Streaming Algorithms

- Streaming Model of Computation
- Streaming Algorithms and DFA
- Stream: Motivation and Applications
- Synopsis: Sliding Window, Histogram, Wavelets
- Sampling from Stream: Reservoir Sampling
- Linear Sketch
- Count-Min Sketch
- AMS Sketch

Imdad ullah Khan

Count-Min Sketch

Count-min sketch

Count-min sketch stores frequencies of random groups of elements

$\triangleright$ Cormode & Muthukrishnan (2004)

$$\mathcal{S} \; = < a_1, a_2, a_3, \ldots, a_m > \qquad a_i \in [n]$$

f_j : frequency of j in $\mathcal{S}$ $\qquad$ $\mathbf{F} \; = \; (f_1, f_2, \ldots, f_n)$

- Cannot store frequency of every element $j \in [n]$
- Store total frequency of random groups in $[n]$ (elements in hash buckets)

Algorithm : Count-Min Sketch (k, ϵ, δ)

$\text{COUNT} \leftarrow \text{ZEROS}(k)$ $\qquad$ $\triangleright$ sketch consists of k integers

Pick a random $h : [n] \mapsto [k]$ from a 2-universal family $\mathcal{H}$

On input a_i

$\quad \text{COUNT}[h(a_i)] \leftarrow \text{COUNT}[h(a_i)] + 1$ $\qquad$ $\triangleright$ increment count at index $h(a_i)$

On query j $\qquad$ $\triangleright$ query: $\mathbf{F}[j] =?$

$\quad$ **return** $\text{COUNT}[h(j)]$

Count-min sketch

Count-min sketch stores frequencies of random groups of elements

Algorithm : Count-Min Sketch (k, ϵ, δ)

$\text{COUNT} \leftarrow \text{ZEROS}(k)$ ▷ sketch consists of k integers

Pick a random $h : [n] \mapsto [k]$ from a 2-universal family $\mathcal{H}$

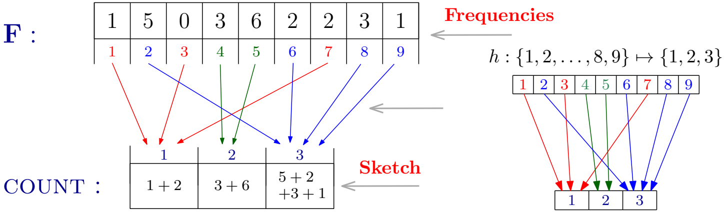
On input a_i

  $\text{COUNT}[h(a_i)] \leftarrow \text{COUNT}[h(a_i)] + 1$ ▷ increment count at index $h(a_i)$

On query j ▷ query: $\mathbf{F}[j] = ?$

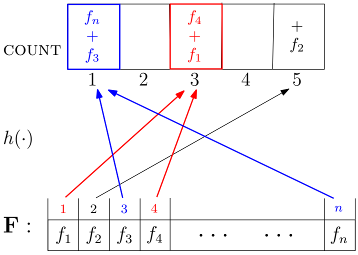
  return $\text{COUNT}[h(j)]$

$\mathcal{S}$: 2, 5, 6, 7, 8, 2, 1, 2, 7, 5, 5, 4, 2, 8, 8, 9, 5, 6, 4, 4, 2, 5, 5



Count-min sketch

Count-min sketch stores frequencies of random groups of elements

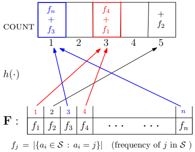


$f_j = |\{a_i \in \mathcal{S} : a_i = j\}|$ (frequency of j in $\mathcal{S}$)

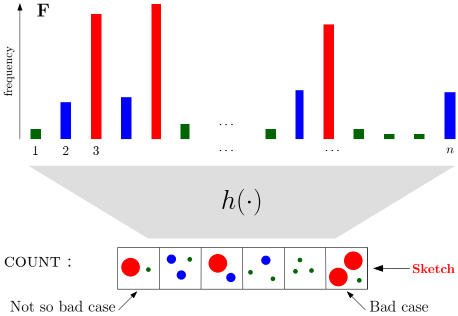
- $k = {}^{2}/_{\epsilon}$
- Large k means better estimate (many smaller groups) but more space
- **$\tilde{f}_j$: estimate for f_j – output of algorithm**

Count-min sketch stores frequencies of random groups of elements

- $k = 2/\epsilon$
- Large k means better estimate but more space
- $\tilde{f}_j$: estimate for f_j – output of algorithm

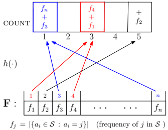


Quality on $\tilde{f}_j$:



Count-min sketch stores frequencies of random groups of elements

- $k = {}^{2}/_{\epsilon}$
- Large k means better estimate but more space
- $\tilde{f}_j$: estimate for f_j – output of algorithm



