Variance

CSE 312 Su23
Lecture 9

Announcements

Midterm: next Monday, July 17th, 12:00-1:00 PCAR 391

Details on Ed

HW3 Due Wednesday; next HW out after the midterm

Where are we?

$X : \Omega \to \mathbb{R}$

A random variable is a way to summarize what outcome you saw.

The Expectation of a random variable is its average value.

A way to summarize a random variable

Expectation is **linear**

$\mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y]$.

$X + Y$ is a random variable – it's a function that outputs a number given an outcome (or, here, a combination of outcomes).

Variance

Another one number summary of a random variable.

But wait, we already have expectation, what's this for?

Consider these two games

$$P_{X_1}(x_1) = \begin{cases} 1/2 & x_1 = 1 \\ 1/2 & x_1 = -1 \end{cases}$$

Would you be willing to play these games?

$$\mathbb{E}[X_1] = \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot -1 = 0$$

Game 1: I will flip a fair coin; if it's heads, I pay you \$1. If it's tails, you pay me \$1. Let X_1 be your profit if you play game 1

Game 2: I will flip a fair coin; if it's heads, I pay you \$10,000. If it's tails, you pay me \$10,000. Let X_2 be your profit if you play game 2.

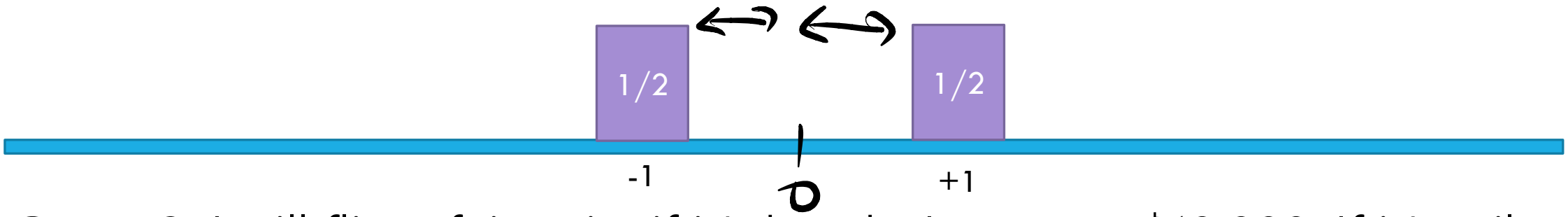
$$P_{X_2}(x_2) = \begin{cases} 1/2 & x_2 = 10000 \\ 1/2 & x_2 = -10000 \end{cases}$$

Both games are "fair" ($\mathbb{E}[X_1] = \mathbb{E}[X_2] = 0$)

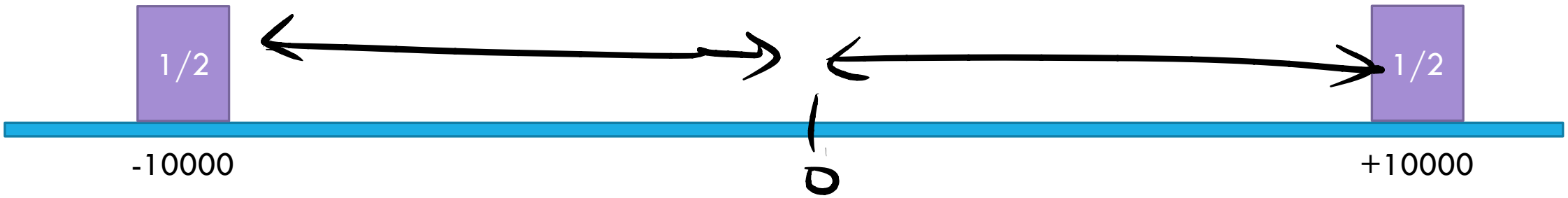
$$\mathbb{E}[X_2] = \frac{1}{2} \cdot 10000 + \frac{1}{2} \cdot (-10000) = 0$$

Consider these two games

Game 1: I will flip a fair coin; if it's heads, I pay you \$1. If it's tails, you pay me \$1. Let X_1 be your profit if you play game 1



Game 2: I will flip a fair coin; if it's heads, I pay you \$10,000. If it's tails, you pay me \$10,000. Let X_2 be your profit if you play game 2.



What's the difference

Expectation tells you what the average will be…

But it doesn't tell you how "extreme" your results could be.

Nor how likely those extreme results are.

Game 2 has many (well, only) very extreme results.

In expectation they "cancel out" but if you can only play once…

…it would be nice to measure that.

