Scalable ML

10605-10805

Count-Min Sketch

Barnabás Póczos

Motivation

Safe password selection:

We want to provide protection against "statistical-guessing awards"

- **Goal**: Create an oracle that can identify undesirably popular passwords
- If a user wants to choose a way-too-popular password, we will not let her do that

What is the right data structure for this problem?

- We want to be able to query the frequency of each password used by our users

We will use the "Count-min sketch" data structure

Details: Schechter et al, Popularity Is Everything: A New Approach to Protecting Passwords from Statistical-Guessing Attacks

An Improved Data Stream Summary: The Count-Min Sketch and its Applications

Graham Cormode, S. Muthukrishnan

Journal of Algorithms
Volume 55, Issue 1, April 2005, Pages 58-75

Problem Statement

Goal:

Introduce a new **sublinear space** data structure for summarizing data streams

Data Streams

Definition: [Data stream]

$$\text{Let } a(t) \doteq \begin{pmatrix} a_1(t) \\ a_2(t) \\ \vdots \\ a_i(t) \\ \vdots \\ a_n(t) \end{pmatrix} \in \mathbb{R}^n$$

n is very-very big, e.g. number of all possible passwords in the world...

Example: $a_i(t)$ = the number of how many times the i^{th} password was used by our users till time step t.

We cannot store $a(t)$ in memory or disk since n is too big.

At time step 0 all components are zero:

$$a(0) \doteq \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{R}^n$$

The time is discrete: t=0,1,2,...

Updates in Data Streams

The t-th update is given in the form of (i_t, c_t).

- $\star$ Here $i_t \in \{1, 2 \ldots, n\}$ indicates where to update the stream,
- $\star$ and $c_t \in \mathbb{R}$ indicates the quantity of update.

In time step $t - 1$ we do an (i_t, c_t) update.
After the (i_t, c_t) update:

$$\text{Let } a(t) \doteq \begin{pmatrix} a_1(t-1) \\ a_2(t-1) \\ \vdots \\ a_{i_t}(t-1) + c_t \\ \vdots \\ a_{n-1}(t-1) \\ a_n(t-1) \end{pmatrix} \in \mathbb{R}^n$$

Query Types

Definition [Point Query]

$Q(i)$: Return an approximation of $a_i(t) \in \mathbb{R}$

$i \in \{1, 2 \ldots, n\}$

Definition [Range Query]

$Q(l, r)$: Return an approximation of $\sum_{i=l}^{r} a_i(t) \in \mathbb{R}$

$l, r \in \{1, 2 \ldots, n\}$

[discussed in the paper, we are not going to discuss it]

Definition [Inner Product Query]

$Q(a, b)$: Given two stream $a(t), b(t) \in \mathbb{R}^n$,

provide an approximation of $\sum_{i=1}^{n} a_i(t) b_i(t) \in \mathbb{R}$

[discussed in the paper, we are not going to discuss it]

Norms

1-norm:

$$\|a\|_1 = \sum_{i=1}^{n} |a_i(t)|$$

p-norm:

$$\|a\|_p = \left(\sum_{i=1}^{n} |a_i(t)|^p \right)^{1/p}$$

Assumptions

- "**Machine words**" can store integers up to $\max\{\|a\|_1, n\}$

$$\text{using } \log(\max\{\|a\|_1, n\}) \text{ bits}$$

- **Space counting**: We count the num of words stored in our data structure (instead of bits)

- **If we need the space count in bits:**

$$\text{num of bits} \leq (\text{num of words}) \times \log(\max\{\|a\|_1, n\})$$

- **Time counting:** We count the operations on words

Goals

Goals:

