CountMin Sketch: Finding Heavy Hitters

COMPCSI 753: Algorithms for Massive Data

Instructor: Ninh Pham

University of Auckland

Auckland, Aug 24, 2020

Basic definitions

- Let $\mathbf{U}$ be a universe of size $\mathbf{n}$, i.e. $\mathbf{U = \{1, 2, 3, \ldots , n\}}$.

- Cash register model stream:
  - Sequence of $\mathbf{m}$ elements $\mathbf{a_1}, \ldots , \mathbf{a_m}$ where $\mathbf{a_i} \in \mathbf{U}$.
  - Elements of $\mathbf{U}$ may or may not occur once or several times in the stream.

- Finding heavy hitters in data stream (today's lecture):
  - Given a stream, finding frequent items.

Frequent items

- Each element of data stream is a tuple.
- Given a stream of $\mathbf{m}$ elements $\mathbf{a_1}, \ldots, \mathbf{a_m}$ where $\mathbf{a_i} \in \mathbf{U}$, finding the most/top-$\mathbf{k}$ frequent elements.

- Example:
  - $\{\underline{1}, 2, \underline{1}, 3, 4, 5\} \rightarrow \mathbf{f} = \{\underline{2}, 1, 1, 1, 1\}$
  - $\{\underline{1}, \underline{2}, \underline{1}, 3, \underline{1}, \underline{2}, 4, 5, \underline{2}, 3\} \rightarrow \mathbf{f} = \{\underline{3}, \underline{3}, 2, 1, 1\}$

- We need an approximation solution with much smaller memory with theoretical guarantees.

Deterministic: Misra Gries

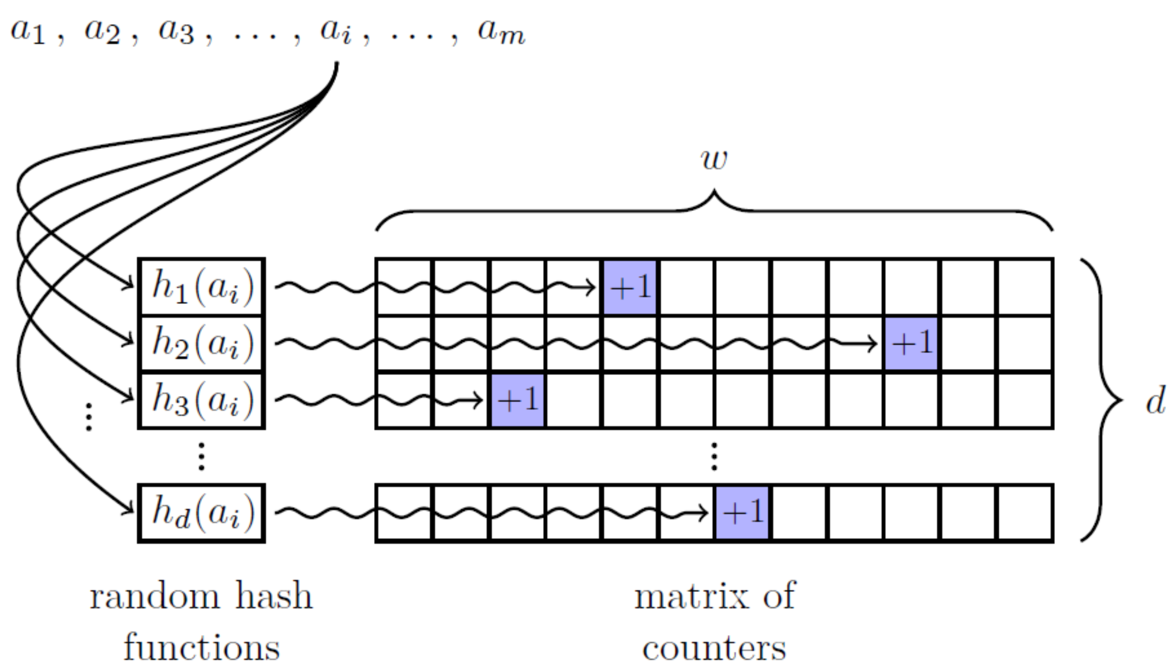
- Process an element **a**:
  - If we already have a counter for **a**, increment it.
  - Else, if there is no counter for **a**, but fewer **k** counters, create a counter for **a** initialized to **1**.
  - Else, decrease all counters by **1**. Remove **0** counters (key step).

