Bloom Filter

Bloom Filter Operations

Bloom Filter Operations

B.add(x)

Bloom Filter Operations

B.add(x)

x in B?

Bloom Filter Operations

B.add(x)

x in B?

No deletion

Bloom Filter Operations

B.add(x)

x in B?

No deletion

No key/value pairs

Bloom Filter Operations

B.add(x)

x in B?

No deletion

No key/value pairs

Main Points

Bloom Filter Operations

B.add(x)

x in B?

No deletion

No key/value pairs

Main Points

- Bloom filters are very small, typically smaller than the data stored.

Bloom Filter Operations

B.add(x)

x in B?

No deletion

No key/value pairs

Main Points

- Bloom filters are very small, typically smaller than the data stored
- There is one-sided error. Bloom filters may say yes to x in B? when the answer is no.

Why is small important?

Why is small important?

- A key way to deal with big data is to make it smaller.

Why is small important?

- A key way to deal with big data is to make it smaller.
- Smaller may allow the data to be

Why is small important?

- A key way to deal with big data is to make it smaller.
- Smaller may allow the data to be
  - In cache rather than RAM

Why is small important?

- A key way to deal with big data is to make it smaller.
- Smaller may allow the data to be
  - In cache rather than RAM
  - In RAM rather than Disc

Why is small important?

- A key way to deal with big data is to make it smaller.
- Smaller may allow the data to be
  - In cache rather than RAM
  - In RAM rather than Disc
  - On a local machine rather than a server

Why is small important?

- A key way to deal with big data is to make it smaller.
- Smaller may allow the data to be
  - In cache rather than RAM
  - In RAM rather than Disc
  - On a local machine rather than a server
  - On a mobile Device

When is one-sided error OK?

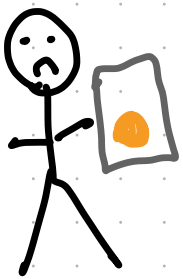
When is one-sided error OK?

Example: A database of Bad URLs, when a user types in one you want a warning with perhaps some info

When is one-sided error OK?

Example: A database of Bad URLs, when a user types in one you want a warning with perhaps some info

Option 1:



Put the DB on each user's device

When is one-sided error OK?

Example: A database of Bad URLs, when a user types in one you want a warning with perhaps some info

Option 1:

Put the DB on each user's device

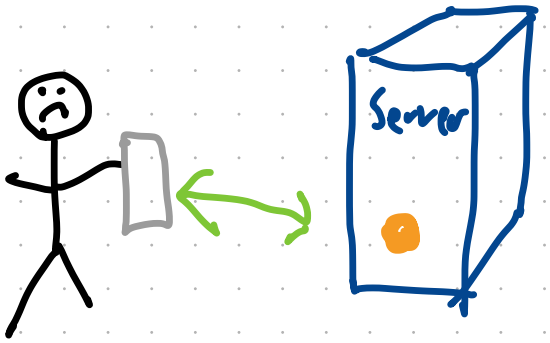
When is one-sided error OK?

Example: A database of Bad URLs, when a user types in one you want a warning with perhaps same info

Option 1:

Put the DB on each user's device

Option 2:



Put the DB on a server

When is one-sided error OK?

Example: A database of Bad URLs, when a user types in one you want a warning with perhaps some info

Option 1:

Put the DB on each user's device

Option 2:

Put the DB on a server

When is one-sided error OK?

Example: A database of Bad URLs, when a user types in one you want a warning with perhaps same info

Option 1:

Put the DB on each user's device

Option 2:

Put the DB on a server

Option 3:

Put a Bloom filter on the phone and a DB on a server.

Bloom Filters, Take 1

Bloom Filters, Take 1

- A Bloom filter is a Bit Array, call it A

Bloom Filters, Take 1

- A Bloom filter is a Bit Array, call it A
- Let k denote the size of the Bitarray
  (usually $k = cn$, for a small c, e.g. 10)

Bloom Filters, Take 1

- A Bloom filter is a Bit Array, call it A
- Let k denote the size of the Bitarray
  (usually $k = cn$, for a small c, e.g. 10)
- To insert x, set $A[hash(x) \% k] = 1$

Bloom Filters, Take 1

- A Bloom filter is a Bit Array, call it A
- Let K denote the size of the Bitarray
  (usually $K = cn$, for a small c, e.g. 10)
- To insert x, set $A[hash(x) \% K] = 1$
- To check if x is in the filter
  we ask is $A[hash(x) \% K] == 1$?