Develop an (ϵ, δ) approximate and probabilistic algorithm

That is, $\forall \epsilon > 0, \delta > 0$,

$$\mathbb{P}(\text{ error in answering querry Q} > \epsilon) < \delta$$

Since *n* is very big, we also want the algorithm to operate in sublinear space, i.e using only

$$\text{Poly}(\log(\max\{n, \|a\|_1\})) < \text{Linear}(\max\{n, \|a\|_1\})$$

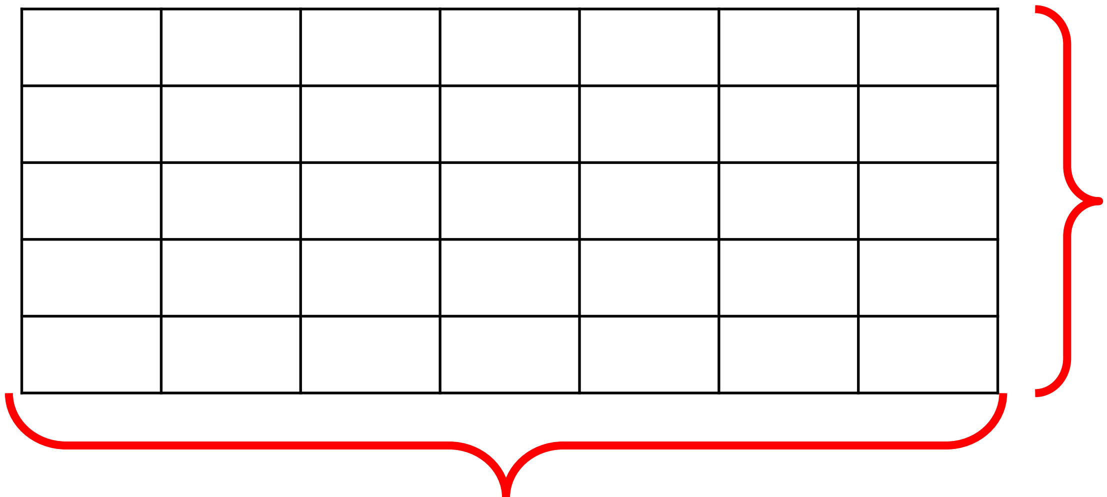
space.

The Count-Min Sketch Data Structure

Definition [CM Data Structure]

The $CM(\epsilon, \delta)$ data structure is represented with a 2D array, called Count matrix

Count=



$d \doteq \left\lceil \log \frac{1}{\delta} \right\rceil$

$(\log = \ln)$

Count $\in \mathbb{R}^{d \times w}$

$w \doteq \left\lceil \frac{e}{\epsilon} \right\rceil$ here $e = \exp(1)$.

The size of $CM(\epsilon, \delta)$ doesn't depend on n.

Pairwise Independent Hash Functions

We are also given *d* hash functions:

$$H = \{h_1, h_2, \ldots, h_d\}$$

$$h_i : \{1, 2, \ldots, n\} \to \{1, 2, \ldots, w\}$$

We assume that *H* is a **pairwise independent** hash family, that is

$$\mathbb{P}_{h \in H}[h(i) = h(j)] \leq \frac{1}{w}, \quad \forall i \neq j \;\; (1 \leq i, j \leq n)$$

uniform

[The probability of collision]

