Lecture 8: Count-min Sketch

CMSC 5705 Advanced Topics in Database Systems

Yufei Tao

Department of Computer Science and Engineering
Chinese University of Hong Kong

November 16, 2010

This lecture will discuss another data structure on streams. As mentioned in the previous lecture, in this context, an infinite number of data items arrive continuously, whereas the memory capacity is bounded by a small size. Every item can be seen only once. The goal is to use such a small memory to answer interesting queries with strong precision guarantees.

Problem definitions

Given an integer x, we denote by $[x]$ the set of integers in $[1, x]$. In this lecture, we assume a *key domain* of $[U]$, where U is an integer.

Definition (Stream)

A *stream* is an infinite sequence of operations, each of which has the form (k, v), where k is in the domain $[U]$, and v is in the real domain $\mathbb{R}$.

Definition (State vector)

A *state vector* A is defined as $[A[1], ..., A[U]]$, where $A[i]$ $(1 \leq i \leq U)$ equals the sum of the v-values of all the past updates (k, v) where $k = i$.

We will assume that all values of A are non-negative at all times, i.e., $A[i] \geq 0$ for all i.

Problem definitions (cont.)

Problem (Point query)

Given a key $k \in [U]$, a *point query* returns an estimated value of $A[k]$.

Problem (Range query)

Given keys $k_1, k_2 \in [U]$ with $k_1 \leq k_2$, a *range query* returns an estimated value of $\sum_{k_1 \leq k \leq k_2} A[k]$.

Clearly, by using $\Omega(U)$ space, we can easily answer both queries exactly. U, however, may be a huge value such that $\Omega(U)$ space may not be affordable in practice. Our goal is to use space significantly less than $O(U)$ and yet still be able to process queries accurately (i.e., minimizing their errors).

Hash function

The count-min sketch we introduce shortly deploys *hash functions*. For our discussion, we focus on hash functions $f : [n] \to [m]$. That is, given a value $x \in [n]$, $f(x)$ falls in the domain $[m]$. The function has the property that, given any $x_1, x_2 \in [n]$ with $x_1 \neq x_2$, the probability for $f(x_1) = f(x_2)$ equals $1/m$.

Note

Many simple functions satisfy the above property very well in practice. One example is $f(x) = 1 + (\alpha x + \beta)\ mod\ m$, where α (β) is randomly selected from $[p]$ with p being a very large prime number.

Count-min sketch

The structure consists of

- a $d \times w$ array $CM[i,j]$, i.e., $1 \leq i \leq d$ and $1 \leq j \leq w$. The value of d (w) is called the *depth* (*width*) of the array.
- d independent hash functions $h_1, ..., h_d$ from $[H]$ to $[w]$, where H is an integer, and called the *hash domain*.

Space consumption $O(dw)$.

Update

Build a count-min sketch with $H = U$. Given an operation (k, v), update the sketch by adding v to $CM[i, h_i(j)]$ for each $i \in [d]$.

Time $= O(d)$.

Answering a point query

Given the query key $k \in [U]$, return:

$$\min_{1 \leq i \leq d} CM[i, h_i(k)].$$

Query time $= O(d)$.

Theoretical guarantee

Theorem

Let $\hat{\gamma}$ be the answer returned by the count-min sketch for a point query whose exact answer is γ. Choosing $w = \lceil e/\epsilon \rceil$ and $d = \lceil \ln(1/\delta) \rceil$, we have: Then:

$$\hat{\gamma} - \epsilon \|A\|_1 \leq \gamma \leq \hat{\gamma}.$$

The first inequality holds with probability at least $1/\delta$, and the second inequality holds with certainty.

We will prove the theorem in the next few slides.

Proof

Our proof requires the Markov inequality:

Markov inequality

Let X be a random variable. It holds that:

$$Pr(|X| \geq c) \leq \frac{E(|X|)}{c}$$

where c is any positive constant.

Proof (cont.)

If we report directly $CM[1, h_1(k)]$ as the answer, the expectation of the maximum error (i.e., with respect to γ) is

$$\|A\|_1/w \leq \epsilon\|A\|_1/e$$

(i.e., on average, $1/w$ of all the updates contributed to $CM[1, h_1(k)]$). By Markov inequality, the error is larger than $\epsilon\|A\|_1$ with probability at most $1/e$.

Proof (cont.)

The same reasoning applies to the $CM[i, h_1(k)]$ of every i.

