Randomized Algorithms III

Concentration

Pat Morin

Carleton University

University of Sydney

National ICT Australia

Types of Randomized Algorithms

Las-Vegas Algorithm: Running time is a random variable

Monte-Carlo Algorithm: Is correct with probability <1

In both cases, randomization is over the actions of the algorithm, not the input.

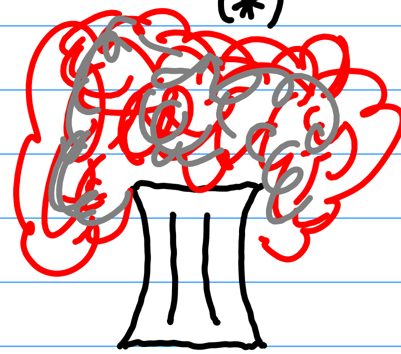
Concentrating Running Time

Markov's Inequality: For any non-negative random variable X

$$\Pr\{X \geq t \cdot E[X]\} \leq 1/t$$

Let A be a Las-Vegas Algorithm, and let X be the running time of A. Then

$$\Pr\{X \geq t \cdot E[X]\} \leq 1/t. \qquad (*)$$



Question: What if (*) is not good enough?

Concentrating A's Running Time

$E[X]$ = expected running time of A

ConcentrAte

```
Repeat forever:
    Run A for 2·E[X] steps
    if A is done return value output by A
    else restart A from scratch.
```

#iterations is dominated by a geometric(½) random variable

$$E[\#\text{iterations}] \le 2$$

$E[\text{running time of ConcentrAte}] \le 2 \cdot E[X]$ ← A little slower

$\Pr\{\text{running time of ConcentrAte} \ge 2 \cdot t\, E[X]\} \le 1/2^t$ ← much more concentrated.

Success Amplification

$p = \text{Prob}\{\text{correct answer}\}$

Yes-Biased Algorithm: An output of 'yes' is always correct

No-Biased Algorithm: An output of 'no' is always correct

Optimization Algorithm: Result is always $\leq$ the optimum

The probability of correctness can be made

$$1 - (1-p)^K$$

by running the algorithm K times.

Monte-Carlo to Las-Vegas

$\boxed{p = \text{Prob}\{\text{correct answer}\}}$

If, in addition to Monte-Carlo algorithm A, we have an algorithm B that can test if the output is correct, then

```
Las Vegas A:
    repeat
        Run A
    until the output of A is correct
```

Las Vegas A is always correct.

Expected running time of Las Vegas A is

$$\frac{\text{Running time of } A + \text{Running time of } B}{p}$$

Chernoff's Bounds

Let $X_i = \begin{cases} 1 & \text{with prob. } p \\ 0 & \text{with prob. } 1-p \end{cases}$

Let $B = \sum_{i=1}^{m} X_i$ where $X_1, \ldots, X_m$ are independent.

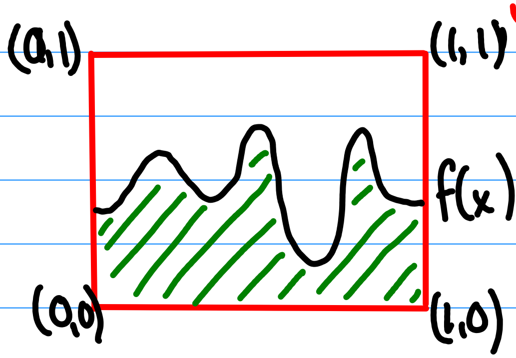
Then

$$\Pr\{B \geq (1+\varepsilon)mp\} \leq 1/e^{\varepsilon^2 mp/3}$$

$$\boxed{mp = E[B]}$$

$$\Pr\{B \leq (1-\varepsilon)mp\} \leq 1/e^{\varepsilon^2 mp/2}$$

Monte-Carlo Integration



Let $p_i = (x_i, y_i)$ be a random point in $[0,1]^2$

Let $I_i = \begin{cases} 1 & \text{if } y_i \le f(x_i) \\ 0 & \text{otherwise} \end{cases}$

Let $B = \left(\sum_{i=1}^{m} I_i\right) / m$

$E[I_i] = \int_0^1 f(x)\,dx \overset{\text{def}}{=} \mu$, $E[B] = \mu$

$\Pr\{B \ge (1+\varepsilon)\mu\} \le \exp(-\varepsilon^2 \mu m / 3)$

$\Pr\{B \le (1-\varepsilon)\mu\} \le \exp(-\varepsilon^2 \mu m / 2)$

Gives $O(m)$ time algorithm to approximate $\int_0^1 f(x)\,dx$.

Landslide Finding

Suppose A is a Monte-Carlo algorithm that is correct with probability 2/3.

