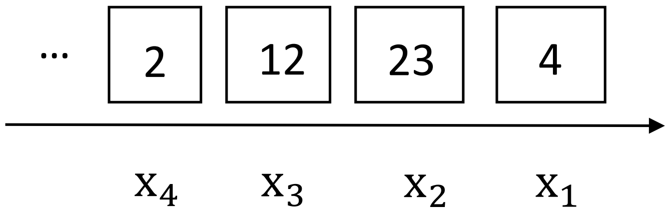
SDSC3001 Tutorial 6

Count-Min Sketch

2024.11.7

Background

- Stream Data: Element x_t arrives at time t.

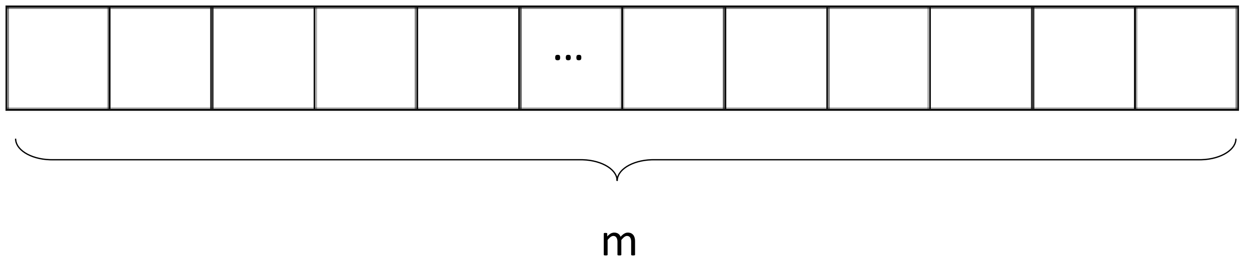


- Task: Count the number of times elements appear in a stream of data.

Background

- A naive solution: maintain a count array that maps elements to their frequencies.
- There are n data points, which are numbers ranging from 1 to m.

Count array:

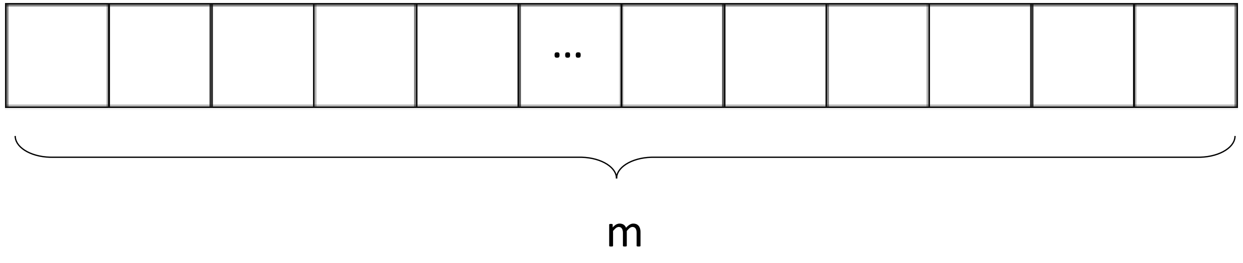


- Space complexity is O(m), Time complexity is O(n)

Why do we use hash?

Data are tel-numbers with 8 digits.

- Space complexity is O(m). m=10^8.
- Not every number in the range [1, 10^8] is a valid telephone number! We're wasting memory!



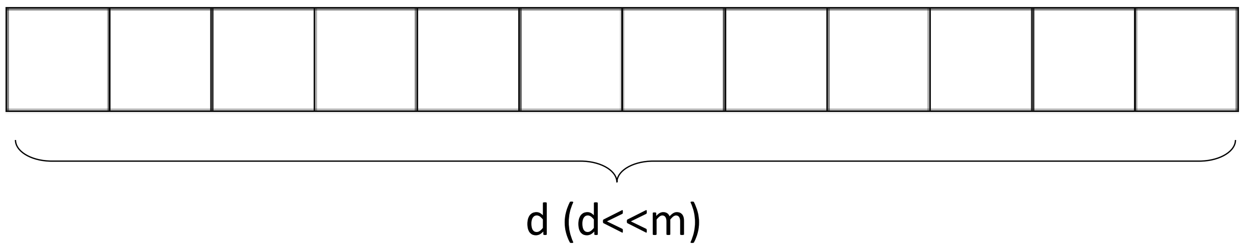
Why do we use hash?

Data are email addresses.

- m=? (An email address can be arbitrarily long, and we may not know the whole set of address)
- [“Alice@cityu.edu.hk” , “Bob@cityu.edu.hk” , ….]

Why do we use hash?

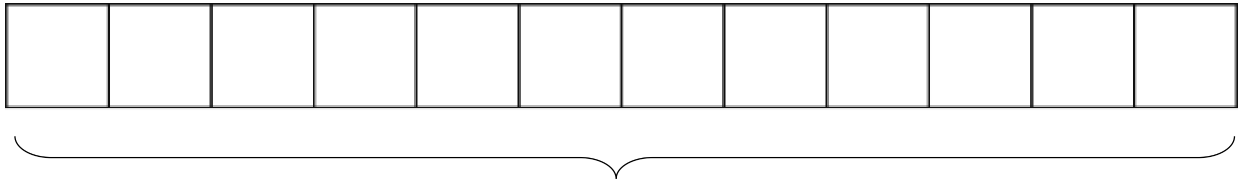
Approximate Counts with Hashing: given that we only have limited space availability.



