CSE 312

Foundations of Computing II

Lecture 10: More on Discrete RVs

PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

Aleks Jovcic

Slide Credit: Based on Stefano Tessaro's slides for 312 19au
incorporating ideas from Anna Karlin, Alex Tsun, Rachel Lin, Hunter Schafer & myself ☺

Agenda

- Linearity Recap
- LOTUS
- Variance
  - Properties of Variance
- Independent Random Variables
  - Properties of Independent Random Variables
- Application: Bloom Filter
  - Read textbook, if time permits we'll go over it in lecture

Recap Linearity of Expectation

Theorem. For any two random variables X and Y *(X, Y do not need to be independent)*

$$\mathbb{E}(X+Y) = \mathbb{E}(X) + \mathbb{E}(Y).$$

Theorem. For any random variables $X_1, \ldots, X_n$, and real numbers $a_1, \ldots, a_n \in \mathbb{R}$,

$$\mathbb{E}(a_1 X_1 + \cdots + a_n X_n) = a_1 \mathbb{E}(X_1) + \cdots + a_n \mathbb{E}(X_n).$$

For any event A, can define the indicator random variable X for A

$$X = \begin{cases} 1 & \text{if event } A \text{ occurs} \\ 0 & \text{if event } A \text{ does not occur} \end{cases}$$

$$\mathbb{P}(X=1) = \mathbb{P}(A)$$
$$\mathbb{P}(X=0) = 1 - \mathbb{P}(A)$$

Rotating the table

n people are sitting around a circular table. There is a name tag in each place. Nobody is sitting in front of their own name tag.

Rotate the table by a random number k of positions between 1 and n-1 (equally likely).

X is the number of people that end up front of their own name tag.

What is E(X)?

$X_i = 1$ if person i is @ their tag

Decompose: $X = X_1 + X_2 \cdots + X_n$

LOE: $E[X] = E[X_1 + \cdots + X_n] = E[X_1] + \cdots + E[X_n]$

Conquer: $E[X_i] = \boxed{\frac{1}{n-1}}$ or $\frac{1}{n}$

$\boxed{\frac{n}{n-1}} = E[X]$

Agenda

- Linearity Recap
- LOTUS
- Variance
  - Properties of Variance
- Independent Random Variables
  - Properties of Independent Random Variables
- Application: Bloom Filter
  - Read textbook, if time permits we'll go over it in lecture

Linearity is special!

$g(X) = X^2$ $\quad E(g(X)) \stackrel{?}{=} g(E[X])$

$E[X^2] = E[X]^2$

$= \quad\quad 0$

$1 \neq 0$

In general $\mathbb{E}(g(X)) \neq g(\mathbb{E}(X))$

E.g., $X = \begin{cases} 1 \;\; with\, prob\;\; 1/2 \\ -1 \;\; with\, prob\; 1/2 \end{cases}$

$E[X] = 1 \cdot \frac{1}{2} + (-1) \cdot \frac{1}{2}$

$= 0$

- $\mathbb{E}(X^2) \neq \mathbb{E}(X)^2$

How DO we compute $\underline{\mathbb{E}(g(X))}$?

Expectation of $g(X)$ LOTUS

X^2 X^3+75 $\sqrt{X}$

Definition. Given a discrete RV $X: \Omega \to \mathbb{R}$, the expectation or expected value of the random variable $g(X)$ is

$g(X) = X^2$
$= 3X+2$
$= X \bmod 8$

$$E[g(X)] = \sum_{\omega \in \Omega} g(X(\omega)) \cdot \Pr(\omega)$$

or equivalently

$\Pr(X=3)$
$\Pr(X=7)$

$$E[g(X)] = \sum_{x \in X(\Omega)} \boxed{g(x)} \cdot \Pr(X = x)$$

Example: Expectation of $g(X)$

$\Omega_X = \{1, 2, 3, 4, 5, 6\}$ $\quad g(X) = 10X^3$

Suppose we rolled a fair, 6-sided die in a game. Your winnings will be the cube of the number rolled, times 10. Let X be the result of the dice roll. What is your expected winnings?

$$E[10X^3] = \sum_{x \in \Omega_X} g(x) \cdot \mathbb{P}(X = x) = \sum_{x \in \Omega_X} 10x^3 \cdot \frac{1}{6} = \frac{10}{6} \sum x^3$$

$$\boxed{\frac{10}{6}\left(1^3 + 2^3 + 3^3 + 4^3 + 5^3 + 6^3\right)} \approx \boxed{735}$$

Agenda

- Linearity Recap
- LOTUS
- Variance
  - Properties of Variance
- Independent Random Variables
  - Properties of Independent Random Variables
- Application: Bloom Filter
  - Read textbook, if time permits we'll go over it in lecture

Two Games

Game 1: *In every round, you win \$2 with probability 1/3, lose \$1 with probability 2/3.*

W_1 = payoff in a round of Game 1

$$\mathbb{P}(W_1 = 2) = \frac{1}{3}, \mathbb{P}(W_1 = -1) = \frac{2}{3}$$

Two Games

Game 1: *In every round, you win \$2 with probability 1/3, lose \$1 with probability 2/3.*

W_1 = payoff in a round of Game 1

$\mathbb{P}(W_1 = 2) = \frac{1}{3}$, $\mathbb{P}(W_1 = -1) = \frac{2}{3}$

Game 2: *In every round, you win \$10 with probability 1/3, lose \$5 with probability 2/3.*

W_2 = payoff in a round of Game 2

$\mathbb{P}(W_2 = 10) = \frac{1}{3}$, $\mathbb{P}(W_2 = -5) = \frac{2}{3}$

Which game would you rather play?