Bloom Filters, Take 1

- A Bloom filter is a Bit Array, call it A
- Let k denote the size of the Bitarray
  (usually $k = cn$, for a small c, e.g. 10)
- To insert x, set $A[hash(x) \% k] = 1$
- To check if x is in the filter
  we ask is $A[hash(x) \% k] == 1$?

Chance of a false positive is high!

Bloom Filters, Take 1

- A Bloom filter is a Bit Array, call it A
- Let K denote the size of the Bitarray
  (usually $K = cn$, for a small c, e.g. 10)
- To insert x, set $A[hash(x) \% K] = 1$
- To check if x is in the filter
  we ask is $A[hash(x) \% K] == 1$?

Chance of a false positive is high!

Can we improve?

Bloom Filters, For Real

- A Bloom filter is a Bit Array, call it A
- Let k denote the size of the Bitarray
  (usually $k = cn$, for a small c, e.g. 10)
- Let h denote the number of hash functions
- To insert x, set $A[hash_i(x) \bmod k] = 1$
  for all i
- To check if x is in the filter
  we ask is $A[hash_i(x) \bmod k] == 1$?
  for all i?

Code

Analysis:

Analysis:

- What is the false positive rate given k, n, h?

Analysis:

- What is the false positive rate given k, n, h?

- How should we choose the right number of hash function

Analysis:

- What is the false positive rate given k, n, h?
- How should we choose the right number of hash function
- What is the traveaff between error rate and k (size)?

False Positive Rate

False Positive Rate

- N items inserted, each sets h random bits to one

False Positive Rate

- N items inserted, each sets h random bits to one
- What is the chance that if we look at h random bits, they are all one?

False Positive Rate

- N items inserted, each sets h random bits to one
- What is the chance that if we look at h random bits, they are all one?
- What is the chance that if we look at a single random bit it is zero

False Positive Rate

- N items inserted, each sets h random bits to one
- What is the chance that if we look at h random bits, they are all one?
- What is the chance that if we look at a single random bit it is zero

P[single bit zero] =

False Positive Rate

- N items inserted, each sets h random bits to one
- What is the chance that if we look at h random bits, they are all one?
- What is the chance that if we look at a single random bit it is zero

$$P[\text{single bit zero}] = \left(1 - \frac{1}{k}\right)^{hn}$$

False Positive Rate

- N items inserted, each sets h random bits to one
- What is the chance that if we look at h random bits, they are all one?
- What is the chance that if we look at a single random bit it is zero

$$P[\text{single Bit zero}] = \left(1-\frac{1}{K}\right)^{hn} \approx \boxed{\left(1-\frac{1}{K}\right)^{K\cdot\frac{hn}{K}}}$$

False Positive Rate

- N items inserted, each sets h random bits to one
- What is the chance that if we look at h random bits, they are all one?
- What is the chance that if we look at a single random bit it is zero

$$P[\text{single bit zero}] = \left(1-\frac{1}{K}\right)^{hn} = \underbrace{\left(1-\frac{1}{K}\right)^{K}}_{\approx e}{}^{\frac{hn}{K}} \approx e^{\frac{hn}{K}}$$

False Positive Rate

- N items inserted, each sets h random bits to one
- What is the chance that if we look at h random bits, they are all one?
- What is the chance that if we look at a single random bit it is zero

$$P[\text{single bit zero}] = \left(1-\frac{1}{K}\right)^{hn} \approx \underbrace{\left(1-\frac{1}{K}\right)^{K}}_{\approx e}{}^{\frac{hn}{K}} \approx e^{\frac{hn}{K}}$$

$$P[\underset{\text{(false positive)}}{\overset{\text{Random}}{h \text{ Bits one}}}]$$

False Positive Rate

- N items inserted, each sets h random bits to one
- What is the chance that if we look at h random bits, they are all one?
- What is the chance that if we look at a single random bit it is zero

$$P[\text{single bit zero}] = \left(1-\frac{1}{K}\right)^{hn} = \left[\left(1-\frac{1}{K}\right)^{K}\right]^{\frac{hn}{K}} \approx e^{\frac{hn}{K}}$$

$$\left(1-\frac{1}{K}\right)^{K} \approx e$$

$$P[\underset{\text{(false positive)}}{\text{h Random Bits one}}] \approx \left(1-e^{hn/K}\right)^{h}$$

