Building a Better Bloom Filter

Adam Kirsch
and
Michael Mitzenmacher

TR-02-05

Computer Science Group
Harvard University
Cambridge, Massachusetts

Building a Better Bloom Filter

Adam Kirsch* Michael Mitzenmacher †

Division of Engineering and Applied Sciences
Harvard University
{kirsch,michaelm}@eecs.harvard.edu

Abstract

A technique from the hashing literature is to use two hash functions $h_1(x)$ and $h_2(x)$ to simulate additional hash functions of the form $g_i(x) = h_1(x) + ih_2(x)$. We demonstrate that this technique can be usefully applied to Bloom filters and related data structures. Specifically, only two hash functions are necessary to effectively implement a Bloom filter without any loss in the asymptotic false positive probability. This leads to less computation and potentially less need for randomness in practice.

1 Introduction

A Bloom filter is a simple space-efficient randomized data structure for representing a set in order to support membership queries. Although Bloom filters allow false positives, the space savings often outweigh this drawback. The Bloom filter and its many variations have proven increasingly important for many applications (see, for example, [3]). For those who are not familiar with the data structure, we review it below in Section 2.

In this paper, we show that applying a standard technique from the hashing literature can simplify the implementation of Bloom filters significantly. The idea is the following: two hash functions $h_1(x)$ and $h_2(x)$ can simulate more than two hash functions of the form $g_i(x) = h_1(x) + ih_2(x)$. (See, for example, Knuth's discussion of open addressing with double hashing [10].) In our context i will range from 0 up to some number $k-1$ to give k hash functions, and the hash values are taken modulo the size of the relevant hash table. We demonstrate that this technique can be usefully applied to Bloom filters and related data structures. Specifically, only two hash functions are necessary to effectively implement a Bloom filter without an increase in the asymptotic false positive probability. This leads to less computation and potentially less need for randomness in practice. This improvement was found empirically in the work of Dillinger and Manolios [5, 6]; here we provide a full theoretical analysis of this technique.

After reviewing the Bloom filter data structure, we begin with a specific example, focusing on a useful but somewhat idealized Bloom filter construction that provides the main insights. We then move to a more general setting that covers several issues that might arise in practice, such as when the size of the hash table is a power of two as opposed to a prime. Finally, we demonstrate the utility of this approach beyond the simple Bloom filter by showing how it can be used to reduce the number of hash functions required for the Count-Min sketches of [4].

*Supported in part by an NSF Graduate Research Fellowship and NSF grants CCR-9983832 and CCR-0121154.
†Supported in part by NSF grants CCR-9983832 and CCR-0121154.

2 Standard Bloom filters

We begin by reviewing the fundamentals of Bloom filters, based on the presentation of the survey [3], which we refer to for further details. A Bloom filter for representing a set $S = \{x_1, x_2, \ldots, x_n\}$ of n elements from a large universe U consists of an array of m bits, initially all set to 0. The filter uses k independent hash functions $h_1, \ldots, h_k$ with range $\{1, \ldots, m\}$, where it assumed that these hash functions map each element in the universe to a random number uniform over the range. While the randomness of the hash functions is clearly an optimistic assumption, it appears to be suitable in practice [8, 13]. For each element $x \in S$, the bits $h_i(x)$ are set to 1 for $1 \leq i \leq k$. (A location can be set to 1 multiple times.) To check if an item y is in S, we check whether all $h_i(y)$ are set to 1. If not, then clearly y is not a member of S. If all $h_i(y)$ are set to 1, we assume that y is in S, and hence a Bloom filter may yield a *false positive*.

The probability of a false positive for an element not in the set, or the *false positive probability*, can be estimated in a straightforward fashion, given our assumption that hash functions are perfectly random. After all the elements of S are hashed into the Bloom filter, the probability that a specific bit is still 0 is

$$p' = \left(1 - \frac{1}{m}\right)^{kn} \approx \mathrm{e}^{-kn/m}.$$

In this section, we generally use the approximation $p = \mathrm{e}^{-kn/m}$ in place of p' for convenience.

If ρ is the proportion of 0 bits after all the n elements are inserted in the table, then conditioned on ρ the probability of a false positive is

$$(1 - \rho)^k \approx (1 - p')^k \approx (1 - p)^k = \left(1 - \mathrm{e}^{-kn/m}\right)^k.$$

These approximations follow since $\mathbf{E}[\rho] = p'$, and ρ can be shown to be highly concentrated around p' using standard techniques. It is easy to show that the expression $\left(1 - \mathrm{e}^{-kn/m}\right)^k$ is minimized when $k = \ln 2 \cdot (m/n)$, giving a false positive probability f of

$$f = \left(1 - \mathrm{e}^{-kn/m}\right)^k = (1/2)^k \approx (0.6185)^{m/n}.$$

In practice, k must be an integer, and a smaller, sub-optimal k might be preferred since this reduces the number of hash functions that have to be computed.

This analysis provides us (roughly) with the probability that a single item $z \notin S$ gives a false positive. We would like to make a broader statement, that in fact this gives a false positive *rate*. That is, if we choose a large number of distinct elements not in S, the fraction of them that yield false positives is approximately f. But this result follows immediately from the fact that ρ is highly concentrated around p', and for this reason, the false positive probability is sometimes called the *false positive rate*.

Before moving on, we note that sometimes Bloom filters are described slightly differently, with each hash function having a disjoint range of m/k consecutive bit locations instead of having one shared array of m bits. Repeating the analysis above, we find that in this case the probability that a specific bit is 0 is

$$\left(1 - \frac{k}{m}\right)^n \approx \mathrm{e}^{-kn/m}.$$

Asymptotically the performance is the same as the original scheme, although since for $k \geq 1$,

$$\left(1 - \frac{k}{m}\right)^n \leq \left(1 - \frac{1}{m}\right)^{kn},$$

this modification never decreases the false positive probability.

3 A Simple Construction Using Two Hash Functions

As an instructive example case, we consider the following specific application of the general technique described in the introduction. We devise a Bloom filter that uses k hash functions, each with range $\{0, 1, 2, \ldots, p-1\}$ for a prime p. Our hash table consists of $m = kp$ bits; each hash function is assigned a disjoint subarray of p bits in the filter, that we treat as numbered $\{0, 1, 2, \ldots, p-1\}$. Our k hash functions will be of the form

$$g_i(x) = h_1(x) + ih_2(x) \bmod p,$$

where $h_1(x)$ and $h_2(x)$ are two independent, uniform random hash functions on the universe with range $\{0, 1, 2, \ldots, p-1\}$, and throughout we assume that i ranges from 0 to $k-1$.

In this setting, for any two elements x and y, exactly one of the following three cases occurs:

1. $g_i(x) \neq g_i(y)$ for all i; or
2. $g_i(x) = g_i(y)$ for exactly one i; or
3. $g_i(x) = g_i(y)$ for all i.

That is, if $g_i(x) = g_i(y)$ for at least two values of i, then it is clear that we must have $h_1(x) = h_1(y)$ and $h_2(x) = h_2(y)$, so all hash values are the same. It is this property that implies the analysis and makes this an instructive example; in Section 4, we consider more general cases where other non-trivial collisions can occur.

As a first step, we consider a set $S = \{x_1, x_2, \ldots, x_n\}$ of n elements from U and another element $z \notin S$, and compute the probability that z yields a false positive. A false positive corresponds to the event $\mathcal{F}$ that for each i there is (at least) one j such that $g_i(z) = g_i(x_j)$. Obviously, one way this can occur is if $h_1(x_j) = h_1(z)$ and $h_2(x_j) = h_2(z)$ for some j. The probability of this event $\mathcal{E}$ is

$$\mathbf{Pr}(\mathcal{E}) = 1 - \left(1 - \frac{1}{p^2}\right)^n = 1 - \left(1 - \frac{k^2}{m^2}\right)^n.$$

Notice that when $k = cm/n$ for some constant c, as is standard for a Bloom filter, we have $\mathbf{Pr}(\mathcal{E}) = o(1)$. Now since

$$\begin{aligned}\mathbf{Pr}(\mathcal{F}) &= \mathbf{Pr}(\mathcal{F} \mid \mathcal{E})\,\mathbf{Pr}(\mathcal{E}) + \mathbf{Pr}(\mathcal{F} \mid \neg\mathcal{E})\,\mathbf{Pr}(\neg\mathcal{E}) \\ &= \mathbf{Pr}(\mathcal{E}) + \mathbf{Pr}(\mathcal{F} \mid \neg\mathcal{E})\,\mathbf{Pr}(\neg\mathcal{E}) \\ &= o(1) + \mathbf{Pr}(\mathcal{F} \mid \neg\mathcal{E})(1 - o(1)),\end{aligned}$$

it suffices to consider $\mathbf{Pr}(\mathcal{F} \mid \neg\mathcal{E})$ to obtain the asymptotic false positive probability, which is a constant when m/n and k are constants.

Conditioned on $\neg\mathcal{E}$ and $(h_1(z), h_2(z))$, the pair $(h_1(x_j), h_2(x_j))$ is uniformly distributed over the $p^2 - 1$ values in $V = \{0, \ldots, p-1\}^2 - (h_1(z), h_2(z))$. Of these, for each $i^* \in \{0, \ldots, k-1\}$, the $p-1$ pairs in

$$V' = \{(a, b) \in V \,:\, a \equiv i^*(h_2(z) - b) + h_1(z) \bmod p,\ b \not\equiv h_2(z) \bmod p\}$$

are the ones such that if $(h_1(x_j), h_2(x_j)) \in V'$, then i^* is the unique value of i such that $g_i(x_j) = g_i(z)$. We can therefore view the conditional probability as a variant of a balls-and-bins problem. There are n balls, and k bins. With probability $k(p-1)/(p^2-1) = k/(p+1)$ a ball lands in a bin, and with the remaining probability it is discarded; when a ball lands in a bin, the bin it

lands in is chosen uniformly at random. What is the probability that all of the bins have at least one ball?

This can be expressed in various ways. First, we may recall that the number of surjections from a set of size a to a set of size b is given by $b!S(a,b)$, where $S(a,b)$ refers to the Stirling number of second kind. Then directly we have

$$\mathbf{Pr}(\mathcal{F} \mid \neg\mathcal{E}) = \sum_{a=k}^{n} \binom{n}{a} \left(\frac{k}{p+1}\right)^a \left(1 - \frac{k}{p+1}\right)^{n-a} \frac{k!S(a,k)}{k^a}.$$

One could proceed by taking the limit of this expression as $n \to \infty$ (see, for example, the discussion of [2]).

Alternatively, we may note that for a standard Bloom filter, we have a similar problem. Assuming each of the k hash values for an element $z \notin S$ are distinct (which occurs with high probability), in this case there are nk balls (one for each hash of each item), each with probability k/m of landing in a bin, which corresponds to a hash value for z. It is clear that in the limit as m and n grow large and k is held as a fixed constant, the distribution of the number of balls landing in a bin converges to the same distribution in both cases, and hence the probability of a false positive converges to

$$f = \left(1 - \mathrm{e}^{-kn/m}\right)^k$$

in both cases. As we have stated, a more formal and general argument will be given in Section 4.

Now, as in Section 2, we must argue that f is not only the asymptotic false positive probability, but that it also acts like a false positive *rate*. Similar to the case for the standard Bloom filter, this boils down to a concentration argument. Once the set S is hashed, there is a set

$$B = \{(b_1, b_2) : h_1(z) = b_1 \text{ and } h_2(z) = b_2 \text{ implies } z \text{ gives a false positive}\}.$$

Conditioned on $|B|$, the probability of a false positive for any element in $U - S$ is $|B|/p^2$, and these events are independent. If we show that $|B|$ is concentrated around its expectation, it follows easily that the fraction of false positives in a set of distinct elements not in S is concentrated around f.

A simple Doob martingale argument suffices. Each hashed element of S can change the number of pairs in B by at most kp in either direction. From [12, Section 12.5], it follows that for any $\epsilon > 0$,

$$\mathbf{Pr}(|B - \mathbf{E}[B]| \geq \epsilon p^2) \leq 2\exp\left[\frac{-2\epsilon^2 p^2}{nk^2}\right].$$

It is now easy to derive the desired conclusion. We defer the details until Section 7, where we give a more rigorous proof of a more general result.

4 A General Framework

In this section, we introduce a general framework for analyzing non-standard Bloom filter schemes, such as the one examined in Section 3. We show that under very broad conditions, the asymptotic false positive probability of a scheme is the same as for a standard Bloom filter.

Before delving into details, we must introduce some notation. For any integer ℓ, we define the set $[\ell] = \{0, 1, \ldots, \ell - 1\}$ (note that this definition is slightly non-standard). For a random variable X, we denote the support of X by $\mathrm{Supp}(X)$, and if Y is another random variable, then $X \sim Y$ denotes that X and Y have the same distribution. In addition, we use $\mathrm{Po}(\lambda)$ to denote the Poisson distribution with parameter λ.

We will also need some notation concerning multi-sets. For a multi-set M, we use $|M|$ to denote the number of distinct elements of M, and $\|M\|$ to denote the number of elements of M with multiplicity. For two multi-sets M and M', we define $M \cap M'$ and $M \cup M'$ to be, respectively, the intersection and union of M' as *multi-sets*. Furthermore, in an abuse of standard notation, we define the statement $i, i \in M$ as meaning that i is an element of M of multiplicity at least 2.

We are now ready to define the framework. As in the previous sections, U denotes the universe of items and $S \subseteq U$ denotes the set of n items for which the Bloom filter will answer membership queries. We define a *scheme* to be a method of assigning hash locations to every element of U. More formally, a scheme is specified by a joint distribution of discrete random variables $\{H(u) : u \in U\}$ (implicitly parameterized by n), where for $u \in U$, $H(u)$ represents the multi-set of hash-locations assigned to u by the scheme. We do not require a scheme to be defined for every value of n, but we do insist that it be defined for infinitely many values of n, so that we may take limits as $n \to \infty$. For example, for the class of schemes discussed in Section 3, we think of the constants k and c as being fixed to give a particular scheme, which is only defined for values of n such that $p \stackrel{\text{def}}{=} m/k$ is a prime, where $m \stackrel{\text{def}}{=} cn$. Since there are infinitely many primes, the asymptotic behavior of this scheme as $n \to \infty$ is well-defined and is exactly the same as discussed in Section 3, where we let m be a free parameter and analyzed the behavior as $n, m \to \infty$ subject to m/n and k being fixed constants, and m/k being prime.