Designing a Measure – Try 1

Well let's measure how far all the events are away from the center, and how likely they are

dist

$$\sum_{\omega} \mathbb{P}(\omega) \cdot (X(\omega) - \mathbb{E}[X])$$

Designing a Measure – Try 1

Well let's measure how far all the events are away from the center, and how likely they are

$$\sum_{\omega} \mathbb{P}(\omega) \cdot (X(\omega) - \mathbb{E}[X])$$

What happens with Game 1?

$$\frac{1}{2} \cdot (1 - 0) + \frac{1}{2} \cdot (-1 - 0)$$

$$= \frac{1}{2} - \frac{1}{2} = 0$$

What happens with Game 2?

$$\frac{1}{2} \cdot (100000 - 0) + \frac{1}{2} \cdot (-100000 - 0)$$

$$5000 - 5000 = 0$$

Designing a Measure – Try 2

How do we prevent cancelling? Squaring makes everything positive.

$$\sum_{\omega} \mathbb{P}(\omega) \cdot (X(\omega) - \mathbb{E}[X])^2$$

What happens with Game 1?

$$\frac{1}{2} \cdot (1 - 0)^2 + \frac{1}{2} \cdot (-1 - 0)^2$$

$$\frac{1}{2} + \frac{1}{2} = 1$$

What happens with Game 2?

$$\frac{1}{2} \cdot (100000 - 0)^2 + \frac{1}{2} \cdot (-100000 - 0)^2$$

$$5{,}000{,}000{,}000 + 5{,}000{,}000{,}000 = 10^{10}$$

Variance

Variance

The variance of a random variable X is

$$\mathrm{Var}(X) = \sum_{\omega} \mathbb{P}(\omega) \cdot (X(\omega) - \mathbb{E}[X])^2 = \mathbb{E}[(X - \mathbb{E}[X])^2] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$$

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

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

$$= \mathbb{E}[X^2] - \mathbb{E}[X]^2$$

$\mathbb{E}[\mathbb{E}[X]^2]$

Variance

Variance

The variance of a random variable X is

$$\mathrm{Var}(X) = \sum_{\omega} \mathbb{P}(\omega) \cdot (X(\omega) - \mathbb{E}[X])^2 = \mathbb{E}[(X - \mathbb{E}[X])^2] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$$

def $\mathbb{E}[X]$

$$Y_i = \begin{cases} 0 \\ 1 \end{cases}$$

Variance

Variance

The variance of a random variable X is

$$\mathrm{Var}(X) = \sum_{\omega} \mathbb{P}(\omega) \cdot (X(\omega) - \mathbb{E}[X])^2 = \mathbb{E}[(X - \mathbb{E}[X])^2] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$$

The first form forms are the definition. The last one is an algebra trick.

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

Standard Deviation

$$Var(X) = E[X^2] - (E[X])^2$$

cm^2

Standard Deviation

$$\sigma(X) = \sqrt{Var(X)}$$

$\downarrow$

cm

Also measures spread of a random variable X.

Variance squares the units of X, while standard deviation has the same units: this can be useful.

Variance of a die

Let X be the result of rolling a fair die.

$$\mathrm{Var}(X) = \mathbb{E}[(X - \mathbb{E}[X])^2] = \mathbb{E}[(X - 3.5)^2]$$

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

$$= \frac{35}{12} \approx 2.92.$$

Variance of a die

Let X be the result of rolling a fair die.

$$\mathrm{Var}(X) = \mathbb{E}[(X - \mathbb{E}[X])^2] = \mathbb{E}[(X - 3.5)^2]$$

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

$$= \frac{35}{12} \approx 2.92.$$

1 - 6

Or $\mathbb{E}[X^2] - (E[X])^2 = \sum_{k=1}^{6} \frac{1}{6} \cdot k^2 - 3.5^2 = \frac{91}{6} - 3.5^2 \approx 2.92$

LOTUS

Variance of n Coin Flips

Flip a coin n times, where it comes up heads with probability p each time (independently). Let X be the total number of heads.

We saw last time $\mathbb{E}[X] = np$.

$$X_i = \begin{cases} 1 & \text{if flip } i \text{ is heads} \\ 0 & \text{otherwise} \end{cases}$$

$$X = \sum_{i=1}^{n} X_i$$

$$\mathbb{E}[X] = \mathbb{E}\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} \mathbb{E}[X_i] = \sum_{i=1}^{n} p = np.$$

LOE

Variance of n Coin Flips

Flip a coin n times, where it comes up heads with probability p each time (independently). Let X be the total number of heads.

What about $\mathbf{Var}(X)$

$$\mathbb{E}[(X - \mathbb{E}[X])^2] = \sum_{\omega} \mathbb{P}(\omega)(X(\omega) - np)^2$$

$$= \sum_{k=0}^{n} \binom{n}{k} \cdot p^k (1-p)^{n-k} \cdot (k - np)^2$$

Algebra time?