Two Games

Game 1: *In every round, you win \$2 with probability 1/3, lose \$1 with probability 2/3.*

W_1 = payoff in a round of Game 1

$\mathbb{P}(W_1 = 2) = \frac{1}{3}$, $\mathbb{P}(W_1 = -1) = \frac{2}{3}$

$\mathbb{E}(W_1) = 0$

Game 2: *In every round, you win \$10 with probability 1/3, lose \$5 with probability 2/3.*

W_2 = payoff in a round of Game 2

$\mathbb{P}(W_2 = 10) = \frac{1}{3}$, $\mathbb{P}(W_2 = -5) = \frac{2}{3}$

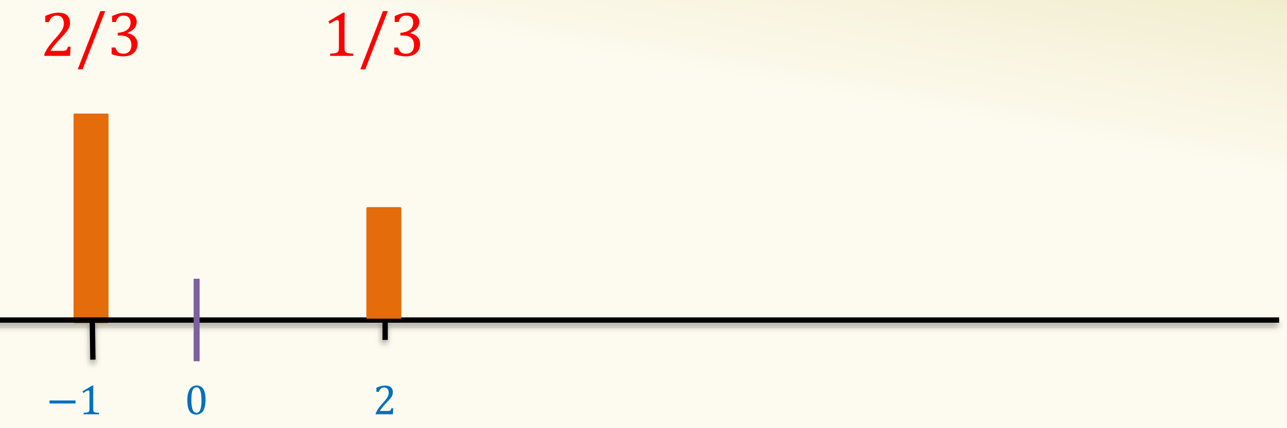
$\mathbb{E}(W_2) = 0$

Which game would you <u>rather</u> play?

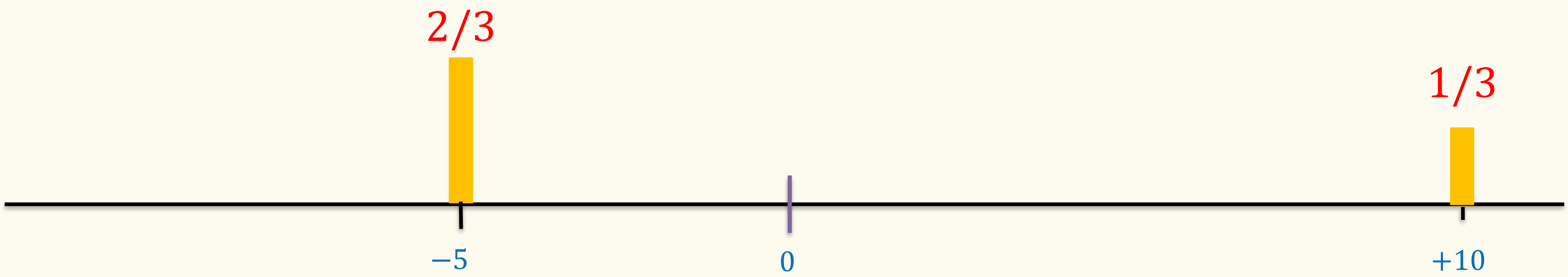
Somehow, Game 2 has higher volatility!

Two Games

$\mathbb{P}(W_1 = 2) = \frac{1}{3}$, $\mathbb{P}(W_1 = -1) = \frac{2}{3}$



$\mathbb{P}(W_2 = 10) = \frac{1}{3}$, $\mathbb{P}(W_2 = -5) = \frac{2}{3}$



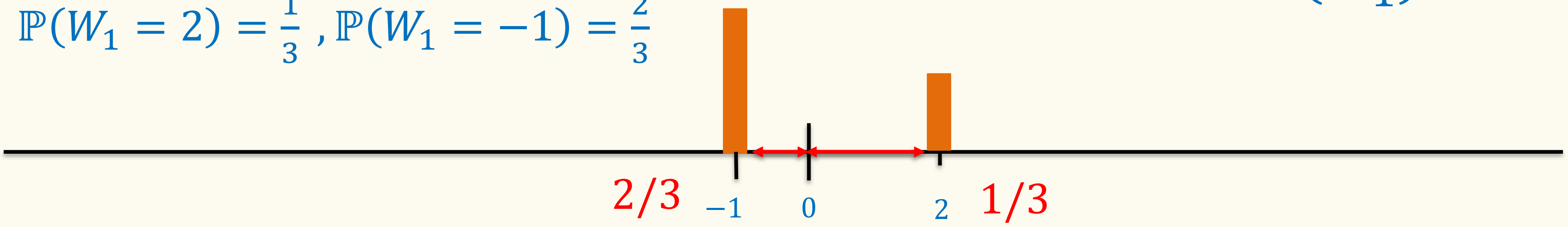
Same expectation, but clearly very different distribution.

We want to capture the difference – New concept: **Variance**

Variance (Intuition, First Try)

$\mathbb{P}(W_1 = 2) = \frac{1}{3}, \mathbb{P}(W_1 = -1) = \frac{2}{3}$

$\mathbb{E}(W_1) = 0$



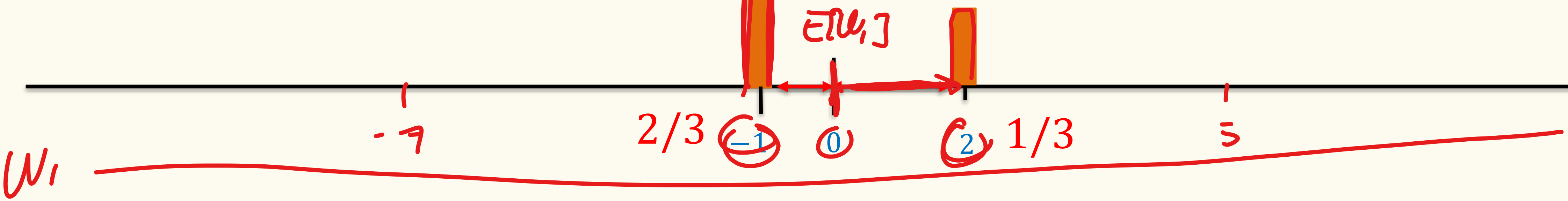
New quantity (random variable): How far from the expectation?

$$\Delta(W_1) = W_1 - E[W_1]$$

Variance (Intuition, First Try)

$\mathbb{E}(W_1) = 0$

$\mathbb{P}(W_1 = 2) = \frac{1}{3}$, $\mathbb{P}(W_1 = -1) = \frac{2}{3}$



New quantity (random variable): How far from the expectation?

