Weekplan: Bloom Filters

Philip Bille Inge Li Gørtz Eva Rotenberg

References and Reading

[1] Notes on Bloom filters, section 6.1 and 6.2, Jeff Erickson

[2] Scribe notes from Georgia Tech, except sections 5 and 6.

[3] Notes on Discrete Probability, Section 1.1-1.3, Jeff Erickson

[4] Notes on Hashing, Section 5.1-5.4, Jeff Erickson

[5] Network Applications of Bloom Filters: A Survey. Andrei Broder and Michael Mitzenmacher, Internet Mathematics Vol. 1, No. 4: 485-509

[6] Probability and Computing. Michael Mitzenmacher and Eli Upfal. Cambride University Press

We recommend reading the specified sections and sections of [1] and [2] in detail. If your probability theory is a little rusty we recommend that you read [3]. You can read up on hash functions in [4].

Exercises

1 [w] **Bloom filter** Let $m = 17$, $h_1(x) = (x + 15) \mod m$, $h_2(x) = (4x + 11) \mod m$, and $h_3(x) = (7x + 2) \mod m$. Insert the keys 23, 7, 50, and 91 into the bit vector, and show the resulting vectors content. Then, find a key that is a false positive.

2 Expected number of zeroes Let p be the probability that a specific bit b is zero after inserting n elements in the Bloom filter (we calculated the value of p in class). Let Z be the number of bits in the Bloom filter that are zero after inserting n elements. Show that the expected value of Z is mp.

3 Union and intersection Suppose you have two Bloom filters F_A and F_B representing two sets A and B. The two bloom filters are created using the same number of bits m and the same k hash functions.

3.1 Let F_{OR} be a new Bloom filter formed by computing the bitwise OR of F_A and F_B. Is this the same as the Bloom filter constructed by adding the elements of $A \cup B$ one at a time?

3.2 Let F_{AND} be the Bloom filter formed by computing the bitwise AND of F_A and F_B. Argue that this is not the same as the Bloom filter constructed by adding the elements of $A \cap B$ one at a time.

3.3 Argue that F_{AND} can be used to check if an element x is in the set $A \cap B$ with one-sided error. That is, give an algorithm that always returns TRUE if $x \in A \cap B$, and explain how we can get false-positives.

4 Dynamic size Bloom filters can easily be halved in size, allowing an application to dynamically shrink a Bloom filter. Suppose that the size of the filter is a power of 2. To halve the size of the filter, just OR the first and second halves together. Explain how to to a lookup in the new table.

5 Set differences (from [6]) Bloom filters can be used to estimate set differences. Suppose you have two sets X and Y representing two peoples 100 favorite songs. Let F_X and F_Y be the Bloom filters of the two sets created using the same number of bits m and the same k hash functions.

5.1 What is the probability that a given bit b is 1 in F_X and 0 in F_Y?

5.2 Determine the expected number of bits where the two Bloom filters differ as a function of m, k, and $|X \cap Y|$.

5.3 Explain how this could be used as a tool to find people with the same taste in music more easily than comparing the lists of songs directly.

6 Deletion (from [6]) Suppose that we want to extend Bloom filters to allow deletions as well as insertions of items into the underlying set. We could modify the Bloom filter to be an array of counters instead of an array of bits. Each time an item is inserted into a Bloom filter, the counters given by the hashes of the item are increased by one. To delete an item, one can simply decrement the counters. To keep space small, the counters should be a fixed length, such as 4 bits.

Explain how errors can arise when using fixed-length counters.