Having defined the notion of a scheme, we may now formalize some important concepts with new notation (all of which is implicitly parameterized by n). We define H to be the set of all hash locations that can be assigned by the scheme (formally, H is the set of elements that appear in some multi-set in the support of $H(u)$, for some $u \in U$). For $x \in S$ and $z \in U - S$, define $C(x, z) = H(x) \cap H(z)$ to be the multi-set of hash collisions of x with z. We let $\mathcal{F}(z)$ denote the *false positive event* for $z \in U - S$, which occurs when each of z's hash locations is also a hash location for some $x \in S$.

In the schemes that we consider, $\{H(u) : u \in U\}$ will always be independent and identically distributed. In this case, $\mathbf{Pr}(\mathcal{F}(z))$ is the same for all $z \in U - S$, as is the joint distribution of $\{C(x, z) : x \in S\}$. Thus, to simplify the notation, we may fix an arbitrary $z \in U - S$ and simply use $\mathbf{Pr}(\mathcal{F})$ instead of $\mathbf{Pr}(\mathcal{F}(z))$ to denote the false positive probability, and we may use $\{C(x) : x \in S\}$ instead of $\{C(x, z) : x \in S\}$ to denote the joint probability distribution of the multi-sets of hash collisions of elements of S with z.

The main technical result of this section is the following key theorem, which is a formalization and generalization of the argument given in Section 3 for showing that the asymptotic false positive probability for the scheme analyzed there is the same as for a standard Bloom filter with the same parameters.

Theorem 4.1. *Fix a scheme. Suppose that there are constants λ and k such that:*

1. *$\{H(u) : u \in U\}$ are independent and identically distributed.*
2. *For $u \in U$, $\|H(u)\| = k$.*
3. *For $x \in S$*
$$\mathbf{Pr}(\|C(x)\| = i) = \begin{cases} 1 - \frac{\lambda}{n} + o(1/n) & i = 0 \\ \frac{\lambda}{n} + o(1/n) & i = 1 \\ o(1/n) & i > 1 \end{cases}.$$
4. *For $x \in S$,*
$$\max_{i \in H} \left| \mathbf{Pr}(i \in C(x) \mid \|C(x)\| = 1,\ i \in H(z)) - \frac{1}{k} \right| = o(1) \qquad \text{as } n \to \infty.$$

Then

$$\lim_{n\to\infty} \mathbf{Pr}(\mathcal{F}) = \left(1 - \mathrm{e}^{-\lambda/k}\right)^k.$$

Proof. For ease of exposition, we assign every element of $H(z)$ a unique number in $[k]$ (treating multiple instances of the same hash location as distinct elements). More formally, we define an arbitrary bijection f_M from M to $[k]$ for every multi-set $M \subseteq H$ with $\|M\| = k$ (where f_M treats multiple instances of the same hash location in M as distinct elements), and label the elements of $H(z)$ according to $f_{H(z)}$. This convention allows us to identify the elements of $H(z)$ by numbers $i \in [k]$, rather than hash locations $i \in H$.

For $i \in [k]$ and $x \in S$, define $X_i(x) = 1$ if $i \in C(x)$ and 0 otherwise, and define $X_i \stackrel{\text{def}}{=} \sum_{x\in S} X_i(x)$. Note that $i \in C(x)$ is an abuse of notation; what we really mean is $f^{-1}_{H(z)}(i) \in C(x)$, although we will continue using the former since it is much less cumbersome.

We show that $X^n \stackrel{\text{def}}{=} (X_0, \ldots, X_{k-1})$ converges in distribution to a vector $P \stackrel{\text{def}}{=} (P_0, \ldots, P_{k-1})$ of k independent Poisson random variables with parameter λ/k, as $n \to \infty$. To do this, we make use of moment generating functions. For a random variable R, the moment generating function of R is defined by $M_R(t) \stackrel{\text{def}}{=} \mathbf{E}[\exp(tR)]$. We show that for any $t_0, \ldots, t_k$,

$$\lim_{n\to\infty} M_{\sum_{i=0}^{k-1} t_i X_i}(t_k) = M_{\sum_{i=0}^{k-1} t_i P_i}(t_k),$$

which is sufficient by [1, Theorem 29.4 and p. 390], since

$$\begin{aligned}
M_{\sum_{i=0}^{k-1} t_i P_i}(t_k) &= \mathbf{E}\left[\mathrm{e}^{t_k \sum_{i\in[k]} t_i P_i}\right] \\
&= \prod_{i\in k} \mathbf{E}\left[\mathrm{e}^{t_k t_i \mathrm{Po}(\lambda/k)}\right] \\
&= \prod_{i\in k} \sum_{j=0}^{\infty} \mathrm{e}^{-\lambda/k} \frac{\lambda^j}{k^j j!} \mathrm{e}^{t_k t_i j} \\
&= \prod_{i\in k} \mathrm{e}^{\frac{\lambda}{k}\left(\mathrm{e}^{t_k t_i} - 1\right)} \\
&= \mathrm{e}^{\frac{\lambda}{k}\left(\sum_{i\in k} \mathrm{e}^{t_k t_i} - 1\right)} < \infty,
\end{aligned}$$

where the first step is just the definition of the moment generating function, the second step follows from independence of the $t_i P_i(\lambda_k)$'s, the third step is just the definition of the Poisson distribution, the fourth step follows from the Taylor series for e^x, and the fifth step is obvious.

Proceeding, we write

$$
\begin{aligned}
&M_{\sum_{i\in[k]} t_i X_i}(t_k) \\
&= M_{\sum_{i\in[k]} t_i \sum_{x\in S} X_i(x)}(t_k) \\
&= M_{\sum_{x\in S} \sum_{i\in[k]} t_i X_i(x)}(t_k) \\
&= \left(M_{\sum_{i\in[k]} t_i X_i(x)}(t_k)\right)^n \\
&= \Bigg(\mathbf{Pr}(\|C(x)\| = 0) \\
&\qquad + \sum_{j=1}^{k} \mathbf{Pr}(\|C(x)\| = j) \sum_{T\subseteq[k]:|T|=j} \mathbf{Pr}(C(x) = f_{H(z)}^{-1}(T) \mid \|C(x)\| = j) \mathrm{e}^{t_k \sum_{i\in T} t_i}\Bigg)^n \\
&= \left(1 - \frac{\lambda}{n} + \frac{\lambda}{n} \sum_{i\in[k]} \mathbf{Pr}(i \in C(x) \mid \|C(x)\| = 1) \mathrm{e}^{t_k t_i} + o(1/n)\right)^n \\
&= \left(1 - \frac{\lambda}{n} + \frac{\lambda}{n} \sum_{i\in[k]} \left(\frac{1}{k} + o(1)\right) \mathrm{e}^{t_k t_i} + o(1/n)\right)^n \\
&= \left(1 - \frac{\lambda}{n} + \frac{\lambda \sum_{i\in[k]} \mathrm{e}^{t_k t_i}}{kn} + o(1/n)\right)^n \\
&\to \mathrm{e}^{-\lambda + \frac{\lambda}{k} \sum_{i\in[k]} \mathrm{e}^{t_k t_i}} \qquad \text{as } n \to \infty \\
&= \mathrm{e}^{\frac{\lambda}{k}\left(\sum_{i\in[k]} \left(\mathrm{e}^{t_k t_i} - 1\right)\right)} \\
&= M_{\sum_{i\in[k]} t_i \mathrm{Po}_i(\lambda_k)}(t_k).
\end{aligned}
$$

The first two steps are obvious. The third step follows from the fact that the $H(x)$'s are independent and identically distributed (for $x \in S$) conditioned on $H(z)$, so the $\sum_{i\in[k]} t_i X_i(x)$'s are too, since each is a function of the corresponding $H(x)$. The fourth step follows from the definition of the moment generating function. The fifth and sixth steps follow from the assumptions on the distribution of $C(x)$ (in the sixth step, the conditioning on $i \in H(z)$ is implicit in our convention that associates integers in $[k]$ with the elements of $H(z)$). The seventh, eighth, and ninth steps are obvious, and the tenth step follows from a previous computation.

Now fix some bijection $g : \mathbb{Z}_{\geq 0}^k \to \mathbb{Z}_{\geq 0}$, and define $h : \mathbb{Z}_{\geq 0} \to \{0, 1\} : h(x) = 1$ if and only if every coordinate of $g^{-1}(x)$ is greater than 0. Since $\{X^n\}$ converges to P in distribution, $\{g(X^n)\}$ converges to $g(P)$ in distribution, because g is a bijection and X^n and P have discrete distributions. Skorohod's Representation Theorem [1, Theorem 25.6] now implies that there is some probability space where one may define random variables $\{Y_n\}$ and P', where $Y_n \sim g(X^n)$ and $P' \sim g(P)$, and $\{Y_n\}$ converges to P' almost surely. Of course, since the Y_n's only take integer values, whenever $\{Y_n\}$ converges to P', there must be some n_0 such that $Y_{n_0} = Y_{n_1} = P'$ for any $n_1 > n_0$, and so $\{h(Y_n)\}$ trivially converges to $h(P')$. Therefore, $\{h(Y_n)\}$ converges to

$h(P')$ almost surely, so

$$\begin{aligned}
\mathbf{Pr}(\mathcal{F}) &= \mathbf{Pr}(\forall i \in [k], X_i > 0) \\
&= \mathbf{E}[h(g(X^n))] \\
&= \mathbf{E}[h(Y_n)] \\
&\to \mathbf{E}[h(P')] \qquad \text{as } n \to \infty \\
&= \mathbf{Pr}(\mathrm{Po}(\lambda/k) > 0)^k \\
&= \left(1 - \mathrm{e}^{-\lambda/k}\right)^k,
\end{aligned}$$

where the fourth step is the only nontrivial one, and it follows from [1, Theorem 5.4]. □

It turns out that the conditions of Theorem 4.1 can be verified very easily in many cases.

Lemma 4.1. *Fix a scheme. Suppose that there are constants λ and k such that:*

1. *$\{H(u) : u \in U\}$ are independent and identically distributed.*
2. *For $u \in U$, $\|H(u)\| = k$.*
3. *For $u \in U$,*
$$\max_{i \in H} \left| \mathbf{Pr}(i \in H(u)) - \frac{\lambda}{kn} \right| = o(1/n).$$
4. *For $u \in U$,*
$$\max_{i_1, i_2 \in H} \mathbf{Pr}(i_1, i_2 \in H(u)) = o(1/n).$$
5. *The set of all possible hash locations H satisfies $|H| = O(n)$.*

Then the conditions of Theorem 4.1 hold (with the same value for λ), and so the conclusion does as well.

Remark. Recall that, under our notation, the statement $i, i \in H(u)$ is true if and only if i is an element of $H(u)$ of multiplicity at least 2.

Proof. We adopt the convention introduced in the proof of Theorem 4.1 where the elements of $H(z)$ are identified by the integers in $[k]$.

The first two conditions of Theorem 4.1 are trivially satisfied. For the third condition, observe that for any $j \in \{2, \ldots, k\}$ and $x \in S$,

$$\begin{aligned}
\mathbf{Pr}(\|C(x)\| = j) &\le \mathbf{Pr}(\|C(x)\| > 1) \\
&= \mathbf{Pr}\left(\exists i_1 \le i_2 \in [k] : i_1, i_2 \in H(x) \text{ or } \exists i \in H : i \in H(x), i, i \in H(z)\right) \\
&\le \sum_{i_1 \le i_2 \in [k]} \mathbf{Pr}(i_1, i_2 \in H(x)) + \sum_{i \in H} \mathbf{Pr}(i \in H(x))\, \mathbf{Pr}(i, i \in H(z)) \\
&\le k^2 o(1/n) + |H| \left(\frac{\lambda}{kn} + o(1/n) \right) o(1/n) \\
&= o(1/n) + |H| o(1/n^2) \\
&= o(1/n) + O(n) o(1/n^2) \\
&= o(1/n)
\end{aligned}$$

and

$$\mathbf{Pr}(\|C(x)\| = 1) \leq \mathbf{Pr}(|C(x)| \geq 1) \leq \sum_{i \in [k]} \mathbf{Pr}(i \in H(x)) \leq k \left(\frac{\lambda}{kn} + o(1/n) \right) = \frac{\lambda}{n} + o(1/n),$$

and

$$\begin{aligned}
\mathbf{Pr}(\|C(x)\| \geq 1) &= \mathbf{Pr}\left(\bigcup_{i \in [k]} i \in H(x) \right) \\
&\geq \sum_{i \in [k]} \mathbf{Pr}(i \in H(x)) - \sum_{i_1 < i_2 \in [k]} \mathbf{Pr}(i_1, i_2 \in H(x)) \\
&\geq k \left(\frac{\lambda}{kn} + o(1/n) \right) - k^2 o(1/n) \\
&= \frac{\lambda}{n} + o(1/n),
\end{aligned}$$

so

$$\begin{aligned}
\mathbf{Pr}(\|C(x)\| = 1) &= \mathbf{Pr}(\|C(x)\| \geq 1) - \mathbf{Pr}(\|C(x)\| > 1) \\
&\geq \frac{\lambda}{n} + o(1/n) - o(1/n) \\
&= \frac{\lambda}{n} + o(1/n).
\end{aligned}$$

Therefore,

$$\mathbf{Pr}(\|C(x)\| = 1) = \frac{\lambda}{n} + o(1/n),$$

and

$$\mathbf{Pr}(\|C(x)\| = 0) = 1 - \sum_{j=1}^{k} \mathbf{Pr}(\|C(x)\| = j) = 1 - \frac{\lambda}{n} + o(1/n).$$

We have now shown that the third condition of Theorem 4.1 is satisfied.