Variance

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

$1^2 \cdot p + 0^2 \cdot (1-p) = p \qquad = p(1-p)$

Variance Adds for Independent RVs

If X and Y are independent, then $\text{Var}(X+Y) = \text{Var}(X) + \text{Var}(Y)$

Are the X_i independent? Yes! In this problem X_i is independent of X_j for $i \neq j$ where

$$X_i = \begin{cases} 1 & \text{if flip } i \text{ was heads} \\ 0 & \text{otherwise} \end{cases}$$

$$X = \sum_{i=1}^{n} X_i$$

$$\text{Var}(X) = \text{Var}\left(\sum_{i=1}^{n} X_i\right) = \sum_{i=1}^{n} \text{Var}(X_i) = \sum_{i=1}^{n} p(1-p) = np(1-p)$$

Variance

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

What's the $\mathrm{Var}(X_i)$?

$\mathbb{E}[(X_i - \mathbb{E}[X_i])^2]$

$= \mathbb{E}[(X_i - p)^2]$

$= p(1-p)^2 + (1-p)(0-p)^2$

$= p(1-p)[(1-p) + p] = p(1-p).$

OR $Var(X_i) = \mathbb{E}\left[X_i^2\right] - \mathbb{E}[X_i]^2 = \mathbb{E}[X_i] - p^2 = p - p^2 = p(1-p)$.

Plugging In

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

What's the $\mathrm{Var}(X_i)$?

$p(1-p)$.

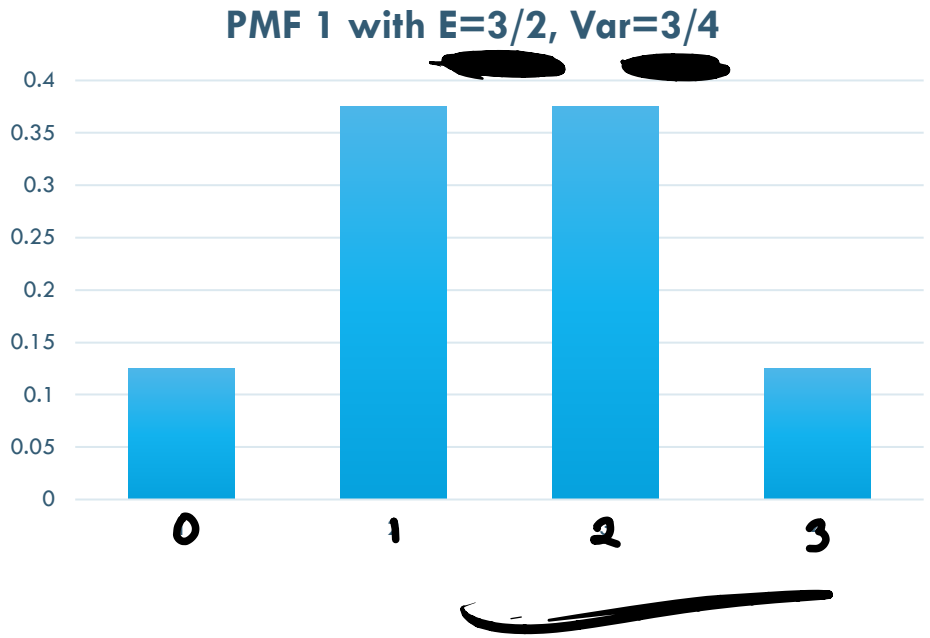
$\mathrm{Var}(X) = \sum_{i=1}^{n} p(1-p) = np(1-p)$.