- i = hash("Alice@cityu.edu.hk"), where hash() represents a hash function.
- Count[i] = Count[i] + 1.
- Always O(1) time to search where we count.
- Time complexity is O(n).

Why do we use hash?

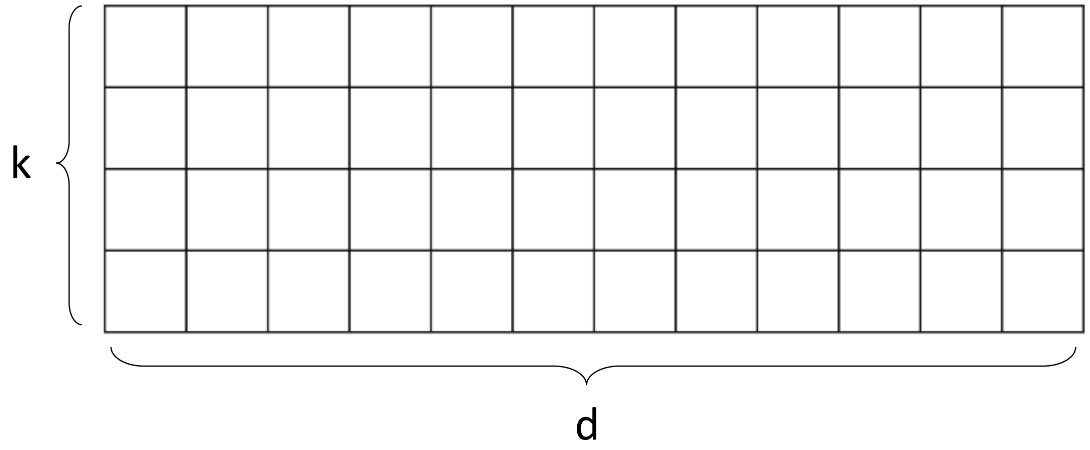
Approximate Counts with Hashing: given that we only have limited space availability.



d (d<<m)

Drawback: hash conflicts.
Solution: use more hash functions

Use k hash functions



- Initialization: count$[i,\ j]$=0 $\forall i \in [k], \forall j \in [d]$
- Increment count (of element **a**): count$[i, h_i($**a**$)]$+=1, $\forall i \in [k]$
- Retrieve count (of element **a**): $min_{i \in [k]}$ count$[i, h_i($**a**$)]$
- Sketch: use relatively finite statistics information to represent the all stream data.

Example

	0	1	2
h_1(x)= x mod 3			
h_2(x)= 2x mod 3			
h_3(x)= x^2 mod 3			

Example

	0	1	2
h_1(x)= x mod 3		1	
h_2(x)= 2x mod 3			1
h_3(x)= x^2 mod 3		1	

… 6 4 2 5 3 4

Example

	0	1	2
h_1(x)= x mod 3	1	1	
h_2(x)= 2x mod 3	1		1
h_3(x)= x^2 mod 3	1	1	

… | 6 | 4 | 2 | 5 | 3 | 4 | →

Example

	0	1	2
h_1(x)= x mod 3	1	3	2
h_2(x)= 2x mod 3	1	3	2
h_3(x)= x^2 mod 3	1	5	0

… 5 4 2 5 3 4

Example

A=

	0	1	2	
	1	3	2	h_1(x)= x mod 3
	1	3	2	h_2(x)= 2x mod 3
	1	5	0	h_3(x)= x^2 mod 3

If we want to know the frequency of number 5:

- h_1(5)=2, h_2(5)=h_3(5)=1.
- Count(5)=min{A[i, h_i(5)]}=min{2, 3, 5}=2

Algorithm

Count-Min Sketch

- We sample hash functions $h_1, h_2, \ldots, h_k$ independently and uniformly at random from a universal hashing family
  - $h_i : [m] \to [d]$, $\Pr\{h_i(s_1) = h_i(s_2)\} \leq \frac{1}{d}$ for $s_1 \neq s_2$
  - We will figure out k and d later
- When a_t comes, $\mathrm{count}_i[h_i(a_t)]{+}{+}$ for all $i = 1, \ldots, k$
- When estimating f_s^t, return $\mathrm{est}_s^t = \min\{\mathrm{count}_1[h_1(s)], \ldots, \mathrm{count}_k[h_k(s)]\}$

Algorithm

- $\text{est}_s^t \geq f_s^t$ is trivial
- $h_1, \ldots, h_k$ are independent to each other
  - $\Pr\{\text{est}_s^t \geq f_s^t + \epsilon t\} \leq (\frac{1}{\epsilon d})^k$
- Set $k = 2/\epsilon$ and $d = \log \frac{1}{\delta} \Rightarrow (\frac{1}{\epsilon d})^k = \delta$
  - With probability at least $1 - \delta$, $f_s^t \leq \text{est}_s^t \leq f_s^t + \epsilon t$