- Example: {1, 2 ,3, 1, 4, 2, 1, 4, 5, 2, 6}, **n=6, k=3, m=11**

  {1, 2, 3, 1, 4, 2, 1, 4, 5, 2, 6}

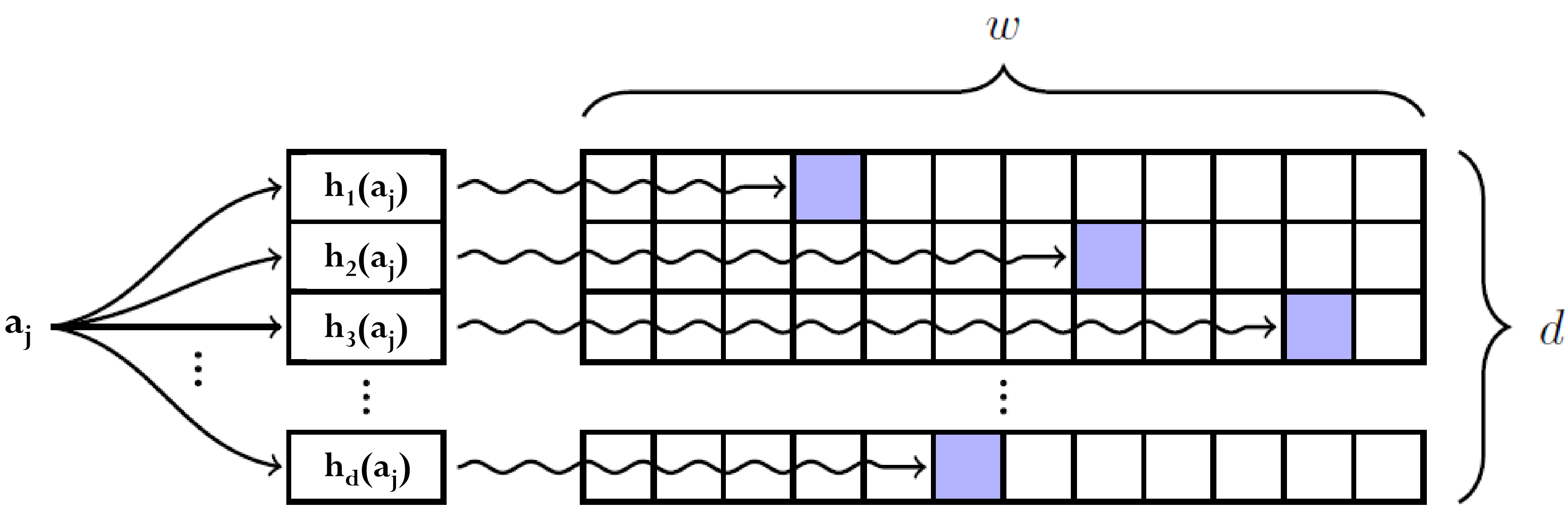
Randomized: CountMin sketch

- Setup:
  - **d** independent **universal** hash functions **h** over range **[0, w)**
  - **d** different counters, $\mathbf{C_1, \ldots, C_d}$. Each of size **w** initialized with **0**s.
- Process an element $\mathbf{a_j}$:
  - For each hash function, compute $\mathbf{h_i(a_j)}$ and increment $\mathbf{C_i[h_i(a_j)]}$ by **1**



Randomized: CountMin sketch

- Query: How many times $\mathbf{a_j}$ occurred?
  - For each hash function, compute $\mathbf{h_i(a_j)}$ and get $\mathbf{C_i[h_i(a_j)]}$
  - Return $\mathbf{min(C_1[h_1(a_j)], \dots , C_d[h_d(a_j)])}$



return the minimum of values in blue cells

Universal hash function family

- Universal hash function definition:
  - A family of hash function $\mathbf{H = \{h : U \rightarrow \{0, 1, \ldots, w\text{-}1\}\}}$ is **universal** if for any **2** distinct keys $\mathbf{x_i \neq x_j \in U}$, we have

