CS 7280: Data Str & Alg Scalable Comp	Spring 2025

Lecture 15 — March 10, 2025

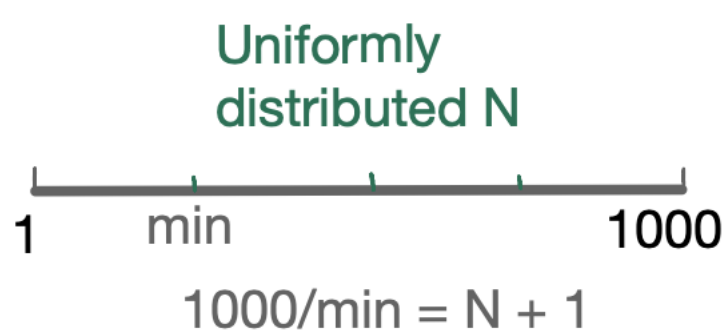
Prof. Prashant Pandey — *Scribe: Yirong Wang*

1 Overview

In the last lecture, we covered sketch-based streaming algorithms for frequency estimation of streaming data. In this lecture, we will discuss cardinality estimation techniques and related problems, including the Hat Problem, Mash (Bottom-k Minhash Sketch), the LogLog and HyperLogLog algorithms.

2 Hat Problem

Suppose I take cards labeled from 1 to 1000 and choose a random subset of size N to hide in my hat. You are allowed to see one representative card from the cards in the hat such as the *median*, *minimum*, or *maximum*, and you need to estimate N.



For example, if the minimum card you see is 40, you can estimate N using the approximation:

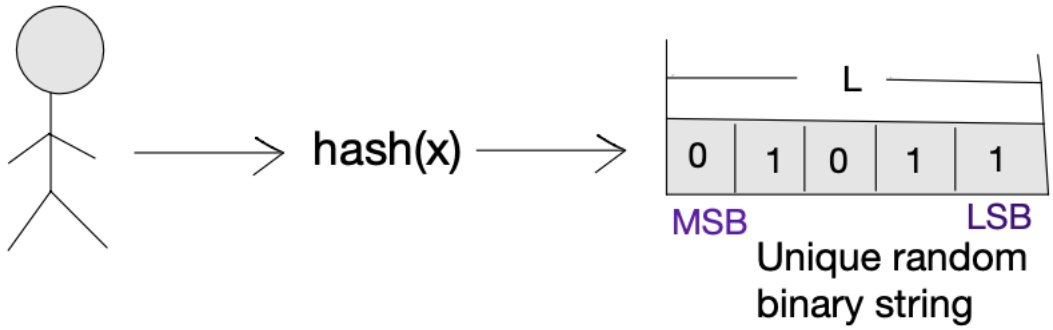
$$40 \approx \frac{1000}{N+1} \implies N \approx 24$$

This method provides a simple estimation that fits in 10 bits of memory.

3 LogLog Cardinality Estimation

Simple solution: a hash table takes $\Omega(N)$ space.

We are going to use randomness.



Hash function:

$$P_\lambda(0) = \frac{1}{2}$$
$$P_\lambda(1) = \frac{1}{2}$$

Coin game: flip a coin, if `H`, flip again. If `T`, stop.

- Probability of getting exactly 3 `H`s is:

$$\frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} = \frac{1}{16}$$

- Exactly 16 attempts on sequence will have exactly 3 `H`s.

- Another way of saying, if we see a sequence of 3 `H`s, we know there are probably 16 items.

- If in all hash values, the longest streak of `H`s is L, then on average there are 2^{L+1} items.

- Count leading 0s in hashes. Estimate the total number of distinct items. E.g., if leading 0s = 2, then the cardinality estimation is $2^{2+1} = 8$ items.

- Let M be the maximum number of unique elements.

$$2^{L+1} \leq M \Rightarrow L \leq lgM$$

Bit length of $L = lgL = lglgM$.

Problems:

- Will only estimate powers of 2.
- Too much luck involved.

Solution: use multiple approximate counters (ACs) and average the results.

$$\frac{AC_1 + AC_2 + \cdots + AC_k}{k}$$

can have bias.

3.1 HyperLogLog Algorithm

Takes the harmonic mean of k ACs.

Definition 1. *The harmonic mean of k numbers $x_1, x_2, \ldots, x_k$ is defined as:*

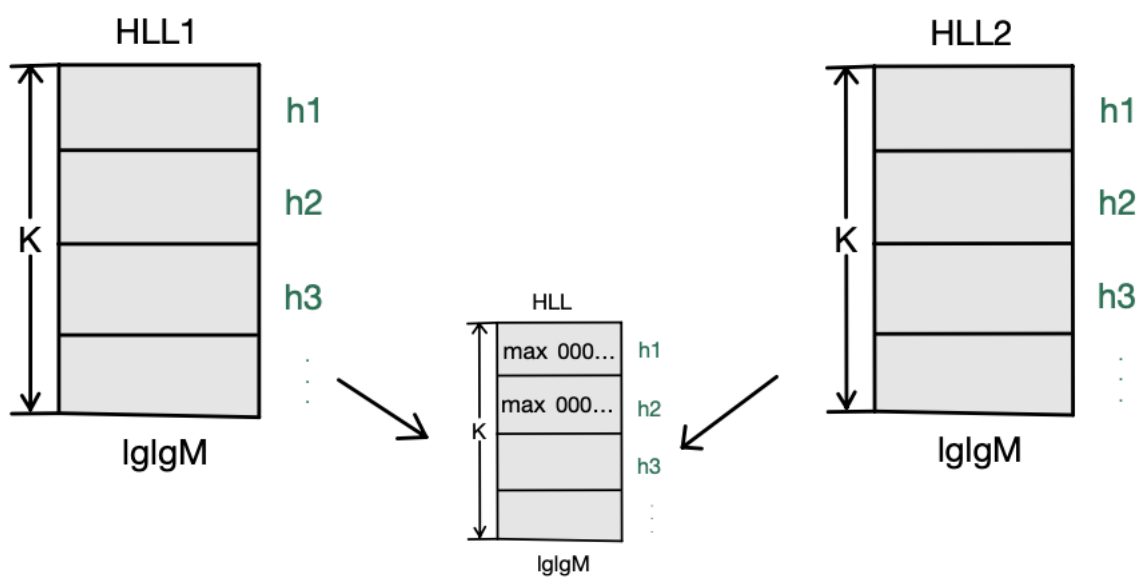
$$H = \left(\frac{{x_1}^{-1} + {x_2}^{-1} + \cdots + {x_k}^{-1}}{k} \right)^{-1}$$

