Bloom filters

Ch. 8.4

Bloom filters

False positive (filter error) may occur.

Rejection is always accurate.

- When?
  - Returning "Maybe" and "No" as answers is acceptable.
- What?
  - Bit array

0	1	2	3		m-1
0	1	0	0	…	1

  - Uniform and independent hash functions $f_1, f_2, \ldots, f_h$

$$0 \le f_i(k) \le m\text{-}1, \textit{ where } k \text{ is key } \textit{and } i = 1, 2, \ldots, h$$

- Operations:
  - Insert an element into the set
  - Member: Is the element in the set?

Operation: Insertion

Given m bits of memory BF and h hash functions.

	0	1	2	3		m-1
BF	0	0	0	0	…	0

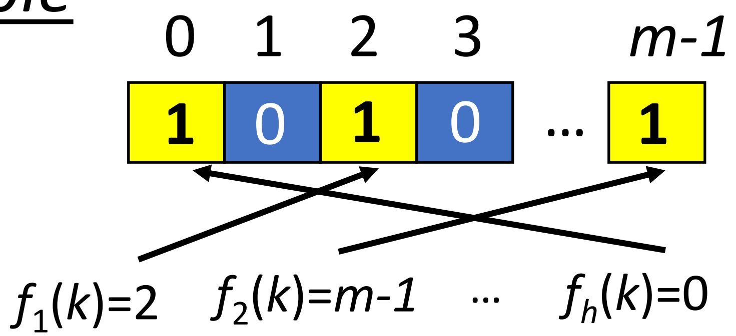
$f_1, f_2, \ldots, f_h$

$0 \leq f_i(k) \leq m\text{-}1$

- Initialize all m bits to be 0.
- To insert key k, set bits $\boxed{f_1(k), f_2(k), \ldots, f_h(k)}$ to be 1.

Signature of key k

Example



Example

Given m = 11 (Normally, very larger m is used.)

h = 2 (2 hash functions)

- $f_1(k) = k \bmod m$
- $f_2(k) = (2k) \bmod m$

0	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0

- Operation 1: Insert k = 15

0	0	0	0	**1**	0	0	0	**1**	0	0

$f_1(15) = 15 \bmod 11 = 4$

$f_2(15) = (2*15) \bmod 11 = 8$

- Operation 2: Insert k = 17

0	**1**	0	0	**1**	0	**1**	0	**1**	0	0

$f_1(17) = 17 \bmod 11 = 6$

$f_2(17) = (2*17) \bmod 11 = 1$

Operation: Member(k, BF)

Search for key k

- Any BF[$f_i(k)$] = 0 → k is not in the set.
- All BF[$f_i(k)$] = 1 → k may be in the set.

Following the previous example:

$f_1(k) = k \bmod m$, $f_2(k) = (2k) \bmod m$, where m=11

0	1	2	3	4	5	6	7	8	9	10
0	1	0	0	1	0	1	0	1	0	0

The 11-bit bloom filter contains the set S={15, 17}.

- Operation 3: Member k = 15

  BF[$f_1(15)$] = 1 and BF[$f_2(15)$] = 1 return "Maybe"

- Operation 4: Member k = 6

  BF[$f_1(6)$] = 1 and BF[$f_2(6)$] = 1 return "Maybe"

  Filter error or False positive

Exercise

Please reply your answers of Q3-Q5 via the following link:

https://forms.gle/rzSHma4tLUURKEzw5

Group members: 2~4 people

Given m = 13 (size of bit array for the bloom filter BF)

h = 3 (number of hash functions)

- $f_1(k) = (3k) \bmod m$
- $f_2(k) = (2k) \bmod m$
- $f_3(k) = k^2 \bmod m$

Q3: Please write out the bit array after inserting 11.

Q4: (Continue of Q3) Please write out the bit array after inserting 1.

Q5: (Continue of Q4) What are the results of Member(3, BF)?

Design of bloom filters

- Choose *m* (filter size in bits)
  - Large *m* to reduce filter error
- Pick *h* (number of hash functions)
  - *h* is too small: Probability of different keys having same signature is high.
  - *h* is too large: The bloom filter fills with ones quickly.
- Select *h* hash functions
  - Hash functions should be relatively independent.

