Bloom Filters Project
CS 6505

Zizheng Wu

May 1, 2016

1 Introduction

In this project I implemented a Bloom filter and then analyzed the false positive rate of my implementation.

2 Settings

N denotes the size of the universe. m items are added to the subset S that my Bloom filter is maintaining. My Bloom filter has a table of size n. Let $c = \frac{n}{m}$, $k = c \ln 2$, and k denote the number of hash functions used. The hash functions are $h(x) = (ax + b) \mod n$, where $a, b \in \{0, 1, \cdots, n-1\}$.

Now we have k hash functions $h_1, h_2, \cdots, h_k$ and for each i, $h_i : U \to \{0, 1, \cdots, n-1\}$. First we initialize our table H to all 0's. Then we add m items to the subset S. For each item x:

- For all $i \in \{1, 2, \cdots, k\}$
- Set $H[h_i(x))] = 1$

To query if $y \in S$:

- if for all $i \in \{1, 2, \cdots, k\}$, $H[h(y_i)] = 1$, output YES
- else (if $\exists i$ where $H[h(y_i)] = 0$), output NO

In the experiment, the false positive rate is calculated by checking whether my Bloom filter erroneously says YES for an element $x \notin S$:

$$P(\text{false positive}) = \frac{\text{\# of elements erroneously said to be in subset } S}{\text{total \# of elements not in subset } S}$$

Theoretically, the desired false positive probabability is:

$$P = e^{-\ln^2 2 \frac{n}{m}}$$

3 Result

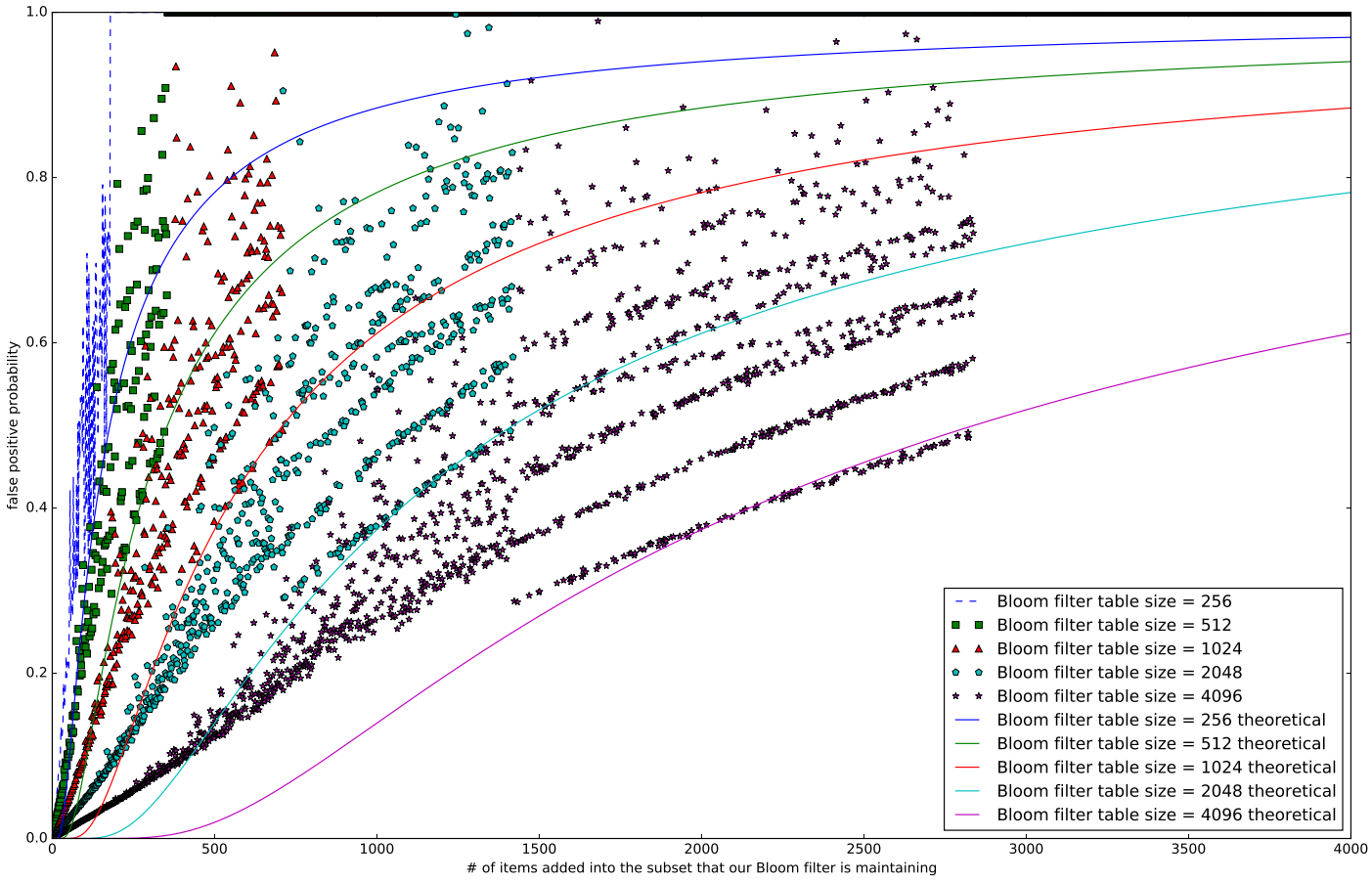

Figure 1: The experimental and theoretical result (the ranges of x are the same $[1, 4000]$ for every size, lots of false positive probability values are squeezed at 1.0)

4 Analysis

Not surprisingly, Bloom filter performs better with larger table size and less number of items inserted into its subset.

Generally speaking, my implementation roughly achieves the theoretical claim. Still, the theoretical claim is noticeably better than my real implementation. I believe the reason behind that is because the theoretical claim assumes independence for the probabilities of each hash function and the bit being set. However, in the real experiment, the hash functions randomly generated are not entirely independent of each other.

5 Conclusion

In this project, I implemented the Bloom filter and then analyzed its performance. The experimental result is in accordance with the theoretical approximation.