Algorithms and Probability (FS2025)
Week 8

Rui Zhang

April 9, 2025

Contents

pword: variance

$\log = \ln$

$\log = \ln$

1 Mini Quiz

6. $\mathrm{Var}[X] = \sigma^2$, $\mathbb{E}[X] = 0$, then: $\mathbb{P}[X > \lambda\sigma] \leq \frac{1}{\lambda^2}$, $\lambda > 0$

$$\mathbb{P}[|X - \mathbb{E}[X]| \geq t] \leq \frac{\mathrm{Var}[X]}{t^2}$$

$$\Rightarrow \mathbb{P}[|X| \geq \lambda\sigma] \leq \frac{\cancel{\sigma^2}}{\lambda^2 \cancel{\sigma^2}} = \frac{1}{\lambda^2}$$

$$\frac{1}{\lambda^2} \geq \mathbb{P}[|X| \geq \lambda\sigma] = \mathbb{P}[\{X \geq \lambda\sigma\} \cup \{X \leq -\lambda\sigma\}] \geq \mathbb{P}[X \geq \lambda\sigma] + \cancel{\mathbb{P}[X \leq -\lambda\sigma]}$$

$$= \mathbb{P}[\{X > \lambda\sigma\}] + \underbrace{\mathbb{P}[\{X = \lambda\sigma\}]}_{\geq 0}$$

$$\geq \mathbb{P}[X > \lambda\sigma] (+\ 0)$$

7. If $A(x)$ is s.t. $\mathbb{P}[A \text{ correct}] = \frac{3}{4}$ then with $O(\log(\frac{1}{\delta}))$ independent repetitions, returning most common answer is correct with $p = 1-\delta$

8. $X \geq 0$, $\mathbb{E}[X] > 100 \Rightarrow \mathbb{P}[X > 100] \geq \frac{1}{2}$

9. r.v. X & constant a, $\mathrm{Var}[X + a] = \mathrm{Var}[X] + a$

2 Exercise Feedback

3 Content

3.1 Variance

Definition: Let X be a random variable with expectation $\mu = \mathbb{E}[X]$. We then define the variance of X to be

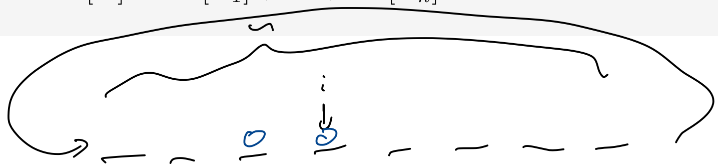
$$\mathrm{Var}[X] := \mathbb{E}[(X - \mu)^2] = \sum_{x \in W_X} (x - \mu)^2 \cdot \mathbb{P}[X = x]$$

and we call the square root of this value the standard deviation $\sigma = \sqrt{\mathrm{Var}[X]}$

Theorem: $\mathrm{Var}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$

Theorem: $\mathrm{Var}[a \cdot X + b] = a^2 \cdot \mathrm{Var}[X]$

> **Theorem:** Let $X_1, \dots X_n$ be n mutually independent random variables and X equal to the sum of all of those random variables. Then we have that:
>
> $$\mathrm{Var}[X] = \mathrm{Var}[X_1] + \cdots + \mathrm{Var}[X_n]$$



<u>Exercise:</u> (Doublesums)

We have blue and red beads in a pot and want to build a necklace with n beads. To do this, we randomly draw a bead from our pot and add it to the necklace. We assume that we have an infinite supply of beads and that each draw results in a red bead with probability 0.25 $\frac{1}{4}$ and a blue bead with probability 0.75 $\frac{3}{4}$. After adding n beads in sequence to our necklace (where the beads are numbered in order), we close it into a loop. We now ask how many color transitions occur in the necklace, meaning how many positions exist where the i-th bead has a different color than the $i+1$-th bead (where the $n+1$-th bead is the same as the 1st bead, since it forms a loop). Let

$$X := \text{Number of color transitions in the necklace}$$

To start, we first define indicator variables for all subproblems (which is almost always a good approach) to simplify the problem. One can verify (or look it up in a reference) that X does not follow a binomial distribution.

$$X_i := \text{Indicatorvariable for the event "There is a color transition at the i-th bead"}$$

(a) What is $\mathbb{E}[X]$?