CM Data Structure Update

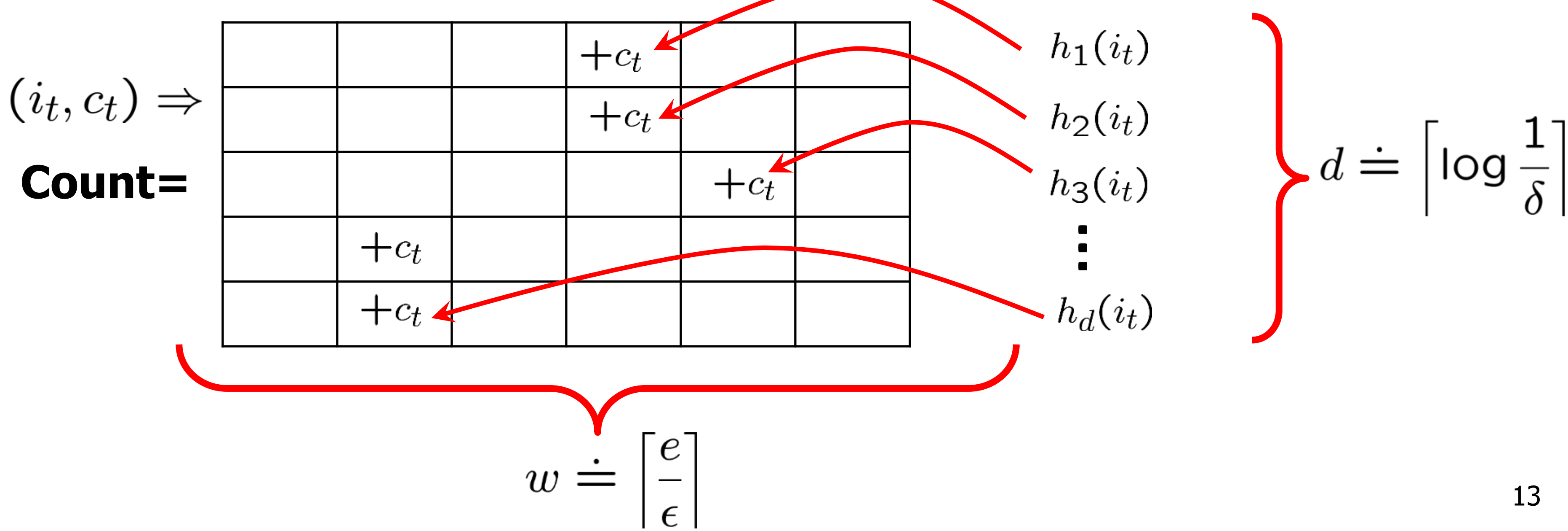
After the (i_t, c_t) update:

$$a_{i_t}(t) = a_{i_t}(t-1) + c_t \qquad a(t) = \begin{pmatrix} a_1(t-1) \\ a_2(t-1) \\ \vdots \\ a_{i_t}(t-1) + c_t \\ \vdots \\ a_{n-1}(t-1) \\ a_n(t-1) \end{pmatrix} \in \mathbb{R}^n$$

CM Update Procedure

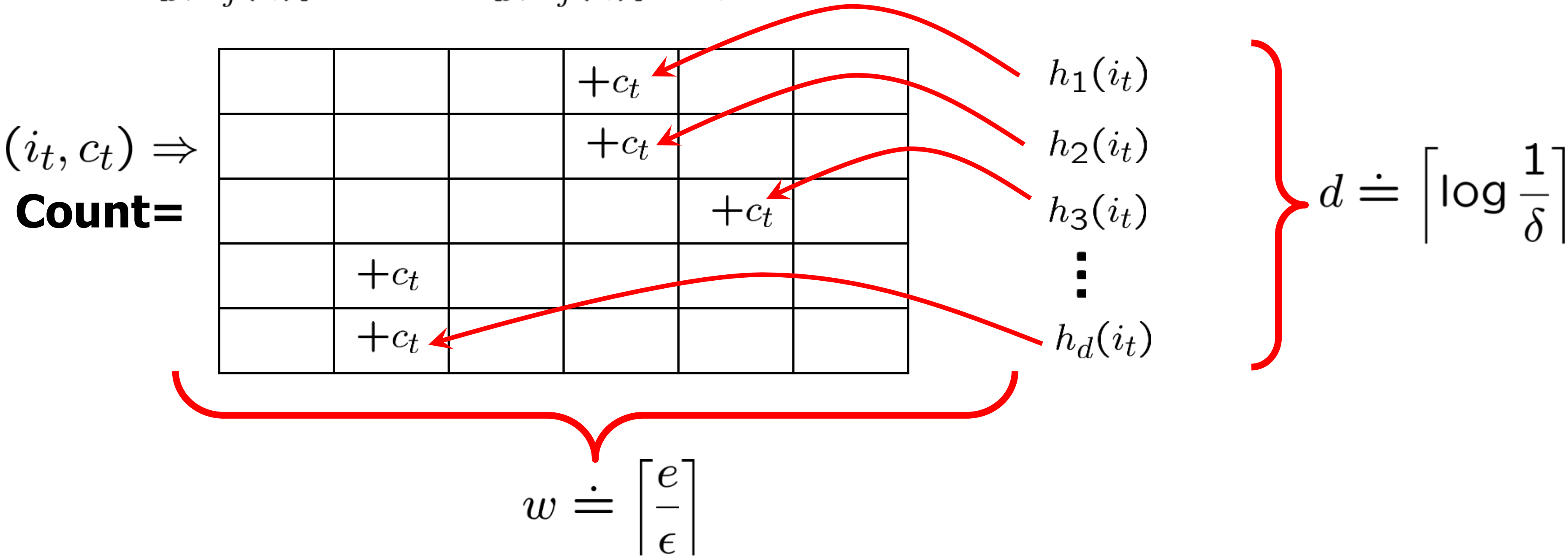
Update $\underbrace{\text{one entry}}_{h_j(i_t)}$ in each $\underbrace{\text{row}}_{j}$ of the Count sketch matrix.

