{ csci 3|6907 | Lecture 3 }

Hoeteck Wee · hoeteck@gwu.edu

Announcements

- Homework 2 to be out by Fri
- online form on course webpage

PART I | tail bounds

the case against expectation

QUESTION. How early should I arrive at the airport?

STATISTIC. The expected security wait time is 30 mins.

- perhaps... 50% : wait is 5 mins, 50% : wait is 55 mins
- more meaningful: 99% : wait $\leq$ 35 mins

  interpretation: if I arrive at airport 45 mins early, I'll miss one flight in every 100 flights I take.

PHILOSOPHY.

" If you've never missed a flight, you're spending too much time in airports. "

TAIL BOUNDS.

"with high prob., a r.v. X assumes values close to $E[X]$."

Tail bounds, I

Markov's Inequality

Let X be a non-negative r.v. Then, for all $a > 0$,

$$\Pr[X \geq a] \leq \mathrm{E}[X]/a$$

- EXAMPLE: $\mathrm{E}[\text{wait}] = 30\,\text{mins} \implies \Pr[\text{wait} \geq 5\,\text{hrs}] \leq \frac{30}{5 \cdot 60} = 0.1$
- PROOF:

$$\begin{aligned} \mathrm{E}[X] &= \sum_{i=0}^{\infty} i \Pr[X = i] \\ &= \sum_{0 \leq i < a} i \Pr[X = i] + \sum_{i \geq a} i \Pr[X = i] \\ &\geq 0 + \sum_{i \geq a} a \Pr[X = i] = a \cdot \Pr[X \geq a] \end{aligned}$$

Tail bounds, II

Chebyshev's Inequality

For any $a > 0$,
$$\Pr\left[\left|X - \mathrm{E}[X]\right| \geq a\right] \leq \mathrm{Var}[X]/a^2$$

- EXAMPLE: suppose $\mathrm{Var}[\text{wait}] = 5\,\text{mins}^2$. Then,
$$\Pr[|\text{wait} - 30| \geq 10] \leq \frac{5}{10^2} = 0.05$$
$$\implies \quad 95\%\text{: wait between 20 and 40 mins}$$

- PROOF: apply Markov's to the non-negative r.v. $\Upsilon = (X - \mathrm{E}[X])^2$
$$\Pr[\Upsilon \geq a^2] \leq \mathrm{E}[\Upsilon]/a^2 = \mathrm{Var}[X]/a^2$$

- COROLLARY: $\Pr[X \geq E[X] + a] \leq \mathrm{Var}[X]/a^2$

Example: coin flips

- X : # heads in a sequence of n independent flips of an unbiased coin.
- $X \sim B(n, \frac{1}{2})$, so $\mathrm{E}[X] = \frac{n}{2}$ and $\mathrm{Var}[X] = \frac{n}{4}$.
- By Markov's, $\Pr[X \geq \frac{3n}{4}] \leq \frac{2}{3}$

  $n = 200$: 33% chance # heads less than 150
- By Chebyshev's, $\Pr[|X - \frac{n}{2}| \geq \frac{n}{4}] \leq \frac{n}{4}/(\frac{n}{4})^2 = \frac{4}{n}$.

  $n = 200$: 98% chance # heads between 50 and 150
- In fact, can replace $\frac{4}{n}$ with $2^{-\Omega(n)}$!

  $n = 200$: 99.95% chance # heads between 50 and 150

  exploit full independence, c.f. Chernoff bound next week

Comparison of tail bounds

- GENERALITY: Markov's $\gg$ Chebyshev's

  (non-negative · bounded variance)

- "ERROR": Markov's $\ll$ Chebyshev's

  (constant · $1/\text{poly}$)

- "DEVIATION": Markov's $\ll$ Chebyshev's

  (one-sided · two-sided)

PART 2 | randomized median finding

Median Finding

MEDIAN FINDING Problem

Input: a set S of n values from some totally ordered universe

Goal: output the median element m of S

- WHAT'S KNOWN: "easier" than sorting – there is a deterministic linear-time algorithm.
- TODAY: a simple randomized $O(n)$ time algorithm
- WARM-UP: approximate median finding in $O(n)$ time

  GOAL: output x s.t. $|\operatorname{rank}_S(x) - n/2| \leq \delta n$

  (e.g. $\delta = 0.1$ or $\delta = \frac{1}{\sqrt{n}}$)

  NOTE: allow algorithm to err with small probability

Approximate Median Finding

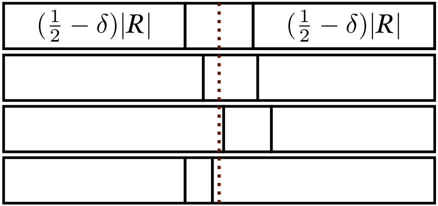
Approx Median Finding Problem

Input: a set S of n values from some totally ordered universe

Goal: output x in S such that $|\operatorname{rank}_S(x) - n/2| \leq \delta n$

too small $(\frac{1}{2} - \delta)n$	good	too big $(\frac{1}{2} - \delta)n$	S

- IDEA. pick a *small* random subset R of S



median(R) is good

median(R) is good

median(R) is too small

median(R) is too big

Approximate Median Finding

Approx Median Finding Problem

Input: a set S of n values from some totally ordered universe

Goal: output x in S such that $|\,\text{rank}_S(x) - n/2| \leq \delta n$

too small $(\frac{1}{2} - \delta)n$	good	too big $(\frac{1}{2} - \delta)n$	S

- IDEA. pick a *small* random subset R of S, where $|R| \leq n/\log n$
- HOPE. with prob ≈ 1, median element in R is good
- QUESTION. how to find median element in R?
- RUNNING TIME. sort in $O(|R| \log |R|) = O(n)$ time

