Announcements

• Midterm feedback/evaluation is open till next Tuesday. Please take a few mins to fill it out.
Agenda

• Bloom Filters Example & Analysis
• Zoo of Discrete RVs
  – Uniform Random Variables
  – Bernoulli Random Variables
  – Binomial Random Variables
  – Applications
Bloom Filters

to the rescue

(Named after Burton Howard Bloom)
Bloom Filters

- Stores information about a set of elements \( S \subseteq U \).
- Supports two operations:
  1. \( \text{add}(x) \) - adds \( x \in U \) to the set \( S \)
  2. \( \text{contains}(x) \) – ideally: true if \( x \in S \), false otherwise

Possible false positives

Combine with fallback mechanism – can distinguish false positives from true positives with extra cost

Two goals:
1. **Very fast** (ideally constant time) answers to queries “Is \( x \in S \)” for any \( x \in U \).
2. **Minimal storage** requirements.
Bloom Filters – Ingredients

Basic data structure is a $k \times m$ binary array “the Bloom filter”
- $k$ rows $t_1, \ldots, t_k$, each of size $m$
- Think of each row as an $m$-bit vector

$k$ different hash functions $h_1, \ldots, h_k : U \to [m]$

We idealize each hash function $h_1$ as assigning each input $x$ to a random output $y$ in $[m]$
Bloom Filters – Three operations

- Set up Bloom filter for $S = \emptyset$

- Update Bloom filter for $S \leftarrow S \cup \{x\}$

- Check if $x \in S$

**function** \text{INITIALIZE}(k, m)

\begin{verbatim}
for i = 1, ..., k: do 
    ti = new bit vector of m 0s 
\end{verbatim}

**function** \text{ADD}(x)

\begin{verbatim}
for i = 1, ..., k: do 
    ti[hi(x)] = 1 
\end{verbatim}

**function** \text{CONTAINS}(x)

\begin{verbatim}
return t1[h1(x)] == 1 \land t2[h2(x)] == 1 \land \cdots \land tk[hk(x)] == 1 
\end{verbatim}
Bloom Filters: Example

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

```plaintext
function INITIALIZE($k, m$)
    for $i = 1, \ldots, k$: do
        $t_i = \text{new bit vector of } m \text{ 0s}$
```

<table>
<thead>
<tr>
<th>Index →</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$t_2$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$t_3$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Bloom Filters: Example

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

<table>
<thead>
<tr>
<th>Index</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$t_2$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$t_3$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**function** `ADD(x)`

```plaintext
for i = 1, ..., k: do
    $t_i[h_i(x)] = 1$
```

add(“thisisavirus.com”)

$h_1(“thisisavirus.com”) → 2$
Bloom Filters: Example

Bloom filter \( t \) of length \( m = 5 \) that uses \( k = 3 \) hash functions

```
function \( \text{ADD}(x) \)
for \( i = 1, \ldots, k \): do
\( t_i[h_i(x)] = 1 \)
```

- \( h_1(\text{"thisisavirus.com"}) \rightarrow 2 \)
- \( h_2(\text{"thisisavirus.com"}) \rightarrow 1 \)

![Table showing Bloom filter state after adding "thisisavirus.com".](image)

<table>
<thead>
<tr>
<th>Index ( \rightarrow )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_1 )</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( t_2 )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( t_3 )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
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

$t_i[h_i(x)] = 1$

<table>
<thead>
<tr>
<th>Index →</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$t_2$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$t_3$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

add(“thisisavirus.com”)

$h_1(“thisisavirus.com”) \rightarrow 2$

$h_2(“thisisavirus.com”) \rightarrow 1$

$h_3(“thisisavirus.com”) \rightarrow 4$
Bloom Filters: Example

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

**function** `ADD(x)`

```plaintext
for $i = 1, \ldots, k$: do
  $t_i[h_i(x)] = 1$
```

`add("thisisavirus.com")`

- $h_1(\text{"thisisavirus.com"}) \rightarrow 2$
- $h_2(\text{"thisisavirus.com"}) \rightarrow 1$
- $h_3(\text{"thisisavirus.com"}) \rightarrow 4$

<table>
<thead>
<tr>
<th>Index →</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$t_2$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$t_3$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Bloom Filters: Example

Bloom filter \( t \) of length \( m = 5 \) that uses \( k = 3 \) hash functions

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

Since all conditions satisfied, returns True (correctly)

<table>
<thead>
<tr>
<th>Index ( \rightarrow )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_1 ) ( h_1 )</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( t_2 ) ( h_2 )</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( t_3 ) ( h_3 )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

contains("thisisavirus.com")

\( h_1("thisisavirus.com") \rightarrow 2 \)