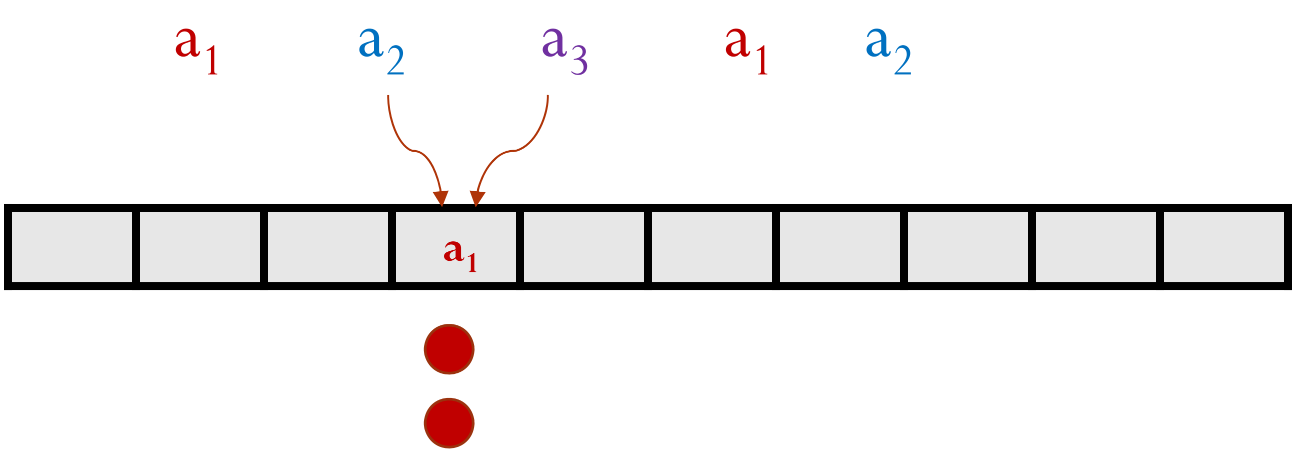
    $$\mathbf{Pr_h[h(x_i) = h(x_j)] \leq 1/w}$$

- In our CountMin sketch:
  - Given two different items $\mathbf{a_i \neq a_j}$, what is the prob. $\mathbf{a_i}$ and $\mathbf{a_j}$ collide?

    $$\mathbf{Pr_h[h(a_i) = h(a_j)] \leq 1/w}$$

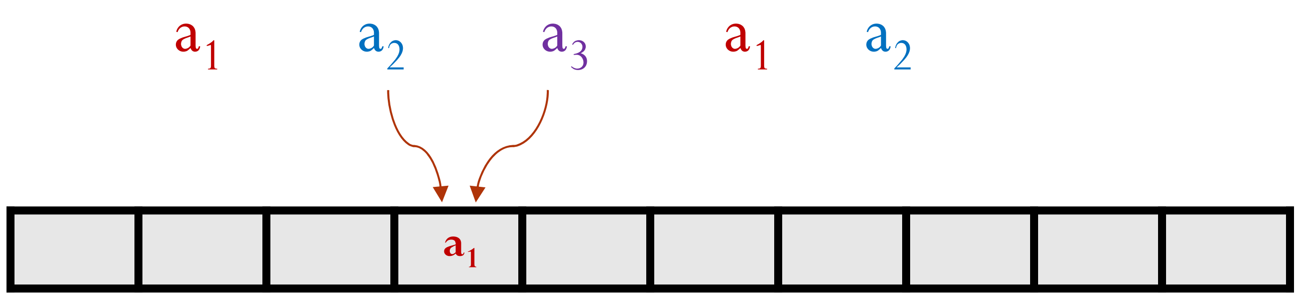
Analysis on 1 array of counters

- Notation:
  - Stream of **m** items $\{\mathbf{a_1, \ldots, a_m}\}$ from the universe **U** of size **n**.
  - Frequency vector $\mathbf{f} = \{\mathbf{f_1, \ldots, f_n}\}$ and $\|\boldsymbol{f}\|_1 = \mathbf{m}$.
- Question:
  - Given a particular item $\mathbf{a_1}$, how many times $\mathbf{a_i} \neq \mathbf{a_1}$ collide by **h**?



Analysis on 1 array of counters

- Question:
  - Given a particular item $\mathbf{a_1}$, how many times $\mathbf{a_i} \neq \mathbf{a_1}$ collide by $\mathbf{h}$.