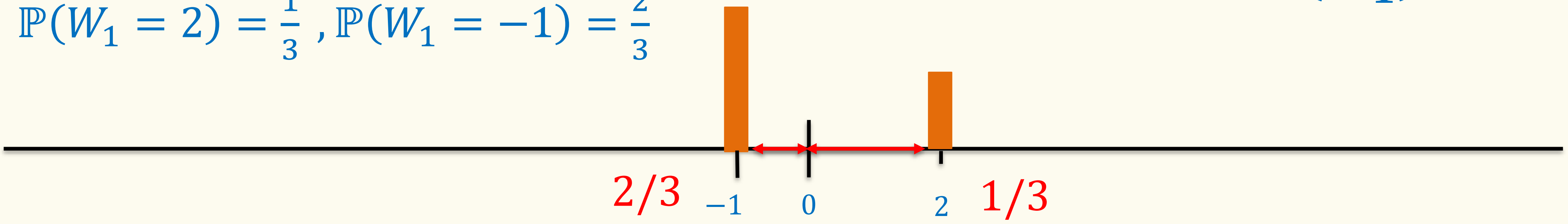
$$\Delta(W_1) = W_1 - E[W_1]$$

$$E[\Delta(W_1)] = E[W_1 - E[W_1]]$$
$$= E[W_1] - E\big[E[W_1]\big]$$
$$= E[W_1] - E[W_1]$$
$$= 0$$

Variance (Intuition, Better Try)

$\mathbb{E}(W_1) = 0$

$\mathbb{P}(W_1 = 2) = \frac{1}{3}, \mathbb{P}(W_1 = -1) = \frac{2}{3}$



A better quantity (random variable): How far from the expectation?

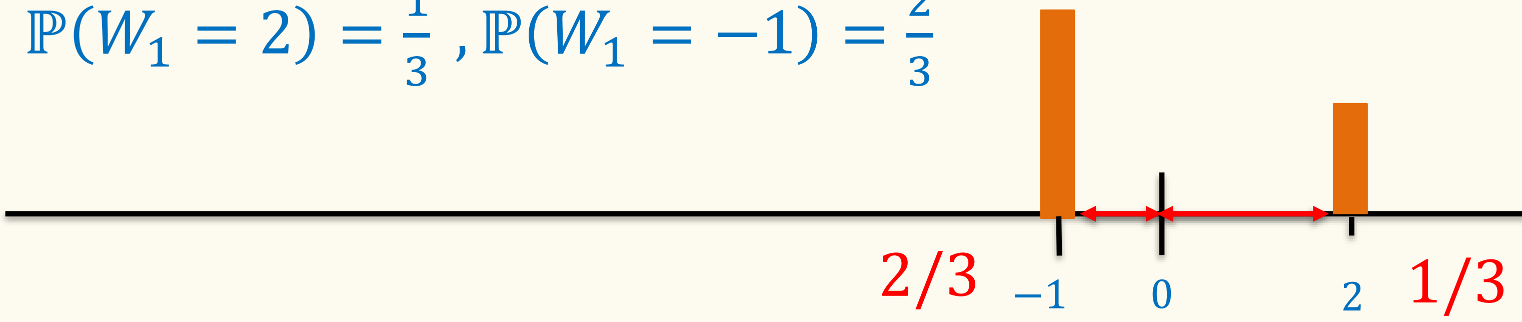
$\Delta(W_1) = (W_1 - E[W_1])^2$

$E[\Delta(W_1)] = E[(W_1 - E[W_1])^2]$

Variance (Intuition, Better Try)

$$\mathbb{E}(W_1) = 0$$

$$\mathbb{P}(W_1 = 2) = \frac{1}{3}, \mathbb{P}(W_1 = -1) = \frac{2}{3}$$



$w_1 = 2$

A better quantity (random variable): How far from the expectation?

$$\Delta(W_1) = (W_1 - E[W_1])^2$$

$$\mathbb{P}(\Delta(W_1) = 1) = \boxed{\frac{2}{3}}$$

$$\mathbb{P}(\Delta(W_1) = 4) = \boxed{\frac{1}{3}}$$

$1 - 0$

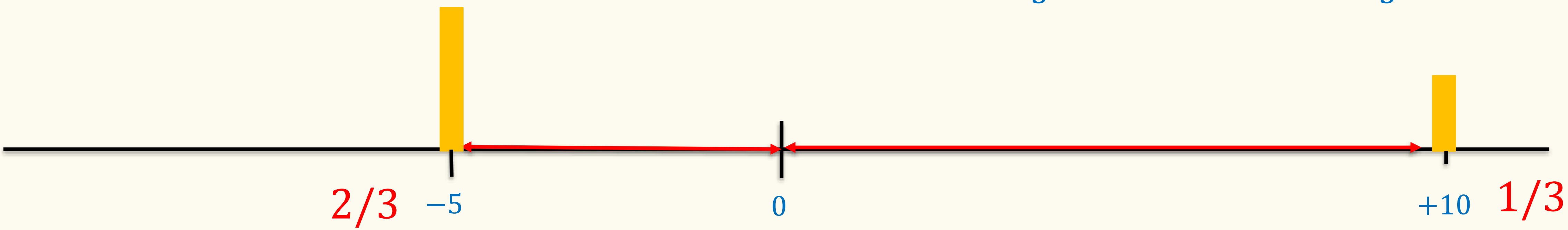
$$E[\Delta(W_1)] = E[(W_1 - E[W_1])^2]$$

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

$$= \boxed{2}$$

Variance (Intuition, Better Try)

$\mathbb{P}(W_2 = 10) = \frac{1}{3}, \mathbb{P}(W_2 = -5) = \frac{2}{3}$



A better quantity (random variable): How far from the expectation?