For the fourth condition, we observe that for any $i \in [k]$ and $x \in S$,

$$\mathbf{Pr}(i \in C(x), \|C(x)\| = 1) \leq \mathbf{Pr}(i \in H(x)) = \frac{\lambda}{kn} + o(1/n),$$

and

$$\begin{aligned}
\mathbf{Pr}(i \in C(x), \|C(x)\| = 1) &= \mathbf{Pr}(i \in H(x)) - \mathbf{Pr}(i \in H(x), \|C(x)\| > 1) \\
&\geq \mathbf{Pr}(i \in H(x)) - \mathbf{Pr}(\|C(x)\| > 1) \\
&= \frac{\lambda}{kn} + o(1/n) - o(1/n),
\end{aligned}$$

so

$$\mathbf{Pr}(i \in C(x), \|C(x)\| = 1) = \frac{\lambda}{kn} + o(1/n),$$

implying that

$$\mathbf{Pr}(i \in C(x) \mid \|C(x)\| = 1) = \frac{\mathbf{Pr}(i \in C(x), \|C(x)\| = 1)}{\mathbf{Pr}(\|C(x)\| = 1)} = \frac{\frac{\lambda}{kn} + o(1/n)}{\frac{\lambda}{n} + o(1/n)} = \frac{1}{k} + o(1),$$

completing the proof (the conditioning on $i \in H(z)$ is once again implied by the convention that associates elements of $[k]$ with the hash locations in $H(z)$). □

5 Some Specific Schemes

We are now ready to analyze some specific schemes. In particular, we examine a natural generalization of the scheme described in Section 3, as well as the double hashing and extended double hashing schemes introduced in [5, 6].

In both of these cases, we consider a Bloom filter consisting of an array of $m = cn$ bits and k hash functions, where $c > 0$ and $k \geq 1$ are fixed constants. The nature of the hash functions depends on the particular scheme under consideration.

5.1 Partition Schemes

First, we consider the class of *partition schemes*, where the Bloom filter is defined by an array of m bits that is partitioned into k disjoint arrays of $m' = m/k$ bits (we require that m be divisible by k), and an item $u \in U$ is hashed to location

$$h_1(u) + ih_2(u) \bmod m'$$

of array i, for $i \in [k]$, where h_1 and h_2 are independent fully random hash functions with codomain $[m']$. Note that the scheme analyzed in Section 3 is a partition scheme where m' is prime (and so is denoted by p in Section 3) .

Unless otherwise stated, henceforth we do all arithmetic involving h_1 and h_2 modulo m'.

We prove the following theorem concerning partition schemes.

Theorem 5.1. *For a partition scheme,*

$$\lim_{n\to\infty} \mathbf{Pr}(\mathcal{F}) = \left(1 - \mathrm{e}^{-k/c}\right)^k.$$

Proof. We will show that $H(u)$'s satisfy the conditions of Lemma 4.1 with $\lambda = k^2/c$. For $i \in [k]$ and $u \in U$, define

$$\begin{aligned} g_i(u) &= (i, h_1(u) + ih_2(u)) \\ H(u) &= (g_i(u) \,:\, i \in [k]). \end{aligned}$$

That is, $g_i(u)$ is u's ith hash location, and $H(u)$ is the multi-set of u's hash locations. This notation is obviously consistent with the definitions required by Lemma 4.1.

Since h_1 and h_2 are independent and fully random, the first two conditions are trivial. The last condition is also trivial, since there are $m = cn$ possible hash locations. For the remaining two conditions, fix $u \in U$. Observe that for $(i, r) \in [k] \times [m']$,

$$\mathbf{Pr}((i,r) \in H(u)) = \mathbf{Pr}(h_1(u) = r - ih_2(u)) = \frac{1}{m'} = \frac{k^2/c}{kn},$$

and that for distinct $(i_1, r_1), (i_2, r_2) \in [k] \times [m']$, we have

$$\begin{aligned} \mathbf{Pr}((i_1,r_1),(i_2,r_2) \in H(u)) &= \mathbf{Pr}(i_1 \in H(u))\,\mathbf{Pr}(i_2 \in H(u) \mid i_1 \in H(u)) \\ &= \frac{1}{m'}\,\mathbf{Pr}(h_1(u) = r_2 - i_2h_2(u) \mid h_1(u) = r_1 - i_1h_2(u)) \\ &= \frac{1}{m'}\,\mathbf{Pr}((i_1 - i_2)h_2(u) = r_1 - r_2) \\ &\leq \frac{1}{m'} \cdot \frac{\gcd(|i_2 - i_1|, m')}{m'} \\ &\leq \frac{k}{(m')^2} \\ &= o(1/n) \end{aligned}$$

where the fourth step is the only nontrivial step, and it follows from the standard fact that for any $r, s \in [m]$, there are at most $\gcd(r, m)$ values $t \in [m]$ such that $rt \equiv s \bmod m$ (see, for example, [9, Proposition 3.3.1]). Finally, since it is clear that from the definition of the scheme that $|H(u)| = k$ for all $u \in U$, we have that for any $(i, r) \in [k] \times [m']$,

$$\mathbf{Pr}((i, r), (i, r) \in H(u)) = 0.$$

□

5.2 (Extended) Double Hashing Schemes

Next, we consider the class of double hashing and extended double hashing schemes, which are analyzed empirically in [5, 6]. In these schemes, an item $u \in U$ is hashed to location

$$h_1(u) + ih_2(u) + f(i) \bmod m$$

of the array of m bits, for $i \in [k]$, where h_1 and h_2 are independent fully random hash functions with codomain $[m]$, and $f : [k] \to [m]$ is an arbitrary function. When $f(i) \equiv 0$, the scheme is called a *double hashing scheme*. Otherwise, it is called an *extended double hashing scheme (with f)*.

Unless otherwise stated, we do all arithmetic involving h_1 and h_2 modulo m.

We prove the following theorem concerning double hashing schemes.

Theorem 5.2. *For any (extended) double hashing scheme,*

$$\lim_{n\to\infty} \mathbf{Pr}(\mathcal{F}) = \left(1 - \mathrm{e}^{-k/c}\right)^k.$$

Remark. The result holds for any choice of f. In fact, f can even be drawn from an arbitrary probability distribution over $[m]^{[k]}$, so long as it is drawn independently of the two random hash functions h_1 and h_2.

Proof. We proceed by showing that the conditions of Lemma 4.1 are satisfied (for $\lambda = k^2/c$) by this scheme. Since h_1 and h_2 are independent and fully random, the first two conditions trivially hold. The last condition is also trivial, since there are $m = cn$ possible hash locations.

Showing that the third and fourth conditions hold requires more effort. First, we need some notation. For $u \in U$, $i \in [k]$, define

$$g_i(u) = h_1(u) + ih_2(u) + f(i)$$
$$H(u) = (g_i(u) \,:\, i \in [k]).$$

That is, $g_i(u)$ is u's ith hash location, and $H(u)$ is the multi-set of u's hash locations. This notation is obviously consistent with the definitions required by Lemma 4.1. Fix $u \in U$. For $r \in [m]$,

$$\mathbf{Pr}(\exists j \in [k] : g_j(u) = r) \le \sum_{j\in[k]} \mathbf{Pr}(h_1(u) = r - jh_2(u) - f(j)) = \frac{k}{m}.$$

Furthermore, for any $j_1, j_2 \in [k]$ and $r_1, r_2 \in [m]$

$$\begin{aligned}
\mathbf{Pr}(g_{j_1}(u) = r_1, g_{j_2}(u) = r_2) &= \mathbf{Pr}(g_{j_1}(u) = r_1)\,\mathbf{Pr}(g_{j_2}(u) = r_2 \mid g_{j_1}(u) = r_1) \\
&= \frac{1}{m}\,\mathbf{Pr}(g_{j_2}(u) = r_2 \mid g_{j_1}(u) = r_1) \\
&= \frac{1}{m}\,\mathbf{Pr}((j_1 - j_2)h_2(u) = r_1 - r_2 + f(j_2) - f(j_1)) \\
&\leq \frac{1}{m} \cdot \frac{\gcd(|j_1 - j_2|, m)}{m} \\
&\leq \frac{1}{m} \cdot \frac{k}{m} \\
&= \frac{k}{m^2} \\
&= o(1/n),
\end{aligned}$$

where the fourth step is the only nontrivial step, and it follows from the standard fact that for any $r, s \in [m]$, there are at most $\gcd(r, m)$ values $t \in [m]$ such that $rt \equiv s \bmod m$ (see, for example, [9, Proposition 3.3.1]). Therefore, for $r \in [m]$,

$$\begin{aligned}
\mathbf{Pr}(\exists j \in [k] : g_j(u) = r) &\geq \sum_{j \in [k]} \mathbf{Pr}(g_j(u) = r) - \sum_{j_1 < j_2 \in [k]} \mathbf{Pr}(g_{j_1}(u) = r, g_{j_2}(u) = r) \\
&\geq \frac{k}{m} - k^2 o(1/n) \\
&= \frac{k}{m} + o(1/n),
\end{aligned}$$

implying that

$$\mathbf{Pr}(r \in H(u)) = \mathbf{Pr}(\exists j \in [k] : g_j(u) = r) = \frac{k}{m} + o(1/n),$$

so the third condition of Lemma 4.1 holds. For the fourth condition, fix any $r_1, r_2 \in [m]$. Then

$$\mathbf{Pr}(r_1, r_2 \in H(u)) \leq \sum_{j_1, j_2 \in [k]} \mathbf{Pr}(g_{j_1}(u) = r_1, g_{j_2}(u) = r_2) \leq k^2 o(1/n) = o(1/n),$$

completing the proof. □

6 Rate of Convergence

In the previous sections, we identified a broad class of non-standard Bloom filter schemes that have the same asymptotic false positive probability as a standard Bloom filter. Unfortunately, these results are not particularly compelling in settings with very limited space, since it is reasonable to think that the rate of convergence in the conclusion of Theorem 4.1 might be fairly slow. This problem is compounded by the fact that Bloom filters are particularly attractive in applications where space is extremely limited (for example, see [3]), since they give a fairly small error rate while using only a small constant number of bits per item. Thus, with these applications in mind, we provide a detailed analysis of the rate of convergence in Theorem 4.1.

Before proceeding with the results, we introduce some useful notation. For functions $f(n)$ and $g(n)$, we use $f(n) \sim g(n)$ to denote that $\lim_{n\to\infty} f(n)/g(n) = 1$. Similarly, we use $f(n) \lesssim g(n)$ to denote that $\limsup_{n\to\infty} f(n)/g(n) \leq 1$ and $f(n) \gtrsim g(n)$ to denote that $\liminf_{n\to\infty} f(n)/g(n) \geq 1$.

We are now ready to prove the main technical result of this section.

Theorem 6.1. *Under the same conditions as in Theorem 4.1,*

$$\mathbf{Pr}(\mathcal{F}) - \left(1 - \mathrm{e}^{-\lambda/k}\right)^k \sim n\epsilon(n),$$

where

$$\begin{aligned}\epsilon(n) \stackrel{def}{=} &\left(\mathbf{Pr}(\|C(x)\| = 0) - 1 + \frac{\lambda}{n}\right)\left(1 - \mathrm{e}^{-\frac{\lambda}{k}}\right)^k \\ &+ \left(\mathbf{Pr}(\|C(x)\| = 1) - \frac{\lambda}{n}\right)\left(1 - \mathrm{e}^{-\frac{\lambda}{k}}\right)^{k-1} \\ &+ \sum_{j=2}^{k} \mathbf{Pr}(\|C(x)\| = j)\left(1 - \mathrm{e}^{-\frac{\lambda}{k}}\right)^{k-j}.\end{aligned}$$

Remark. This result is intuitively pleasing, since it says that the portion of the false positive probability represented by the asymptotic error term is essentially the probability that $\|C(x,z)\| > 1$ for exactly one $x \in S$ and z's other $k - \|C(x,z)\|$ hash locations are hit by the other elements of S in the "asymptotic" filter (that is, in the limit as $n-1 \to \infty$), which happens with probability $(1 - \mathrm{e}^{-\lambda/k})^{k-\|C(x,z)\|}$. (This *almost* follows from Theorem 4.1. The difference is that now z has only $k - \|C(x,z)\|$ hash locations, while the elements of $S - \{x\}$ each have k hash locations; however, it should be clear from the proof of Theorem 4.1 that the limiting false positive probability in this case is $(1 - \mathrm{e}^{-\lambda/k})^{k-\|C(x,z)\|}$.)

Proof. We begin along the same lines as in the proof of Theorem 4.1. First, we adopt the convention introduced there that allows us to associate the elements of $H(z)$ (with multiplicity) with the elements of $[k]$. Next, for $i \in [k]$ and $x \in S$, we define $X_i(x) = 1$ if $i \in C(x)$ and $X_i(x) = 0$ otherwise, $X_i \stackrel{\text{def}}{=} \sum_{x \in S} X_i(x)$, and $X \stackrel{\text{def}}{=} (X_0, \ldots, X_{k-1})$. Finally, we define $P \stackrel{\text{def}}{=} (P_0, \ldots, P_{k-1})$ to be a vector of k independent $\mathrm{Po}(\lambda/k)$ random variables.

Define

$$\begin{aligned} f(n) &\stackrel{\text{def}}{=} \mathbf{Pr}(\|C(x)\| = 0) - 1 + \frac{\lambda}{n} \\ g_i(n) &\stackrel{\text{def}}{=} \mathbf{Pr}(i \in C(x), \|C(x)\| = 1) - \frac{\lambda}{kn} \qquad \text{for } i \in [k] \\ h_T(n) &\stackrel{\text{def}}{=} \mathbf{Pr}(C(x) = f^{-1}_{H(z)}(T)) \qquad \text{for } T \subseteq [k] : |T| > 1, \end{aligned}$$

and note that they are all $o(1/n)$ by the hypotheses of the lemma. For $T \subseteq [k]$, we may now

write

$$\begin{aligned}
\mathbf{Pr}\left(\bigcap_{i\in T} X_i = 0\right) &= \prod_{x\in S} \mathbf{Pr}\left(\{i\in[k] \,:\, i\in C(x)\}\subseteq \overline{T}\right) \\
&= \left(\mathbf{Pr}(\|C(x)\|=0) + \sum_{i\in\overline{T}} \mathbf{Pr}(i\in C(x), \|C(x)\|=1)\right. \\
&\qquad \left. + \sum_{T'\subseteq\overline{T}:|T'|>1} \mathbf{Pr}(C(x) = f^{-1}_{H(z)}(T'))\right)^n \\
&= \left(1 - \frac{\lambda|T|}{kn} + f(n) + \sum_{i\in\overline{T}} g_i(n) + \sum_{T'\subseteq\overline{T}:|T'|>1} h_{T'}(n)\right)^n \\
&\sim \exp\left[-\frac{\lambda|T|}{k} + nf(n) + \sum_{i\in\overline{T}} ng_i(n) + \sum_{T'\subseteq\overline{T}:|T'|>1} nh_{T'}(n)\right] \\
&= \mathrm{e}^{-\frac{\lambda|T|}{k}}\left(\exp\left[nf(n) + \sum_{i\in\overline{T}} ng_i(n) + \sum_{T'\subseteq\overline{T}:|T'|>1} nh_{T'}(n)\right]\right) \\
&\sim \mathrm{e}^{-\frac{\lambda|T|}{k}}\left(1 + nf(n) + \sum_{i\in\overline{T}} ng_i(n) + \sum_{T'\subseteq\overline{T}:|T'|>1} nh_{T'}(n)\right),
\end{aligned}$$

where the first two steps are obvious, the third step follows from the definition of f, the g_i's, and the $h_{T'}$'s, and the fourth and sixth steps follow from the assumption that all of those functions are $o\left(1/n\right)$ (since $\mathrm{e}^{t(n)} \sim 1 + t(n)$ if $t(n) = o(1)$).