$$E[X] = \sum_{i=1}^{n} E[X_i] = \sum_{i=1}^{n} \mathbb{P}[X_i = 1] = \sum_{i=1}^{n} \frac{1}{4}\cdot\frac{3}{4} + \frac{3}{4}\cdot\frac{1}{4} = n \cdot \frac{6}{16} = \frac{3}{8} n$$

(b) What is $\mathbb{E}[X_i \cdot X_j]$ for arbitrary $1 \leq i \leq j \leq n$? (Hint: Case Distinction)

$i = j$: $\quad E[X_i X_j] = E[X_i^2] = E[X_i] = \mathbb{P}[X_i = 1] = \frac{3}{8} \leftarrow n$

i, j s.t.
X_i, X_j indep: $\quad E[X_i X_j] = E[X_i]E[X_j] = \left(\frac{3}{8}\right)^2 = \frac{9}{64} \leftarrow n^2 - n - 2n$

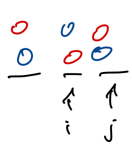
i, j s.t.
X_i, X_j dep: $\quad E[X_i X_j] = 1 \cdot \mathbb{P}[X_i X_j = 1] + \cancel{0 \cdot \mathbb{P}[X_i = 0]}$

$\Downarrow$

$i \in \{j-1, j+1\}$

1 1 → 1
1 0 → 0
0 1 → 0
0 0 → 0

$$= \frac{3}{4}\cdot\frac{1}{4}\cdot\frac{3}{4} + \frac{1}{4}\cdot\frac{3}{4}\cdot\frac{1}{4} \leftarrow 2n$$



$$= \frac{12}{64}$$

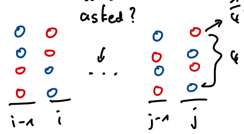
(c) ~~Using (b)~~, what is Var[X]?

$$Var[X] = E[X^2] - E[X]^2 = E[(\sum_{i=1}^{n} X_i)^2] - E[X]^2 = E[\sum_{i=1}^{n} X_i \sum_{j=1}^{n} X_j] - E[X]^2 = E[\sum_{i=1}^{n}\sum_{j=1}^{n} X_i X_j] - E[X]^2$$

$$= \boxed{\sum_{i=1}^{n}\sum_{j=1}^{n} E[X_i X_j]} - E[X]^2 = (n \cdot \tfrac{3}{8} \quad = \ldots = \text{yey}$$

independence if $i \notin \{j-1, j+1\}$

who asked?



$$\frac{1}{4}\cdot\frac{1}{4}\cdot\frac{3}{4}\cdot\frac{3}{4}\cdot 4 = \frac{9}{64} = \mathbb{P}[X_i = 1, X_j = 1] = \underbrace{\mathbb{P}[X_i = 1]}_{\frac{3}{8}}\underbrace{\mathbb{P}[X_j = 1]}_{\frac{3}{8}}$$

3.2 Inequalities

Theorem: (Markov) Let X be a random variable **which only attains non-negative values**. Then we have for all $t \in \mathbb{R}^{>0}$

$$\Pr[X \geq t] \leq \frac{\mathbb{E}[X]}{t}$$

Theorem: (Chebyshev) Let X be a random variable and $t > 0$. Then we have:

$$\Pr[|X - \mathbb{E}[X]| \geq t] \leq \frac{\mathrm{Var}[X]}{t^2}$$

Theorem: (Chernoff). Let $X_1, \ldots, X_n$ be n independent Bernoulli random variables such that $\Pr[X_i = 1] = p_i$ and $\Pr[X_i = 0] = 1 - p_i$. Then for the sum of those independent random variables $X = X_1 + \cdots + X_n$, we have:

$$\Pr[X \geq (1+\delta)\mathbb{E}[X]] \leq e^{-\frac{1}{3}\delta^2\mathbb{E}[X]} \text{ for all } 0 < \delta < 1 \quad (1)$$

$$\Pr[X \leq (1-\delta)\mathbb{E}[X]] \leq e^{-\frac{1}{2}\delta^2\mathbb{E}[X]} \text{ for all } 0 < \delta < 1 \quad (2)$$

$$\Pr[X \geq t] \leq 2^{-t} \text{ for } t \geq 2e\mathbb{E}[X] \quad (3)$$

Exercise: (Continuation of Doublesums)