$$\Delta'(W_2) = (W_2 - E[W_2])^2$$

$$\mathbb{P}(\Delta'(W_2) = 25) = \frac{2}{3}$$

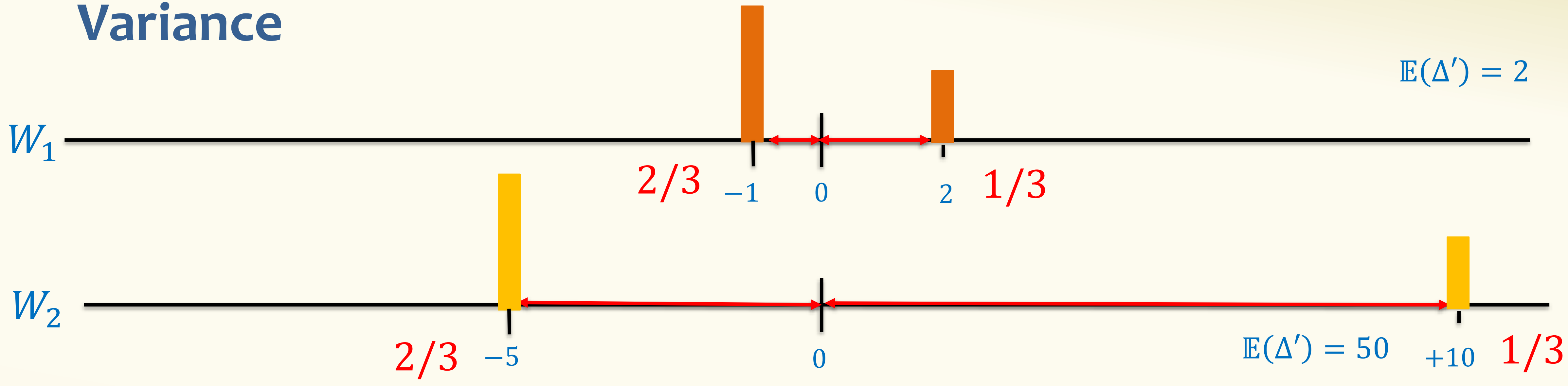
$$\mathbb{P}(\Delta'(W_2) = 100) = \frac{1}{3}$$

$$E[\Delta'(W_2)] = E[(W_2 - E[W_2])^2]$$

$$= \frac{2}{3} \cdot 25 + \frac{1}{3} \cdot 100$$

$$= 50$$

Variance



We say that W_2 has **"higher variance"** than W_1.

Variance

$\text{Var}(X)$

Definition. The **variance** of a (discrete) RV X is

$$\text{Var}(X) = \mathbb{E}\left[\left(X - \mathbb{E}(X)\right)^2\right] = \sum_x \mathbb{p}_X(x) \cdot \left(x - \mathbb{E}(X)\right)^2$$

Recall $\mathbb{E}(X)$ is a **constant**, not a random variable itself.

Intuition: Variance is a quantity that measures, in expectation, how "far" the random variable is from its expectation.

Variance

Definition. The **variance** of a (discrete) RV X is

$$\mathrm{Var}(X) = \mathbb{E}\left[\left(X - \mathbb{E}(X)\right)^2\right] = \sum_x \mathbb{p}_X(x) \cdot \left(x - \mathbb{E}(X)\right)^2$$

Recall $\mathbb{E}(X)$ is a **constant**, not a random variable itself.

Standard deviation: $\sigma(X) = \sqrt{\mathrm{Var}(X)}$

Intuition: Variance (or standard deviation) is a quantity that measures, in expectation, how "far" the random variable is from its expectation.

Variance – Example 1

X fair die

- $\mathbb{P}(X = 1) = \cdots = \mathbb{P}(X = 6) = 1/6$
- $\mathbb{E}(X) = 3.5$

$$\mathrm{Var}(X) =? = \sum_{x \in \Omega_X} \mathbb{P}(X = x)(x - \mathbb{E}[X])^2$$

$$= \frac{1}{6}(1 - 3.5)^2 + \frac{1}{6}(2 - 3.5)^2 + \frac{1}{6}(3 - 3.5)^2$$

$$+ \ldots =$$

Variance – Example 1

X fair die

- $\mathbb{P}(X = 1) = \cdots = \mathbb{P}(X = 6) = 1/6$
- $\mathbb{E}(X) = 3.5$

$$\mathrm{Var}(\mathrm{X}) = \sum_x \mathbb{P}(X = x) \cdot \left(x - \mathbb{E}(X)\right)^2$$

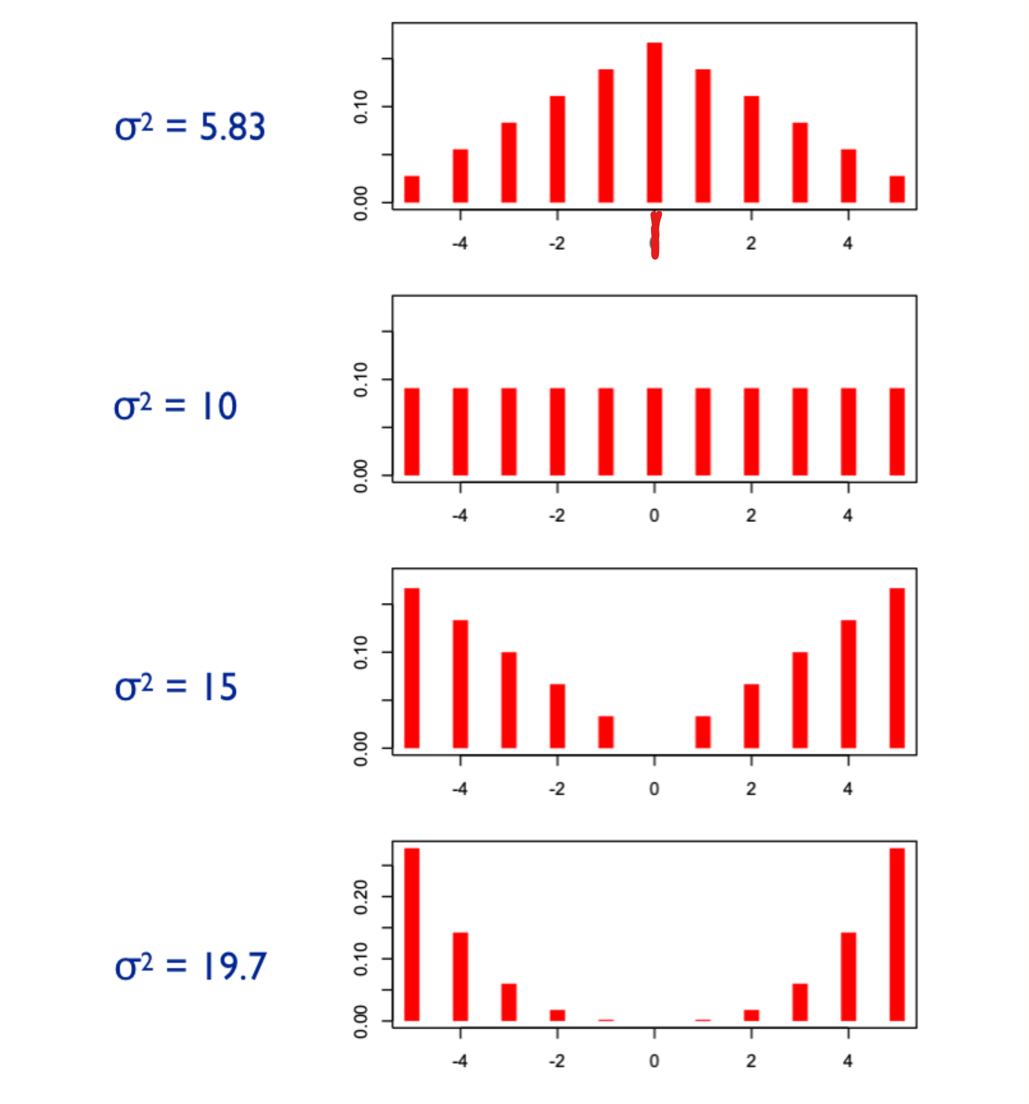
$$= \frac{1}{6}[(1-3.5)^2 + (2-3.5)^2 + (3-3.5)^2 + (4-3.5)^2 + (5-3.5)^2 + (6-3.5)^2]$$