Thus, the inclusion/exclusion principle implies that

$$\begin{aligned}
\mathbf{Pr}(\mathcal{F}) - \mathbf{Pr}(\forall i : P_i > 0) &= -\left(\mathbf{Pr}(\exists i : X_i = 0) - \mathbf{Pr}(\exists i : P_i = 0)\right) \\
&= -\sum_{\emptyset\subset T\subseteq[k]} (-1)^{|T|+1}\left(\mathbf{Pr}\left(\bigcap_{i\in T} X_i = 0\right) - \mathbf{Pr}\left(\bigcap_{i\in T} P_i = 0\right)\right) \\
&= \sum_{\emptyset\subset T\subseteq[k]} (-1)^{|T|}\left(\mathbf{Pr}\left(\bigcap_{i\in T} X_i = 0\right) - \mathrm{e}^{-\frac{\lambda|T|}{k}}\right) \\
&\sim n\sum_{\emptyset\subset T\subseteq[k]} (-1)^{|T|}\mathrm{e}^{-\frac{\lambda|T|}{k}}\left(f(n) + \sum_{i\in\overline{T}} g_i(n) + \sum_{T'\subseteq\overline{T}:|T'|>1} h_{T'}(n)\right).
\end{aligned}$$

To evaluate the sum on the last line, we write

$$\begin{aligned}
M &\stackrel{\text{def}}{=} \sum_{\emptyset \subset T \subseteq [k]} (-1)^{|T|} \mathrm{e}^{-\frac{\lambda |T|}{k}} \left(f(n) + \sum_{i \in \overline{T}} g_i(n) + \sum_{T' \subseteq \overline{T} : |T'| > 1} h_{T'}(n) \right) \\
&= \sum_{j=1}^{k} \left(-\mathrm{e}^{-\frac{\lambda}{k}} \right)^j \sum_{T \subseteq [k] : |T| = j} f(n) \\
&\quad + \sum_{j=1}^{k} \left(-\mathrm{e}^{-\frac{\lambda}{k}} \right)^j \sum_{T \subseteq [k] : |T| = j} \sum_{i \in \overline{T}} g_i(n) \\
&\quad + \sum_{j=1}^{k} \left(-\mathrm{e}^{-\frac{\lambda}{k}} \right)^j \sum_{T \subseteq [k] : |T| = j} \sum_{T' \subseteq \overline{T} : |T'| > 1} h_{T'}(n),
\end{aligned}$$

and evaluate each term separately. First, we compute

$$\begin{aligned}
\sum_{j=1}^{k} \left(-\mathrm{e}^{-\frac{\lambda}{k}} \right)^j \sum_{T \subseteq [k] : |T| = j} f(n) &= f(n) \sum_{j=1}^{k} \binom{k}{j} \left(-\mathrm{e}^{-\frac{\lambda}{k}} \right)^j \\
&= \left(\mathbf{Pr}(\|C(x)\| = 0) - 1 + \frac{\lambda}{n} \right) \left(\left(1 - \mathrm{e}^{-\frac{\lambda}{k}} \right)^k - 1 \right)
\end{aligned}$$

Next, we see that

$$\begin{aligned}
\sum_{j=1}^{k} \left(-\mathrm{e}^{-\frac{\lambda}{k}} \right)^j \sum_{T \subseteq [k] : |T| = j} \sum_{i \in \overline{T}} g_i(n) &= \sum_{j=1}^{k} \left(-\mathrm{e}^{-\frac{\lambda}{k}} \right)^j \sum_{i \in [k]} g_i(n) \left| \{ T \subseteq [k] \,:\, |T| = j, i \notin T \} \right| \\
&= \left(\sum_{i \in [k]} g_i(n) \right) \sum_{j=1}^{k} \binom{k-1}{j} \left(-\mathrm{e}^{-\frac{\lambda}{k}} \right)^j \\
&= \left(\sum_{i \in [k]} g_i(n) \right) \left(\left(1 - \mathrm{e}^{-\frac{\lambda}{k}} \right)^{k-1} - 1 \right) \\
&= \left(\mathbf{Pr}(\|C(x)\| = 1) - \frac{\lambda}{n} \right) \left(\left(1 - \mathrm{e}^{-\frac{\lambda}{k}} \right)^{k-1} - 1 \right),
\end{aligned}$$

where we have used the convention that $\binom{k-1}{k} = 0$. Now, for the last term, we compute

$$\begin{aligned}
\sum_{T \subseteq [k] : |T| = j} \sum_{T' \subseteq \overline{T} : |T'| > 1} h_{T'}(n) &= \sum_{\ell=2}^{k-j} \sum_{T' \subseteq [k] : |T'| = \ell} h_{T'}(n) \left| \left\{ T \subseteq [k] \,:\, |T| = j, T' \subseteq \overline{T} \right\} \right| \\
&= \sum_{\ell=2}^{k-j} \binom{k-\ell}{j} \mathbf{Pr}(\|C(x)\| = \ell),
\end{aligned}$$

so

$$\sum_{j=1}^{k}\left(-\mathrm{e}^{-\frac{\lambda}{k}}\right)^{j}\sum_{T\subseteq[k]:|T|=j}\sum_{T'\subseteq\overline{T}:|T'|>1}h_{T'}(n)=\sum_{j=1}^{k}\left(-\mathrm{e}^{-\frac{\lambda}{k}}\right)^{j}\sum_{\ell=2}^{k-j}\binom{k-\ell}{j}\mathbf{Pr}(\|C(x)\|=\ell)$$
$$=\sum_{j=1}^{k}\sum_{\ell=2}^{k-j}\left(-\mathrm{e}^{-\frac{\lambda}{k}}\right)^{j}\binom{k-\ell}{j}\mathbf{Pr}(\|C(x)\|=\ell)$$
$$=\sum_{j=1}^{k}\sum_{r=j}^{k-2}\left(-\mathrm{e}^{-\frac{\lambda}{k}}\right)^{j}\binom{r}{j}\mathbf{Pr}(\|C(x)\|=k-r)$$
$$=\sum_{r=1}^{k-2}\sum_{j=1}^{r}\left(-\mathrm{e}^{-\frac{\lambda}{k}}\right)^{j}\binom{r}{j}\mathbf{Pr}(\|C(x)\|=k-r)$$
$$=\sum_{r=1}^{k-2}\mathbf{Pr}(\|C(x)\|=k-r)\sum_{j=1}^{r}\binom{r}{j}\left(-\mathrm{e}^{-\frac{\lambda}{k}}\right)^{j}$$
$$=\sum_{r=1}^{k-2}\mathbf{Pr}(\|C(x)\|=k-r)\left(\left(1-\mathrm{e}^{-\frac{\lambda}{k}}\right)^{r}-1\right)$$
$$=\sum_{j=2}^{k-1}\mathbf{Pr}(\|C(x)\|=j)\left(\left(1-\mathrm{e}^{-\frac{\lambda}{k}}\right)^{k-j}-1\right).$$

Adding the terms together gives

$$\begin{aligned}M=&\left(\mathbf{Pr}(\|C(x)\|=0)-1+\frac{\lambda}{n}\right)\left(1-\mathrm{e}^{-\frac{\lambda}{k}}\right)^{k}\\&+\left(\mathbf{Pr}(\|C(x)\|=1)-\frac{\lambda}{n}\right)\left(1-\mathrm{e}^{-\frac{\lambda}{k}}\right)^{k-1}\\&+\sum_{j=2}^{k-1}\mathbf{Pr}(\|C(x)\|=j)\left(1-\mathrm{e}^{-\frac{\lambda}{k}}\right)^{k-j}\\&-\left(\mathbf{Pr}(\|C(x)\|=0)+\mathbf{Pr}(\|C(x)\|=1)+\sum_{j=2}^{k-1}\mathbf{Pr}(\|C(x)\|=j)-1\right).\end{aligned}$$

Of course,

$$-\left(\mathbf{Pr}(\|C(x)\|=0)+\mathbf{Pr}(\|C(x)\|=1)+\sum_{j=2}^{k-1}\mathbf{Pr}(\|C(x)\|=j)-1\right)=\mathbf{Pr}(\|C(x)\|=k),$$

so

$$\begin{aligned}M=&\left(\mathbf{Pr}(\|C(x)\|=0)-1+\frac{\lambda}{n}\right)\left(1-\mathrm{e}^{-\frac{\lambda}{k}}\right)^{k}\\&+\left(\mathbf{Pr}(\|C(x)\|=1)-\frac{\lambda}{n}\right)\left(1-\mathrm{e}^{-\frac{\lambda}{k}}\right)^{k-1}\\&+\sum_{j=2}^{k-1}\mathbf{Pr}(\|C(x)\|=j)\left(1-\mathrm{e}^{-\frac{\lambda}{k}}\right)^{k-j}\\&+\mathbf{Pr}(\|C(x)\|=k)\\=&\,\epsilon(n).\end{aligned}$$

Since

$$\mathbf{Pr}(\mathcal{F}) - \left(1 - \mathrm{e}^{-\lambda/k}\right)^k = \mathbf{Pr}(\mathcal{F}) - \mathbf{Pr}(\forall i : P_i > 0) \sim nM = n\epsilon(n),$$

the result follows. □

Unfortunately, the schemes that we discuss in this paper are often too messy to apply Theorem 6.1 generally; the values $\mathbf{Pr}(\|C(x)\| = j)$ depend on the specifics of the hash functions being use. For example, whether the size of the range is prime or not affects $\mathbf{Pr}(\|C(x)\| = j)$. The result can be applied in cases to examine specific schemes; for example, in the partitioned scheme, when m' is prime, $\mathbf{Pr}(\|C(x)\| = j) = 0$ for $j = 2, \ldots, k-1$, and so the expression becomes easily computable. To achieve general results, we derive some simple bounds that are sufficient to draw some interesting conclusions.

Lemma 6.1. *Assume the same conditions as in Theorem 4.1. Furthermore, suppose that for $x \in S$, it is possible to define events $E_0, \ldots, E_{\ell-1}$ such that*

1. $\mathbf{Pr}(\|C(x)\| \geq 1) = \mathbf{Pr}\left(\bigcup_{i\in[\ell]} E_i\right)$
2. $\sum_{i\in[\ell]} \mathbf{Pr}(E_i) = \lambda/n$
3. $\mathbf{Pr}(\|C(x)\| \geq 2) \leq \sum_{i<j\in[\ell]} \mathbf{Pr}(E_i \cap E_j)$.

then

$$n\left[\mathbf{Pr}(\|C(x)\| = k) - \left(1 - \mathrm{e}^{-\frac{\lambda}{k}}\right)^{k-1}\left(1 + \mathrm{e}^{-\frac{\lambda}{k}}\right)\sum_{i<j\in[\ell]} \mathbf{Pr}(E_i \cap E_j)\right] \lesssim \mathbf{Pr}(\mathcal{F}) - \left(1 - \mathrm{e}^{-\lambda/k}\right)^k \lesssim n\sum_{i<j\in[\ell]} \mathbf{Pr}(E_i \cap E_j)$$

Proof. As in Theorem 6.1, we define

$$\begin{aligned}\epsilon(n) \stackrel{\text{def}}{=} &\left(\mathbf{Pr}(\|C(x)\| = 0) - 1 + \frac{\lambda}{n}\right)\left(1 - \mathrm{e}^{-\frac{\lambda}{k}}\right)^k \\ &+ \left(\mathbf{Pr}(\|C(x)\| = 1) - \frac{\lambda}{n}\right)\left(1 - \mathrm{e}^{-\frac{\lambda}{k}}\right)^{k-1} \\ &+ \sum_{j=2}^{k} \mathbf{Pr}(\|C(x)\| = j)\left(1 - \mathrm{e}^{-\frac{\lambda}{k}}\right)^{k-j},\end{aligned}$$

so that

$$\mathbf{Pr}(\mathcal{F}) - \left(1 - \mathrm{e}^{-\lambda/k}\right)^k \sim n\epsilon(n).$$

Now,

$$\begin{aligned}M &\stackrel{\text{def}}{=} \left(\mathbf{Pr}(\|C(x)\| = 0) - 1 + \frac{\lambda}{n}\right)\left(1 - \mathrm{e}^{-\frac{\lambda}{k}}\right)^k + \left(\mathbf{Pr}(\|C(x)\| = 1) - \frac{\lambda}{n}\right)\left(1 - \mathrm{e}^{-\frac{\lambda}{k}}\right)^{k-1} \\ &= \left(1 - \mathrm{e}^{-\frac{\lambda}{k}}\right)^{k-1}\left(\left(\mathbf{Pr}(\|C(x)\| = 0) - 1 + \frac{\lambda}{n}\right)\left(1 - \mathrm{e}^{-\frac{\lambda}{k}}\right) + \left(\mathbf{Pr}(\|C(x)\| = 1) - \frac{\lambda}{n}\right)\right) \\ &= \left(1 - \mathrm{e}^{-\frac{\lambda}{k}}\right)^{k-1}\left((\mathbf{Pr}(\|C(x)\| = 0) + \mathbf{Pr}(\|C(x)\| = 1) - 1) - \mathrm{e}^{-\frac{\lambda}{k}}\left((\mathbf{Pr}(\|C(x)\| = 0) - 1) + \frac{\lambda}{n}\right)\right) \\ &= \left(1 - \mathrm{e}^{-\frac{\lambda}{k}}\right)^{k-1}\left(-\mathbf{Pr}(\|C(x)\| \geq 2) - \mathrm{e}^{-\frac{\lambda}{k}}\left(-\mathbf{Pr}(\|C(x)\| \geq 2) - \mathbf{Pr}(\|C(x)\| = 1) + \frac{\lambda}{n}\right)\right) \\ &= -\left(1 - \mathrm{e}^{-\frac{\lambda}{k}}\right)^k \mathbf{Pr}(\|C(x)\| \geq 2) + \mathrm{e}^{-\frac{\lambda}{k}}\left(1 - \mathrm{e}^{-\frac{\lambda}{k}}\right)^{k-1}\left(\mathbf{Pr}(\|C(x)\| = 1) - \frac{\lambda}{n}\right).\end{aligned}$$

In particular, we have that $M \leq 0$ since

$$\mathbf{Pr}(\|C(x)\| = 1) \leq \mathbf{Pr}(\|C(x)\| \geq 1) = \mathbf{Pr}\left(\bigcup_{i\in[\ell]} E_i\right) \leq \sum_{i\in[l]} \mathbf{Pr}(E_i) = \lambda/n.$$

Therefore

$$\epsilon(n) = M + \sum_{j=2}^{k} \mathbf{Pr}(\|C(x)\| = j)\left(1 - \mathrm{e}^{-\frac{\lambda}{k}}\right)^{k-j} \leq \mathbf{Pr}(\|C(x)\| \geq 2) \leq \sum_{i<j\in[\ell]} \mathbf{Pr}(E_i \cap E_j),$$

establishing the upper bound in the lemma.