(d) Find a best possible upper bound for $\Pr[X \geq t + \mathbb{E}[X]]$

Short Questions: (Inequalities):

(a) Which of the following claims holds for every random varable $X \geq 0$:

(1) $\Pr[X \geq 10] \leq \frac{\mathbb{E}[X]}{10}$ ✓

(2) $\Pr[X \geq -10] \leq \frac{\mathbb{E}[X]}{10}$

(3) $\Pr[X \leq -10] \leq \frac{\mathbb{E}[X]}{10}$ ✓

(b) Let X be a random variable with $\mathbb{E}[X] = 5$ and $\mathrm{Var}[X] = 0.9$. What is the best upper bound for $\Pr[X \geq 6]$ which you can derive?

(1) 9/10

(2) 0

(3) 1/2

(4) 1

(5) 5/6

$$\mathbb{P}[|X-5| \geq 1] = \mathbb{P}[|X-\mathbb{E}[X]| \geq 1] \leq \mathrm{Var}[X] = \frac{9}{10}$$

$$\vee|$$

$$\mathbb{P}[X-5 \geq 1] = \mathbb{P}[X \geq 6]$$

(c) Bob has a coin which shows "heads" with probability $1/n$ and "tails" with probability $1 - 1/n$. He throws the coin $5n$ times. Let X be the random variable which counts how often he gets "heads". In that case we have $\Pr[X \geq 30] \leq 2^{-30}$

$\mathbb{E}[X] = 5$

$2 \cdot e \cdot 5 = 10 \cdot e \leq 30 = t$

Random Algorithms

$$I \to \boxed{\quad A \quad} \to A(I)$$

$$\Downarrow$$

$$I \to \boxed{\quad A \quad} \to A(I, R)$$

$$\uparrow$$

$$R$$

$$\text{Correctness: For all inputs } I, \text{ we have } \Pr[\mathbb{A}(I,R) \text{ is correct}] \geq \text{something that is almost 1} \tag{4}$$

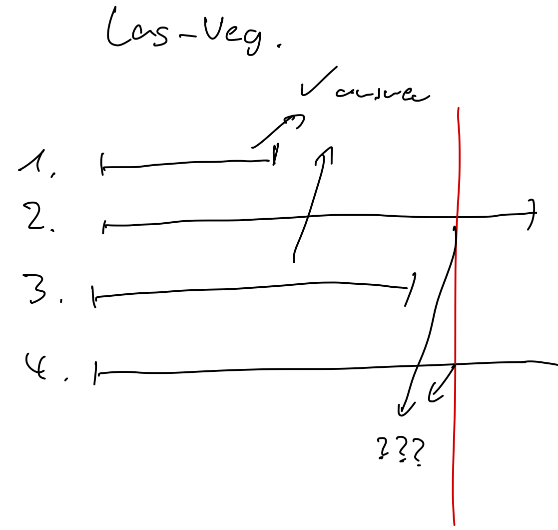
$$\text{Runtime: For all inputs } I, \text{ we have } \Pr[\text{Algo. Runtime} \leq O(f(n))] \geq \text{something that is almost 1} \tag{5}$$

Figure 1: 99% of gamblers stop before they win big. Never give up.

Random Algorithms are divided up into two types:

Monte-Carlo-Algorithm:	Las-Vegas-Algorithms
Correctness of the result is a random variable.	Runtime is a random variable.

3.3.1 Las-Vegas Algorithms



> **Theorem:** Let A be an algorithm which never gives a wrong result but sometimes returns "???", such that
>
> $$\Pr[A(I) \text{ correct}] \geq \epsilon$$
>
> Then for all $\delta \geq 0$ let A_δ be the algorithm which repeats A until a non-"???" value is returned or until at most $N = \epsilon^{-1} \ln \delta^{-1}$ repetitions have occurred (in which case A_δ gives up and returns "???" too.) Then, A_δ has the correctness probability:
>
> $$\Pr[A_\delta(I) \text{ correct}] \geq 1 - \delta$$

Proof. The probability that "???" is returned N times (and A_δ thus also returns some nonsense), is

$$\begin{aligned} \Pr[A_\delta(I) incorrect] &= (1-\epsilon)^N \\ &\leq e^{-\epsilon\epsilon^{-1}\ln\delta^{-1}} \qquad 1 - x \leq e^{-x} \\ &= e^{\ln \delta} \\ &= \delta \end{aligned}$$

