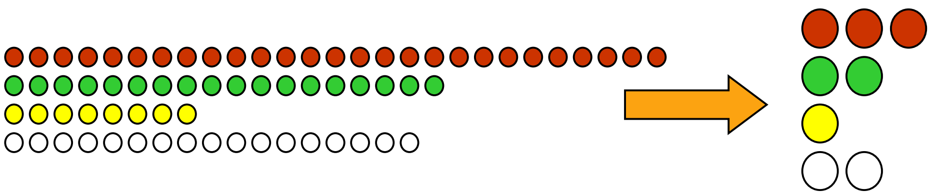
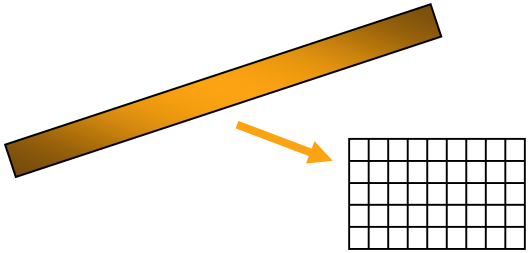
Data Sketches





Lecturer: Jiaheng Lu

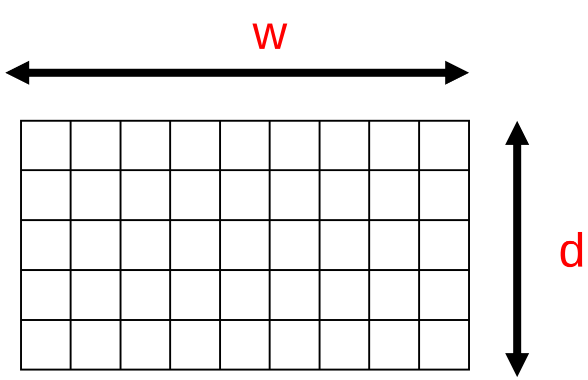
Autumn 2016

Outline

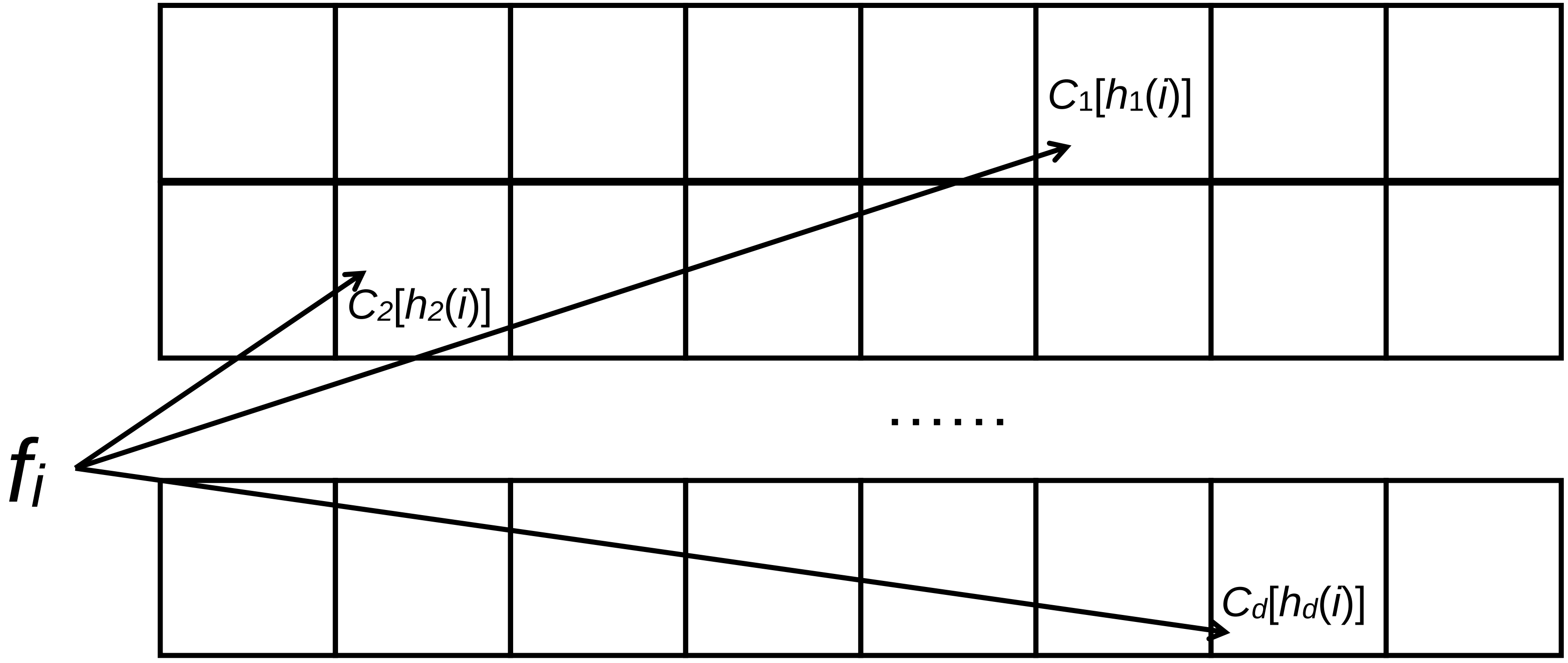
- Massive data stream
- Bloom filter
- Count-min
- Count-sketch
- FM-sketch