False Positive Rate

- N items inserted, each sets h random bits to one
- What is the chance that if we look at h random bits, they are all one?
- What is the chance that if we look at a single random bit it is zero

$$P[\text{single bit zero}] = \left(1-\frac{1}{K}\right)^{hn} = \left[\left(1-\frac{1}{K}\right)^{K}\right]^{\frac{hn}{K}} \approx e^{\frac{hn}{K}}$$

($\left(1-\frac{1}{K}\right)^{K} \approx e$)

$$P[\text{h random bits one}] \approx \left(1-e^{hn/K}\right)^{h}$$

(false positive)

This is minimized when $h = \frac{K}{n} \ln 2$

False Positive Rate

$$P[\text{single Bit zero}] = \left(1-\frac{1}{k}\right)^{hn} \approx \boxed{\left(1-\frac{1}{k}\right)^{k}}^{\frac{hn}{k}} \approx e^{\frac{hn}{k}}$$

$\approx e$

$$P[\underset{\text{(false positive)}}{h \text{ Random Bits one}}] \approx \left(1-e^{hn/k}\right)^{h}$$

This is minimized when $h = \frac{k}{n}\ln 2$

$$P[k \text{ Random Bits one}] \approx$$

with optimal h
(false positive)

False Positive Rate

$$P[\text{single Bit zero}] = \left(1-\frac{1}{k}\right)^{hn} = \left[\left(1-\frac{1}{k}\right)^{k}\right]^{\frac{hn}{k}} \approx e^{\frac{hn}{k}}$$

$\left(1-\frac{1}{k}\right)^{k} \approx e$

$$P[\underset{\text{(false positive)}}{h \overset{\text{Random}}{\text{Bits}} \text{ one}}] \approx \left(1-e^{hn/k}\right)^{h}$$

This is minimized when $h = \frac{k}{n}\ln 2$

$$P[k \overset{\text{Random}}{\text{Bits}} \text{ one}] \approx \left(1-e^{hn/k}\right)^{h}$$

with optimal h
(false positive)

False Positive Rate

$$P[\text{single Bit zero}] = \left(1-\frac{1}{k}\right)^{hn} = \boxed{\left(1-\frac{1}{k}\right)^{k}}^{\frac{hn}{k}} \approx e^{\frac{hn}{k}}$$

$\approx e$

$$P[\underset{\text{(false positive)}}{h \text{ Random Bits one}}] \approx \left(1-e^{hn/k}\right)^{h}$$

This is minimized when $h = \frac{k}{n}\ln 2$

$$P[h \text{ Random Bits one}] \approx \left(1-e^{hn/k}\right)^{h} = \left(1-e^{\left(\frac{k}{n}\ln 2\right)\frac{n}{k}}\right)^{\frac{k}{n}\ln 2}$$

with optimal h
(false positive)

False Positive Rate

$$P[\text{single Bit zero}] = \left(1-\frac{1}{k}\right)^{hn} = \underbrace{\left(1-\frac{1}{k}\right)^{k}}_{\approx e}{}^{\frac{hn}{k}} \approx e^{\frac{hn}{k}}$$

$$P[\underset{\text{(false positive)}}{h \text{ Random Bits one}}] \approx \left(1-e^{hn/k}\right)^{h}$$

This is minimized when $h = \frac{k}{n}\ln 2$

$$P[h \text{ Random Bits one}] \approx \left(1-e^{hn/k}\right)^{h} = \underbrace{\left(1-e^{\left(\frac{k}{n}\ln 2\right)\frac{n}{k}}\right)}_{1/2}{}^{\frac{k}{n}\ln 2} = \left(\frac{1}{2}\right)^{\frac{k}{n}\ln 2}$$

with optimal h (false positive)

False Positive Rate

$$P[\text{single bit zero}] = \left(1-\frac{1}{k}\right)^{hn} = \left[\left(1-\frac{1}{k}\right)^{k}\right]^{\frac{hn}{k}} \approx e^{\frac{hn}{k}}$$

($\left(1-\frac{1}{k}\right)^{k} \approx e$)

$$P[\underset{\text{(false positive)}}{\text{Random } h \text{ bits one}}] \approx \left(1-e^{hn/k}\right)^{h}$$

This is minimized when $h = \frac{k}{n}\ln 2$

$$P[\underset{\text{with optimal } h}{\underset{\text{(false positive)}}{\text{Random } h \text{ bits one}}}] \approx \left(1-e^{hn/k}\right)^{h} = \left(\underbrace{1-e^{\left(\frac{k}{n}\ln 2\right)\frac{n}{k}}}_{1/2}\right)^{\frac{k}{n}\ln 2} = \left(\frac{1}{2}\right)^{\frac{k}{n}\ln 2}$$

