Lecture 14

Mergeable summaries, Sketching

Course: Algorithms for Big Data

Instructor: Hossein Jowhari

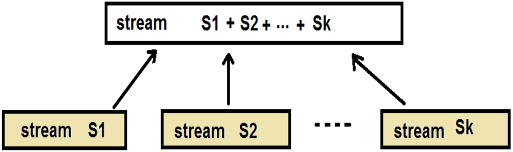
Department of Computer Science and Statistics
Faculty of Mathematics
K. N. Toosi University of Technology

Spring 2021

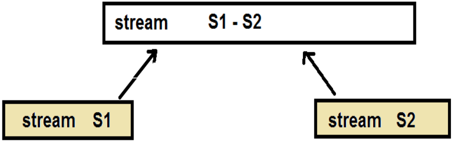
Operations on data streams

In many settings, we like to able to perform operations on data such as merging (adding) or subtracting two (or multiple) data sets.

- Merge (union)



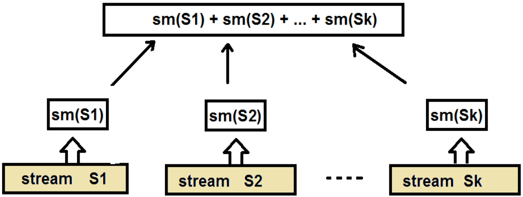
- Subtract



However, often these data streams are generated in different sites. Transmitting a large data stream, even if we pay the communication cost, might not be feasible.

Therefore it is much desirable if we transmit a small summary of the data stream.

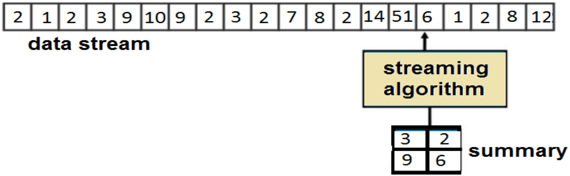
Naturally, we want the summaries to be mergeable or subtractable.



How to build mergeable summaries of data?

- Sampling is simple and easy but many problems have large sampling complexity.

- Streaming algorithms can do much more. In a way, an streaming algorithm builds a summary of the input stream after processing every data item. The constructed summary is used to answer a specific question about data. For example the summary is used to answer (approximately) how many distinct elements are there in the stream.



A specific example: Misra-Gries algorithm

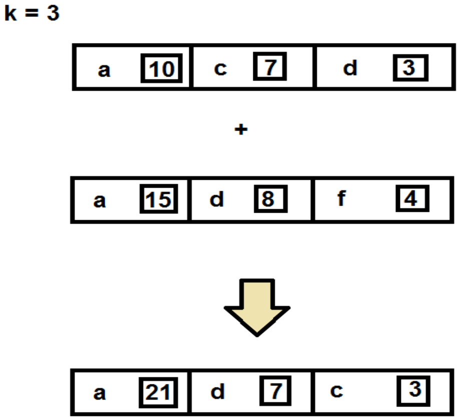
Recall that Misri-Gries algorithm (the majority-based algorithm) keeps at most k elements along with k counters.

element	counter	element	counter	element	counter	next item
	0		0		0	f
f	1		0		0	g
f	1	g	1		0	h
f	1	g	1	h	1	d
	0		0		0	c
c	1		0		0	c
c	2		0		0	d
c	2	d	1		0	a
c	2	d	1	a	1	b
c	1		0		0	t
c	1	t	1		0	a
c	1	t	1	a	1	w
	0		0		0	a
a	1		0		0	s
a	1	s	1		0	a
a	2	s	1		0	b
a	2	s	1	b	1	a
a	3	s	1	b	1	b
a	3	s	1	b	2	c
a	2		0	b	1	n
a	2	n	1	b	1	a
a	3	n	1	b	1	c
a	2		0		0	c
a	2	c	1		0	a
a	3	c	1		0	a
a	4	c	1		0	b
a	4	c	1	b	1	f
a	3		0		0	c
a	3	c	1		0	a
a	4	c	1		0	c
a	4	c	2		0	c
a	4	c	3		0	c
a	4	c	4		0	

stream

summary

The Misra-Gries summary is deterministic and easily mergeable. Why?



Subtracting Streams

For the occurrence streams, we subtract the associated frequency vectors. Here subtracting the element i means deleting i from the first stream.

Suppose we are able to subtract the streams $S2$ from $S1$. In other words $\forall i,\ f_{S1}(i) \geq f_{S2}(i)$. We are in the strict turnstile model.

$$S1 = a, a, a, a, a, a, a, a, b, b, b, b, c, c, c, d$$

$$f_{S1}(a) = 8, \quad f_{S1}(b) = 4, \quad f_{S1}(c) = 3, \quad f_{S1}(d) = 1$$

$$S2 = a, a, a, a, a, a, a, a, d$$

$$f_{S2}(a) = 8, \quad f_{S2}(b) = 0, \quad f_{S2}(c) = 0, \quad f_{S2}(d) = 1$$

$$f_{S1-S2}(a) = 0,\ \ f_{S1-S2}(b) = 4,\ \ f_{S1-S2}(c) = 3,\ \ f_{S1-S2}(d) = 0$$

Is Misra-Gries summary subtractable?