Run A K times and output the most frequent answer

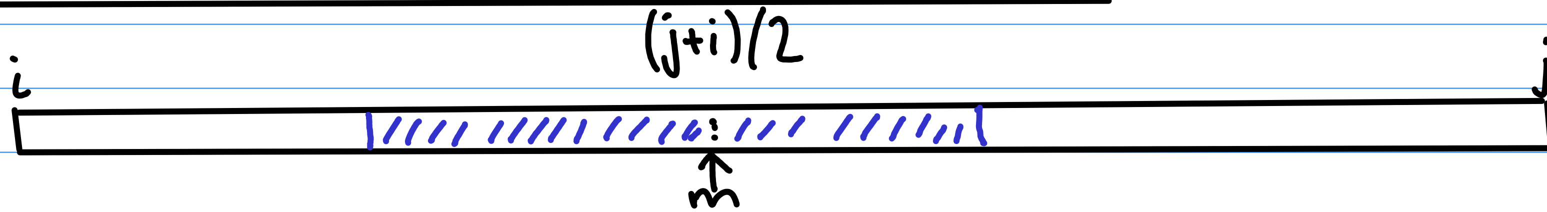
Let $I_i = \begin{cases} 1 & \text{if } A \text{ is correct on iteration } i \\ 0 & \text{otherwise} \end{cases}$, $B = \sum_{i=1}^{K} I_i$

$$\Pr\{B \le \tfrac{1}{2}K\} = \Pr\{B \le (1-\tfrac{1}{4})\tfrac{2}{3}K\} \le \exp(-2K/96)$$

∴ This works for any Monte-Carlo algorithm whose success probability is $p \ge \frac{1}{2} + \varepsilon$ for constant $\varepsilon > 0$.

Random Binary Search

```
RandomBinarySearch(i, j)
    if i > j return i
    p ← random(i, ..., j)
    if A_p = x then return p
    if A_p > x then return Search(i, p-1)
    if A_p < x then return Search(p+1, j)
```



Success: p is within distance $j-i/4$ of m

Success implies $j'-i' \leq (j-i)\cdot 3/4 \Rightarrow$ At most $\log_{4/3} n$ successes needed

$\Pr\{\text{success}\} = \frac{1}{2}$

Let $I_i = \begin{cases} 1 & \text{if } i\text{th step is a success} \\ 0 & \text{otherwise} \end{cases}$

Let $B_k = \sum_{i=1}^{k} I_i$, let $K = (c \cdot 2 \log_{4/3} n)/(1+\varepsilon)$

$$\Pr\{B_K \leq \log_{4/3} n\} \leq \exp(-\varepsilon^2 K/6) = (1/n)^{\varepsilon^2 \Omega(c))}$$

But, if $B_K \geq \log_{4/3} n$ then $i - j \leq 1$

Theorem: The probability that Random Binary Search requires more than $c \cdot 2 \log_{4/3} n)/(1+\varepsilon)$ iterations is at most

$$(1/n)^{\varepsilon^2 \Omega(c))}$$

Theorem: The probability that a RBST performs more than $c \cdot 2\log_{4/3} n)/(1+\varepsilon)$ comparisons when searching for a particular element is at most

$$(1/n)^{\varepsilon^2 \Omega(c))}$$

Theorem: The probability that the height of a RBST is more than $c \cdot 2\log_{4/3} n$ is at most

$$(1/n)^{\varepsilon^2 \Omega(c))}$$

McDiarmid's Inequality

Let $X_1, \ldots, X_n$ be independent random variables.

Let $f: \mathbb{R}^n \to \mathbb{R}$ be such that

$$|f(x_1, x_2, \ldots, x_i, \ldots, x_n) - f(x_1, x_2, \ldots, \tilde{x}_i, \ldots, x_n)| \leq c_i \quad \forall i, x_1 \ldots x_n, \tilde{x}_i$$

Then

$$\Pr\{|f(X) - E[f(X)]| \geq t\} \leq 2\exp\left(-2t^2 / \sum_{i=1}^{n} c_i^2\right)$$

Counting Inversions

Let $X_1, \dots, X_n$ be i.i.d in $[0,1]$

$$I_{i,j} = \begin{cases} 1 & \text{if } x_i > x_j \\ 0 & \text{otherwise} \end{cases}$$

$$B = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} I_{i,j} \qquad E[B] = \frac{1}{2}\binom{n}{2}$$

Changing X_i changes only n terms in B

$$\therefore \Pr\left\{ \left|B - \tfrac{1}{2}\tbinom{n}{2}\right| \geq t \right\} \leq 2\exp\left(-2t^2/n^2\tbinom{n}{2}\right)$$

$$\therefore \Pr\left\{ \left|B - \tfrac{1}{2}\tbinom{n}{2}\right| \geq c \cdot n^2 \right\} \leq 2/e^{\Omega(c^2)}$$

#comparisons in insertion sort is very tightly concentrated around $(\frac{1}{2})\binom{n}{2}$

Summary

Different Kinds of randomized algorithms

- Monte-Carlo
- Las-Vegas

A number is either prime or it's not!

Different Kinds of guarantees

- Expected running time
- Tail estimates on running time

- Failure probability (biased, min, max)
- Correct more than half the time?