Review Count-Min Sketch

- Model input data as a matrix
  - Create a small summary as an array of $w \times d$ in size
  - Use d hash function to map vector entries to $[1..w]$



Review Count-Min Sketch



Count-sketch: Dot product of two vectors

- Problem: Estimate the value of dot product of two vectors
- Example: V1: (1,0,1,2,0) and V2 (0,0,2,1,0)
- V1 • V2 = 0+0+2+2+0 = 4

Use Count-min for computing dot product?

- Map each vector to a $w \times d$ matrix
- Select the min value of the dot product

- See an example to illustrate the method

Use Count-min for computing dot product? See an example.

V1: (1,0,1,2,0), Index (0,1,2,3,4)

Initial

0	0	0
0	0	0
0	0	0

h1(j) $= j \ mod \ 3$
h2(j) $= (j \ mod \ 4) \ mod \ 3$
h3(j) $= (2*j) \ mod \ 3$

Use Count-min for computing dot product? See an example.

V1: (1,0,1,2,0), Index (0,1,2,3,4)

Initial

1	0	0
1	0	0
1	0	0

h1(j) = $j \ mod \ 3$
h2(j) = $(j \ mod \ 4) \ mod \ 3$
h3(j) = $(2*j) \ mod \ 3$

Use Count-min for computing dot product? See an example.

V1: (1,0,1,2,0), Index (0,1,2,3,4)

Initial

1	0	1
1	0	1
1	1	0

h1(j) = $j \; mod \; 3$
h2(j) = $(j \; mod \; 4) \; mod \; 3$
h3(j) = $(2*j) \; mod \; 3$

Use Count-min for computing dot product? See an example.

V1: (1,0,1,2,0), Index (0,1,2,3,4)

Initial

3	0	1
3	0	1
3	1	0

h1(j) = $j \ mod \ 3$
h2(j) = $(j \ mod \ 4) \ mod \ 3$
h3(j) = $(2*j) \ mod \ 3$

Use Count-min for computing dot product? See an example.

V2 (0,0,2,1,0), Index (0,1,2,3,4)

Initial

0	0	0
0	0	0
0	0	0

h1(j) = $j\ mod\ 3$
h2(j) = $(j\ mod\ 4)\ mod\ 3$
h3(j) = $(2*j)\ mod\ 3$

Use Count-min for computing dot product? See an example.

V2 (0,0,2,1,0), Index (0,1,2,3,4)

Initial

0	0	2
0	0	2
0	2	0

h1(j) = $j \; mod \; 3$
h2(j) = $(j \; mod \; 4) \; mod \; 3$
h3(j) = $(2*j) \; mod \; 3$

Use Count-min for computing dot product? See an example.

