n, so at least n steps
- Nn edges for N length n paths, no implied congestion bound

Algorithm: Natural routing: bit fixing (left to right)

Example: 1011 to 1100:

$$1011 \rightarrow 1111 \rightarrow 1101 \rightarrow 1100$$

Claim: Bit-fixing takes $\Omega(\sqrt{N})$ steps.

Proof: Transpose permutation: for each i, let $i = a_i \circ b_i$ where $|a_i| = |b_i| = n/2$. Then set

$$d(i) = b_i \circ a_i.$$

Consider nodes i such that $a_i = 0$ and b_i is odd.

- destination: $b_i \circ 0$
- at some point must pass from $1 \circ 0$ to $0 \circ 0$
- so $2^{n/2}/2$ such i, so at least $\Omega(\sqrt{N})$ steps