Hence, $\hat{\gamma} = \min_{1 \le i \le d} CM[i, h_i(k)]$ has an error greater than $\epsilon \|A\|_1$ if and only if the $CM[i, h_1(k)]$ of *all* i have an error exceeding $\epsilon \|A\|_1$. The independence of $h_1, ..., h_d$ indicates that this can happen with probability at most

$$(1/e)^d \le \delta.$$

□

Now we switch our attention to range queries. A naive way to answer such a query with search interval $[k_1, k_2]$ is to issue $k_2 - k_1 + 1$ point queries, and combine their results. This approach, however, is not likely to be accurate. More space is needed to boost the precision.

Without loss of generality, assume U is a power of 2. Partition the domain $[U]$ into $\log_2 U$ sets of intervals:

- S_1: evenly partition U into intervals of length 1.
- S_2: evenly partition U into intervals of length 2.
- S_3: evenly partition U into intervals of length 4.

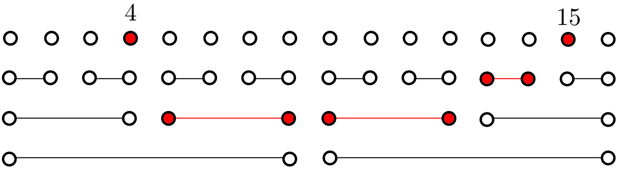
  ...

- $S_{\log_2 U}$: evenly partition U into intervals of length $2^{(\log_2 U)-1} = U/2$.

Each interval (in any of these sets) is called a *dyadic interval*.

An observation

Any range $[k_1, k_2]$ in $[U]$ can be broken into at most $2 \log U$ dyadic intervals. The following figure shows an example where $[4, 15]$ is broken into 5 dyadic ranges shown in red.



Each set of dyadic intervals can be regarded as a coarse version of the domain $[U]$.

Structure for range queries

For S_i $(1 \leq i \leq log_2 U)$, maintain a count-min sketch CM_i with

- $H = U/2^{i-1}$ (i.e., the number of intervals in the set)
- $w = \lceil 2e \log U/\epsilon \rceil$
- $d = \lceil 1/\delta \rceil$.

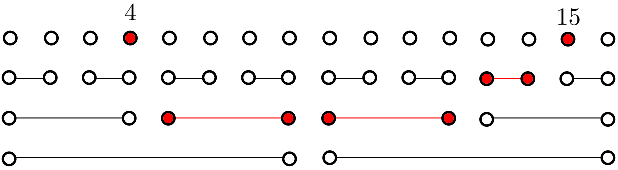
Space $= O(\frac{log^2 U}{\epsilon} \log \frac{1}{\delta})$.

Update

Given an operation (k, v), update CM_i $(1 \leq i \leq log_2 U)$ in the same way as in point queries by converting the operation to $(\lceil k/2^{i-1} \rceil, v)$. Namely, for each $1 \leq j \leq d$, add v to $CM_i[j, h_j(k')]$, where $k' = \lceil k/2^{i-1} \rceil$.

Answering a range query

Recall that a query range can be partitioned into at most $2 \log U$ dyadic ranges, two in each S_i $(1 \le i \le \log_2 U)$. Perform at most 2 point queries in each CM_i, corresponding to the dyadic intervals in S_i



Sum up the answers of all point queries.

Theoretical guarantee

Theorem

Let $\hat{\gamma}$ be the answer returned by the count-min sketch for a range query whose exact answer is γ. Then:

$$\hat{\gamma} - \epsilon \|A\|_1 \leq \gamma \leq \hat{\gamma}.$$

The first inequality holds with probability at least $1/\delta$, and the second inequality holds with certainty.

Proof

The key observation is that, if we had set $d = 1$, then the estimate we get would have an expected maximum error of $\epsilon \|A\| / e$. The rest of the proof proceeds in the same way as that of the theorem for point queries.

Playback of this lecture

- Count-min sketch.
- Space $O(\frac{1}{\epsilon} \cdot \ln \frac{1}{\delta})$ for point queries (each error bounded by $\epsilon\|A\|_1$ with probability at least $1 - \delta$).
- Space $O(\frac{\log^2 U}{\epsilon} \cdot \ln \frac{1}{\delta})$ for range queries (each error bounded by $\epsilon\|A\|_1$ with probability at least $1 - \delta$).