V2 (0,0,2,1,0), Index (0,1,2,3,4)

Initial

1	0	2
1	0	2
1	2	0

h1(j) = $j \ mod \ 3$
h2(j) = $(j \ mod \ 4) \ mod \ 3$
h3(j) = $(2*j) \ mod \ 3$

Use Count-min for computing dot product? See an example.

V1: (1,0,1,2,0), V2: (0,0,2,1,0),

3	0	1
3	0	1
3	1	0

1	0	2
1	0	2
1	2	0

$3 \times 1 + 1 \times 2 = 5$
$3 \times 1 + 1 \times 2 = 5$
$3 \times 1 + 1 \times 2 = 5$

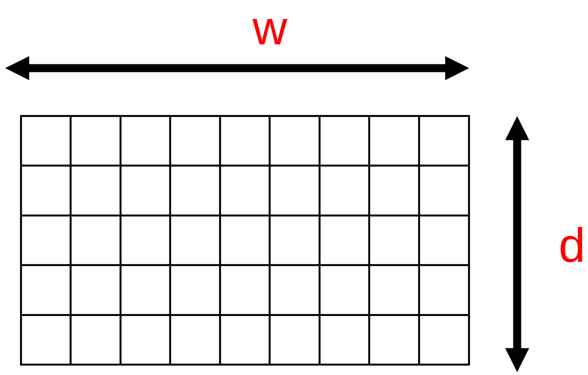
Min (5,5,5) = 5

The accurate value is 2+2=4.
Over-estimation!

