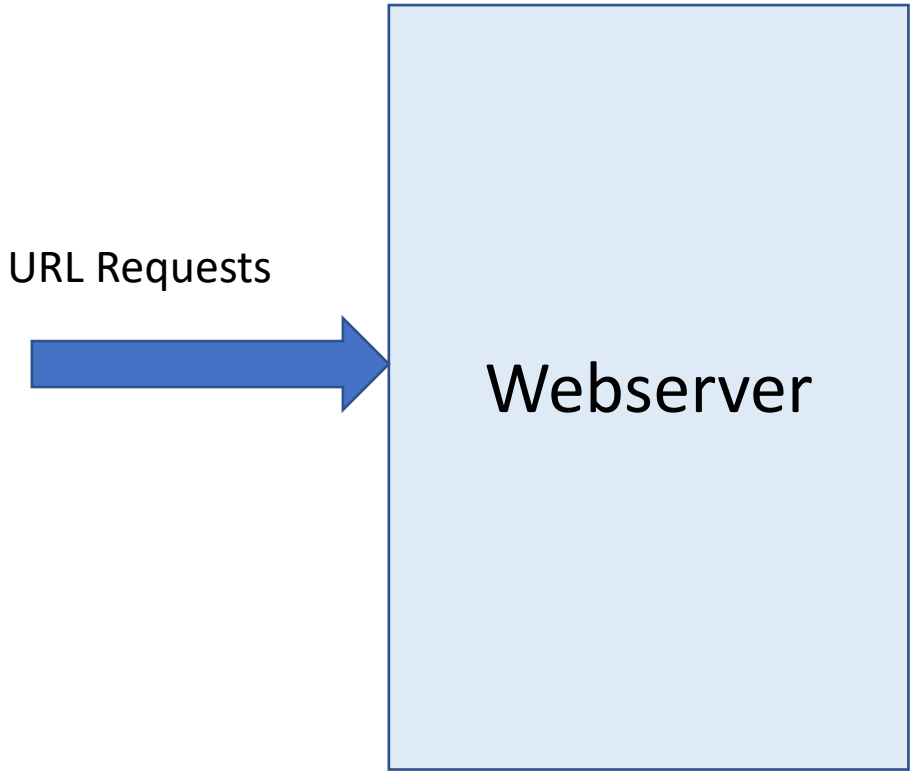
Count-Min Sketches

Sriram Sankaranarayanan

Data Structures and Algorithms

Problem: Count Items in a Data Stream



Problem: Keep count of how often each URL is requested.

Unique URLs: U_1 U_2 U_3 U_4 U_5 U_6 … U_N

Have N distinct counters: C(1),…, C(N)

Each time URL U_j requested: Increment counter C(j)

Problem: N is humungous.
Most URLs are requested very few times.
A few URLs are requested a lot of times.

Advantage: *Approximate count within 10% of actual answer (say) is acceptable.*

Approximate Counting Data-Structure

Stream of data: $x_1 x_2 x_3 x_4, \cdots, x_W$

Each element of the stream : $x_j \in \{1, \ldots, N\}$

Return $approxCount(j)$: must be within ϵW of *true count*
with probability at least δ

- Typical Numbers: $N \sim 10^8, W \sim 10^9, \quad \epsilon \sim 10^{-6}, \quad \delta = 0.99$
- From a stream of nearly 1 billion items each having a number between 1 and 100Million, count how often each item occurs where the count is within 1000 of the true count at least 99% of the time.

Basic Idea of Count-Min Sketch

Use m counters: $C(1) \ldots C(m)$

We will choose m later (expect $m \ll M$)

Draw a hash function at random from family H = $\{ h_1, h_2, \ldots \}$

$$h_i : \{1, .., M\} \rightarrow \{1, \ldots m\}$$

Stream Item x_j: $\text{Increment}(C\left(h\left(x_j\right)\right))$

$approxCount(k) = C(h(k))$

Count-Min Sketch Error Analysis

- $approxCount(j) \geq count(j)$

Count-Min Sketch: Choosing m

Count-Min Sketch: Reducing Error Probability

Count-Min Sketch: Overall Algorithm

- Initialize K counter-banks with hash functions : $h_1, h_2, \dots, h_K$
- Stream item x_j
  - Increment $C_1(h_1(x_j)), C_2(h_2(x_j)), \dots, C_K(h_K(x_j))$
- Query count of k
  - $approxCount(k) = \min(C_1(h(k)), C_2(h(k)), \dots C_K(h(k)))$

Count-Min Sketch: Some actual numbers

- Stream of 1 Billion Items
- $\epsilon = 10^{-6}$ (tolerate error of upto 1000)
- $\delta = 0.9$

- $m = \frac{e}{\epsilon} \approx 3 \times 10^6, K = -\ln(1-\delta) \approx 3$
- Use 3 banks of 3 million counters, each.
  - Guarantees that `approxCount` will be within 1000 of true at least answer 90% of the time.