For the lower bound, we note that

$$\begin{aligned}
\mathbf{Pr}(\|C(x)\| = 1) - \frac{\lambda}{n} &= \mathbf{Pr}(\|C(x)\| \geq 1) - \mathbf{Pr}(\|C(x)\| \geq 2) - \frac{\lambda}{n} \\
&= \mathbf{Pr}\left(\bigcup_{i\in[\ell]} E_i\right) - \mathbf{Pr}(\|C(x)\| \geq 2) - \frac{\lambda}{n} \\
&\geq \sum_{i\in[\ell]} \mathbf{Pr}(E_i) - \sum_{i<j\in[\ell]} \mathbf{Pr}(E_i \cap E_j) - \mathbf{Pr}(\|C(x)\| \geq 2) - \frac{\lambda}{n} \\
&= -\sum_{i<j\in[\ell]} \mathbf{Pr}(E_i \cap E_j) - \mathbf{Pr}(\|C(x)\| \geq 2) \\
&\geq -2\sum_{i<j\in[\ell]} \mathbf{Pr}(E_i \cap E_j),
\end{aligned}$$

so

$$\begin{aligned}
M &= -\left(1 - \mathrm{e}^{-\frac{\lambda}{k}}\right)^{k} \mathbf{Pr}(\|C(x)\| \geq 2) + \mathrm{e}^{-\frac{\lambda}{k}}\left(1 - \mathrm{e}^{-\frac{\lambda}{k}}\right)^{k-1}\left(\mathbf{Pr}(\|C(x)\| = 1) - \frac{\lambda}{n}\right) \\
&\geq -\left(1 - \mathrm{e}^{-\frac{\lambda}{k}}\right)^{k} \mathbf{Pr}(\|C(x)\| \geq 2) - \mathrm{e}^{-\frac{\lambda}{k}}\left(1 - \mathrm{e}^{-\frac{\lambda}{k}}\right)^{k-1} 2\sum_{i<j\in[\ell]} \mathbf{Pr}(E_i \cap E_j) \\
&\geq -\left(1 - \mathrm{e}^{-\frac{\lambda}{k}}\right)^{k} \sum_{i<j\in[\ell]} \mathbf{Pr}(E_i \cap E_j) - \mathrm{e}^{-\frac{\lambda}{k}}\left(1 - \mathrm{e}^{-\frac{\lambda}{k}}\right)^{k-1} 2\sum_{i<j\in[\ell]} \mathbf{Pr}(E_i \cap E_j) \\
&= -\left(1 - \mathrm{e}^{-\frac{\lambda}{k}}\right)^{k-1}\left(1 + \mathrm{e}^{-\frac{\lambda}{k}}\right) \sum_{i<j\in[\ell]} \mathbf{Pr}(E_i \cap E_j).
\end{aligned}$$

Therefore,

$$\begin{aligned}
\epsilon(n) &= \sum_{j=2}^{k} \mathbf{Pr}(\|C(x)\| = j)\left(1 - \mathrm{e}^{-\frac{\lambda}{k}}\right)^{k-j} + M \\
&\geq \mathbf{Pr}(\|C(x)\| = k) - \left(1 - \mathrm{e}^{-\frac{\lambda}{k}}\right)^{k-1}\left(1 + \mathrm{e}^{-\frac{\lambda}{k}}\right) \sum_{i<j\in[\ell]} \mathbf{Pr}(E_i \cap E_j),
\end{aligned}$$

completing the proof. □

Lemma 6.1 is easily applied to the schemes discussed in Sections 5.1 and 5.2.

Theorem 6.2. *For the partition scheme discussed in Section 5.1,*

$$\frac{k^2}{c^2 n}\left[1-\left(1-\mathrm{e}^{-\frac{\lambda}{k}}\right)^{k-1}\left(1+\mathrm{e}^{-\frac{\lambda}{k}}\right)\frac{k^3}{2}\right] \lesssim \mathbf{Pr}(\mathcal{F}) - \left(1-\mathrm{e}^{-\lambda/k}\right)^k \lesssim \frac{k^5}{2c^2 n}$$

Proof. We wish to apply Lemma 6.1. To this end, we fix $x \in S$, and for $i \in [k]$, we define E_i to be the event that $i \in C(x)$ (once again, we use the convention introduced in the proof of Theorem 4.1 that allows us to associate the elements of $H(z)$ with the elements of $[k]$). Then

$$\mathbf{Pr}(\|C(x)\| \geq 1) = \mathbf{Pr}\left(\bigcup_{i\in[k]} E_i\right).$$

Recall from the proof of Theorem 5.1 that the partition scheme satisfies the conditions of Theorem 4.1 with $\lambda = k^2/c$. Furthermore, (as we saw in the proof of Theorem 5.1),

$$\sum_{i\in[k]} \mathbf{Pr}(E_i) = \sum_{i\in[k]} \frac{1}{m'} = \frac{\lambda}{n}.$$

The proof of Theorem 5.1 also tells us that for $i \neq j \in [k]$,

$$\mathbf{Pr}(E_i \cap E_j) \leq \frac{k}{(m')^2} = \frac{k^3}{c^2 n^2},$$

so

$$\mathbf{Pr}(\|C(x)\| \geq 2) \leq \sum_{i<j\in[k]} \mathbf{Pr}(E_i \cap E_j) \leq \frac{k^5}{2c^2 n^2},$$

where we have used the (obvious) fact that every $u \in U$ is assigned k *distinct* hash locations in the partition scheme. Finally, we note that $\|C(x)\| = k$ if $h_1(x) = h_1(z)$ and $h_2(x) = h_2(z)$, so

$$\mathbf{Pr}(\|C(x)\| = k) \geq \frac{1}{(m')^2} = \frac{k^2}{c^2 n^2}.$$

Plugging these bounds into the result from Lemma 6.1 gives the result. □

Theorem 6.3. *For the double hashing schemes discussed in Section 5.2,*

$$\frac{1}{c^2 n}\left[1-\left(1-\mathrm{e}^{-\frac{\lambda}{k}}\right)^{k-1}\left(1+\mathrm{e}^{-\frac{\lambda}{k}}\right)\frac{k^5}{2}\right] \lesssim \mathbf{Pr}(\mathcal{F}) - \left(1-\mathrm{e}^{-\lambda/k}\right)^k \lesssim \frac{k^5}{2c^2 n}$$

Proof. We wish to apply Lemma 6.1. First, recall from the proof of Theorem 5.2 that every double hashing scheme satisfies the conditions of Theorem 4.1 with $\lambda = k^2/c$. Now fix $x \in S$. We reintroduce some notation from the proof of Theorem 5.2. For $u \in U$ and $i \in [k]$, we define

$$g_i(u) = h_1(u) + ih_2(u) + f(i)$$

(where we continue to use the convention that all arithmetic involving the hash functions h_1 and h_2 is done modulo m).

Proceeding, for $i, j \in [k]$, we define $E_{i,j}$ to be the event that $g_j(x) = g_i(z)$. Then

$$\mathbf{Pr}(\|C(x)\| \geq 1) = \mathbf{Pr}\left(\bigcup_{i,j\in[k]} E_{i,j}\right),$$

and, as we saw in the proof of Theorem 5.2,

$$\sum_{i,j\in[k]} \mathbf{Pr}(E_{i,j}) = \sum_{i,j\in[k]} \mathbf{Pr}(g_j(x) = g_i(z)) = \sum_{i,j\in[k]} \frac{1}{m} = \frac{\lambda}{n}.$$

Furthermore, fixing any ordering $<$ on $[k]^2$,

$$\begin{aligned}
\mathbf{Pr}(\|C(x)\| \geq 2) &= \mathbf{Pr}(\exists i_1, i_2, j_1, j_2 \in [k] : \forall \ell \in \{1,2\}, g_{j_\ell}(x) = g_{i_\ell}(x)) \\
&= \mathbf{Pr}\left(\bigcup_{(i_1,j_1)<(i_2,j_2)\in[k]^2} E_{i_1,j_1} \cap E_{i_2,j_2} \right) \\
&\leq \sum_{(i_1,j_1)<(i_2,j_2)\in[k]^2} \mathbf{Pr}(E_{i_1,j_1} \cap E_{i_2,j_2}),
\end{aligned}$$

so the conditions of Lemma 6.1 are satisfied. To complete the proof, we note that for any $(i_1, j_1), (i_2, j_2) \in [k^2]$,

$$\begin{aligned}
\mathbf{Pr}(E_{i_1,j_1} \cap E_{i_2,j_2}) &= \mathbf{Pr}(g_{j_1}(x) = g_{i_1}(z), g_{j_2}(x) = g_{i_2}(z)) \\
&\leq \frac{1}{m} \cdot \frac{k}{m} \\
&= \frac{k}{c^2 n^2},
\end{aligned}$$

where the computation in the second step was done in the proof of Theorem 5.2. Therefore,

$$\sum_{(i_1,j_1)<(i_2,j_2)\in[k]^2} \mathbf{Pr}(E_{i_1,j_1} \cap E_{i_2,j_2}) \leq \sum_{(i_1,j_1)<(i_2,j_2)\in[k]^2} \frac{k}{c^2 n^2} \leq \frac{k^5}{2c^2 n^2}.$$

Finally,

$$\mathbf{Pr}(\|C(x)\| = k) \geq \mathbf{Pr}(h_1(x) = h_1(z), h_2(x) = h_2(z)) = \frac{1}{m^2} = \frac{1}{c^2 n^2}.$$

Plugging these bounds into the result of Lemma 6.1 yields the result. □

It remains to investigate whether the error term analyzed in Theorems 6.2 and 6.3 is negligible in practice. Recall that for all of the schemes considered so far, the asymptotic false positive probability is $(1 - \exp[-k/c])^k$, the same as for a standard Bloom filter. We would like to minimize this probability. The easiest way to do this is to maximize c given the application-specific constraints on the size of the filter, and then optimize k subject to that value of c, which results in setting $k = c \ln 2$ (this is a standard result for Bloom filters which is easily obtained using calculus; see, for example, [3]), yielding an asymptotic false positive probability of $2^{-c \ln 2}$. Applying Theorems 6.2 and 6.3, we have that for all of the examined schemes, this setting of k results in

$$\mathbf{Pr}(\mathcal{F}) - 2^{-c \ln 2} \lesssim \frac{(\ln 2)^5}{2} \frac{c^3}{n} \qquad \text{as } n \to \infty.$$

We now give a heuristic argument that the above error term is negligible in practice. Suppose that the asymptotic inequality above held for every n, and not just in the limit as $n \to \infty$. Then

for any $\epsilon > 0$,

$$\begin{aligned}
\mathbf{Pr}(\mathcal{F}) - 2^{-c\ln 2} \geq \epsilon 2^{-c\ln 2} &\Rightarrow \frac{(\ln 2)^5}{2}\frac{c^3}{n} \geq \epsilon 2^{-c\ln 2} \\
&\Rightarrow \frac{(\ln 2)^5}{2}\frac{c^3}{n} \geq \epsilon 2^{-c} \\
&\Rightarrow 2^{c+3\ln c} \geq \frac{2n\epsilon}{(\ln 2)^5} \\
&\Rightarrow 2^{2c+1} \geq \frac{2n\epsilon}{(\ln 2)^5} \\
&\Rightarrow c \geq \frac{1}{2}\log_2\left(\frac{n\epsilon}{(\ln 2)^5}\right).
\end{aligned}$$

The first step is the only non-rigorous step, and it follows from the assumption that the asymptotic inequality above holds for every n. The second step holds since $\ln 2 < 1$, the third step is simple algebra, the fourth step follows from the fact that $3\ln c < c + 1$ for all $c > 0$, and the fifth step is also simple algebra. From this heuristic argument, we conclude that the asymptotic error term analyzed above is negligible unless $c \gtrapprox \log_2 n$. In these cases, however, it might be more appropriate to use a hash table or fingerprints rather than a Bloom filter (see, for example, [12, Section 5.5]).

7 Multiple Queries

In the previous sections, we analyzed the behavior of $\mathbf{Pr}(\mathcal{F}(z))$ for some fixed z and moderately sized n. Unfortunately, this quantity is not directly of interest in most applications. Instead, one is usually concerned with certain characteristics of the distribution of the number of, say, $z_1, \ldots, z_\ell \in U - S$ for which $\mathcal{F}(z)$ occurs. In other words, rather than being interested in the probability that a particular false positive occurs, we are concerned with, for example, the fraction of distinct queries on elements of $U - S$ posed to the filter for which it returns false positives. Since $\{\mathcal{F}(z) : z \in U - S\}$ are not independent, the behavior of $\mathbf{Pr}(\mathcal{F})$ alone does not directly imply results of this form. This section is devoted to overcoming this difficulty.