$$Count[j, h_j(i_t)] := Count[j, h_j(i_t)] + c_t$$



CM Data Structure Update

$$Count[j, h_j(i_t)] := Count[j, h_j(i_t)] + c_t$$



For each i_t, cells in a specific pattern (dependening on $h_1, \ldots, h_d$) get increased by c_t.

Point Query Approximation

Point Query:

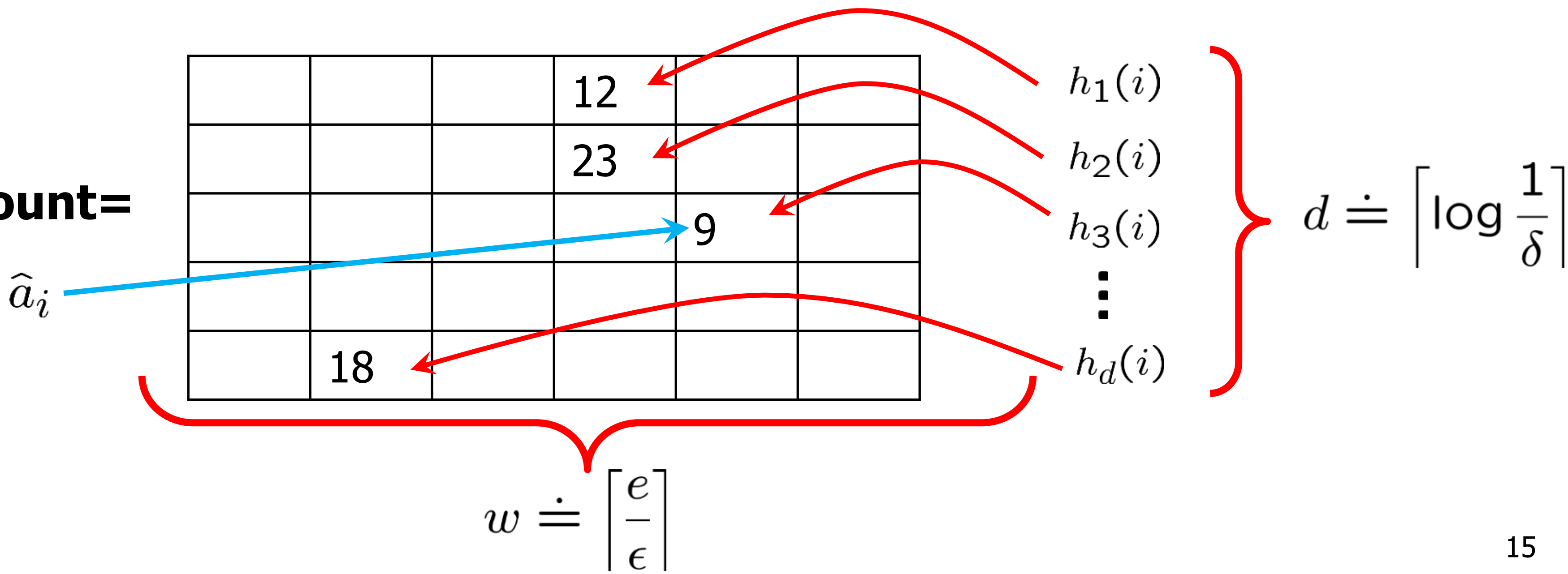
$$Q(i)\text{: Return an approximation of } a_i(t) \in \mathbb{R}$$

$$i \in \{1, 2 \ldots, n\}$$

Approximation:

$$\text{Let } \hat{a}_i \doteq \min_{1 \le j \le d} Count[j, h_j(i)] \quad i \in \{1, 2 \ldots, n\}$$