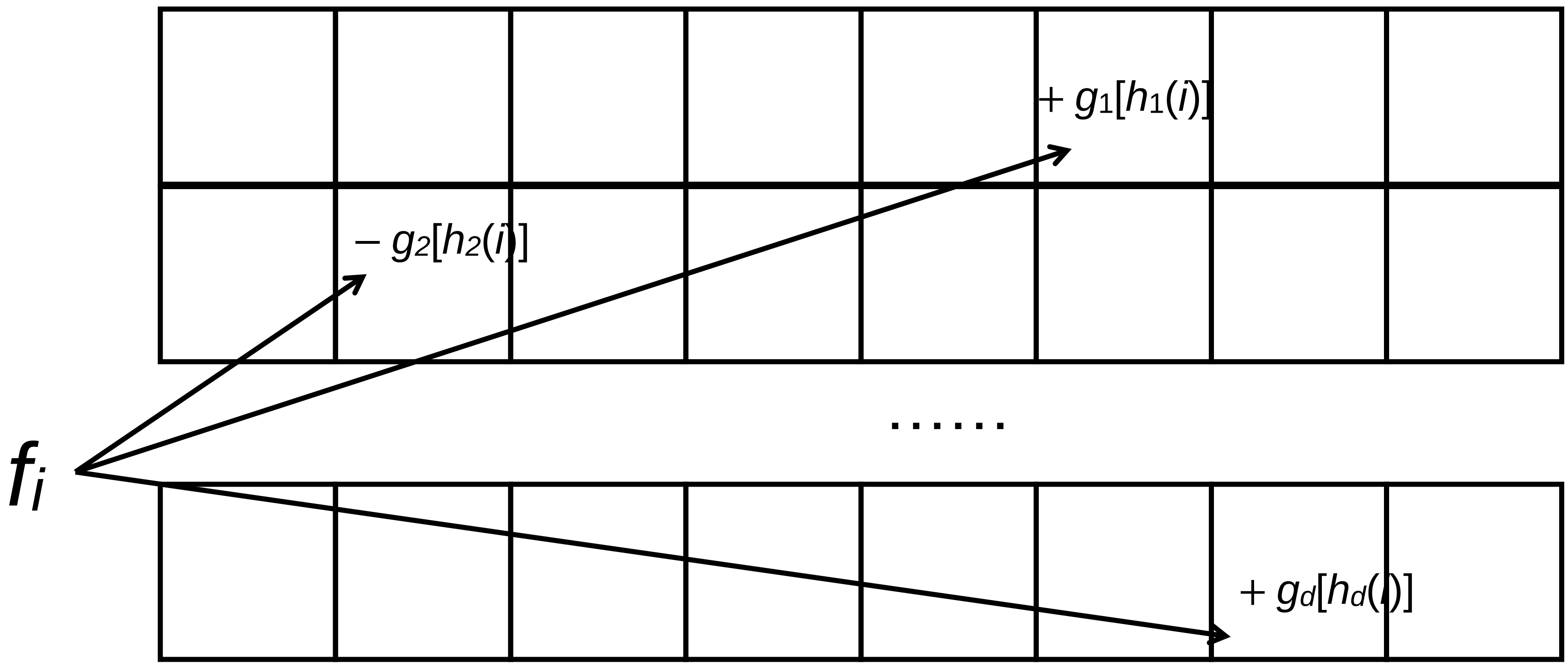
Count-Sketch

- Model input vector $V = (v_0, v_1, \ldots, v_{n-1})$ as a matrix
  - Create a small summary as a matrix of $w \times d$ in size
  - Use d hash function pairs to map vector entries to $[1..w]$

Two hash
functions h and g



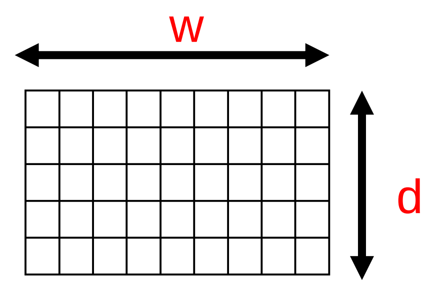
Count sketch



- Second hash function: g_i maps each i to $+1$ or -1

Count-sketch Algorithm

- Input: vector $V=(v_0, v_1, \ldots, v_{n-1})$
- Output: a matrix M of w × d in size
-



1. $C[0, 0] \ldots C[d-1, w-1] = 0$;
2. for $j \leftarrow 0$ to $d-1$ do

   Initialize h_j, g_j;

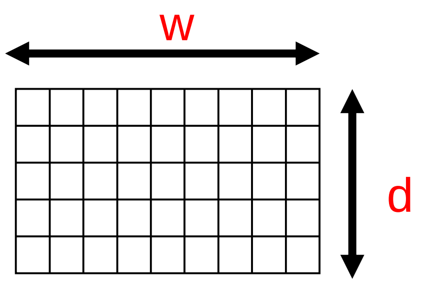
3. for $j \leftarrow 0$ to $d-1$ do

   for $i \leftarrow 0$ to $w-1$

   do $C[j, h_j(i)] \leftarrow C[j, h_j(i)] + g_j(i) * v_i$;

Count-sketch Algorithm (Cont.)

- Input: two matrixes M1 and M2 for two vectors
- Output: the estimation of the dot product of two vectors



for j ← 0 to d-1 do

E(j) = $\sum_{i=0}^{w-1}(M1(j,i) * M2(j,i))$

Return Median { E(j) | j=0,1,…,d-1}

Example of count-sketch (V1)

V1: (1,0,1,2,0), Index (0,1,2,3,4)

Initial

0	0	0
0	0	0
0	0	0

h1(j) = $j \mathit{mod}\ 3$
h2(j) = $(j \mathit{mod}\ 4) \mathit{mod}\ 3$
h3(j) = $(2{*}j) \mathit{mod}\ 3$

Map index domain with *h* and *g* functions,

g hash functions

	0	1	2	3	4
g1	+1	-1	-1	+1	-1
g2	-1	+1	-1	+1	+1
g3	+1	-1	+1	-1	+1

Example of count sketch (V1)

V1: (1,0,1,2,0), Index (0,1,2,3,4)

Initial

1	0	0
0	0	0
0	0	0

g hash functions

	0	1	2	3	4
g1	+1	-1	-1	+1	-1
g2	-1	+1	-1	+1	+1
g3	+1	-1	+1	-1	+1

h1(j) = $j \ mod \ 3$
h2(j) = $(j \ mod \ 4) \ mod \ 3$
h3(j) = $(2*j) \ mod \ 3$