Expectation and Variance aren't everything

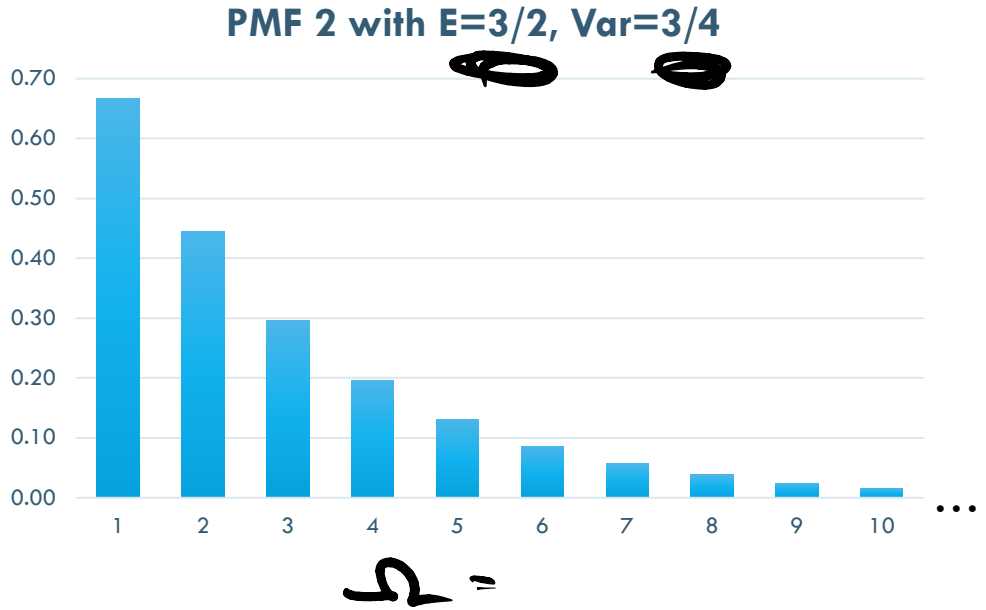
Alright, so expectation and variance is everything right?

No!

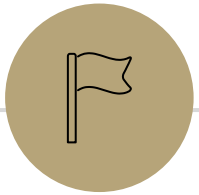
Flip a fair coin 3 times indep. Count heads.



Flip a biased coin (prob heads=2/3) until heads. Count flips.



A PMF or CDF *does* fully describe a random variable.



Useful Facts

Make a prediction

How should $\mathrm{Var}(X + c)$ relate to $Var(X)$ if c is a constant?

How should $\mathrm{Var}(\mathrm{aX})$ relate to $\mathrm{Var}(X)$ is a is a constant?

$c > 0$

$\longrightarrow$

$\mathbb{E}[X + c] = \mathbb{E}[X] + c$

Facts About Variance

$\mathrm{Var}(X + c) = \mathrm{Var}(X)$

Facts About Variance

$\mathrm{Var}(X + c) = \mathrm{Var}(X)$

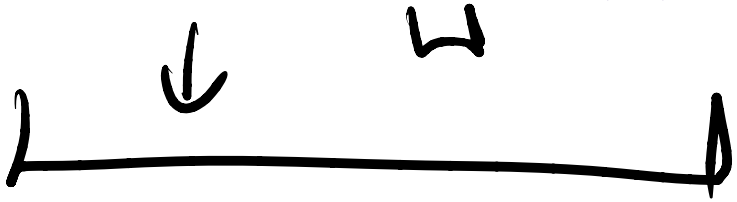
Proof:

$$\mathrm{Var}(X + c) = \mathbb{E}[(X + c)^2] - \mathbb{E}[X + c]^2$$
$$= \mathbb{E}[X^2] + \mathbb{E}[2Xc] + \mathbb{E}[c^2] - (\mathbb{E}[X] + c)^2$$
$$= \mathbb{E}[X^2] + 2c\mathbb{E}[X] + c^2 - \mathbb{E}[X]^2 - 2c\mathbb{E}[X] - c^2$$
$$= \mathbb{E}[X^2] - \mathbb{E}[X]^2$$
$$= \mathrm{Var}(X)$$

Facts about Variance

$\mathrm{Var}(aX) = a^2\mathrm{Var}(X)$

$E[aX] = aE[X]$



$-1 \quad 0 \quad 1$ $\quad (3)$

$-3 \quad 0 \quad 3$ $\quad \mathrm{Var}(-X) = (-1)^2\mathrm{Var}(X)$

$= \mathrm{Var}(X)$

Facts about Variance

$$\mathrm{Var}(aX) = a^2\mathrm{Var}(X)$$

Proof:

$$= \mathbb{E}[(aX)^2] - (\mathbb{E}[aX])^2$$
$$= a^2\mathbb{E}[X^2] - (a\mathbb{E}[X])^2$$
$$= a^2\mathbb{E}[X^2] - a^2\mathbb{E}[X]^2$$
$$= a^2(\mathbb{E}[X^2] - \mathbb{E}[X]^2)$$

Facts about Variance

In general, $Var(X + Y) \neq Var(X) + Var(Y)$

Proof by counterexample:

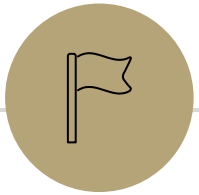
Let X be a RV with PMF P(X=1) = P(X=-1) = 1/2.

$\mathbb{E}[X] = 0,\ Var(X) = 1$

Let Y = -X. $\mathbb{E}[Y] = 0,\ Var(Y) = 1$

So, Var(X) + Var(Y) = 2.