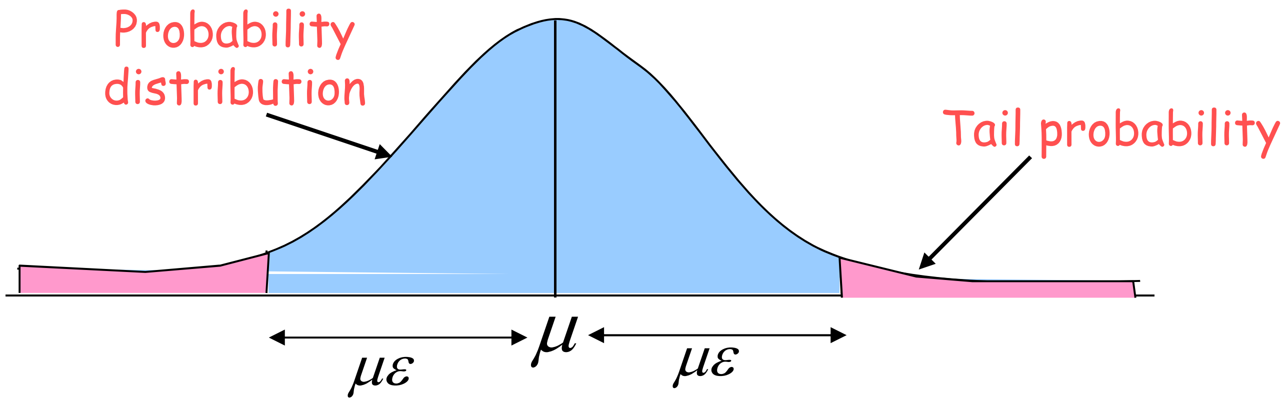
- Answer:
  - Let $\mathbf{X_2, \ldots, X_n}$ be contributions of $\mathbf{a_2, \ldots, a_n}$ in the bucket $\mathbf{h(a_1)}$.

$$\mathbf{Pr\ [h(a_i) = h(a_1)] \leq 1/w \rightarrow E\ [X_i] \leq f_i/w}$$

  - Let $\mathbf{Y = X_2 + \ldots + X_n}$ be the total increments by other item $\mathbf{a_i} \neq \mathbf{a_1}$.

$$\mathbf{E[Y] = E[X_2] + \ldots + E[X_n] = (f_2 + \ldots + f_n)/w \leq m/w}$$

Basic tools: Tail inequality



- Markov's inequality for **E[Y] = μ**:

$$\Pr[Y \geq 1+\varepsilon] \leq \frac{\mu}{1+\varepsilon} \text{ for any } \varepsilon > 0.$$

$$\Pr[Y \geq (1+\varepsilon)\,\mu] \leq \frac{1}{1+\varepsilon} \text{ for any } \varepsilon > 0.$$

Analysis on 1 array of counters

- Observation: We always over-estimate.
- Question: How large we over estimate?

- Analysis for a particular item $\mathbf{a_1}$:
  - Let $\mathbf{Y = X_2 + \ldots + X_n}$ be the total increments by other item $\mathbf{a_i \neq a_1}$.
  - The value of counter $\mathbf{h(a_1)}$: $\mathbf{f'_1 = f_1 + Y}$.
  - Using Markov's inequality:

  $$\mathbf{Pr\,[f'_1 \geq f_1 + \varepsilon m] = Pr\,[Y \geq \varepsilon m] \leq E[Y] / \varepsilon m \leq m/(w * \varepsilon m)}$$

  - Choose $\mathbf{w = 2/\varepsilon}$, we have this prob. is at most $\mathbf{1/2}$.

- Choose $\mathbf{w = 2/\varepsilon}$, the probability that our error is larger than $\mathbf{\varepsilon m}$ is smaller than $\mathbf{1/2}$.

Analysis on d arrays of counters

- Boosting the accuracy:
  - Using **d** independent hash functions corresponding to **d** independent arrays of counters.
  - $\mathbf{F_1 = \min(C_1[h_1(a_1)], \ldots , C_d[h_d(a_1)]) = \min(f'_1, f'_2, \ldots , f'_d)}$.
- Analysis:
  - $\mathbf{Pr\ [\ F_1 \geq f_1 + \varepsilon m] \ = Pr\ [f'_1 \geq f_1 + \varepsilon m \wedge \ldots \wedge f'_d \geq f_1 + \varepsilon m] \leq 1/2^d}$
  - Choose $\mathbf{d = \log(1/\delta)}$, we have $\mathbf{Pr\ [\ F_1 \geq f_1 + \varepsilon m] \ \leq \delta}$.
- With probability at least $\mathbf{1 - \delta}$, we have

$$\mathbf{F_1 < f_1 + \varepsilon m = f_1 + \varepsilon \|\boldsymbol{f}\|_1}$$

Homework

- Implement the CountMin Sketch algorithm on the dataset from assignment 1:
  - Description: Each line (doc ID, word ID, freq.) as a stream tuple.
  - Query: What are the most and top-**10** frequent word ID have been used?