Quality on $\tilde{f}_j$:

1. $\tilde{f}_j \geq f_j$
   - Other elements that hash to $h(j)$ contribute to $\tilde{f}_j$
2. $Pr\left[\, \tilde{f}_j \leq f_j + \epsilon \|F\|_1 \,\right] \geq {}^{1}/_{2}$
   - $\tilde{f}_j = f_j + X_j$ ▷ X_j: Excess in $\tilde{f}_j$ (error)
   - $X_j = \sum_{i \in [n] \setminus j} f_i \cdot 1_{h(i)=h(j)}$ ▷ $1_{condition}$ is indicator of condition

$$\mathbb{E}\,(X_j) = \mathbb{E}\Bigg(\sum_{i \in [n] \setminus j} f_i \cdot 1_{h(i)=h(j)} \Bigg) = \sum_{i \in [n] \setminus j} f_i \cdot \frac{1}{k} \leq \|F\|_1 \cdot \frac{\epsilon}{2}$$

   - By Markov inequality we get the bound

Amplifying the probability of basic Count-Min Sketch

Keep t over-estimates, $t = \log(1/\delta)$, $k = 2/\epsilon$ and return their minimum

Unlikely that all t functions hash j with very frequent elements

---

Algorithm : Count-Min Sketch (k, ϵ, δ)

$\textsc{count} \leftarrow \textsc{zeros}(t \times k)$ ▷ sketch consists of t rows of k integers

Pick t random functions $h_1, \ldots, h_t : [n] \mapsto [k]$ from a 2-universal family

On input a_i

for $r = 1$ to t **do**

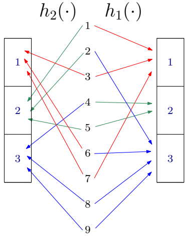
$\textsc{count}[r][h_r(a_i)] \leftarrow \textsc{count}[r][h_r(a_i)] + 1$

▷ increment $\textsc{count}[r]$ at index $h_r(a_i)$

On query j ▷ query: $\mathbf{F}[j] = ?$

return $\underset{1 \le r \le t}{\textsc{min}} \ \textsc{count}[r][h_r(j)]$

Amplifying the probability of basic Count-Min Sketch



$\mathcal{S}$: 2, 5, 6, 7, 8, 2, 1, 2, 7, 5, 5, 4, 2, 8, 8, 9, 5, 6, 4, 4, 2, 5, 5

$h_1(\cdot)$

COUNT :

1	2	3
$0+1+2$	$3+6$	$5+2$ $+3+1$
$0+2+2$	$1+5+6$	$3+3+1$

$h_2(\cdot)$

Sketch

F :

1	2	3	4	5	6	7	8	9
1	5	0	3	6	2	2	3	1

True Frequencies

On input a

$h_1(a)$ $h_t(a)$

	1	2	3	...	...	k
COUNT[1][·]		+1				
COUNT[2][·]					+1	
COUNT[3][·]				+1		
⋮						
COUNT[t][·]						+1

On query a $\quad$ $\text{MIN}_i \ \text{COUNT}[\,i\,][\,h_i(a)]$

Count-min sketch

Amplifying the probability of basic Count-Min Sketch

1. $\tilde{f}_j \;\geq\; f_j$
   - For every r, other elements that hash to $h_r(j)$ contribute to $\tilde{f}_j$
2. $Pr\big[\; \tilde{f}_j \;\leq\; f_j + \epsilon\|F\|_1 \;\big] \;\geq\; 1 - \delta$
   - X_{jr} : contribution of other elements to $Count[r][h_r(j)]$
   - $\Pr\left[X_{jr} \geq \epsilon\|F\|_1\right] \;\leq\; 1/2 \quad$ for $k = 2/\epsilon$
   - The event $\tilde{f}_j \;\geq\; f_j + \epsilon\|F\|_1 \quad$ is $\;\forall\; 1 \leq r \leq t \quad X_{jr} \;\geq\; \epsilon\|F\|_1$
   - $\Pr\left[\; \forall\, r \;\; X_{jr} \geq \epsilon\|F\|_1 \;\right] \;\leq\; (1/2)^t$
   - $t = \log(1/\delta) \;\implies\; \Pr\left[\; \forall\, r\; X_{jr} \geq \epsilon\|F\|_1 \;\right] \;\leq\; \left(\frac{1}{2}\right)^{\log 1/\delta} \;=\; \delta$

- Count-Min sketch is an $(\epsilon\|F\|_1, \delta)$-<u>additive</u> approximation algorithm
- Space required is $k \cdot t$ integers $= O(1/\epsilon \log(1/\delta) \log n)$ (plus constant)