$\square$

3.3.2 Monte-Carlo Algorithms

For Monte-Carlo-Algorithms, we cannot always make the same improvements. The main problem here is that we cannot know if the returned value from the algorithm is correct or not. However, under certain assumptions, we can improve via the following two theorems:

> **Theorem:** Let A be a randomized algorithm which gives a binary output: Either "Yes" or "No", where we have a "one-sided" error, i.e.:
>
> $$\Pr[A(I) = \text{"}Yes\text{"}] = 1 \text{ if } I \text{ is a "Yes-Instance" (so the algorithm should return "Yes")} \tag{6}$$
> $$\Pr[A(I) = \text{"}No\text{"}] \geq \epsilon \text{ if } I \text{ is a "No-Instance" (so the algorithm should return "No")} \tag{7}$$
>
> Then for $\delta > 0$ let A_δ be an algorithm which repeats A until either "No" is returned (in which case A_δ returns "No") or until $N = \epsilon^{-1} \ln \delta^{-1}$ iterations have passed, which all resulted in a "Yes". Then we have for all inputs I:
>
> $$\Pr[A_\delta(I) \text{ correct}] \geq 1 - \delta$$

Proof. Left as an exercise.  It is basically the same as with Las-Vegas-Algorithms □

What about if we have errors on both sides?

Theorem: Let $\epsilon > 0$ and A be a randomized algorithm which either returns "Yes" or "No". Where, independent of the instance, we have:

$$\Pr[A(I) \text{ correct}] \geq 1/2 + \epsilon$$

Then we have for all $\delta > 0$ Let A_δ be the algorithm that repeats $N = 4\epsilon^{-2} \ln \delta^{-1}$ iterations of A and outputs the majority of answers, then we have that

$$\Pr[A_\delta(I) \text{ correct}] \geq 1 - \delta$$

3.3.3 Optimization Algorithms

Finally, I would like to visit the case where we would like to maximize the result of a random algorithm. In this case, the returned value of an algorithm should be larger than some baseline $f(I)$. We may now repeat this algorithm until we find a value bigger than $f(I)$ just like with the ideas from before:

Theorem: Let $\epsilon > 0$ and A be a randomized algorithm for an optimization problem, where we have

$$\Pr[A(I) \geq f(I)] \geq \epsilon$$

Then for all $\delta > 0$ let A_δ be the algorithm which repeats A $N = \epsilon^{-1} \ln \delta^{-1}$ times and returns the best result of all iterations. Then we have for A_δ that

$$\Pr[A_\delta(I) \geq f(I)] \geq 1 - \delta$$

Short Questions: (Randomized Algorithms:)

(a) Every deterministic algorithm can be seen as a random algorithm. ✓

(b) Every random algorithm can be seen as a deterministic algorithm. ✗

(c) Let $n = pq$ with $n > 10001$ be a product of two different primes $p, q \leq 10001$. We don't know p, q, but would like to calculate them. To this end, we develop the followig algorithm: Given an input n, choose a number $x \in [2, 10001]$ randomly. If n is divisible by x then the algorithm returns $(x, n/x)$. Otherwise it repeats from the beginning. Which type of random-algorithm is this?

$$\frac{2}{10000} = \frac{1}{5000} = p_{\text{correct}}$$

$$\frac{4999}{5000} = p_{\text{error}}$$

(d) Take the same algorithm from (c), instead of waiting until the algorithm is done, we would like to stop the algorithm after at most m iterations and return "???". We would like to guarantee that the algorithm returns a correct (non-"???") answer with probability at least 3/4. Which is the minimum m which guarantees this.

prob. of being wrong m times in a row: $\left(1-\frac{1}{5000}\right)^m$. We want that this probability is $\leq 1-\frac{3}{4}$

$\Rightarrow \left(1-\frac{1}{5000}\right)^m \leq \frac{1}{4}$

$\Rightarrow m \geq \log_{\frac{4999}{5000}}\left(\frac{1}{4}\right) \approx 6930.$ Note that using the $1-x \leq e^{-x}$ approx. here would result in a less accurate result!

(1) 200
(2) 7000
(3) 10000
(4) 2000