Now, it is easy to see that in the schemes that we analyze here, once the hash locations for every $x \in S$ have been determined, the events $\{\mathcal{F}(z) : z \in U - S\}$ are independent and occur with equal probability. More formally, letting $\mathbf{1}(\cdot)$ denote the indicator function, $\{\mathbf{1}(\mathcal{F}(z)) : z \in U - S\}$ are conditionally independent and identically distributed given $\{H(x) : x \in S\}$. Thus, conditioned on $\{H(x) : x \in S\}$, an enormous number of classical convergence results (e.g. the law of large numbers and the central limit theorem) can be applied to $\{\mathbf{1}(\mathcal{F}(z)) : z \in U - S\}$.

These observations motivate a general technique for deriving the sort of convergence results for $\{\mathbf{1}(\mathcal{F}(z)) : z \in U - S\}$ that one might desire in practice. First, we show that with high probability over the set of hash locations used by elements of S (that is, $\{H(x) : x \in S\}$), the random variables $\{\mathbf{1}(\mathcal{F}(z)) : z \in U - S\}$ are essentially independent Bernoulli trials with success probability $\lim_{n\to\infty} \mathbf{Pr}(\mathcal{F})$. From a technical standpoint, this result is the most important in this section. Next, we show how to use that result to prove counterparts to the classical convergence theorems mentioned above that hold in our setting.

Proceeding formally, we begin with a critical definition.

Definition 7.1. *Consider any scheme where $\{H(u) : u \in U\}$ are independent and identically distributed. Write $S = \{x_1, \ldots, x_n\}$. The* false positive rate *is defined to be the random variable*

$$R = \mathbf{Pr}(\mathcal{F} \mid H(x_1), \ldots, H(x_n)).$$

The false positive rate gets its name from the fact that, conditioned on R, the random variables $\{\mathbf{1}(\mathcal{F}(z)) : z \in U - S\}$ are independent Bernoulli trials with common success probability R. Thus, the fraction of a large number of queries on elements of $U - S$ posed to the filter for which it returns false positives is very likely to be close to R. In this sense, R, while a random variable, acts like a rate for $\{\mathbf{1}(\mathcal{F}(z)) : z \in U - S\}$.

It is important to note that in much of literature concerning standard Bloom filters, the false positive rate is not defined as above. Instead the term is often used as a synonym for the false positive probability. Indeed, for a standard Bloom filter, the distinction between the two concepts as we have defined them is unimportant in practice, since, as mentioned in Section 2, one can easily show that R is very close to $\mathbf{Pr}(\mathcal{F})$ with extremely high probability (see, for example, [11]). It turns out that this result generalizes very naturally to the framework presented in this paper, and so the practical difference between the two concepts is largely unimportant even in our very general setting. However, the proof is more complicated than in the case of a standard Bloom filter, and so we will be very careful to use the terms as we have defined them.

Theorem 7.1. *Consider a scheme where the conditions of Lemma 4.1 hold. Furthermore, assume that there is some function g and independent identically distributed random variables $\{V_u : u \in U\}$, such that V_u is uniform over $\mathrm{Supp}(V_u)$, and for $u \in U$, we have $H(u) = g(V_u)$. Define*

$$\begin{aligned} p &\stackrel{def}{=} \left(1 - \mathrm{e}^{-\lambda/k}\right)^k \\ \Delta &\stackrel{def}{=} \max_{i \in H} \mathbf{Pr}(i \in H(u)) - \frac{\lambda}{nk} \quad (= o(1/n)) \\ \xi &\stackrel{def}{=} nk\Delta(2\lambda + k\Delta) \quad (= o(1)) \end{aligned}$$

Then for any $\epsilon = \epsilon(n) > 0$ with $\epsilon = \omega(|\,\mathbf{Pr}(\mathcal{F}) - p|)$, for n sufficiently large so that $\epsilon > |\,\mathbf{Pr}(\mathcal{F}) - p|$,

$$\mathbf{Pr}(|R - p| > \epsilon) \leq 2\exp\left[\frac{-2n(\epsilon - |\,\mathbf{Pr}(\mathcal{F}) - p|)^2}{\lambda^2 + \xi}\right].$$

Furthermore, for any function $h(n) = o(\min(1/|\,\mathbf{Pr}(\mathcal{F}) - p|, \sqrt{n}))$, we have that $(R - p)h(n)$ converges to 0 in probability as $n \to \infty$.

Remark. Since $|\,\mathbf{Pr}(\mathcal{F}) - p| = o(1)$ by Lemma 4.1, we may take $h(n) = 1$ in Theorem 7.1 to conclude that R converges to p in probability as $n \to \infty$.

Remark. From the proofs of Theorems 5.1 and 5.2, it is easy to see that for both the partition and (extended) double hashing schemes, $\Delta = 0$, so $\xi = 0$ for both schemes as well.

Remark. We have added a new condition on the distribution of $H(u)$, but it trivially holds in all of the schemes that we discuss in this paper (since, for independent fully random hash functions h_1 and h_2, the random variables $\{(h_1(u), h_2(u)) : u \in U\}$ are independent and identically distributed, and $(h_1(u), h_2(u))$ is uniformly distributed over its support).

Proof. The proof is essentially a standard application of Azuma's inequality to an appropriately defined Doob martingale. Specifically, we employ the technique discussed in [12, Section 12.5].

For convenience, write $S = \{x_1, \ldots, x_n\}$. For $h_1, \ldots, h_n \in \mathrm{Supp}(H(u))$, define

$$f(h_1, \ldots, h_n) \stackrel{\mathrm{def}}{=} \mathbf{Pr}(\mathcal{F} \mid H(x_1) = h_1, \ldots, H(x_n) = h_n),$$

and note that $R = f(H(x_1), \ldots, H(x_n))$. Now consider some c such that for any $h_1, \ldots, h_j$, h'_j, $h_{j+1}, \ldots, h_n \in \mathrm{Supp}(H(u))$,

$$|f(h_1, \ldots, h_n) - f(h_1, \ldots, h_{j-1}, h'_j, h_{j+1}, \ldots, h_n)| \leq c.$$

Since the $H(x_i)$'s are independent, we may apply the result of [12, Section 12.5] to obtain

$$\mathbf{Pr}(|R - \mathbf{E}[R]| \geq \delta) \leq 2\mathrm{e}^{-2\delta^2/nc^2},$$

for any $\delta > 0$.

To find a small choice for c, we write

$$\begin{aligned}
&|f(h_1, \ldots, h_n) - f(h_1, \ldots, h_{j-1}, h'_j, h_{j+1}, \ldots, h_n)| \\
&= |\,\mathbf{Pr}(\mathcal{F} \mid H(x_1) = h_1, \ldots, H(x_n) = h_n) \\
&\qquad - \mathbf{Pr}(\mathcal{F} \mid H(x_1) = h_1, \ldots, H(x_{j-1}) = h_{j-1}, H(x_j) = h'_j, H(x_{j+1}) = h_{j+1}, \ldots H(x_n) = h_n)| \\
&= \frac{\left| |\{v \in \mathrm{Supp}(V_u) \,:\, g(v) \subseteq \bigcup_{i=1}^n h_i\}| - \left| \left\{ v \in \mathrm{Supp}(V_u) \,:\, g(v) \subseteq \bigcup_{i=1}^n \left\{ \begin{array}{ll} h'_j & i = j \\ h_i & i \neq j \end{array} \right\} \right| \right|}{|\mathrm{Supp}(V_u)|} \\
&\leq \frac{\max_{v' \in \mathrm{Supp}(V_u)} |\,\{v \in \mathrm{Supp}(V_u) \,:\, |g(v) \cap g(v')| \geq 1\}\,|}{|\mathrm{Supp}(V_u)|} \\
&= \max_{M' \in \mathrm{Supp}(H(u))} \mathbf{Pr}(|H(u) \cap M'| \geq 1),
\end{aligned}$$

where the first step is just the definition of f, the second step follows from the definitions of V_u and g, the third step holds since changing one of the h_i's to some $M' \in \mathrm{Supp}(H(u))$ cannot change

$$\left| \left\{ v \in \mathrm{Supp}(V_u) \,:\, g(v) \subseteq \bigcup_{i=1}^n h_i \right\} \right|$$

by more than

$$\left| \left\{ v \in \mathrm{Supp}(V_u) \,:\, |g(v) \cap M'| \geq 1 \right\} \right|,$$

and the fourth step follows from the definitions of V_u and g.

Now consider any fixed $M' \in \mathrm{Supp}(H(u))$, and let $y_1, \ldots, y_{|M'|}$ be the distinct elements of M'. Recall that $\|M'\| = k$, so $|M'| \leq k$. Applying a union bound, we have that

$$\begin{aligned}
\mathbf{Pr}(|H(u) \cap M'| \geq 1) &= \mathbf{Pr}\left(\bigcup_{i=1}^{|M'|} y_i \in H(u) \right) \\
&\leq \sum_{i=1}^{|M'|} \mathbf{Pr}(y_i \in H(u)) \\
&\leq \sum_{i=1}^{|M'|} \frac{\lambda}{kn} + \Delta \\
&\leq \frac{\lambda}{n} + k\Delta.
\end{aligned}$$

Therefore, we may set $c = \frac{\lambda}{n} + k\Delta$ to obtain

$$\mathbf{Pr}(|R - \mathbf{E}[R]| > \delta) \leq 2\exp\left[\frac{-2n\delta^2}{\lambda^2 + \xi}\right],$$

for any $\delta > 0$. Since $\mathbf{E}[R] = \mathbf{Pr}(\mathcal{F})$, we write (for sufficiently large n so that $\epsilon > |\,\mathbf{Pr}(\mathcal{F}) - p|$)

$$\begin{aligned}
\mathbf{Pr}(|R - p| > \epsilon) &\leq \mathbf{Pr}(|R - \mathbf{Pr}(\mathcal{F})| > \epsilon - |\,\mathbf{Pr}(\mathcal{F}) - p|) \\
&\leq 2\exp\left[\frac{-2n(\epsilon - |\,\mathbf{Pr}(\mathcal{F}) - p|)^2}{\lambda^2 + \xi}\right].
\end{aligned}$$

To complete the proof, we see that for any constant $\delta > 0$,

$$\mathbf{Pr}(|R-p|h(n) > \delta) = \mathbf{Pr}(|R-p| > \delta/h(n)) \to 0 \quad \text{as } n \to \infty,$$

where the second step follows from the fact that $|\mathbf{Pr}(\mathcal{F}) - p| = o(1/h(n))$, so for sufficiently large n,

$$\begin{aligned}\mathbf{Pr}(|R-p| > \delta/h(n)) &\le 2\exp\left[\frac{-2n(\delta/h(n) - |\mathbf{Pr}(\mathcal{F})-p|)^2}{\lambda^2+\xi}\right] \\ &\le 2\exp\left[-\frac{\delta^2}{\lambda^2+\xi}\cdot\frac{n}{h(n)^2}\right] \\ &\to 0 \quad \text{as } n\to\infty,\end{aligned}$$

and the last step follows from the fact that $h(n) = o(\sqrt{n})$. □

Since, conditioned on R, the events $\{\mathcal{F}(z) : z \in U - S\}$ are independent and each occur with probability R, Theorem 7.1 suggests that $\{\mathbf{1}(\mathcal{F}(z)) : z \in U - S\}$ are essentially independent Bernoulli trials with success probability p. The next result is a formalization of this idea.

Lemma 7.1. *Consider a scheme where the conditions of Theorem 7.1 hold. Let $\mathcal{F}_{n_0}(z)$ denote $\mathcal{F}(z)$ in the case when the scheme is used with $n = n_0$. Similarly, let R_{n_0} denote R in the case where $n = n_0$. Let $\{X_n\}$ be a sequence of real-valued random variables, where each X_n can be expressed as some function of $\{\mathbf{1}(\mathcal{F}_n(z)) : z \in U - S\}$, and let Y be any probability distribution on $\mathbb{R}$. Then for every $x \in \mathbb{R}$ and $\epsilon = \epsilon(n) > 0$ with $\epsilon = \omega(|\mathbf{Pr}(\mathcal{F}) - p|)$, for sufficiently large n so that $\epsilon > |\mathbf{Pr}(\mathcal{F}) - p|$,*

$$\begin{aligned}|\mathbf{Pr}(X_n \le x) - \mathbf{Pr}(Y \le x)| \le{}& |\mathbf{Pr}(X_n \le x \mid |R_n - p| \le \epsilon) - \mathbf{Pr}(Y \le x)| \\ &+ 2\exp\left[\frac{-2n(\epsilon - |\mathbf{Pr}(\mathcal{F})-p|)^2}{\lambda^2+\xi}\right].\end{aligned}$$

Proof. The proof is a straightforward application of Theorem 7.1. Fix any $x \in \mathbb{R}$, and choose some ϵ satisfying the conditions of the lemma. Then

$$\begin{aligned}\mathbf{Pr}(X_n \le x) &= \mathbf{Pr}(X_n \le x, |R_n - p| > \epsilon) + \mathbf{Pr}(X_n \le x, |R_n - p| \le \epsilon) \\ &= \mathbf{Pr}(X_n \le x \mid |R_n - p| \le \epsilon) \\ &\quad + \mathbf{Pr}(|R_n - p| > \epsilon)\left[\mathbf{Pr}(X_n \le x \mid |R_n - p| > \epsilon) - \mathbf{Pr}(X_n \le x \mid |R_n - p| \le \epsilon)\right],\end{aligned}$$

implying that

$$|\mathbf{Pr}(X_n \le x) - \mathbf{Pr}(X_n \le x \mid |R_n - p| \le \epsilon)| \le \mathbf{Pr}(|R_n - p| > \epsilon).$$

Therefore,

$$\begin{aligned}&|\mathbf{Pr}(X_n \le x) - \mathbf{Pr}(Y \le x)| \\ &\le |\mathbf{Pr}(X_n \le x) - \mathbf{Pr}(X_n \le x \mid |R_n - p| \le \epsilon)| + |\mathbf{Pr}(X_n \le x \mid |R_n - p| \le \epsilon) - \mathbf{Pr}(Y_n \le x)| \\ &\le \mathbf{Pr}(|R_n - p| > \epsilon) + |\mathbf{Pr}(X_n \le x \mid |R_n - p| \le \epsilon) - \mathbf{Pr}(Y_n \le x)|,\end{aligned}$$

so for sufficiently large n so that $\epsilon > |\mathbf{Pr}(\mathcal{F}) - p|$,

$$\begin{aligned}|\mathbf{Pr}(X_n \le x) - \mathbf{Pr}(Y \le x)| \le{}& |\mathbf{Pr}(X_n \le x \mid |R_n - p| \le \epsilon) - \mathbf{Pr}(Y \le x)| \\ &+ 2\exp\left[\frac{-2n(\epsilon - |\mathbf{Pr}(\mathcal{F})-p|)^2}{\lambda^2+\xi}\right],\end{aligned}$$

by Theorem 7.1. □

To illustrate the power of Theorem 7.1 and Lemma 7.1, we use them to prove versions of the strong law of large numbers, the weak law of large numbers, Hoeffding's inequality, and the central limit theorem.