Example of count sketch (V1)

V1: (1,0,1,2,0), Index (0,1,2,3,4)

Initial

1	0	0
-1	0	0
0	0	0

h1(j) = $j \ mod \ 3$
h2(j) = $(j \ mod \ 4) \ mod \ 3$
h3(j) = $(2*j) \ mod \ 3$

g hash functions

	0	1	2	3	4
g1	+1	-1	-1	+1	-1
g2	-1	+1	-1	+1	+1
g3	+1	-1	+1	-1	+1

Example of count sketch (V1)

V1: (1,0,1,2,0), Index (0,1,2,3,4)

Initial

1	0	0
-1	0	0
1	0	0

h1(j) = $j \; mod \; 3$
h2(j) = $(j \; mod \; 4) \; mod \; 3$
h3(j) = $(2*j) \; mod \; 3$

g hash functions

	0	1	2	3	4
g1	+1	-1	-1	+1	-1
g2	-1	+1	-1	+1	+1
g3	+1	-1	+1	-1	+1

Example of count sketch (V1)

V1: (1,0,1,2,0), Index (0,1,2,3,4)

Initial

1	0	-1
-1	0	-1
1	1	0

h1(j) = $j \bmod 3$
h2(j) = $(j \bmod 4) \bmod 3$
h3(j) = $(2 * j) \bmod 3$

g hash functions

	0	1	2	3	4
g1	+1	-1	-1	+1	-1
g2	-1	+1	-1	+1	+1
g3	+1	-1	+1	-1	+1

Example of count sketch (V1)

V1: (1,0,1,2,0), Index (0,1,2,3,4)

Initial

3	0	-1
1	0	-1
-1	1	0

h1(j) = $j \bmod 3$
h2(j) = $(j \bmod 4) \bmod 3$
h3(j) = $(2*j) \bmod 3$

g hash functions

	0	1	2	3	4
g1	+1	-1	-1	+1	-1
g2	-1	+1	-1	+1	+1
g3	+1	-1	+1	-1	+1

Example of count sketch (V2)

V2: (0,0,2,1,0), Index (0,1,2,3,4)

Initial

0	0	-2
0	0	-2
0	2	0

h1(j) = $j \bmod 3$
h2(j) = $(j \bmod 4) \bmod 3$
h3(j) = $(2^*j) \bmod 3$

g hash functions

	0	1	2	3	4
g1	+1	-1	-1	+1	-1
g2	-1	+1	-1	+1	+1
g3	+1	-1	+1	-1	+1

Example of count sketch (V2)

V2: (0,0,2,1,0), Index (0,1,2,3,4)

Initial

1	0	-2
1	0	-2
-1	2	0

$h1(j) = j \bmod 3$
$h2(j) = (j \bmod 4) \bmod 3$
$h3(j) = (2^*j) \bmod 3$

g hash functions

	0	1	2	3	4
g1	+1	-1	-1	+1	-1
g2	-1	+1	-1	+1	+1
g3	+1	-1	+1	-1	+1

Example of count sketch (result)

V1: (1,0,1,2,0),

V2: (0,0,2,1,0),

3	0	-1
1	0	-1
-1	1	0

1	0	-2
1	0	-2
-1	2	0

$3 \times 1 + (-1) \times (-2) = 5$
$1 \times 1 + (-1) \times (-2) = 3$
$(-1) \times (-1) + 1 \times 2 = 3$

Median (5,3,3) = 3

The accurate value is 2+2=4. Underestimation!

Questions?