Count=



Point Query Approximation

$$\hat{a}_i \doteq \min_{1 \le j \le d} Count[j, h_j(i)]$$

Theorem 1

Let $c_t \geq 0$.

$\hat{a}_i$ has the following approximation guarantees

- $\star$ $a_i \leq \hat{a}_i \quad \forall i \in \{1, 2 \ldots, n\}$
- $\star$ $\hat{a}_i \leq a_i + \epsilon \|a\|_1$ with prob. at least $1 - \delta \quad \forall i \in \{1, 2 \ldots, n\}$

Storage cost: $wd = \left\lceil \frac{e}{\epsilon} \right\rceil \left\lceil \log \frac{1}{\delta} \right\rceil$ words

Comput time complexity: $O(d) = O\left(\left\lceil \log \frac{1}{\delta} \right\rceil\right)$

Example

We have 4 hash functions: $h_1, \ldots, h_4$, and $n = 3$

Collision matrix

		2	1	3
		2	13	
		1	23	
				123

Hash Functions

			1		$h_1(1)$
			1		$h_2(1)$
		1			$h_3(1)$
				1	$h_4(1)$

		2			$h_1(2)$
		2			$h_2(2)$
			2		$h_3(2)$
				2	$h_4(2)$

				3	$h_1(3)$
			3		$h_2(3)$
			3		$h_3(3)$
				3	$h_4(3)$

Example

Let the stream be

$$
\begin{array}{c}
a_1 \\ a_2 \\ a_3
\end{array}
\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}
\overset{(2, c_1) \Rightarrow}{}
\begin{pmatrix} 0 \\ c_1 \\ 0 \end{pmatrix}
\overset{(1, c_2) \Rightarrow}{}
\begin{pmatrix} c_2 \\ c_1 \\ 0 \end{pmatrix}
\overset{(2, c_3) \Rightarrow}{}
\begin{pmatrix} c_2 \\ c_1 + c_3 \\ 0 \end{pmatrix}
\overset{(3, c_4) \Rightarrow}{}
\begin{pmatrix} c_2 \\ c_1 + c_3 \\ c_4 \end{pmatrix}
$$

$$t = 0 \qquad t = 1 \qquad t = 2 \qquad t = 3 \qquad t = 4$$

Let the us calculate the values in the CM sketch matrix

Example

Stream updates:

$$\begin{array}{c} a_1 \\ a_2 \\ a_3 \end{array}\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \xrightarrow{(2, c_1)} \begin{pmatrix} 0 \\ c_1 \\ 0 \end{pmatrix}$$

$t = 0$ $\qquad$ $t = 1$

Hash functions

		2			$h_1(2)$
		2			$h_2(2)$
			2		$h_3(2)$
				2	$h_4(2)$

Count=

		c1		
		c1		
			c1	
				c1

Collision matrix

		2	1	3
		2	13	
		1	23	
				123

Example

Stream updates:

$$\begin{array}{c} a_1 \\ a_2 \\ a_3 \end{array} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \xrightarrow{(2, c_1) \Rightarrow} \begin{pmatrix} 0 \\ c_1 \\ 0 \end{pmatrix} \xrightarrow{(1, c_2) \Rightarrow} \begin{pmatrix} c_2 \\ c_1 \\ 0 \end{pmatrix}$$