$$= \frac{2}{6}[2.5^2 + 1.5^2 + 0.5^2] = \frac{2}{6}\left[\frac{25}{4} + \frac{9}{4} + \frac{1}{4}\right] = \frac{35}{12} \approx 2.91677\ldots$$

Variance in Pictures

Captures how much "spread" there is in a pmf

All pmfs in picture have same expectation



Agenda

- Linearity Recap
- LOTUS
- Variance
  - Properties of Variance
- Independent Random Variables
  - Properties of Independent Random Variables
- Application: Bloom Filter
  - Read textbook, if time permits we'll go over it in lecture

Variance – Properties

$Var(X+b) = Var(X)$

Definition. The **variance** of a (discrete) RV X is

$$\mathrm{Var}(X) = \mathbb{E}\left[\left(X - \mathbb{E}(X)\right)^2\right] = \sum_x \mathbb{p}_X(x) \cdot \left(x - \mathbb{E}(X)\right)^2$$

Theorem. For any $a, b \in \mathbb{R}$, $\mathrm{Var}(a \cdot X + b) = a^2 \cdot \mathrm{Var}(X)$

(Proof: Exercise!)

Theorem. $\mathrm{Var}(X) = \mathbb{E}(X^2) - \mathbb{E}(X)^2$

Variance

Theorem. $\mathrm{Var}(X) = \mathbb{E}(X^2) - \mathbb{E}(X)^2$

Proof: $\mathrm{Var}(X) = \mathbb{E}\left[\left(X - \mathbb{E}(X)\right)^2\right]$

Recall $\mathbb{E}(X)$ is a **constant**

$$= \mathbb{E}[X^2 - 2\mathbb{E}(X) \cdot X + \mathbb{E}(X)^2]$$

$$= \mathbb{E}(X^2) - 2\mathbb{E}(X)\mathbb{E}(X) + \mathbb{E}(X)^2$$

(linearity of expectation!)

$$= \mathbb{E}(X^2) - \mathbb{E}(X)^2$$

$\mathbb{E}(X^2)$ and $\mathbb{E}(X)^2$ are different !

Variance – Example 1

X fair die

- $\mathbb{P}(X = 1) = \cdots = \mathbb{P}(X = 6) = 1/6$
- $\mathbb{E}(X) = \frac{21}{6}$
- $\mathbb{E}(X^2) = \frac{91}{6}$

$$\mathrm{Var}(X) = \mathbb{E}(X^2) - \mathbb{E}(X)^2 = \frac{91}{6} - \left(\frac{21}{6}\right)^2 = \frac{105}{36} \approx \underline{2.91677}$$

$E[g(X)]$

$g(X) = X^2$

In General, $\mathrm{Var}(X+Y) \neq \mathrm{Var}(X) + \mathrm{Var}(Y)$

Example to show this:

- Let X be a r.v. with pmf $\mathbb{P}(X=1) = \mathbb{P}(X=-1) = 1/2$
  - What is $\mathrm{E}[X]$ and $\mathrm{Var}(X)$?

In General, $\mathrm{Var}(X+Y) \neq \mathrm{Var}(X) + \mathrm{Var}(Y)$

Example to show this:

- Let X be a r.v. with pmf $\mathbb{P}(X=1) = \mathbb{P}(X=-1) = 1/2$
  - $\mathrm{E}[X] = 0$ and $\mathrm{Var}(X) = 1$
- Let $Y = -X$
  - What is $\mathrm{E}[Y]$ and $\mathrm{Var}(Y)$?

In General, $\mathrm{Var}(X+Y) \neq \mathrm{Var}(X) + \mathrm{Var}(Y)$

Example to show this:

- Let X be a r.v. with pmf $\mathbb{P}(X=1) = \mathbb{P}(X=-1) = 1/2$
  - $\mathrm{E}[X] = 0$ and $\mathrm{Var}(X) = 1$
- Let $Y = -X$
  - $\mathrm{E}[Y] = 0$ and $\mathrm{Var}(Y) = 1$

What is $\mathrm{Var}(X+Y)$?

Agenda

- Linearity Recap
- LOTUS
- Variance
  - Properties of Variance
- Independent Random Variables
  - Properties of Independent Random Variables
- Application: Bloom Filter
  - Read textbook, if time permits we'll go over it in lecture

Random Variables and Independence

$\mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B)$

Definition. Two random variables X, Y are **(mutually) independent** if for all x, y,

$$\mathbb{P}(X = x, Y = y) = \mathbb{P}(X = x) \cdot \mathbb{P}(Y = y)$$

Intuition: Knowing X doesn't help you guess Y and vice versa

Definition. The random variables $X_1, \ldots, X_n$ are **(mutually) independent** if for all $x_1, \ldots, x_n$,

$$\mathbb{P}(X_1 = x_1, \ldots, X_n = x_n) = \mathbb{P}(X_1 = x_1) \cdots \mathbb{P}(X_n = x_n)$$

Example

$9 \% 2 = 1$

$4 \% 2 = 0$

$n = 4$

Let X be the number of heads in n *independent* coin flips of the same coin with probability p of coming up Heads. Let $Y = \mathrm{X \bmod 2}$ be the parity (even/odd) of X.

Are X and Y independent?