\( h_2("thisisavirus.com") \rightarrow 1 \)

\( h_3("thisisavirus.com") \rightarrow 4 \)
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`)

- \( h_1(`totallynotsuspicious.com`) \) \( \rightarrow 1 \)
- \( h_2(`totallynotsuspicious.com`) \) \( \rightarrow 0 \)
- \( h_3(`totallynotsuspicious.com`) \) \( \rightarrow 4 \)

<table>
<thead>
<tr>
<th>Index ( \rightarrow )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_1 )</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( t_2 )</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( t_3 )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Bloom Filters: False Positives

Bloom filter \( t \) of length \( m = 5 \) that uses \( k = 3 \) hash functions

**function** `CONTAINS(x)`

```plaintext
return \( t_1[h_1(x)] = 1 \land t_2[h_2(x)] = 1 \land \cdots \land t_k[h_k(x)] = 1 \)
```

Since all conditions satisfied, returns **True** (incorrectly)

<table>
<thead>
<tr>
<th>Index ( \rightarrow )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>t_1</code> (`verynormalsite.com``)</td>
<td>0</td>
<td>1</td>
<td><strong>1</strong></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><code>t_2</code> (`verynormalsite.com``)</td>
<td>1</td>
<td><strong>1</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><code>t_3</code> (`verynormalsite.com``)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td><strong>1</strong></td>
</tr>
</tbody>
</table>
Analysis: False positive probability

**Question:** For an element \( x \in U \), what is the probability that \( \text{contains}(x) \) returns true if \( \text{add}(x) \) was never executed before?

Probability over what?!

Over the choice of the \( h_1, \ldots, h_k \)

Assumptions for the analysis (somewhat stronger than for ordinary hashing):

- Each \( h_i(x) \) is uniformly distributed in \([m]\) for all \( x \) and \( i \)
- Hash function outputs for each \( h_i \) are mutually independent (not just in pairs)
- Different hash functions are independent of each other
False positive probability – Events

Assume we perform \(\text{add}(x_1), \ldots, \text{add}(x_n)\)

\(\text{contains}(x)\) for \(x \notin \{x_1, \ldots, x_n\}\)

Event \(E_i\) holds iff \(h_i(x) \in \{h_i(x_1), \ldots, h_i(x_n)\}\)

\[
P(\text{false positive}) = P(E_1 \cap E_2 \cap \cdots \cap E_k) = \prod_{i=1}^{k} P(E_i)
\]

\(h_1, \ldots, h_k\) independent
False positive probability – Events

Event $E_i$ holds iff $h_i(x) \in \{h_i(x_1), ..., h_i(x_n)\}$

Event $E_i^c$ holds iff $h_i(x) \neq h_i(x_1)$ and ... and $h_i(x) \neq h_i(x_n)$

$$P(E_i^c) = \sum_{z=1}^{m} P(h_i(x) = z) \cdot P(E_i^c | h_i(x) = z)$$

LTP
False positive probability – Events

Event $E_i^c$ holds iff $h_i(x) \neq h_i(x_1)$ and ... and $h_i(x) \neq h_i(x_n)$

\[
P(E_i^c \mid h_i(x) = z) = P(h_i(x_1) \neq z, ..., h_i(x_n) \neq z \mid h_i(x) = z)
\]

\[
= P(h_i(x_1) \neq z, ..., h_i(x_n) \neq z)
\]

\[
= \prod_{j=1}^{n} P(h_i(x_j) \neq z)
\]

\[
= \prod_{j=1}^{n} \left(1 - \frac{1}{m}\right) = \left(1 - \frac{1}{m}\right)^n
\]

Independence of values of $h_i$ on different inputs

Outputs of $h_i$ uniformly spread

\[
P(E_i^c) = \sum_{z=1}^{m} P(h_i(x) = z) \cdot P(E_i^c \mid h_i(x) = z) = \left(1 - \frac{1}{m}\right)^n
\]
False positive probability – Events

Event $E_i$ holds iff $h_i(x) \in \{h_i(x_1), \ldots, h_i(x_n)\}$

Event $E_i^c$ holds iff $h_i(x) \neq h_i(x_1)$ and ... and $h_i(x) \neq h_i(x_n)$

$$P(E_i^c) = \left(1 - \frac{1}{m}\right)^n$$

$$\text{FPR} = \prod_{i=1}^{k} \left(1 - P(E_i^c)\right) = \left(1 - \left(1 - \frac{1}{m}\right)^n\right)^k$$
False Positivity Rate – Example

\[ \text{FPR} = \left( 1 - \left( 1 - \frac{1}{m} \right)^{\frac{n}{k}} \right) \]

e.g., \( n = 5,000,000 \)
\( k = 30 \)
\( m = 2,500,000 \)

\[ \text{FPR} = 1.28\% \]
Comparison with Hash Tables - Space

- Google storing 5 million URLs, each URL 40 bytes.
- Bloom filter with $k = 30$ and $m = 2,500,000$

