The Count-Min Sketch with Applications

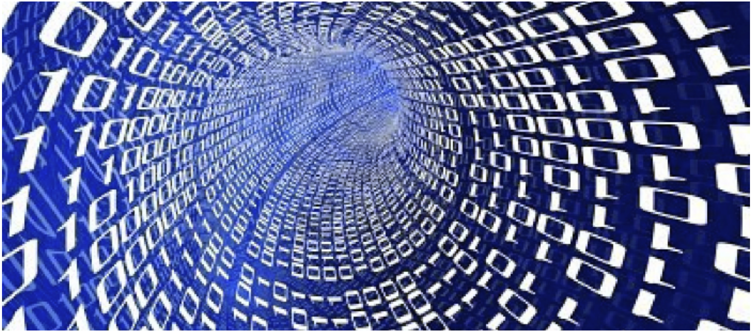
Steven Wu

University of Pennsylvania

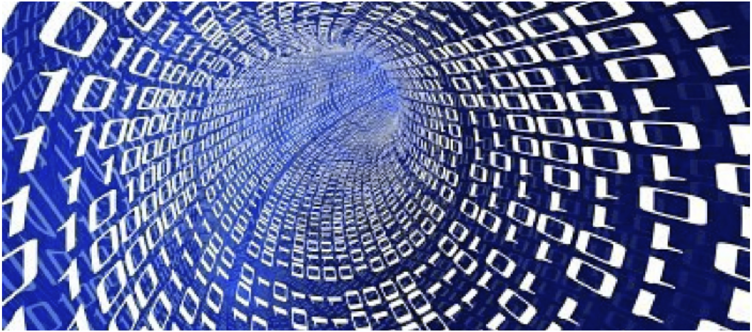
December 6, 2014

Paper by G. Cormode and S. Muthukrishnan
(awarded the 2014 Imre Simon Test-of-Time Award)

Data Streams



Data Streams



- Approach: take one pass over data, summarize the data (to answer some class of queries)

Data Stream Model

1. Data stream represents a high-dimensional vector a, initially all zero: for $1 \leq i \leq U$, $a[i] = 0$

Data Stream Model

1. Data stream represents a high-dimensional vector a, initially all zero: for $1 \leq i \leq U$, $a[i] = 0$
2. n items in the stream: t-th update is $(i(t), c(t))$, meaning $a[i(t)]$ is updated to $a[i] + c(t)$

Sketches

Figure: Sketches are a class of data summaries

Sketches

Figure: Sketches are a class of data summaries

- For example, linear projection of source data with appropriate random vectors

Count-Min Sketch

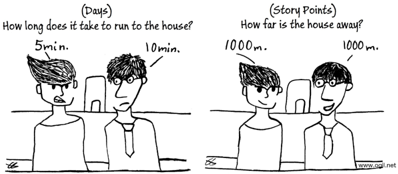
CM Sketch solve the following problems

- Point Estimation : $a[i]$
- Range Sums : $\sum_{i=j}^{k} a[i]$
- Inner Product : $\langle a, b \rangle = \sum_i a[i] \times b[i]$

Point Estimation

Problem: given i, return $a[i]$

- Let $N = \sum c(t) = \|a\|_1$
- Replace vector a with small sketch which approximates each $a[i]$ up to εN with probability $1 - \delta$



Tools

- 2-wise independent hash functions $h_1, \ldots, h_{\log(1/\delta)} \colon [U] \to \left[\frac{2}{\varepsilon}\right]$

Tools

- 2-wise independent hash functions $h_1, \ldots, h_{\log(1/\delta)} \colon [U] \to \left[\frac{2}{\varepsilon}\right]$
- A family H mapping $A \to B$ is 2-wise independent if for any distinct i, j, and any values u, v

$$\Pr_{h \in_R H}[h(i) = u \text{ and } h(j) = v] = 1/|B|^2$$

Tools

- 2-wise independent hash functions $h_1, \ldots, h_{\log(1/\delta)} \colon [U] \to \left[\frac{2}{\varepsilon}\right]$
- A family H mapping $A \to B$ is 2-wise independent if for any distinct i, j, and any values u, v

$$\Pr_{h \in_R H}[h(i) = u \text{ and } h(j) = v] = 1/|B|^2$$

- Example:

$$h(j) = a \cdot j + b \mod |B|$$

a, b are chosen independently from B and $|B|$ is prime

Update Algorithm

Update an array of counters:

Update Algorithm

Update an array of counters:
(i, count) comes in: $C[j][h_j(i)]+$ count

Update Algorithm

Update an array of counters:
(i, count) comes in: $C[j][h_j(i)] +$ count

h_1			+count				
h_2				+count			
$\vdots$		+count					
$h_{\log(1/\delta)}$							+count

Table: Array of counters, dimension: $\log(1/\delta) \times 2/\varepsilon$

Estimate

$$\hat{a}[i] = \min_j C[j][h_j(i)]$$

Estimate

$$\hat{a}[i] = \min_j C[j][h_j(i)]$$

Analysis

For the j-th counter,

$$C[j][h_j(i)] = a[i] + X_{i,j}$$

Estimate

$$\hat{a}[i] = \min_j C[j][h_j(i)]$$

Analysis

For the j-th counter,

$$C[j][h_j(i)] = a[i] + X_{i,j}$$

where $X_{i,j} = \sum_k a[k]$ such that $h_j(i) = h_j(k)$

Estimate

$$\hat{a}[i] = \min_j C[j][h_j(i)]$$

Analysis

For the j-th counter,

$$C[j][h_j(i)] = a[i] + X_{i,j}$$

where $X_{i,j} = \sum_k a[k]$ such that $h_j(i) = h_j(k)$

$$\begin{aligned}\mathbb{E}\left[X_{i,j}\right] &= \sum_{k \neq i} a[k] \times \Pr[h_j(i) = h_j(k)] \\ &\leq \varepsilon/2 \times \sum_{k \neq i} a[k] \\ &\leq \varepsilon N/2\end{aligned}$$

With high probability...

Markov Inequality:

$$\Pr[X_{i,j} \geq \varepsilon N] = \Pr[X_{i,j} \geq 2\mathbb{E}[X_{i,j}]] \leq 1/2$$

With high probability...

Markov Inequality:

$$\Pr[X_{i,j} \geq \varepsilon N] = \Pr[X_{i,j} \geq 2\mathbb{E}[X_{i,j}]] \leq 1/2$$

And so

$$\begin{aligned}\Pr[\hat{a}[i] \geq a[i] + \varepsilon N] &= \Pr[\forall j, X_{i,j} > \varepsilon N] \\ &\leq (1/2)^{\log(1/\delta)} = \delta\end{aligned}$$

With high probability...

Markov Inequality:

$$\Pr[X_{i,j} \geq \varepsilon N] = \Pr[X_{i,j} \geq 2\mathbb{E}[X_{i,j}]] \leq 1/2$$

And so

$$\begin{aligned} \Pr[\hat{a}[i] \geq a[i] + \varepsilon N] &= \Pr[\forall j, X_{i,j} > \varepsilon N] \\ &\leq (1/2)^{\log(1/\delta)} = \delta \end{aligned}$$

- For sure, $a[i] \leq \hat{a}[i]$
- With probability at least $1 - \delta$,

$$\hat{a}[i] < a[i] + \varepsilon N$$

Dyadic Intervals

$\log n$ partitions of $[n]$

- $I_0 = \{1, 2, 3, \ldots, n\}$
- $I_1 = \{\{1, 2\}, \{3, 4\} \ldots, \{n-1, n\}\}$
- $I_2 = \{\{1, 2, 3, 4\}, \{5, 6, 7, 8\}, \ldots, \{n-3, n-2, n-1, n\}\}$
- $\ldots$
- $I_{\log n} = \{[n]\}$

Dyadic Intervals

$\log n$ partitions of $[n]$

- $I_0 = \{1, 2, 3, \ldots, n\}$
- $I_1 = \{\{1, 2\}, \{3, 4\} \ldots, \{n-1, n\}\}$
- $I_2 = \{\{1, 2, 3, 4\}, \{5, 6, 7, 8\}, \ldots, \{n-3, n-2, n-1, n\}\}$
- $\ldots$
- $I_{\log n} = \{[n]\}$

Any interval (i, j) can be written as a disjoint union of at most $2 \log n$ such intervals.