But, Var(X+Y) = Var(X + (-X)) = Var(0) = 0.



Bloom Filters

Bloom Filters Intro

Our very first **probabilistic** data structure ☺

Can be looked at as a probabilistic alternative to hash tables

Rely on **hashing**

Context: Hashing

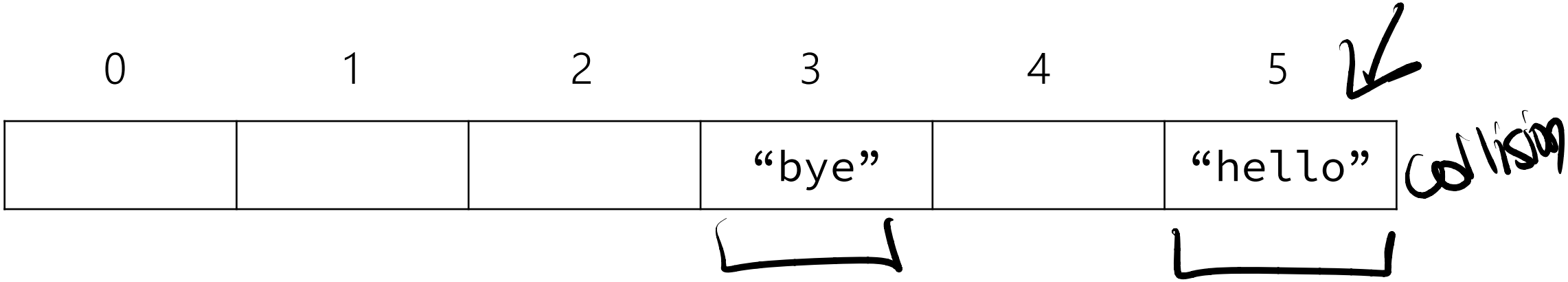
Hash functions “codify” input values, using some mathematical formula

For a hash function h(), we could see an output like:

h(“hello”) = 5, h(“bye”) = 3

h("doggy")=5

Items are then mapped to ‘buckets’ with indexes based on this value



Context: Hashing

Hash functions “codify” input values, using some mathematical formula

E.g. h(“hello”) = 5, h(“bye”) = 3

Items are then mapped to ‘buckets’ with indexes based on this value

Collisions occur when two values have the same code and map to the same index

Ideally: good distribution of values to avoid collisions (uniform!) & fast

Motivating Bloom Filters

Google Chrome has a database of malicious URLs, but it takes a long time to query.

We would like:

Want an in-browser structure, so needs to be **time-efficient and space-efficient**.

We have more safe URLs than malicious URLs. We can improve efficiency by speeding up the dominant case. That is: let's guarantee that our bloom filter always correctly identifies safe URLs.

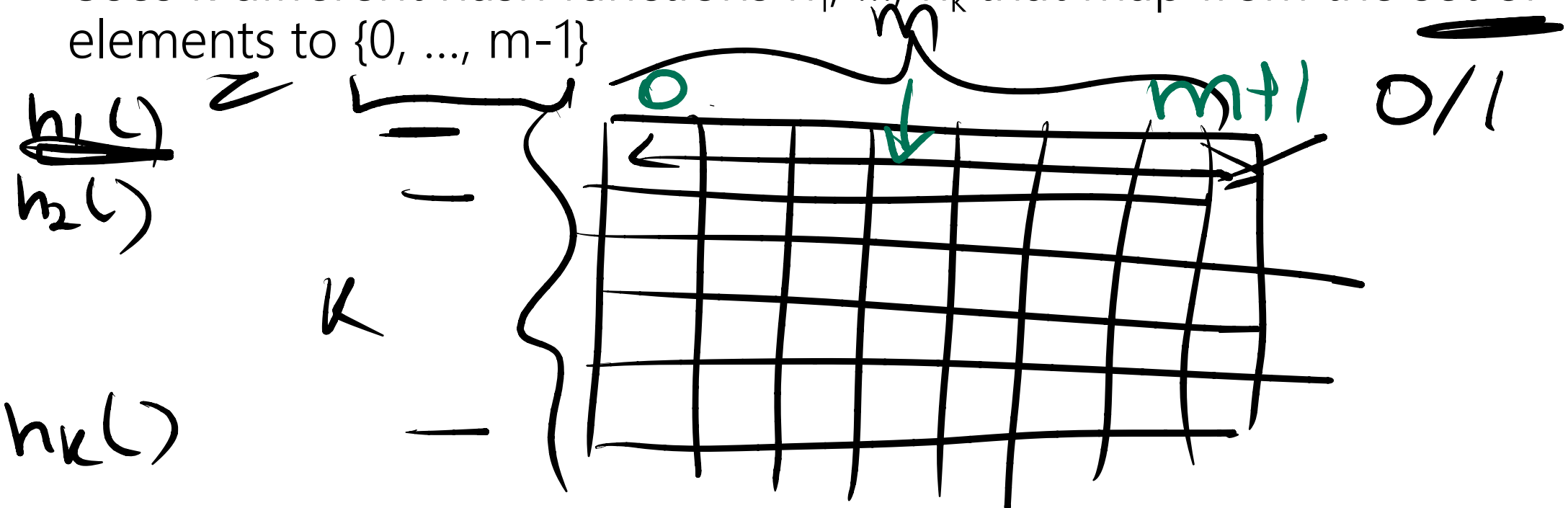
Alright alright… how does this look?