Call this ϵ

False Positive Rate

$$P[\text{single bit zero}] = \left(1-\frac{1}{k}\right)^{hn} = \boxed{\left(1-\frac{1}{k}\right)^{k}}^{\frac{hn}{k}} \approx e^{\frac{hn}{k}}$$

$\approx e$

$$P[h \text{ Random Bits one}] \approx \left(1-e^{hn/k}\right)^{h}$$

(false positive)

This is minimized when $h = \frac{k}{n}\ln 2$

$$P[h \text{ Random Bits one}] \approx \left(1-e^{hn/k}\right)^{h} = \underbrace{\left(1-e^{\left(\frac{k}{n}\ln 2\right)\frac{n}{k}}\right)}_{1/2}^{\frac{k}{n}\ln 2} = \left(\frac{1}{2}\right)^{\frac{k}{n}\ln 2}$$

with optimal h

(false positive)

Call this ϵ

$$\epsilon = 2^{-\frac{k}{n}\ln 2}$$

False Positive Rate

$$P[\text{single bit zero}] = \left(1-\frac{1}{k}\right)^{hn} = \boxed{\left(1-\frac{1}{k}\right)^{k}}^{\frac{hn}{k}} \approx e^{\frac{hn}{k}}$$

$\approx e$

$$P[\underset{\text{(false positive)}}{h \text{ Random Bits one}}] \approx \left(1-e^{hn/k}\right)^{h}$$

This is minimized when $h = \frac{k}{n}\ln 2$

$$P[h \text{ Random Bits one}] \approx \left(1-e^{hn/k}\right)^{h} = \left(1-e^{\left(\frac{k}{n}\ln 2\right)\frac{n}{k}}\right)^{\frac{k}{n}\ln 2} = \left(\frac{1}{2}\right)^{\frac{k}{n}\ln 2}$$

$1/2$

with optimal h (false positive)

Call this ϵ

$$\epsilon = 2^{-\frac{k}{n}\ln 2}$$

$$\ln \epsilon = (\ln 2)\left(-\frac{k}{n}\ln 2\right)$$

False Positive Rate

$$P[\text{single bit zero}] = \left(1-\frac{1}{k}\right)^{hn} = \left[\left(1-\frac{1}{k}\right)^{k}\right]^{\frac{hn}{k}} \approx e^{\frac{hn}{k}}$$

($\left(1-\frac{1}{k}\right)^{k} \approx e$)

$$P[\text{h Random Bits one}] \approx \left(1-e^{hn/k}\right)^{h}$$

(false positive)

This is minimized when $h = \frac{k}{n}\ln 2$

$$P[\text{h Random Bits one}] \approx \left(1-e^{hn/k}\right)^{h} = \left(1-e^{\left(\frac{k}{n}\ln 2\right)\frac{n}{k}}\right)^{\frac{k}{n}\ln 2} = \left(\frac{1}{2}\right)^{\frac{k}{n}\ln 2}$$

($1-e^{\left(\frac{k}{n}\ln 2\right)\frac{n}{k}} = 1/2$)

with optimal h (false positive)

Call this ϵ

$$\epsilon = 2^{-\frac{k}{n}\ln 2}$$

$$\ln \epsilon = (\ln 2)\left(-\frac{k}{n}\ln 2\right)$$

$$\frac{k}{n} = \frac{-1}{\ln^2 2}\ln \epsilon$$

Summary

Choose $\frac{k}{n} = -\frac{1}{\ln^2 2} \ln \epsilon$

(ϵ: error rate; $\frac{k}{n}$: bits per item)

Choose $h = \frac{k}{n} \ln 2 = -\frac{1}{\ln 2} \ln \epsilon$

(h: number of hash functions)

Summary

error rate

Choose $\frac{k}{n} = -\frac{1}{\ln^2 2} \ln \epsilon$

Bits per item

Choose $h = \frac{k}{n} \ln 2 = -\frac{1}{\ln 2} \ln \epsilon$

number of hash functions

Examples	k/n	h
$\epsilon = 1\%$	10	7
$\epsilon = 0.1\%$	19	13

URL Example

- 1 million URL's, 70 bytes on average
  - 70 MB of data

URL Example

- 1 million URL's, 70 bytes on average
  - 70 MB of data
- Bloom Filter 1% error $\frac{k}{n} = 10$
  - 1.2 MB of data