**Hash Table**

(optimistic) $5,000,000 \times 40B = 200$MB

**Bloom Filter**

$2,500,000 \times 30 = 75,000,000$ bits

$< 10$MB
Time

- Say avg user visits 102,000 URLs in a year, of which 2,000 are malicious.
- 0.5 seconds to do lookup in the database, 1ms for lookup in Bloom filter.
- Suppose the false positive rate is 3%

\[
\begin{align*}
1 \text{ms} &+ 100,000 \times 0.03 \times 500 \text{ms} \\
&+ 2,000 \times 500 \text{ms} \\
&\approx 25.51 \text{ms}
\end{align*}
\]

Bloom filter lookup

false positives

0.5 seconds DB lookup

malicious URLs

total URLs
Bloom Filters typical of....

... randomized algorithms and randomized data structures.

- Simple
- Fast
- Efficient
- Elegant
- Useful!
Brain Break
Motivation for “Named” Random Variables

Random Variables that show up all over the place.
  – Easily solve a problem by recognizing it’s a special case of one of these random variables.

Each RV introduced today will show:
  – A general situation it models
  – Its name and parameters
  – Its PMF, Expectation, and Variance
  – Example scenarios you can use it
Agenda

• Bloom Filters Example & Analysis
• Zoo of Discrete RVs, Part I
  – Uniform Random Variables
  – Bernoulli Random Variables
  – Binomial Random Variables
  – Applications
Discrete Uniform Random Variables

A discrete random variable $X$ equally likely to take any (integer) value between integers $a$ and $b$ (inclusive), is uniform.

Notation:

PMF:

Expectation:

Variance:

Example: value shown on one roll of a fair die
Discrete Uniform Random Variables

A discrete random variable $X$ equally likely to take any (integer) value between integers $a$ and $b$ (inclusive), is uniform.

**Notation:** $X \sim \text{Unif}(a, b)$

**PMF:** $P(X = i) = \frac{1}{b - a + 1}$

**Expectation:** $\mathbb{E}[X] = \frac{a + b}{2}$

**Variance:** $\text{Var}(X) = \frac{(b - a)(b - a + 2)}{12}$

**Example:** value shown on one roll of a fair die is $\text{Unif}(1, 6)$:
- $P(X = i) = 1/6$
- $\mathbb{E}[X] = 7/2$
- $\text{Var}(X) = 35/12$
Agenda

• Bloom Filters Example & Analysis
• Zoo of Discrete RVs, Part I
  – Uniform Random Variables
  – Bernoulli Random Variables
  – Binomial Random Variables
  – Applications
Bernoulli Random Variables

A random variable $X$ that takes value 1 ("Success") with probability $p$, and 0 ("Failure") otherwise. $X$ is called a Bernoulli random variable.

**Notation:** $X \sim \text{Ber}(p)$

**PMF:** $P(X = 1) = p, \ P(X = 0) = 1 - p$

**Expectation:**

**Variance:**

Poll:

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.</td>
<td>$p$</td>
<td>$p$</td>
</tr>
<tr>
<td>B.</td>
<td>$p$</td>
<td>$1 - p$</td>
</tr>
<tr>
<td>C.</td>
<td>$p$</td>
<td>$p(1 - p)$</td>
</tr>
<tr>
<td>D.</td>
<td>$p$</td>
<td>$p^2$</td>
</tr>
</tbody>
</table>

[pollev.com/rachel312](pollev.com/rachel312)
Bernoulli Random Variables

A random variable $X$ that takes value 1 (“Success”) with probability $p$, and 0 (“Failure”) otherwise. $X$ is called a Bernoulli random variable.

Notation: $X \sim \text{Ber}(p)$

PMF: $P(X = 1) = p$, $P(X = 0) = 1 - p$

Expectation: $\mathbb{E}[X] = p$, Note: $\mathbb{E}[X^2] = p$

Variance: $\text{Var}(X) = \mathbb{E}[X^2] - [\mathbb{E}[X]]^2 = p - p^2 = p(1 - p)$

Examples:
- Coin flip
- Randomly guessing on a MC test question
- A server in a cluster fails
- Any indicator RV
Agenda

• Bloom Filters Example & Analysis
• Zoo of Discrete RVs, Part I
  – Uniform Random Variables
  – Bernoulli Random Variables
  – Binomial Random Variables
  – Applications
Binomial Random Variables