Performance analysis

- Assume that a bloom filter with
  - m bits of memory
  - h uniform hash functions
  - u elements
- Consider the i-th bit of the bloom filter
  - Probability to be selected by the j-th hash function $f_j(k)$:
    $$P[\, f_j(k) = i \,] = 1/m$$
  - Probability of unselected by the j-th hash function $f_j(k)$:
    $$P[\, f_j(k) \neq i \,] = 1 - 1/m$$
  - Probability of unselected by any of h hash functions:
    $$P[\, f_j(\mathrm{k}) \neq i \;\text{ for } j = 1,\ldots,h] = (1 - 1/m)^h$$
  - After inserting u elements, probability of unselected by any of h hash functions:
    $$p = (1 - 1/m)^{hu}$$

Performance analysis

m bits of memory
h uniform hash functions
u elements

- Consider the i-th bit of the bloom filter
  - After inserting u elements, probability that bit i remains 0: $p = (1 - 1/m)^{hu}$
  - After inserting u elements, probability that bit i is 1: $1 - p$

- Probability of false positives:
  - Take a random element k and check Member(k, BF).
  - What is the probability that it returns *true*?

  - Answer: The probability that all h bits $f_1(k), \ldots, f_h(k)$ in BF are 1

$$f = (1 - p)^h$$

Selection of *h*

> *False positive rate:*
> $f = (1 - p)^h$, where $p = (1 - 1/m)^{hu}$

Two competing forces:

- Large *h*:
  - Test more bits for Member(*k*, BF) → Lower false positive rate
  - More bits in BF are 1 → Higher false positive rate
- Small *h*:
  - Test fewer bits for Member(*k*, BF) → Higher false positive rate
  - More bits in BF are 0 → Lower false positive rate

Minimizing false positive rate (1)

- Assume that the filter size m and the number of elements in the filter u are fixed,

h minimizes false positive rate f if $h = (m \ln 2) / u$

Proof:

$x = e^{\ln x}$

$$\min f = \min (1-p)^h$$
$$= \min e^{h \ln (1-p)}$$
$$= \min h \ln (1 - e^{-hu/m})$$
$$= \min h \ln (1 - a^{-h})$$

Using approximation

$p = (1 - 1/m)^{hu} \approx e^{-hu/m}$

Let $a = e^{u/m}$

Differentiating $h \ln (1- a^{-h})$ with respect to h and setting the result to zeros $\dfrac{d\, h \ln(1 - a^{-h})}{d\, h} = 0$

$\ln(1 - a^{-h}) + h \dfrac{a^{-h} \ln a}{1 - a^{-h}} = 0$ $\quad e^{-hu/m} = ½$ $\quad h = \dfrac{m}{u} \ln(2)$

Minimizing false positive rate (2)

- Assume that the filter size m and the number of elements in the filter u are fixed,

h minimizes false positive rate f if $h = (m \ln 2) / u$

Proof:

$x = e^{\ln x}$

$$\min f = \min (1-p)^h$$
$$= \min e^{h \ln (1-p)}$$
$$= \min h \ln (1-p)$$
$$\approx \min -\frac{m}{u}\ln(p)\ln(1-p)$$

Using approximation

$p = (1 - 1/m)^{hu} \approx e^{-hu/m}$

$p = e^{-hu/m} \rightarrow h = -\frac{m}{u}\ln(p)$

When $p = ½$, the value of f is minimum.

$p = e^{-hu/m} = ½ \rightarrow h = -\frac{m}{u}\ln(p) = -\frac{m}{u}\ln\left(\frac{1}{2}\right) = \frac{m}{u}\ln(2)$

Design of bloom filter

Given m bits of memory and u elements

Choose $h = (m \ln 2) / u$ hash functions

- Probability that some bit i is 1

$$\boxed{p \approx e^{-hu/m} = 1/2}$$

- Expected distribution

$$\boxed{m/2 \text{ bits } 1,\ m/2 \text{ bits } 0}$$

- Probability of false positives

$$\boxed{f = (1-p)^h \approx \left(\frac{1}{2}\right)^h = \left(\frac{1}{2}\right)^{(\ln 2)m/u} \approx 0.6185^{m/u}}$$

Limitations of bloom filters

- The naive implementation of the Bloom filter doesn’t support the delete operation.

- The false-positives rate can be reduced but can’t be reduced to zero.

Exercise

- Given a bloom filter with m bits of memory size and storing u elements. We set $m = 8u$.
  - Q6: Please compute the optimum number of hash functions that minimizes the false positive probability f.
  - Q7: (Continue of Q6) Please compute the false positive probability f.

Please reply your answers of Q6-Q7 via the following link:

https://forms.gle/rzSHma4tLUURKEzw5

Group members: 2~4 people

Hint: $f = (1-p)^h$ $\qquad h = \frac{m}{u}\ln(2)$

Summary

- What is bloom filter?
- Operations: Insert, Member
- Performance analysis