$t = 0$ $\quad$ $t = 1$ $\quad$ $t = 2$

Hash functions

			1		$h_1(1)$
			1		$h_2(1)$
		1			$h_3(1)$
				1	$h_4(1)$

Count=

		c1	c2	
		c1	c2	
		c2	c1	
				c1+c2

Collision matrix

		2	1	3
		2	13	
		1	23	
				123

Example

Stream updates:

$$
\begin{array}{c}a_1\\a_2\\a_3\end{array}\begin{pmatrix}0\\0\\0\end{pmatrix} \xRightarrow{(2,c_1)} \begin{pmatrix}0\\c_1\\0\end{pmatrix} \xRightarrow{(1,c_2)} \begin{pmatrix}c_2\\c_1\\0\end{pmatrix} \xRightarrow{(2,c_3)} \begin{pmatrix}c_2\\c_1+c_3\\0\end{pmatrix}
$$

$$t=0 \qquad t=1 \qquad t=2 \qquad t=3$$

Hash functions

		2			$h_1(2)$
		2			$h_2(2)$
			2		$h_3(2)$
				2	$h_4(2)$

Count=

		c1+c3	c2	
		c1+c3	c2	
		c2	c1+c3	
				c1+c2 +c3

Collision matrix

		2	1	3
		2	13	
		1	23	
				123

Example

Stream updates:

$$\begin{pmatrix} c_2 \\ c_1 \\ 0 \end{pmatrix} \overset{(2, c_3) \Rightarrow}{} \begin{pmatrix} c_2 \\ c_1 + c_3 \\ 0 \end{pmatrix} \overset{(3, c_4) \Rightarrow}{} \begin{pmatrix} c_2 \\ c_1 + c_3 \\ c_4 \end{pmatrix}$$

$$t = 2 \qquad t = 3 \qquad t = 4$$

				3	$h_1(3)$
			3		$h_2(3)$
			3		$h_3(3)$
				3	$h_4(3)$

Count=

		c1+c 3	c2	c4
		c1+c 3	c2+c4	
		c2	c1+c3+ c4	
				c1+c2+ c3+c4

Collision matrix

		2	1	3
		2	13	
		1	23	
				123

Example

Hash Functions

			1		$h_1(1)$
			1		$h_2(1)$
		1			$h_3(1)$
				1	$h_4(1)$

		2			$h_1(2)$
		2			$h_2(2)$
			2		$h_3(2)$
				2	$h_4(2)$

				3	$h_1(3)$
			3		$h_2(3)$
			3		$h_3(3)$
				3	$h_4(3)$

Count=

		c1+c3	c2	c4
		c1+c3	c2+c4	
		c2	c1+c3+ c4	
				c1+c2+ c3+c4

$$t = 4$$

$$\begin{array}{c} a_1 \\ a_2 \\ a_3 \end{array} \begin{pmatrix} c_2 \\ c_1 + c_3 \\ c_4 \end{pmatrix}$$

$$\hat{a}_i \doteq \min_{1 \leq j \leq d} Count[j, h_j(i)]$$

$$\hat{a}_1 = \min\{c_2, c_2 + c_4, c_2, c_1 + c_2 + c_3 + c_4\}$$
$$= c_2$$

$$\hat{a}_2 = \min\{c_1 + c_3, c_1 + c_3, c_1 + c_3 + c_4, c_1 + c_2 + c_3 + c_4\}$$
$$= c_1 + c_3$$

$$\hat{a}_3 = \min\{c_4, c_2 + c_4, c_1 + c_3 + c_4, c_1 + c_2 + c_3 + c_4\}$$
$$= c_4$$

Point Query Approximation v2

Theorem 2

Let $c_t \in \mathbb{R}$. It doesn't need to be nonnegative anymore.

Let $\hat{a}_i \doteq \text{median}_{1 \leq j \leq d} Count[j, h_j(i)]$ (Instead of min)

$\hat{a}_i$ has the following approximation guarantees

$a_i - 3\epsilon\|a\|_1 \leq \hat{a}_i \leq a_i + 3\epsilon\|a\|_1$ with prob. at least $1 - \delta^{1/4}$

$$\forall i \in \{1, 2 \ldots, n\}$$

Thanks for your Attention! ☺