A $k \times m$ bit array

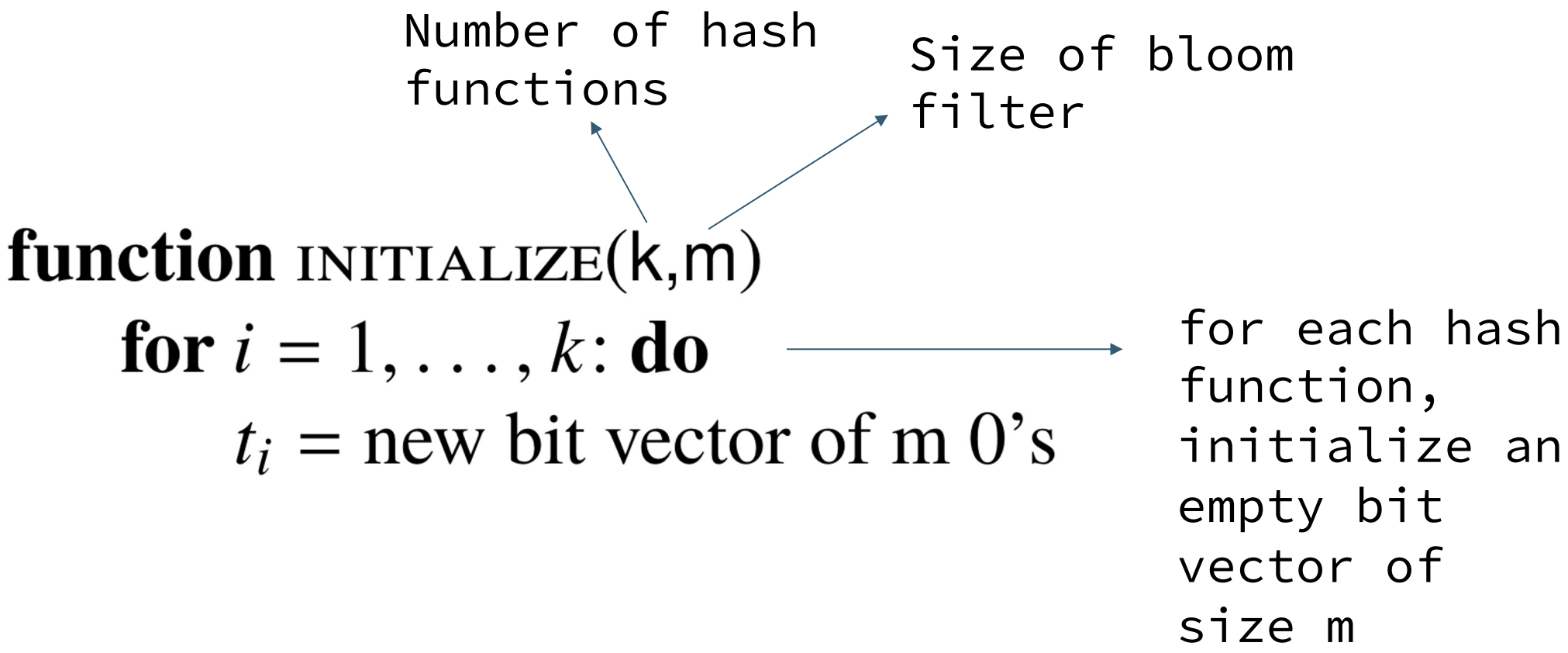
k rows t_1, …, t_k

Each row is an m-bit vector

Uses k different hash functions h_1, …, h_k that map from the set of elements to {0, …, m-1}



Initialization



Initialization

bloom filter t of length 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

Add

function ADD(x)