A discrete random variable $X$ that is the number of successes in $n$ independent random variables $Y_i \sim \text{Ber}(p)$. $X$ is a Binomial random variable where $X = \sum_{i=1}^{n} Y_i$

Examples:
- # of heads in $n$ coin flips
- # of 1s in a randomly generated $n$ bit string
- # of servers that fail in a cluster of $n$ computers
- # of bit errors in file written to disk
- # of elements in a bucket of a large hash table

Poll:

pollev.com/rachel312

$P(X = k)$

A. $p^k (1 - p)^{n-k}$
B. $np$
C. $\binom{n}{k} p^k (1 - p)^{n-k}$
D. $\binom{n}{n-k} p^k (1 - p)^{n-k}$
Binomial Random Variables

A discrete random variable $X$ that is the number of successes in $n$ independent random variables $Y_i \sim \text{Ber}(p)$. $X$ is a Binomial random variable where $X = \sum_{i=1}^{n} Y_i$

Notation: $X \sim \text{Bin}(n, p)$

PMF: $P(X = k) = \binom{n}{k} p^k (1 - p)^{n-k}$

Expectation: $E(X) = np$

Variance: $\text{Var}(X) = np(1-p)$

Poll:

pollev.com/Rachel312

<table>
<thead>
<tr>
<th>Mean</th>
<th>Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>A. (p)</td>
<td>(p)</td>
</tr>
<tr>
<td>B. (np)</td>
<td>(np(1 - p))</td>
</tr>
<tr>
<td>C. (np)</td>
<td>(np^2)</td>
</tr>
<tr>
<td>D. (np)</td>
<td>(n^2p)</td>
</tr>
</tbody>
</table>
Binomial Random Variables

A discrete random variable $X$ that is the number of successes in $n$ independent random variables $Y_i \sim \text{Ber}(p)$. $X$ is a Binomial random variable where $X = \sum_{i=1}^{n} Y_i$

**Notation:** $X \sim \text{Bin}(n, p)$

**PMF:** $P(X = k) = \binom{n}{k} p^k (1 - p)^{n-k}$

**Expectation:** $\mathbb{E}[X] = np$

**Variance:** $\text{Var}(X) = np(1 - p)$
Mean, Variance of the Binomial

“i.i.d.” is a commonly used phrase.
It means “independent & identically distributed”

If $Y_1, Y_2, \ldots, Y_n \sim \text{Ber}(p)$ and independent (i.i.d.), then

$X = \sum_{i=1}^{n} Y_i$, $X \sim \text{Bin}(n, p)$

Claim $\mathbb{E}[X] = np$

$$\mathbb{E}[X] = \mathbb{E} \left[ \sum_{i=1}^{n} Y_i \right] = \sum_{i=1}^{n} \mathbb{E}[Y_i] = n\mathbb{E}[Y_1] = np$$

Claim $\text{Var}(X) = np(1 - p)$

$$\text{Var}(X) = \text{Var} \left( \sum_{i=1}^{n} Y_i \right) = \sum_{i=1}^{n} \text{Var}(Y_i) = n\text{Var}(Y_1) = np(1 - p)$$
Binomial PMFs

PMF for $X \sim \text{Bin}(10, 0.5)$

PMF for $X \sim \text{Bin}(10, 0.25)$
Binomial PMFs

PMF for $X \sim \text{Bin}(30, 0.5)$

PMF for $X \sim \text{Bin}(30, 0.1)$
Example

Sending a binary message of length 1024 bits over a network with probability 0.999 of correctly sending each bit in the message without corruption (independent of other bits).

Let $X$ be the number of corrupted bits.

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

Poll:

pollev.com/rachel312

a. 1022.99
b. 1.024
c. 1.02298
d. 1
e. Not enough information to compute
Welcome to the Zoo! (today)

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Formulae</th>
</tr>
</thead>
</table>
| $X \sim \text{Unif}(a, b)$ | \[
\begin{align*}
P(X = k) &= \frac{1}{b - a + 1} \\
\mathbb{E}[X] &= \frac{a + b}{2} \\
\text{Var}(X) &= \frac{(b - a)(b - a + 2)}{12}
\end{align*}
\] |
| $X \sim \text{Ber}(p)$ | \[
\begin{align*}
P(X = 1) &= p, \ P(X = 0) = 1 - p \\
\mathbb{E}[X] &= p \\
\text{Var}(X) &= p(1 - p)
\end{align*}
\] |
| $X \sim \text{Bin}(n, p)$ | \[
\begin{align*}
P(X = k) &= \binom{n}{k} p^k (1 - p)^{n-k} \\
\mathbb{E}[X] &= np \\
\text{Var}(X) &= np(1 - p)
\end{align*}
\] |