Theorem 7.2. *Consider a scheme that satisfies the conditions of Theorem 7.1. Let $Z \subseteq U - S$ be countably infinite, and write $Z = \{z_1, z_2, \ldots\}$. Then for any $\epsilon > 0$, for n sufficiently large so that $\epsilon > |\operatorname{\mathbf{Pr}}(\mathcal{F}) - p|$, we have:*

1. $$\operatorname{\mathbf{Pr}}\left(\lim_{\ell\to\infty} \frac{1}{\ell}\sum_{i=1}^{\ell} \mathbf{1}(\mathcal{F}_n(z_i)) = R_n\right) = 1.$$

2. *For any $\epsilon > 0$, for n sufficiently large so that $\epsilon > |\operatorname{\mathbf{Pr}}(\mathcal{F}) - p|$,*

$$\operatorname{\mathbf{Pr}}\left(\left|\lim_{\ell\to\infty} \frac{1}{\ell}\sum_{i=1}^{\ell} \mathbf{1}(\mathcal{F}_n(z_i)) - p\right| > \epsilon\right) \le 2\exp\left[\frac{-2n(\epsilon - |\operatorname{\mathbf{Pr}}(\mathcal{F}) - p|)^2}{\lambda^2 + \xi}\right].$$

   In particular, $\lim_{\ell\to\infty} \frac{1}{\ell}\sum_{i=1}^{\ell} \mathbf{1}(\mathcal{F}_n(z_i))$ converges to p in probability as $n \to \infty$.

3. *For any function $Q(n)$, $\epsilon > 0$, and n sufficiently large so that $\epsilon/2 > |\operatorname{\mathbf{Pr}}(\mathcal{F}) - p|$,*

$$\operatorname{\mathbf{Pr}}\left(\left|\frac{1}{Q(n)}\sum_{i=1}^{Q(n)} \mathbf{1}(\mathcal{F}_n(z_i)) - p\right| > \epsilon\right) \le 2\mathrm{e}^{-Q(n)\epsilon^2/2} + 2\exp\left[\frac{-2n(\epsilon/2 - |\operatorname{\mathbf{Pr}}(\mathcal{F}) - p|)^2}{\lambda^2 + \xi}\right].$$

4. *For any function $Q(n)$ with $\lim_{n\to\infty} Q(n) = \infty$ and $Q(n) = o(\min(1/|\operatorname{\mathbf{Pr}}(\mathcal{F}) - p|^2, n))$,*

$$\sum_{i=1}^{Q(n)} \frac{\mathbf{1}(\mathcal{F}_n(z_i)) - p}{\sqrt{Q(n)p(1-p)}} \to \mathrm{N}(0,1) \text{ in distribution as } n \to \infty.$$

Remark. By Theorems 6.2 and 6.3, $|\operatorname{\mathbf{Pr}}(\mathcal{F}) - p| = \Theta(1/n)$ for both the partition and double hashing schemes introduced in Section 5. Thus, for each of the schemes, the condition $Q(n) = o(\min(1/|\operatorname{\mathbf{Pr}}(\mathcal{F}) - p|^2, n))$ in the fourth part of Theorem 7.2 becomes $Q(n) = o(n)$.

Proof. Since, given R_n, the random variables $\{\mathbf{1}(\mathcal{F}_n(z)) : z \in Z\}$ are conditionally independent Bernoulli trials with common success probability R_n, a direct application of the strong law of large numbers yields the first item.

For the second item, we note that the first item implies that

$$\lim_{\ell\to\infty} \frac{1}{\ell}\sum_{i=1}^{\ell} \mathbf{1}(\mathcal{F}_n(z_i)) \sim R_n.$$

A direct application of Theorem 7.1 then gives the result.

The remaining two items are slightly more difficult. However, they can be dealt with using straightforward applications of Lemma 7.1.

For the third item, define

$$X_n \stackrel{\text{def}}{=} \left|\frac{1}{Q(n)}\sum_{i=1}^{Q(n)} \mathbf{1}(\mathcal{F}_n(z_i)) - p\right|.$$

and $Y \stackrel{\text{def}}{=} 0$. Let $\delta = \epsilon/2$ to obtain

$$\begin{aligned}
&\mathbf{Pr}(X_n > \epsilon \mid |R_n - p| \le \delta) \\
&= \mathbf{Pr}\left(\left|\sum_{i=1}^{Q(n)} \mathbf{1}(\mathcal{F}_n(z_i)) - Q(n)p\right| > Q(n)\epsilon \;\middle|\; |R_n - p| \le \delta\right) \\
&\le \mathbf{Pr}\left(\left|\sum_{i=1}^{Q(n)} \mathbf{1}(\mathcal{F}_n(z_i)) - Q(n)R_n\right| > Q(n)\left(\epsilon - |R_n - p|\right) \;\middle|\; |R_n - p| \le \delta\right) \\
&\le \mathbf{Pr}\left(\left|\sum_{i=1}^{Q(n)} \mathbf{1}(\mathcal{F}_n(z_i)) - Q(n)R_n\right| > \frac{Q(n)\epsilon}{2} \;\middle|\; |R_n - p| \le \delta\right) \\
&\le 2\mathrm{e}^{-Q(n)\epsilon^2/2},
\end{aligned}$$

where the first two steps are obvious, the third step follows from the fact that $\mathbf{Pr}(\mathcal{F}_n \mid R_n) = R_n$, and the fourth step is an application of Hoeffding's Inequality (using the fact that, given R_n, $\{\mathbf{1}(\mathcal{F}_n(z)) : z \in Z\}$ are independent and identically distributed Bernoulli trials with common success probability R_n).

Now, since $\mathbf{Pr}(Y \le \epsilon) = 1$,

$$|\mathbf{Pr}(X_n \le \epsilon \mid |R_n - p| \le \delta) - \mathbf{Pr}(Y \le \epsilon)| = \mathbf{Pr}(X_n > \epsilon \mid |R_n - p| \le \delta) \le 2\mathrm{e}^{-Q(n)\epsilon^2/2}.$$

An application of Lemma 7.1 now gives the third item.

For the fourth item, we write

$$\sum_{i=1}^{Q(n)} \frac{\mathbf{1}(\mathcal{F}_n(z_i)) - p}{\sqrt{Q(n)p(1-p)}} = \sqrt{\frac{R_n(1-R_n)}{p(1-p)}} \left(\sum_{i=1}^{Q(n)} \frac{\mathbf{1}(\mathcal{F}_n(z_i)) - R_n}{\sqrt{Q(n)R_n(1-R_n)}} + (R_n - p)\sqrt{\frac{Q(n)}{R_n(1-R_n)}}\right)$$

By the central limit theorem,

$$\sum_{i=1}^{Q(n)} \frac{\mathbf{1}(\mathcal{F}_n(z_i)) - R_n}{\sqrt{Q(n)R_n(1-R_n)}} \to \mathrm{N}(0,1) \quad \text{in distribution as } n \to \infty,$$

since, given R_n, $\{\mathbf{1}(\mathcal{F}_n(z)) : z \in Z\}$ are independent and identically distributed Bernoulli trials with common success probability R_n. Furthermore, R_n converges to p in probability as $n \to \infty$ by Theorem 7.1, so it suffices to show that $(R_n - p)\sqrt{Q(n)}$ converges to 0 in probability as $n \to \infty$. But $\sqrt{Q(n)} = o(\min(1/|\mathbf{Pr}(\mathcal{F}) - p|, \sqrt{n}))$, so another application of Theorem 7.1 gives the result. □

8 Experiments

In this section, we evaluate the theoretical results of the previous sections empirically for small values of n. We are interested in the following specific schemes: the standard Bloom filter scheme, the partition scheme, the double hashing scheme, and the extended double hashing schemes where $f(i) = i^2$ and $f(i) = i^3$.

For $c \in \{4, 8, 12, 16\}$, we do the following. First, compute the value of $k \in \{\lfloor c \ln 2 \rfloor, \lceil c \ln 2 \rceil\}$ that minimizes $p = (1 - \exp[-k/c])^k$. Next, for each of the schemes under consideration, repeat the following procedure $10,000$ times: instantiate the filter with the specified values of n, c,

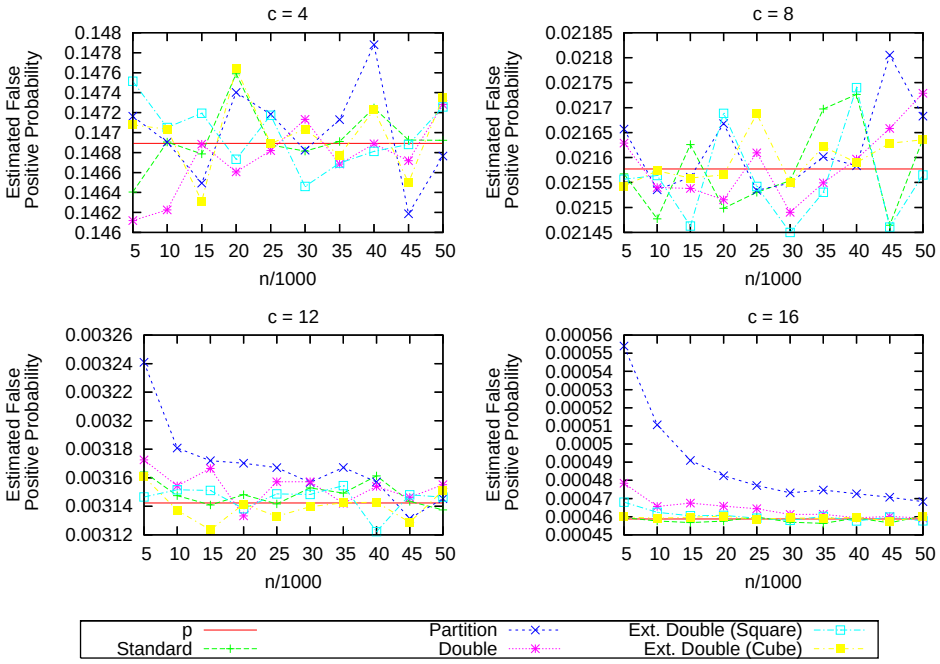

Figure 1: Estimates of the false positive probability for various schemes and parameters.

and k, populate the filter with a set S of n items, and then query $\lceil 10/p \rceil$ elements not in S, recording the number Q of those queries for which the filter returns a false positive. We then approximate the false positive probability of the scheme by averaging the results over all $10,000$ trials. Furthermore, we bin the results of the trials by their values for Q in order to examine the other characteristics of Q's distribution.

The results are shown in Figures 1 and 2. In Figure 1, we see that for small values of c, the different schemes are essentially indistinguishable from each other, and simultaneously have a false positive probability/rate close to p. This result is particularly significant since the filters that we are experimenting with are fairly small, supporting our claim that these schemes are useful even in settings with very limited space. However, we also see that for the slightly larger values of $c \in \{12, 16\}$, the partition scheme is no longer particularly useful for small values of n, while the other schemes are. This result is not particularly surprising, since we know from Section 6 that all of these schemes are unsuitable for small values of n and large values of c. Furthermore, we expect that the partition scheme is the least suited to these conditions, given the observation in Section 2 that the partitioned version of a standard Bloom filter never performs better than the original version. Nevertheless, the partition scheme might still be useful in certain settings, since it gives a substantial reduction in the range of the hash functions.

In Figure 2, we give histograms of the results from our experiments with $n = 5000$ and $c = 8$ for the partition and extended double hashing schemes. Note that for this value of c, optimizing for k yields $k = 6$, so we have $p \approx 0.021577$ and $\lceil 10/p \rceil = 464$. In each plot, we compare the results to $f \stackrel{\text{def}}{=} 10,000\phi_{464p,464p(1-p)}$, where

$$\phi_{\mu,\sigma^2}(x) \stackrel{\text{def}}{=} \frac{\mathrm{e}^{-(x-\mu)^2/2\sigma^2}}{\sigma\sqrt{2\pi}}$$

denotes the density function of $\mathrm{N}(\mu, \sigma^2)$. As one would expect, given central limit theorem in the fourth part of Theorem 7.2, f provides a reasonable approximation to each of the histograms.

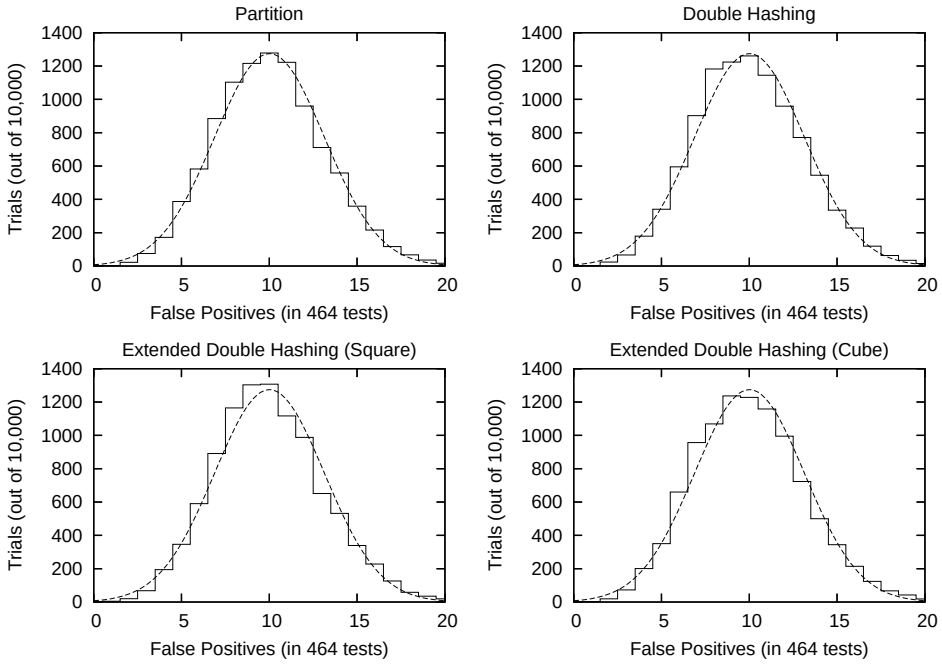

Figure 2: Estimate of distribution of Q (for $n = 5000$ and $c = 8$), compared with f.