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

$t_i[h_i(x)] = 1$

for each hash function h_i

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

Index into ith bit-vector and set to 1

Add

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

```
function ADD(x)
    for i = 1, . . . , 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	1

Add

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	**1**

Contains

function CONTAINS(x)

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

```
Returns True if the bit vector
for each hash function has bit 1
at index determined by that hash
function, otherwise returns
False
```

Contains

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 AND True AND True => return True

contains(“thisisavirus.com”)

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

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**

Collision!

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 for hash function 3, at index 4:
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

False Positives

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

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

contains(“verynormalsite.com”)
h_1(“verynormalsite.com”) → 2
h_2(“verynormalsite.com”) → 0
h_3(“verynormalsite.com”) → 4

Return True AND True And True -> True!

But we never added this website to our bloom filter! This is a **false positive.**

True ∧ T ∧ T
= T

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

Operations

Supports two key operations:

1. add(x) - adds x to bloom filter
2. contains(x) - returns true if x in bloom filter, otherwise returns false

    If return false, definitely not in structure. (don't need to do expensive database lookup, website is safe) true negatives

    If return true, possibly in the structure (some false positives). Have to perform expensive lookup in this rare case. false + true positive

Does not support the following operations:

- removing an element from bloom filter
- giving a collection of elements in the structure

So… how often do we see those false positives?

After adding n URLS to a filter with k hash functions and m columns, we check if URL x is inserted in the bloom filter.

I.e., what is the probability, for some x, that contains(x) returns true if add(x) was never executed before?

Assumptions:

Each $h_i(x)$ is uniformly distributed in {0, …, m-1} for all elements.

Hash functions outputs are mutually independent.

So… how often do we see those false positives?

After adding n URLS to a filter with k hash functions and m columns, we check if URL x is inserted in the bloom filter.

★ $\mathbb{P}(false\ positive)$

$= \mathbb{P}(h_1(x) = 1 \cap h_2(x) = 1 \cap \cdots \cap h_k(x) = 1)$ → mut idp