HHHTHH

$X = 3$

$Y = 1$

$X = 3$

$Y = 0$

$$\underbrace{P(X = 3, Y = 0)}_{0} \stackrel{?}{=} P(X = 3) \cdot P(Y = 0)$$

$$\neq \binom{n}{3} p^3 (1-p)^{n-3} \cdot \sim \frac{1}{2}$$

Poll:

A. Yes
B. No

Example

Make $2n$ *independent* coin flips of the same coin. Let X be the number of heads in the first n flips and Y be the number of heads in the last n flips.

Are X and Y independent?

Poll:

A. Yes
B. No

Agenda

- Linearity Recap
- LOTUS
- Variance
  - Properties of Variance
- Independent Random Variables
  - Properties of Independent Random Variables
- Application: Bloom Filter
  - Read textbook, if time permits we'll go over it in lecture

Important Facts about Independent Random Variables

Theorem. If X, Y independent, $\mathbb{E}(X \cdot Y) = \mathbb{E}(X) \cdot \mathbb{E}(Y)$

Theorem. If X, Y independent, $\mathrm{Var}(X + Y) = \mathrm{Var}(X) + \mathrm{Var}(Y)$

Corollary. If $X_1, X_2, \ldots, X_n$ mutually independent,

$$\mathrm{Var}\left(\sum_{i=1}^{n} X_i\right) = \sum_{i}^{n} \mathrm{Var}(X_i)$$

Independent Random Variables are nice!

Theorem. If X, Y independent, $\mathbb{E}(X \cdot Y) = \mathbb{E}(X) \cdot \mathbb{E}(Y)$

Proof

Let $x_i, y_i, i = 1, 2, \ldots$ be the possible values of X, Y.

$$
\begin{aligned}
E[X \cdot Y] &= \sum_i \sum_j x_i \cdot y_j \cdot P(X = x_i \wedge Y = y_j) \quad \text{independence} \\
&= \sum_i \sum_j x_i \cdot y_j \cdot P(X = x_i) \cdot P(Y = y_j) \\
&= \sum_i x_i \cdot P(X = x_i) \cdot \left(\sum_j y_j \cdot P(Y = y_j) \right) \\
&= E[X] \cdot E[Y]
\end{aligned}
$$

Note: *NOT* true in general; see earlier example $E[X^2] \neq E[X]^2$

Proof not covered

(Not Covered) Proof of $\text{Var}(X + Y) = \text{Var}(X) + \text{Var}(Y)$

Theorem. If X, Y independent, $\text{Var}(X + Y) = \text{Var}(X) + \text{Var}(Y)$

Proof

$$
\begin{aligned}
&Var[X+Y] \\
&= E[(X+Y)^2] - (E[X+Y])^2 \\
&= E[X^2 + 2XY + Y^2] - (E[X] + E[Y])^2 \\
&= E[X^2] + 2E[XY] + E[Y^2] - ((E[X])^2 + 2E[X]E[Y] + (E[Y])^2) \\
&= E[X^2] - (E[X])^2 + E[Y^2] - (E[Y])^2 + 2(E[XY] - E[X]E[Y]) \\
&= Var[X] + Var[Y] + 2(E[X]E[Y] - E[X]E[Y]) \\
&= Var[X] + Var[Y]
\end{aligned}
$$

Proof not covered

Example – Coin Tosses

We flip n *independent* coins, each one heads with probability p

- $X_i = \begin{cases} 1, & i\text{–th outcome is heads} \\ 0, & i\text{–th outcome is tails.} \end{cases}$
- $Z =$ number of heads

Fact. $Z = \sum_{i=1}^{n} X_i$

$\mathbb{P}(X_i = 1) = p$
$\mathbb{P}(X_i = 0) = 1 - p$

What is $\mathrm{E}[Z]$? What is $\mathrm{Var}(Z)$?

np

$\mathrm{Var}(X_i) = E[X_i^2] - E[X_i]^2$

$\mathrm{Var}(Z) = n\mathrm{Var}(X_i)$

$\boxed{n(p - p^2)} = \mathrm{Var}(Z)$

$np(1-p)$

Note: $X_1, \ldots, X_n$ are <u>mutually</u> independent!

Example – Coin Tosses

We flip n *independent* coins, each one heads with probability p

- $X_i = \begin{cases} 1, & i\text{–th outcome is heads} \\ 0, & i\text{–th outcome is tails.} \end{cases}$
- $Z =$ number of heads

Fact. $Z = \sum_{i=1}^{n} X_i$

$\mathbb{P}(X_i = 1) = p$
$\mathbb{P}(X_i = 0) = 1 - p$

$\mathbb{P}(Z = k) = \binom{n}{k} p^k (1-p)^{n-k}$

What is $\mathrm{E}[Z]$? What is $\mathrm{Var}(Z)$?

Note: $X_1, \ldots, X_n$ are <u>mutually</u> independent!

$$\mathrm{Var}(Z) = \sum_{i=1}^{n} \mathrm{Var}(X_i) = n \cdot p(1-p)$$

Note $\mathrm{Var}(X_i) = p(1-p)$

Agenda

- Linearity Recap
- LOTUS
- Variance
  - Properties of Variance
- Independent Random Variables
  - Properties of Independent Random Variables
- Application: Bloom Filter
  - Read textbook, if time permits we'll go over it in lecture

Basic Problem

Problem: Store a subset S of a <u>large</u> set U.

Example. $U =$ set of 128 bit strings $\quad |U| \approx 2^{128}$

$S =$ subset of strings of interest $\quad |S| \approx 1000$

Two goals:

1. **Very fast** (ideally constant time) answers to queries "Is $x \in S$?"
2. **Minimal storage** requirements.

Bloom Filters: Motivation

- Large universe of possible data items.
- Hash table is stored on disk or in network, so any lookup is expensive.
- Many (if not most) of the lookups return “Not found”.

Altogether, this is bad. You’re wasting **a lot of time and space** doing lookups for items that aren’t even present.

Example:

- **Network routers**: want to track source IP addresses of certain packets, .e.g., blocked IP addresses.

Bloom Filters: Motivation

- Probabilistic data structure.
- Close cousins of hash tables.
- Ridiculously space efficient
- To get that, make occasional errors, specifically false positives.

Bloom Filters

- Stores information about a set of elements.
- Supports two operations:
  1. **add(x)** - adds x to bloom filter
  2. **contains(x)** - returns true if x in bloom filter, otherwise returns false
     - If returns false, **definitely** not in bloom filter.
     - If returns true, **possibly** in the structure (some false positives).

Bloom Filters

- Why accept false positives?
  - **Speed** – both operations very very fast.
  - **Space** – requires a miniscule amount of space relative to storing all the actual items that have been added.

○ Often just 8 bits per inserted item!