Approximate Median Finding

Approx Median Finding Algorithm

Input: A list S of n distinct values

1. Pick a random subset R in S with replacement where $|R| \le n/\log n$.
2. Sort R and output median element x in R.

too small $(\frac{1}{2} - \delta)n$	good	too big $(\frac{1}{2} - \delta)n$

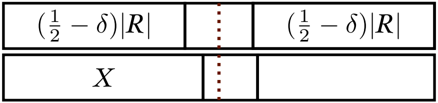
- HOPE. show $\Pr[x \text{ is good }]$ is big.
- FACT. $\Pr[x \text{ is good }] + \Pr[x \text{ is too small }] + \Pr[x \text{ is too big }] = 1$.
- GOAL. show $\Pr[x \text{ is too small }]$ is small.
- let X be r.v. for # elements in R that are too small
- FACT. x is too small $\Leftrightarrow X \ge |R|/2$.

Approximate Median Finding

Approx Median Finding Algorithm

Input: A list S of n distinct values

1. Pick a random subset R in S with replacement where $|R| \leq n/\log n$.
2. Sort R and output median element x in R.



median(R) is good

$\mathrm{E}[X] =$

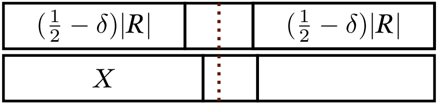
- let X be r.v. for # elements in R that are too small
- FACT. x is too small $\Leftrightarrow X \geq |R|/2$.

Approximate Median Finding

APPROX MEDIAN FINDING Algorithm

Input: A list S of n distinct values

1. Pick a random subset R in S with replacement where $|R| \leq n/\log n$.
2. Sort R and output median element x in R.



median(R) is good

$\mathrm{E}[X] = (\frac{1}{2} - \delta)|R|$

- GOAL. show $\Pr[X \geq |R|/2]$ is small.
- FACT. $X \sim B(|R|, \frac{1}{2} - \delta)$
- $\mathrm{E}[X] = (\frac{1}{2} - \delta)|R|$ and $\mathrm{Var}[X] \leq \frac{1}{4}|R|$.
- Chebyshev's $\Rightarrow \Pr[X \geq E[X] + \delta|R|] \leq \mathrm{Var}[X]/(\delta|R|)^2 \leq 1/(4\delta^2|R|)$

Approximate Median Finding

APPROX MEDIAN FINDING Algorithm

Input: A list S of n distinct values

1. Pick a random subset R in S with replacement where $|R| \leq n/\log n$.
2. Sort R and output median element x in R.

THEOREM. $\Pr[\,|\text{rank}_S(x) - n/2| \leq \delta n\,] \geq 1 - 1/(2\delta^2|R|)$

QUESTION. how to choose $|R|$ to achieve correctness prob ≥ 0.9?

set $1/(2\delta^2|R|) \leq 0.1 \Rightarrow |R| \geq 5/\delta^2$

Randomized Median Finding

RandMedian Algorithm

Input: A list S of n distinct values

1. Find ℓ from S such that $\text{rank}_S(\ell) \approx n/2 - 2n^{3/4}$.
2. Find u from S such that $\text{rank}_S(u) \approx n/2 + 2n^{3/4}$.
3. By comparing with each value in S, compute

   $C = \{y \in S \mid \ell \leq y \leq u\}$;
4. Sort C and output $(\frac{1}{2}n - \text{rank}_S(\ell) + 1)$'th smallest element in C.

- EXAMPLE: $n = 10,001$, $\text{rank}_S(\ell) = 3101$, $\text{rank}_S(u) = 7100$
- sort C where $|C| = 4000$, output element in C with rank 1900

Randomized Median Finding

RandMedian Algorithm

Input: A list S of n distinct values

1. Find ℓ from S such that $\text{rank}_S(\ell) \approx n/2 - 2n^{3/4}$.
2. Find u from S such that $\text{rank}_S(u) \approx n/2 + 2n^{3/4}$.
3. By comparing with each value in S, compute
   $C = \{y \in S \mid \ell \leq y \leq u\}$;
4. Sort C and output $(\frac{1}{2}n - \text{rank}_S(\ell) + 1)$'th smallest element in C.

- CORRECTNESS: $\text{rank}_S(\text{output}) = \text{rank}_C(\text{output}) + (\text{rank}_S(\ell) - 1)$.
- FACT: $|C| \approx 4n^{3/4}$, can sort in $O(n)$ time.
- GOAL: implement steps 1, 2 in $O(n)$ time.

Randomized Median Finding

RandMedian Subroutine

1. Sample ℓ from S such that $\text{rank}_S(\ell)/n \in [\frac{1}{2} - 2\delta, \frac{1}{2}]$;
    1.1 Pick a random subset R of $n^{3/4}$ elements in S with replacement.
    1.2 Output the element x in R whose rank is $(\frac{1}{2} - \delta)|R|$.