Bloom Filters

Anil Maheshwari

anil@scs.carleton.ca
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 $q \in S$ quickly.

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

- Bloom, - *Space/Time tradeoffs in Hash Coding with Allowable Errors*, Communications of ACM 1970
- 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.
- Applications: Estimating size of union/intersection of sets, Avoid cashing ‘one-hit wonders’, Google Bigtable, Chrome’s used 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 $B[h_1(x)] = B[h_2(x)] = \cdots = B[h_k(x)] = 1$.

An Illustration

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

Claim: Let $n = |S|$. After initializing Bloom filter B of size m with k hash-functions for elements of S, $Pr(B[l] = 1) = p = 1 - (1 - \frac{1}{m})^{nk}$, where $l \in \{1, \ldots, m\}$.

Estimating Probability of False-Positives

On query $q \notin S$, 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 $q = y \neq x$.

Independence Assumption?

Implicit assumption that $B[h_2(q)] = 1$ is independent of $B[h_1(q)] = 1$ may not be true . . .

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

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

Summary

Summary of Bloom Filters

1. A simple scheme for testing membership.
   Has one-sided error, i.e., false positives.
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.

A Programming Exercise

A Toy Bloom Filter

Design a Bloom filter B for a small universe U and a subset $S \subseteq U$. Experiment with different sizes of U, S, and k. Evaluate the probability of false positives experimentally and compare with the quantity p^k. (Try to use some library for hash functions.)