Bloom Filters: Initialization

Number of hash functions

Size of array associated to each hash function.

function INITIALIZE(k,m)

  for $i = 1, \ldots, k$: **do**

    t_i = new bit vector of m 0's

for each hash function, initialize an empty bit vector of size m

Bloom Filters: Example

`bloom filter t with m = 5 that uses k = 3 hash functions`

function INITIALIZE(k,m)
  for $i = 1, \ldots, k$: **do**
    t_i = new bit vector of m 0's

Index →	0	1	2	3	4
t_1	0	0	0	0	0
t_2	0	0	0	0	0
t_3	0	0	0	0	0

Bloom Filters: Add

function ADD(X)
  for $i = 1, \ldots, k$: **do**
    $t_i[h_i(x)] = 1$

for each hash function h_i

Index into ith bit-vector, at index produced by hash function and set to 1

$h_i(x)$ → result of hash function h_i on x

Bloom Filters: Example

bloom filter t with m = 5 that uses k = 3 hash functions

function ADD(X)
  for $i = 1, \ldots, k$: **do**
    $t_i[h_i(x)] = 1$

add("thisisavirus.com")

h_1("thisisavirus.com") → 2

Index →	**0**	**1**	**2**	**3**	**4**
t_1	0	0	0	0	0
t_2	0	0	0	0	0
t_3	0	0	0	0	0

Bloom Filters: Example

bloom filter t of length m = 5 that uses k = 3 hash functions

add("thisisavirus.com")

function ADD(X)
  for $i = 1, \ldots, k$: **do**
    $t_i[h_i(x)] = 1$

h_1("thisisavirus.com") → 2

h_2("thisisavirus.com") → 1

Index →	**0**	**1**	**2**	**3**	**4**
t_1	0	0	**1**	0	0
t_2	0	0	0	0	0
t_3	0	0	0	0	0

Bloom Filters: Example

bloom filter t of length m = 5 that uses k = 3 hash functions

function ADD(X)
  for $i = 1, \ldots, k$: **do**
    $t_i[h_i(x)] = 1$

add("thisisavirus.com")

h_1("thisisavirus.com") → 2

h_2("thisisavirus.com") → 1

h_3("thisisavirus.com") → 4

Index →	0	1	2	3	4
t_1	0	0	**1**	0	0
t_2	0	**1**	0	0	0
t_3	0	0	0	0	0

Bloom Filters: Example

bloom filter t of length m = 5 that uses k = 3 hash functions

add("thisisavirus.com")

h_1("thisisavirus.com") → 2
h_2("thisisavirus.com") → 1
h_3("thisisavirus.com") → 4

function ADD(X)
  for $i = 1, \ldots, k$: **do**
    $t_i[h_i(x)] = 1$

Index →	0	1	2	3	4
t_1	0	0	**1**	0	0
t_2	0	**1**	0	0	0
t_3	0	0	0	0	**1**

Bloom Filters: Example

bloom filter t with m = 5 that uses k = 3 hash functions

function CONTAINS(x)
  return $t_1[h_1(x)] == 1 \wedge t_2[h_2(x)] == 1 \wedge \cdots \wedge t_k[h_k(x)] == 1$

contains("thisisavirus.com")

Index →	0	1	2	3	4
t_1	0	0	1	0	0
t_2	0	1	0	0	0
t_3	0	0	0	0	1

Bloom Filters: Example

bloom filter t of length m = 5 that uses k = 3 hash functions

contains("thisisavirus.com")

h_1("thisisavirus.com") → 2

function CONTAINS(x)
    return $t_1[h_1(x)] == 1 \wedge t_2[h_2(x)] == 1 \wedge \cdots \wedge t_k[h_k(x)] == 1$

True

Index →	0	1	2	3	4
t_1	0	0	1	0	0
t_2	0	1	0	0	0
t_3	0	0	0	0	1

Bloom Filters: Example

bloom filter t of length m = 5 that uses k = 3 hash functions

function CONTAINS(x)
  return $t_1[h_1(x)] == 1 \wedge t_2[h_2(x)] == 1 \wedge \cdots \wedge t_k[h_k(x)] == 1$

True   True

contains("thisisavirus.com")

h_1("thisisavirus.com") → 2

h_2("thisisavirus.com") → 1

Index →	0	1	2	3	4
t_1	0	0	**1**	0	0
t_2	0	**1**	0	0	0
t_3	0	0	0	0	**1**

Bloom Filters: Example

bloom filter t of length m = 5 that uses k = 3 hash functions

contains("thisisavirus.com")

h_1("thisisavirus.com") → 2
h_2("thisisavirus.com") → 1
h_3("thisisavirus.com") → 4

function CONTAINS(x)
return $t_1[h_1(x)] == 1 \wedge t_2[h_2(x)] == 1 \wedge \cdots \wedge t_k[h_k(x)] == 1$

True True True

Index →	0	1	2	3	4
t_1	0	0	**1**	0	0
t_2	0	**1**	0	0	0
t_3	0	0	0	0	**1**

Bloom Filters: Example

bloom filter t of length m = 5 that uses k = 3 hash functions

contains("thisisavirus.com")

h_1("thisisavirus.com") → 2
h_2("thisisavirus.com") → 1
h_3("thisisavirus.com") → 4

function CONTAINS(x)
  return $t_1[h_1(x)] == 1 \wedge t_2[h_2(x)] == 1 \wedge \cdots \wedge t_k[h_k(x)] == 1$

True True True

Index	0	1	2	3	4
t_1	0	0	1	0	0
t_2	0	1	0	0	0
t_3	0	0	0	0	1

Since all conditions satisfied, returns True (correctly)

Bloom Filters: Contains

function CONTAINS(x)

$$\textbf{return } t_1[h_1(x)] == 1 \wedge t_2[h_2(x)] == 1 \wedge \cdots \wedge t_k[h_k(x)] == 1$$

Returns True if the bit vector t_i for each hash function has bit 1 at index determined by h_i(x), otherwise returns False

