The Bloom Filter Hashing Solution

Mark E. Lehr Ph.D.

May 2021

1 The sum of Occupancy

What is a Bloom Filter? It is a probabilistic data structure which represents a very efficient lookup procedure. It is typically implemented with a bit vector X of size N that uses k hashes H_{km}^N of m elements from an enumerated list.

Let's start with defining a recursive sequence for the expectation of the sum of elements hashed into a bit array of size N. Let 0/1 represent the occupancy of each position in the array. If multiple elements are hashed to the same index, then this represents a collision. However, this does not increase the binomial condition of occupancy. As the number of hashed elements are added into the array, the conditional expectation becomes $E[\sum X/H_{km}^N] = E_{km} = \sum_{i=1}^{N} x_i P(x_i) = \sum_{i=1}^{N} (1_i P(1_i) + 0_i P(0_i))$ which in recursive incremental form is:

$$E_0 = 0$$

$$E_1 = 1$$

$$E_2 = E_1 + 1 * \frac{N - E_1}{N} + 0 * \frac{E_1}{N}$$

$$E_3 = E_2 + 1 * \frac{N - E_2}{N} + 0 * \frac{E_2}{N}$$

$$E_4 = E_3 + 1 * \frac{N - E_3}{N} + 0 * \frac{E_3}{N}$$

$$\vdots$$

$$E_{km-1} = E_{km-2} + 1 * \frac{N - E_{km-2}}{N} + 0 * \frac{E_{km-2}}{N}$$

$$E_{km} = E_{km-1} + 1 * \frac{N - E_{km-1}}{N} + 0 * \frac{E_{km-1}}{N}$$

The expected sum can easily be computed with algebraic simplification. Notice that we can rewrite this sum by combining like elements and define $C = \frac{N-1}{N}$ with the following:

$$E_0 = 0$$

$$E_1 = 1$$

$$E_2 = CE_1 + 1$$
$$E_3 = CE_2 + 1$$
$$E_4 = CE_3 + 1$$
$$\vdots$$
$$E_{km-1} = CE_{km-2} + 1$$
$$E_{km} = CE_{km-1} + 1$$

By successive summations starting at the base condition and proceeding up the sequence the final form is

$$E_{km} = \sum_{i=0}^{km-1} C^i$$

Of course, the sum of such a series $S_n = \sum_{i=0}^{N} R^i = \frac{1-R^{N+1}}{1-R}$ and using the definition of $C(N)$

$$E_{km} = \frac{1 - C^{km}}{1 - C}$$
$$E_{km} = N(1 - C^{km})$$

We can approximate this by using $e^{(-\frac{1}{N})} \approx C = \frac{N-1}{N} = (1 - \frac{1}{N})$ for $N \gg 1$

$$E_{km} \approx N(1 - e^{\frac{-km}{N}})$$

2 False Positives

The probability of a random hash landing on an occupied position is:

$$P_{km} = \frac{E_{km}}{N} = (1 - C^{km}) \approx (1 - e^{\frac{-km}{N}})$$

However, the probability of random k hashes all corresponding to occupied positions is a like a biased coin from a binomial distribution with an unlikely false positive:

$$P_{fp} = P_{km}^k = (1 - C^{km})^k \approx (1 - e^{\frac{-km}{N}})^k$$

How to interpret this result? We can design a bloom filter with a false positive completely predetermined. A table can be created with k representing the fields and N/m representing the ratio of spaces to hashes in the records. Families of equal probabilities can be mapped to display an optimum k for any desired probability. A theoretical value can initially be calculated. When the m elements are hashed an exact value for P_{km} can be calculated and correspondingly P_{fp}