$= \prod_{i=1}^{k} \mathbb{P}(h_i(x) = 1))$ ←

$= \prod_{i=1}^{k} 1 - \mathbb{P}(h_i(x) = 0)$

$\mathbb{P}(h_1(x)=1) \times$

$\mathbb{P}(h_2(x)=1) \times \ldots$

$\ldots \times \mathbb{P}(h_k(x)=1))$

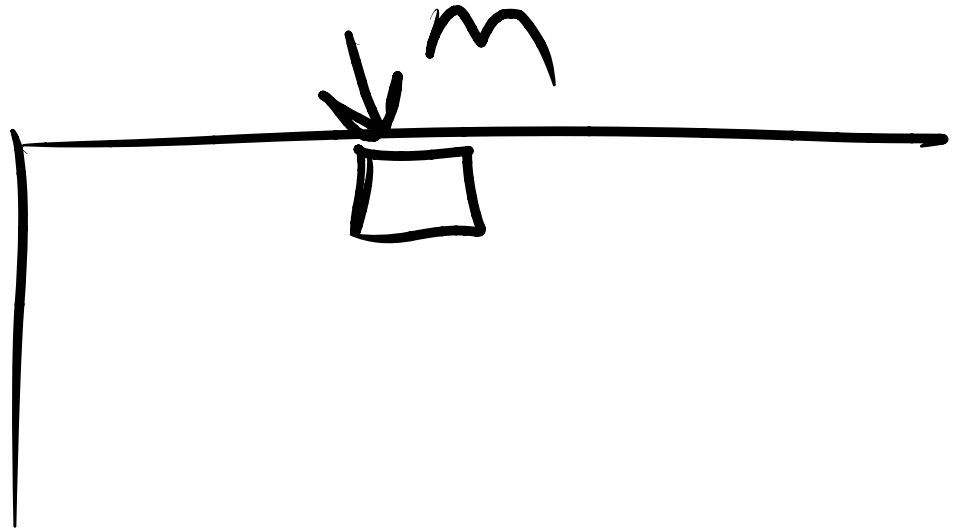
So… how often do we see those false positives?

After adding n URLS to a filter with k hash functions and m columns, we check if URL x is inserted in the bloom filter.

$$= \prod_{i=1}^{k} 1 - \mathbb{P}(h_i(x) = 0)$$

$$= \prod_{i=1}^{k} 1 - \left(\frac{m-1}{m}\right)^{n}$$

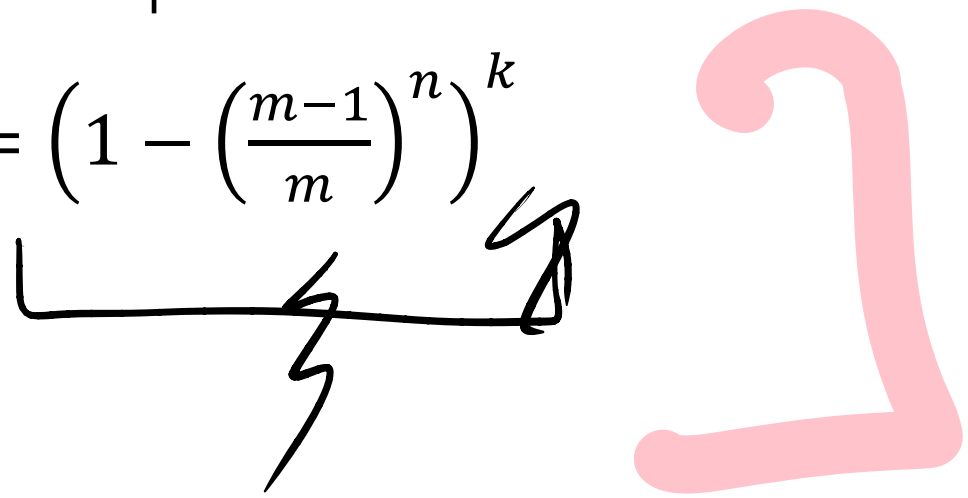
$$= \left(1 - \left(\frac{m-1}{m}\right)^{n}\right)^{k}$$



False positives, false positives everywhere

False positive rate for a k x m bloom filter with n elements:

$$= \left(1 - \left(\frac{m-1}{m}\right)^{n}\right)^{k}$$



If we know how many elements we want to store (n), we should be able to adjust k (number of hash functions) and m (size of bloom filter) in order to minimize the false positive rate.

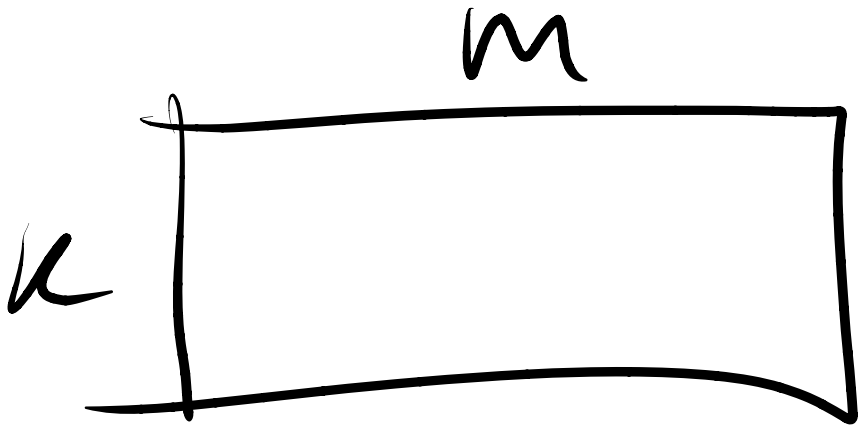
Space

We store 5,000,000 URLs. Each URL is 40 Bytes.

Bloom filter with k = 30, m = 2,500,000. (FPR = 1.28%)

Hash Table: $5{,}000{,}000 \times 40B = 200\ MB$ (optimistic)

Bloom Filter: $2{,}500{,}000 \times 30 = 75{,}000{,}000\ bits < 10\ MB$



Time

Say the average user visits 102,000 URLs in a year, of which 2,000 are malicious.

Lookup time: database – 0.5s, Bloom filter – 1 ms

False positive rate: 3%

Database: $102{,}000 \times 0.5\ s = 51{,}000\ seconds$

Bloom Filter:

$$102{,}000 \times 0.001 + (0.03 \times 100{,}000 + 2000) \times 0.5 = 2{,}602\ seconds$$

Why false positives?

Why accept false positives?

Speed - add and contains have good time complexity

Space - doesn't require much space relative to items to be stored

$k \times m$

We **can distinguish** false positives from true positives with extra cost (of looking at the actual database) – this is OK if we mostly expect negatives and have a low false positive rate!

Comparison

Hash Tables

Store the data itself, so they need as much space as the data takes up.

For e.g., if storing malicious URLs, need space to store the URLs as strings.

Bloom Filters

Don’t store the data itself – use a bit (0/1) array. Bits are small and use much less space.

Other Applications of Bloom Filters

Often used in networking, for e.g., when internet routers want to track blocked IP addresses

- Cache filtering – prevent "one-hit-wonders" from being stored by content delivery networks

In distributed systems, when we want to check consistency of data across different locations -> compare bloom filters instead of full set of data stored

Google BigTable uses Bloom filters to reduce disk lookups