9 A Modified Count-Min Sketch

We now present a modification to the Count-Min sketch introduced in [4] that uses fewer hash functions in a manner similar to our improvement for Bloom filters, at the cost of a small space increase. We begin by reviewing the original data structure.

9.1 Count-Min Sketch Review

The following is an abbreviated review of the description given in [4]. A Count-Min sketch takes as input a stream of *updates* (i_t, c_t), starting from $t = 1$, where each *item* i_t is a member of a universe $U = \{1, \ldots, n\}$, and each *count* c_t is a positive number. (Extensions to negative counts are possible; we do not consider them here for convenience.) The state of the system at time T is given by a vector $\vec{a}(T) = (a_1(T), \ldots, a_n(T))$, where $a_j(T)$ is the sum of all c_t for which $t \leq T$ and $i_t = j$. We generally drop the T when the meaning is clear.

The Count-Min sketch consists of an array Count of width $w \stackrel{\text{def}}{=} \lceil \mathrm{e}/\epsilon \rceil$ and depth $d \stackrel{\text{def}}{=} \lceil \ln 1/\delta \rceil$: $\mathrm{Count}[1,1], \ldots, \mathrm{Count}[d,w]$. Every entry of the array is initialized to 0. In addition, the Count-Min sketch uses d hash functions chosen independently from a pairwise independent family $\mathcal{H} : \{1, \ldots, n\} \to \{1, \ldots, w\}$.

The mechanics of the Count-Min sketch are extremely simple. Whenever an update (i, c) arrives, we increment $\mathrm{Count}[j, h_j(i)]$ by c, for $j = 1, \ldots, d$. Whenever we want an estimate of a_i (called a *point query*), we compute

$$\hat{a}_i \stackrel{\text{def}}{=} \min_{j=1}^{d} \mathrm{Count}[j, h_j(i)].$$

The fundamental result of Count-Min sketches is that for every i,

$$\hat{a}_i \geq a \quad \text{and} \quad \mathbf{Pr}(\hat{a}_i \leq a_i + \epsilon \|\vec{a}\|) \geq 1 - \delta,$$

where the norm is the L_1 norm. Surprisingly, this very simple bound allows for a number of sophisticated estimation procedures to be efficiently and effectively implemented on Count-Min sketches. The reader is once again referred to [4] for details.

9.2 Using Fewer Hash Functions

We now show how the improvements to Bloom filters discussed previously in this paper can be usefully applied to Count-Min sketches. Our modification maintains all of the essential features of Count-Min sketches, but reduces the required number of pairwise independent hash functions to $2\lceil(\ln 1/\delta)/(\ln 1/\epsilon)\rceil$. We expect that, in many settings, ϵ and δ will be related, so that only a constant number of hash functions will be required; in fact, in many such situations only two hash functions are required.

We describe a variation of the Count-Min sketch that uses just two pairwise independent hash functions and guarantees that

$$\hat{a}_i \geq a \quad \text{and} \quad \mathbf{Pr}(\hat{a}_i \leq a_i + \epsilon\|\vec{a}\|) \geq 1 - \epsilon.$$

Given such a result, it is straightforward to obtain a variation that uses $2\lceil(\ln 1/\delta)/(\ln 1/\epsilon)\rceil$ pairwise independent hash functions and achieves the desired failure probability δ: simply build $2\lceil(\ln 1/\delta)/(\ln 1/\epsilon)\rceil$ independent copies of this data structure, and always answer a point query with the minimum estimate given by one of those copies.

Our variation will use d tables numbered $\{0, 1, \ldots, d-1\}$, each with exactly w counters numbered $\{0, 1, \ldots, w-1\}$, where d and w will be specified later. We insist that w be prime. Just as in the original Count-Min sketch, we let $\text{Count}[j, k]$ denote the value of the kth counter in the jth table. We choose hash functions h_1 and h_2 independently from a pairwise independent family $\mathcal{H} : \{0, \ldots, n-1\} \to \{0, 1, \ldots, w-1\}$, and define $g_j(x) = h_1(x) + jh_2(x) \bmod w$ for $j = 0, \ldots, d-1$.

The mechanics of our data structure are the same as for the original Count-Min sketch. Whenever an update (i, c) occurs in the stream, we increment $\text{Count}[j, g_j(i)]$ by c, for $j = 0, \ldots, d-1$. Whenever we want an estimate of a_i, we compute

$$\hat{a}_i \stackrel{\text{def}}{=} \min_{j=0}^{d-1} \text{Count}[j, g_j(i)].$$

We prove the following result:

Theorem 9.1. *For the Count-Min sketch variation described above,*

$$\hat{a}_i \geq a \quad \textit{and} \quad \mathbf{Pr}(\hat{a}_i > a_i + \epsilon\|\vec{a}\|) \leq \frac{2}{\epsilon w^2} + \left(\frac{2}{\epsilon w}\right)^d.$$

In particular, for $w \geq 2\text{e}/\epsilon$ and $\delta \geq \ln 1/\epsilon(1 - 1/2\text{e}^2)$,

$$\hat{a}_i \geq a \quad \textit{and} \quad \mathbf{Pr}(\hat{a}_i > a_i + \epsilon\|\vec{a}\|) \leq \epsilon.$$

Proof. Fix some item i. Let A_i be the total count for all items z (besides i) with $h_1(z) = h_1(i)$ and $h_2(z) = h_2(i)$. Let $B_{j,i}$ be the total count for all items z with $g_j(i) = g_j(z)$, excluding i and items z counted in A_i. It follows that

$$\hat{a}_i = \min_{j=0}^{d-1} \text{Count}[j, g_j(i)] = a_i + A_i + \min_{j=0}^{d-1} B_{j,i}.$$

The lower bound now follows immediately from the fact that all items have nonnegative counts, since all updates are positive. Thus, we concentrate on the upper bound, which we approach by noticing that

$$\mathbf{Pr}(\hat{a}_i \geq a_i + \epsilon\|\vec{a}\|) \leq \mathbf{Pr}(A_i \geq \epsilon\|\vec{a}\|/2) + \mathbf{Pr}\left(\min_{j=0}^{d-1} B_{j,i} \geq \epsilon\|\vec{a}\|/2\right).$$

We first bound A_i. Letting $\mathbf{1}(\cdot)$ denote the indicator function, we have

$$\mathbf{E}[A_i] = \sum_{z \neq i} a_z \, \mathbf{E}[\mathbf{1}(h_1(z) = h_1(i) \wedge h_2(z) = h_2(i))] \leq \sum_{z \neq i} a_z/w^2 \leq \|\vec{a}\|/w^2,$$

where the first step follows from linearity of expectation and the second step follows from the definition of the hash functions. Markov's inequality now implies that

$$\mathbf{Pr}(A_i \geq \epsilon\|\vec{a}\|/2) \leq 2/\epsilon w^2.$$

To bound $\min_{j=0}^{d-1} B_{j,i}$, we note that for any $j \in \{0, \ldots, d-1\}$ and $z \neq i$,

$$\begin{aligned}\mathbf{Pr}((h_1(z) \neq h_1(i) \vee h_2(z) \neq h_2(i)) \wedge g_j(z) = g_j(i)) &\leq \mathbf{Pr}(g_j(z) = g_j(i)) \\ &= \mathbf{Pr}(h_1(z) = h_1(i) + j(h_2(i) - h_2(z))) \\ &= 1/w,\end{aligned}$$

so

$$\mathbf{E}[B_{j,i}] = \sum_{z \neq i} a_z \, \mathbf{E}[\mathbf{1}((h_1(z) \neq h_1(i) \vee h_2(z) \neq h_2(i)) \wedge g_j(z) = g_j(i))] \leq \|\vec{a}\|/w,$$

and so Markov's inequality implies that

$$\mathbf{Pr}(B_{j,i} \geq \epsilon\|\vec{a}\|/2) \leq 2/\epsilon w$$

For arbitrary w, this result is not strong enough to bound $\min_{j=0}^{d-1} B_{j,i}$. However, since w is prime, each item z can only contribute to one $B_{k,i}$ (since if $g_j(z) = g_j(i)$ for two values of j, we must have $h_1(z) = h_1(i)$ and $h_2(z) = h_2(i)$, and in this case z's count is not included in any $B_{j,i}$). In this sense, the $B_{j,i}$'s are negatively dependent [7]. It follows that for any value v,

$$\mathbf{Pr}\left(\min_{j=0}^{d-1} B_{j,i} \geq v\right) \leq \prod_{j=0}^{d-1} \mathbf{Pr}(B_{j,i} \geq v).$$

In particular, we have that

$$\mathbf{Pr}\left(\min_{j=0}^{d-1} B_{j,i} \geq \epsilon\|\vec{a}\|/2\right) \leq (2/\epsilon w)^d,$$

so

$$\begin{aligned}\mathbf{Pr}(\hat{a}_i \geq a_i + \epsilon\|\vec{a}\|) &\leq \mathbf{Pr}(A_i \geq \epsilon\|\vec{a}\|/2) + \mathbf{Pr}\left(\min_{j=0}^{d-1} B_j, i \geq \epsilon\|\vec{a}\|/2\right) \\ &\leq \frac{2}{\epsilon w^2} + \left(\frac{2}{\epsilon w}\right)^d.\end{aligned}$$

And for $w \geq 2\mathrm{e}/\epsilon$ and $\delta \geq \ln 1/\epsilon(1 - 1/2\mathrm{e}^2)$, we have

$$\frac{2}{\epsilon w^2} + \left(\frac{2}{\epsilon w}\right)^d \leq \epsilon/2e^2 + \epsilon(1 - 1/2\mathrm{e}^2) = \epsilon,$$

completing the proof. □

10 Conclusion

Bloom filters are simple randomized data structures that are extremely useful in practice. In fact, they are so useful that any significant reduction in the time required to perform a Bloom filter operation immediately translates to a substantial speedup for many practical applications. Unfortunately, Bloom filters are so simple that they do not leave much room for optimization.

This paper focuses on modifying Bloom filters to use less of the only resource that they traditionally use liberally: (pseudo)randomness. Since the only nontrivial computations performed by a Bloom filter are the constructions and evaluations of pseudorandom hash functions, any reduction in the required number of pseudorandom hash functions yields a nearly equivalent reduction in the time required to perform a Bloom filter operation (assuming, of course, that the Bloom filter is stored entirely in memory, so that random accesses can be performed very quickly).

We have shown that a Bloom filter can be implemented with only two pseudorandom hash functions without any increase in the asymptotic false positive probability, and, for Bloom filters of fixed size with reasonable parameters, without any substantial increase in the false positive probability. We have also shown that the asymptotic false positive probability acts, for all practical purposes and reasonable settings of a Bloom filter's parameters, like a false positive rate. This result has enormous practical significance, since the analogous result for standard Bloom filters is essentially the theoretical justification for their extensive use.

More generally, we have given a general framework for analyzing modified Bloom filters, which we expect will be used in the future to refine the specific schemes that we analyzed in this paper. We also expect that the techniques used in this paper will be usefully applied to other data structures, as demonstrated by our modification to the Count-Min sketch.

Acknowledgements

We are very grateful to Peter Dillinger and Panagiotis Manolios for introducing us to this problem, providing us with advance copies of their work, and also for many useful discussions.

References

[1] P. Billingsley. Probability and Measure, Third Edition. John Wiley & Sons, 1995.

[2] P. Bose, H. Guo, E. Kranakis, A. Maheshwari, P. Morin, J. Morrison, M. Smid, and Y. Tang. *On the false-positive rate of Bloom filters*. Submitted. Temporary version available at `http://cg.scs.carleton.ca/~morin/publications/ds/bloom-submitted.pdf`

[3] A. Broder and M. Mitzenmacher. *Network Applications of Bloom Filters: A Survey*. Internet Mathematics, to appear. Temporary version available at `http://www.eecs.harvard.edu/~michaelm/postscripts/tempim3.pdf`.

[4] G. Cormode and S. Muthukrishnan. *Improved Data Stream Summaries: The Count-Min Sketch and its Applications*. DIMACS Technical Report 2003-20, 2003.

[5] P. C. Dillinger and P. Manolios. *Bloom Filters in Probabilistic Verification*. FMCAD 2004, Formal Methods in Computer-Aided Design, 2004.

[6] P. C. Dillinger and P. Manolios. *Fast and Accurate Bitstate Verification for SPIN*. SPIN 2004, 11th International SPIN Workshop on Model Checking of Software, 2004.

[7] D. P. Dubhashi and D. Ranjan. *Balls and Bins: A Case Study in Negative Dependence*. Random Structures and Algorithms, 13(2):99-124, 1998.

[8] L. Fan, P. Cao, J. Almeida, and A. Z. Broder. *Summary cache: a scalable wide-area Web cache sharing protocol*. IEEE/ACM Transactions on Networking, 8(3):281-293, 2000.

[9] K. Ireland and M. Rosen. A Classical Introduction to Modern Number Theory, Second Edition. Springer-Verlag, New York, 1990.

[10] D. Knuth. The Art of Computer Programming, Volume 3: Sorting and Searching. Addison-Wesley, Reading Massachusetts, 1973.

[11] M. Mitzenmacher. *Compressed Bloom Filters*. IEEE/ACM Transactions on Networking, 10(5):613-620, 2002.

[12] M. Mitzenmacher and E. Upfal. Probability and Computing: Randomized Algorithms and Probabilistic Analysis. Cambridge University Press, 2005.

[13] M. V. Ramakrishna. *Practical performance of Bloom filters and parallel free-text searching*. Communications of the ACM, 32(10):1237-1239, 1989.