Range Queries and Quantiles

- Range: $i, j \in [U]$, estimate $a[i] + \ldots + a[j]$

Range Queries and Quantiles

- Range: $i, j \in [U]$, estimate $a[i] + \ldots + a[j]$
- Approximate median: find j such that

$$a[1] + \ldots + a[j] \geq \frac{N}{2} + \varepsilon N \text{ and}$$
$$a[1] + \ldots + a[j-1] \leq \frac{N}{2} - \varepsilon N$$

Algorithm

Construct $\log U$ Count-Min Sketches, one for each I_i

Algorithm

Construct $\log U$ Count-Min Sketches, one for each I_i

Guarantee

For each $l \in I_i$, we have an estimate $\tilde{a}[l]$ for $a[l]$ such that

$$\Pr[a[l] \leq \tilde{a}[l] \leq a[l] + \varepsilon N] \geq 1 - \delta$$

Algorithm

Construct $\log U$ Count-Min Sketches, one for each I_i

Guarantee

For each $l \in I_i$, we have an estimate $\tilde{a}[l]$ for $a[l]$ such that

$$\Pr[a[l] \leq \tilde{a}[l] \leq a[l] + \varepsilon N] \geq 1 - \delta$$

To estimate range sum for interval $[i, j]$

$$\tilde{a}[i, j] = \tilde{a}[l_1] + \ldots + \tilde{a}[l_{\log U}]$$

Algorithm

Construct $\log U$ Count-Min Sketches, one for each I_i

Guarantee

For each $l \in I_i$, we have an estimate $\tilde{a}[l]$ for $a[l]$ such that

$$\Pr[a[l] \leq \tilde{a}[l] \leq a[l] + \varepsilon N] \geq 1 - \delta$$

To estimate range sum for interval $[i, j]$

$$\tilde{a}[i, j] = \tilde{a}[l_1] + \ldots + \tilde{a}[l_{\log U}]$$

Take a union bound,

$$\Pr\left[a[i, j] \leq \tilde{a}[i, j] \leq a[i, j] + \varepsilon N \log U\right] \geq 1 - \delta \log U$$

Heavy Hitters

Given a sequence of items arriving (or departing) and ϕ, find all items occurring more than ϕN times: find i for which $a[i] > \phi N$

Heavy Hitters

Given a sequence of items arriving (or departing) and ϕ, find all items occurring more than ϕN times: find i for which $a[i] > \phi N$

Approximation

Find all heavy hitters with certainty, with probability at most δ, output an item with $a[i] < (\phi - \varepsilon)N$

Cash Register Case

Figure: All updates are positive

Cash Register Case

Figure: All updates are positive

1. Keep track of $\|a(t)\|_1 = \sum_i \text{count}(t)$

Cash Register Case

Figure: All updates are positive

1. Keep track of $\|a(t)\|_1 = \sum_i \text{count}(t)$
2. (i, count) comes in check if $\hat{a}[i] \geq \phi \|a(t)\|_1$

Cash Register Case

Figure: All updates are positive

1. Keep track of $\|a(t)\|_1 = \sum_i \text{count}(t)$
2. (i, count) comes in check if $\hat{a}[i] \geq \phi\|a(t)\|_1$
3. If so, add i to the heap; scan the heap throw away j if previous estimate $\hat{a}[j] \leq \phi\|a(t)\|_t$
4. Scan the heap again at last to delete items with estimate below $\phi\|a\|_1$

Turnstile case

Figure: Both Departures and Arrivals

Problem becomes harder.

Search Structure

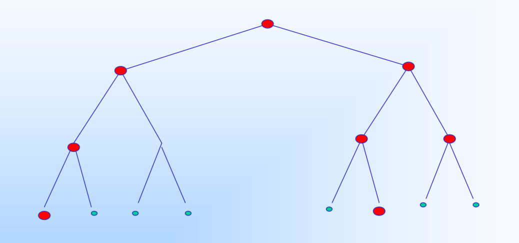


Figure: Binary Search Tree on the Universe $[U]$

- Associate internal nodes with intervals
- Compute Count-Min sketches for each I_i
- Starting from root, level-by-level, mark children l of marked nodes if $\tilde{a}[l] \geq \phi N$

Find heavy-hitters in $O(\phi^{-1} \log n)$ steps

Improved Concentration Bounds for Count-Sketch*

Gregory T. Minton
MIT

Eric Price
MIT

Figure: Improved Analysis

The Count-Min Sketch with Applications

Steven Wu

University of Pennsylvania

December 6, 2014

(Some slides credited to Graham Cormode and Grigory)