- Why does the Count-sketch need to generate the second hash function g_i to produce the random +1 and -1 values?
- Answer the questions in the self-assessment form

Outline

- Massive data stream
- Bloom filter
- Count-min
- Count-sketch
- FM-sketch

Distinct Value Estimation

- Problem *:* Find the *number of distinct values* in a stream of values

Data : 3 2 5 3 2 1 7 5 1 2 3 7

Number of distinct values: 5

FM Sketch

- The algorithm was introduced by Philippe Flajolet and G. Nigel Martin in their 1985 paper "Probabilistic Counting Algorithms for Data Base Applications“

- Download here

- Approximating the number of distinct elements in a stream with a single pass and space efficient

FM Sketch

- Map input x to an integer h(x) in the range $[0;2^L-1]$, where the outputs are sufficiently uniformly distributed.
- Define Tail(h(x)) = number of trailing consecutive 0 from right

Tail(101001) =0
Tail(101010) =1
Tail(101100) =2

- Use a bit array to estimate the number of distinct elements

FM algorithm

- 1. Initialize a bit-array A to be of length L and contain all 0's.
- 2. For each item x :
- 1. index = Tail(h(x))
- 2. A[index] = 1
- 3. Let R denote the smallest index i such that A[i]=0
- 4. Estimate the number of distinct elements as $\frac{2^R}{\varphi}$, where φ= 0.77351.

FM sketch example

Data : 3 4 5 1 2 5 3

h(x) = (x * 7) mod 15

Bit-array A:

0	**0**	**0**	**0**

FM sketch example

Data : 3 4 5 1 2 5 3

h(x) = (x *7) mod 15

Bit-array A:

0	**0**	**1**	**0**

$3*7=21 \xrightarrow{\text{mod 15}} 6=0110 \xrightarrow{\text{Tail}} 1$

FM sketch example

Data : 3 4 5 1 2 5 3

h(x) = (x *7) mod 15

Bit-array A:

0	**0**	**1**	**1**

4*7=28 $\xrightarrow{\text{mod 15}}$ 13=1101 $\xrightarrow{\text{Tail}}$ 0

FM sketch example

Data : 3 4 5 1 2 5 3

h(x) = (x * 7) mod 15

Bit-array A:

0	**0**	**1**	**1**

$5*7=35 \xrightarrow{\text{mod 15}} 5=0101 \xrightarrow{\text{Tail}} 0$

FM sketch example

Data : 3 4 5 1 2 5 3

h(x) = (x * 7) mod 15

Bit-array A:

0	**0**	**1**	**1**

mod 15 Tail

1*7=7 → 7=0111 → 0

FM sketch example

Data : 3 4 5 1 2 5 3

h(x) = (x * 7) mod 15

Bit-array A:

0	**0**	**1**	**1**

mod 15 Tail

2*7=14 → 14=1110 → 1

FM sketch example

Data : 3 4 5 1 2 5 3

Bit-array A:

0	**0**	**1**	**1**

Estimate the number of distinct elements as $\frac{2^R}{\varphi}$, where φ= 0.77351.

R=2, $\frac{2^2}{0.77} = 5.2 \approx 5$

Correct estimation!

Why FM algorithm can work?

- Because hash function Tail(h(x)) mapping input items to i with probability:

  - Pr[Tail(h(x) = 0] = 1/2,
  - Pr[Tail(h(x) = 1] = 1/4,
  - Pr[Tail(h(x) = 2] = 1/8

  - Intuitively, 2^R can be used to estimate the number of distinct elements.

    φ= 0.77351 is a correction ratio for more accurate result.

Hence, the result is $2^R / \varphi$

- Watch a video on FM sketch

- https://www.youtube.com/watch?v=IHitCLlljHo

Summary of FM Sketch

- Computing the number of distinct taking time proportional to the size of the data,
- FM sketch allows approximate computation with much smaller space
  - Logarithmic in the number of distinct elements.

Conclusions on sketches

- We introduced four sketches:
  - Bloom-filter, Count-min, Count-sketch and FM sketch

- Sketch methods: Simple, yet powerful, ideas with great reach

- Public code implementation:

  http://www.cs.rutgers.edu/~muthu/massdal-code-index.html

Reference

- Read the following reference papers about sketch algorithms to have a deeper understanding

- Sketch Techniques for Approximate Query Processing

- https://people.cs.umass.edu/~mcgregor/711S12/sketches1.pdf