less sensitive to large outliers.

Union estimation. HyperLogLog (HLL) maintains a number of "registers" (or "buckets"), each of which records (roughly) the largest number of leading 0s among the elements mapped to that register.

Suppose there are 2 HLLs, each sees its own stream. When merging two HLLs to estimate the cardinality of the union, you merge the per-register values by taking the maximum in each register:

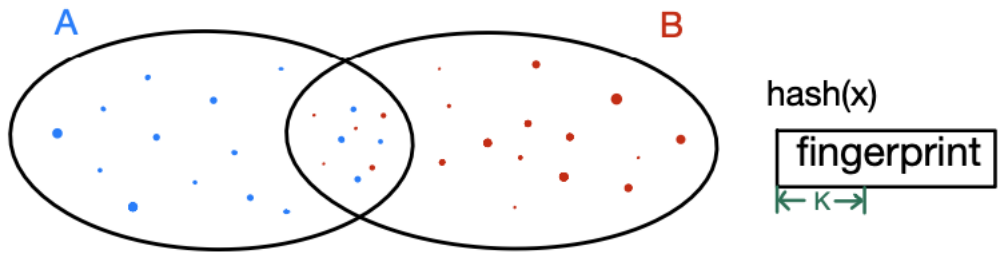
$$\text{merged}[i] = \max(\text{HLL}_1[i],\ \text{HLL}_2[i])$$



Then what about the intersection?

3.2 Mash (Bottom-k Sketch)

The Bottom-k Minhash Sketch, often referred to as Mash, is a sketching technique designed to estimate the Jaccard similarity between sets efficiently, especially in streaming data applications.



Definition 2. *Given two sets A and B, the Jaccard similarity is defined as:*

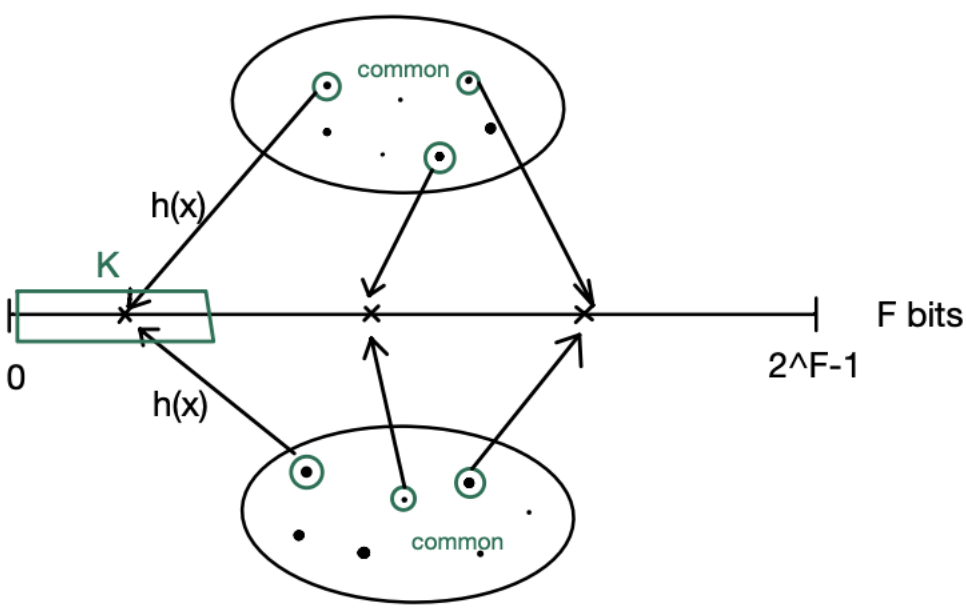
$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

Method: Maintains a sketch by storing the K smallest hash values of the elements in each set, the Jaccard similarity can be approximated as:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \approx \frac{|\text{Sketch}(A \cup B) \cap \text{Sketch}(A) \cap \text{Sketch}(B)|}{|\text{Sketch}(A \cup B)|}$$

Applications: Similiarity search over genomic sequences:

- Genomes
- Metagenomes
- Amino acids



This method is unbiased. Because Sketch($A \cup B$) is a random sample of $A \cup B$, the fraction of elements in Sketch($A \cup B$) that are shared by both Sketch(A) and Sketch(B) is an unbiased estimation of $J(A, B)$.

4 Bibliography

References

[1] Noga Alon, Yossi Matias, Mario Szegedy. The Space Complexity of Approximating the Frequency Moments. *J. Comput. Syst. Sci.*, 58(1):137–147, 1999.