Bloom Filters: False Positives

bloom filter t of length m = 5 that uses k = 3 hash functions

add("totallynotsuspicious.com")

function ADD(x)
  for $i = 1, \ldots, k$: **do**
    $t_i[h_i(x)] = 1$

Index →	0	1	2	3	4
t_1	0	0	1	0	0
t_2	0	1	0	0	0
t_3	0	0	0	0	1

Bloom Filters: False Positives

bloom filter t of length m = 5 that uses k = 3 hash functions

function ADD(X)
  for $i = 1, \ldots, k$: **do**
    $t_i[h_i(x)] = 1$

add("totallynotsuspicious.com")

h_1("totallynotsuspicious.com") → 1

Index →	**0**	**1**	**2**	**3**	**4**
t_1	0	0	1	0	0
t_2	0	1	0	0	0
t_3	0	0	0	0	1

Bloom Filters: False Positives

bloom filter t of length m = 5 that uses k = 3 hash functions

function ADD(X)
  for $i = 1, \ldots, k$: **do**
    $t_i[h_i(x)] = 1$

add("totallynotsuspicious.com")

h_1("totallynotsuspicious.com") → 1

h_2("totallynotsuspicious.com") → 0

Index →	0	1	2	3	4
t_1	0	**1**	1	0	0
t_2	0	1	0	0	0
t_3	0	0	0	0	1

Bloom Filters: False Positives

bloom filter t of length m = 5 that uses k = 3 hash functions

function ADD(X)
  for $i = 1, \ldots, k$: **do**
    $t_i[h_i(x)] = 1$

add("totallynotsuspicious.com")

h_1("totallynotsuspicious.com") → 1

h_2("totallynotsuspicious.com") → 0

h_3("totallynotsuspicious.com") → 4

Index →	0	1	2	3	4
t_1	0	**1**	1	0	0
t_2	**1**	1	0	0	0
t_3	0	0	0	0	1

Bloom Filters: False Positives

bloom filter t of length m = 5 that uses k = 3 hash functions

function ADD(X)
  for $i = 1, \ldots, k$: **do**
    $t_i[h_i(x)] = 1$

add("totallynotsuspicious.com")

h_1("totallynotsuspicious.com") → 1

h_2("totallynotsuspicious.com") → 0

h_3("totallynotsuspicious.com") → 4

Collision, is already set to 1

Index →	0	1	2	3	4
t_1	0	**1**	1	0	0
t_2	**1**	1	0	0	0
t_3	0	0	0	0	1

Bloom Filters: False Positives

bloom filter t of length m = 5 that uses k = 3 hash functions

function ADD(X)
  for $i = 1, \ldots, k$: **do**
    $t_i[h_i(x)] = 1$

add("totallynotsuspicious.com")

h_1("totallynotsuspicious.com") → 1

h_2("totallynotsuspicious.com") → 0

h_3("totallynotsuspicious.com") → 4

Index →	**0**	**1**	**2**	**3**	**4**
t_1	0	**1**	1	0	0
t_2	**1**	1	0	0	0
t_3	0	0	0	0	**1**

Bloom Filters: Example

bloom filter t of length m = 5 that uses k = 3 hash functions

contains("verynormalsite.com")

function CONTAINS(x)
  return $t_1[h_1(x)] == 1 \wedge t_2[h_2(x)] == 1 \wedge \cdots \wedge t_k[h_k(x)] == 1$

Index →	0	1	2	3	4
t_1	0	1	1	0	0
t_2	1	1	0	0	0
t_3	0	0	0	0	1

Bloom Filters: Example

bloom filter t of length m = 5 that uses k = 3 hash functions

contains("verynormalsite.com")

h_1("verynormalsite.com") → 2

function CONTAINS(x)
    return $t_1[h_1(x)] == 1 \wedge t_2[h_2(x)] == 1 \wedge \cdots \wedge t_k[h_k(x)] == 1$

True

Index →	0	1	2	3	4
t_1	0	1	**1**	0	0
t_2	1	1	0	0	0
t_3	0	0	0	0	1

Bloom Filters: Example

bloom filter t of length m = 5 that uses k = 3 hash functions

contains("verynormalsite.com")

h_1("verynormalsite.com") → 2

h_2("verynormalsite.com") → 0

function CONTAINS(x)
return $t_1[h_1(x)] == 1 \wedge t_2[h_2(x)] == 1 \wedge \cdots \wedge t_k[h_k(x)] == 1$

True True

Index →	0	1	2	3	4
t_1	0	1	**1**	0	0
t_2	**1**	1	0	0	0
t_3	0	0	0	0	1

Bloom Filters: Example

bloom filter t of length m = 5 that uses k = 3 hash functions

contains("verynormalsite.com")

function CONTAINS(x)
  return $t_1[h_1(x)] == 1 \land t_2[h_2(x)] == 1 \land \cdots \land t_k[h_k(x)] == 1$

True True True

h_1("verynormalsite.com") → 2
h_2("verynormalsite.com") → 0
h_3("verynormalsite.com") → 4

Index →	0	1	2	3	4
t_1	0	1	**1**	0	0
t_2	**1**	1	0	0	0
t_3	0	0	0	0	**1**

Bloom Filters: Example

bloom filter t of length m = 5 that uses k = 3 hash functions

contains("verynormalsite.com")

function CONTAINS(x)
  return $t_1[h_1(x)] == 1 \wedge t_2[h_2(x)] == 1 \wedge \cdots \wedge t_k[h_k(x)] == 1$

True True True

h_1("verynormalsite.com") → 2
h_2("verynormalsite.com") → 0
h_3("verynormalsite.com") → 4

Since all conditions satisfied, returns True (incorrectly)

Index	**0**	**1**	**2**	**3**	**4**
t_1	0	1	**1**	0	0
t_2	**1**	1	0	0	0
t_3	0	0	0	0	**1**

Bloom Filters: Summary

- An empty bloom filter is an empty k x m bit array with all values initialized to zeros
  - k = number of hash functions
  - m = size of each array in the bloom filter
- add(x) runs in O(k) time
- contains(x) runs in O(k) time
- requires O(km) space (in bits!)
- Probability of false positives from collisions can be reduced by increasing the size of the bloom filter

Bloom Filters: Application

- Google Chrome has a database of malicious URLs, but it takes a long time to query.
- Want an in-browser structure, so needs to be efficient and be space-efficient
- Want it so that can check if a URL is in structure:
  - If return False, then definitely not in the structure (don't need to do expensive database lookup, website is safe)
  - If return True, the URL may or may not be in the structure. Have to perform expensive lookup in this rare case.

Bloom Filters: Many Applications

- Any scenario where space and efficiency are important.
- Used a lot in networking
- In distributed systems when want to check consistency of data across different locations, might send a Bloom filter rather than the full set of data being stored.
- Google BigTable uses Bloom filters to reduce disk lookups
- Internet routers often use Bloom filters to track blocked IP addresses.
- And on and on…

Bloom Filters

It’s typical of randomized algorithms and randomized data structures to be…

- **Simple**
- **Fast**
- **Efficient**
- **Elegant**
- **Useful!**