Bloom Filters

Anil Maheshwari

School of Computer Science
Carleton University
Canada

Outline

Bloom Filter

Problem Definition

Let U be the universe.

Input: A subset $S \subseteq U$.

Query: For any $q \in U$, decide whether $q \in S$.

Objective

Answer queries quickly and use very little extra space.

SPAM Detection

$U =$ All possible email addresses;

$S =$ My collection of non-junk email addresses.

Query: Given any $q \in U$, report whether $q \in S$?

History of Bloom Filters

- Proposed by Bloom in CACM 1970 - *Space/Time tradeoffs in Hash Coding with Allowable Errors*. (7000 Citations)
- Space-Efficient Probabilistic Data Structure for Membership Testing
- May have false positives
- Numerous Variants: Counting Filters, Dynamic Filters with insertion/deletion of elements in S.
- Vast Applications:Estimating size of union/intersection of sets, Avoid cashing ‘one-hit wonders’, Google Bigtable, Chrome’s uses it to detect malicious URLs.
- Refined Analysis in 2008 by members of our school.

Data Structure

Bloom Filter Data Structure

Data Structure

An array B consisting of m bits and k hash functions $h_1, h_2, \ldots, h_k$, where $h_i : U \to \{1, \ldots, m\}$

Initialization

$B \leftarrow 0$.

For all $x \in S$, set $B[h_1(x)] = B[h_2(x)] = \cdots = B[h_k(x)] = 1$.

Illustration: $k = 2$, $m = 12$, $S = \{a, b, c, d\}$

Queries

Queries

Answering Query

For any query $q \in U$,
if $B[h_1(q)] = B[h_2(q)] = \cdots = B[h_k(q)] = 1$, report $q \in S$,
else report $q \notin S$.

Observation

If $q \in S$, the queries are answered correctly.

False Positives

Suppose $q \notin S$
If $B[h_1(q)] = B[h_2(q)] = \cdots = B[h_k(q)] = 1$,
we will report that $q \in S$.

False-Positives

Estimating Probability of False-Positives

Assume $n = |S|$.

- nk times, we attempt to set locations in B to "1".

- What is the probability that $B[l] = 1$?

- Complementary Event: $Pr(B[l] = 0) = (1 - \frac{1}{m})^{nk}$

- $p = Pr(B[l] = 1) = 1 - (1 - \frac{1}{m})^{nk}$

- For False-Positive to occur, all of the k specified locations $B[h_1(q)], \ldots, B[h_k(q)]$ must be "1".

Bloom70

$Pr(B[h_1(q)] = B[h_2(q)] = \cdots = B[h_k(q)] = 1) = p^k$.

Analysis

An Example

Let $n = 1$, $m = 2$, $k = 2$,
$U = \{x, y\}$, $S = \{x\}$ and $q = y \neq x$.

After Initialization B has the following configuration:

B	Pr. of specific config. of B	Given B, Cond. Pr. that $B[h_1(y)] = B[h_2(y)] = 1$
$\boxed{1}\boxed{0}$	$1/2 \times 1/2 = 1/4$	$1/2 \times 1/2 = 1/4$
$\boxed{0}\boxed{1}$	$1/2 \times 1/2 = 1/4$	$1/2 \times 1/2 = 1/4$
$\boxed{1}\boxed{1}$	$2 \times 1/2 \times 1/2 = 1/2$	$1 \times 1 = 1$

Since the three rows are mutually exclusive, the probability of False-Positive is $1/4 \times 1/4 + 1/4 \times 1/4 + 1/2 \times 1 = 10/16$.

An Example Contd.

$n = 1$, $m = 2$, $k = 2$.

Note that Bloom's result states that the probability of false-positive is p^k, where $p = 1 - (1 - \frac{1}{m})^{kn}$.

From Bloom's computation, $p = 1 - (1 - \frac{1}{m})^{kn} = 1 - (1 - \frac{1}{2})^{2\times 1} = 3/4$, and $p^k = p^2 = 9/16$.

But $9/16 \neq 10/16$.

The implicit assumption that $B[h_2(q)] = 1$ is independent of $B[h_1(q)] = 1$ isn't correct.

A Possible Fix

We came up with a fairly technical proof and showed that

Theorem

Let $p_{k,n,m}$ be the false-positive rate for a Bloom filter that stores n elements of a set S in a bit-vector of size m using k hash functions.

1. We can express $p_{k,n,m}$ in terms of the Stirling number of second kind as follows:

$$p_{k,n,m} = \frac{1}{m^{k(n+1)}} \sum_{i=1}^{m} i^k i! \binom{m}{i} \left\{ {kn \atop i} \right\}$$

2. Let $p = 1 - (1 - 1/m)^{kn}$, $k \geq 2$ and $\frac{k}{p}\sqrt{\frac{\ln m - 2k \ln p}{m}} \leq c$ for some $c < 1$. Upper and lower bounds on $p_{k,n,m}$ are given by

$$p^k < p_{k,n,m} \leq p^k \left(1 + O\left(\frac{k}{p}\sqrt{\frac{\ln m - 2k \ln p}{m}}\right)\right)$$

Summary

Summary of Bloom Filters

1. A simple scheme for testing membership.
   Has one-sided error, i.e., false positives.
   It doesn't store the actual items.
2. How to find the right number of hash functions and right size of the filter?
3. Implemented in various search engines, routers, SPAM filters, . . .
4. Unpleasant analysis in our work
   (Reference: P. Bose, H.Guo, E. Kranakis, A. Maheshwari, P. Morin, J. Morrison, M. Smid, Y. Tang: On the false-positive rate of Bloom filters. Inf. Process. Letters 108(4): 210-213 (2008))
5. Challenge: A nicer analysis. Hopefully, this will help with the analysis of variants of Bloom Filters.

Further Remarks

Following is known, but it is somewhat outside the scope of the course. [1]

1. To minimize the false positives, ideal choice for $k = \frac{|B|}{|S|} \ln 2$.
2. An alternate analysis shows that false-positive error rate is $\leq \left(1 - e^{-\frac{k(|S|+0.5)}{|B|-1}}\right)^k \approx \left(1 - e^{-\frac{k|S|}{|B|}}\right)^k$.
3. False-positive under 1% with optimal number of hash functions uses approximately 10 bits per element of S.
4. Over 60 variants of Bloom Filters.

[1] See Wikipedia entry under Bloom Filters.