Unfortunately not. Consider the following example. Suppose number of counter $k = 2$

$S1 = a, a, a, a, a, a, a, a, b, b, b, b, c, c, c, d$

Misra-Gries summary $\Rightarrow$ $a : 4$

$S2 = a, a, a, a, a, a, a, a, d$

Misra-Gries summary $\Rightarrow$ $a : 7,\ d : 1$

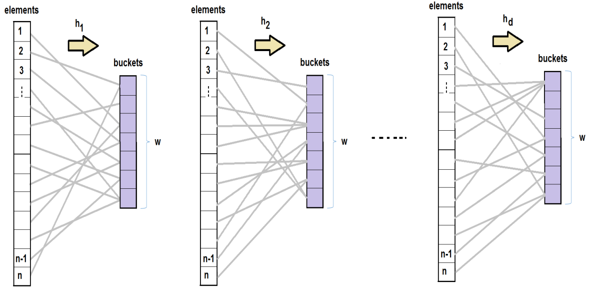
But b is the majority element in $S1 - S2$.

b is missing in both summaries.

CountMin is both mergeable and subtractable

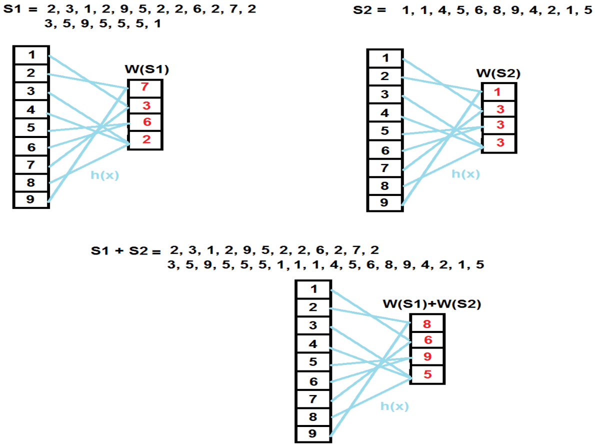
Recall that CountMin randomly hashed the elements $[n]$ into w buckets. For this it uses a series of pairwise independent hash functions $h_i(x) = a_i x + b_i \mod w \qquad i = 1, \ldots, d$

For each bucket, the algorithm counts the number of elements that are hashed to that bucket. The algorithm also stores the hash functions.



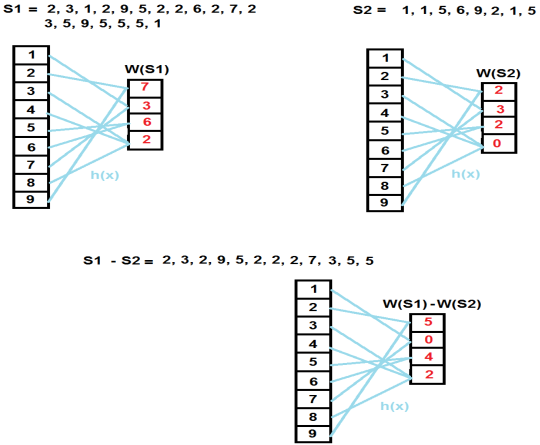
CountMin is mergeable

Assuming all the summaries use the same hash functions, we can easily add CountMin summaries. It is enough to add the corresponding bucket vectors.



CountMin is also subtractable

Again assuming all the summaries use the same hash functions, we can easily subtract two CountMin summaries by subtracting the corresponding bucket vectors.



Sketch/Sketching

- In the literature, the term sketch is often used for data stream summaries. However a sketch usually refers to a mergebale/subtractable summary.
- An alternative definition: A sketch is the image of a (randomized) mapping sk that maps the underlying data vector to a vector with small dimension.

$$sk : \mathbb{R}^n \to \mathbb{R}^t \qquad\qquad t \ll n$$

- A linear sketch is a sketch that can be merged/subtracted by a set of linear operations.

$$sk(\boldsymbol{x} + \boldsymbol{y}) = \alpha\ sk(\boldsymbol{x}) + \beta\ sk(\boldsymbol{y}) + \boldsymbol{c}$$

- A sketching algorithm is an algorithm that builds the sketch of a data.

AMS F_2 sketch

Recall how the AMS (Alon, Matias, Szegedy) algorithm approximated $F_2 = \sum_{i=1}^{n} x_i^2$. Every element i in the stream is adding 1 to the i-th coordinate of an initially zero vector $\boldsymbol{x} \in \mathbb{Z}^n$.

- The algorithms picks a random vector $\boldsymbol{\sigma} \in \{-1, +1\}^n$.
- It processes the stream and computes the inner product $Z = \boldsymbol{\sigma}.\boldsymbol{x}$. This is repeated $t = O(\frac{1}{\epsilon^2})$ number of times independently in parallel.

$$sk : \mathbb{Z}^n \to \mathbb{Z}^t$$

$$\underbrace{\begin{bmatrix} -1 & +1 & \dots & +1 \\ +1 & -1 & \dots & +1 \\ \vdots & \vdots & \vdots & \vdots \\ -1 & -1 & \dots & -1 \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}}_{\boldsymbol{x}} = \underbrace{\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_t \end{bmatrix}}_{sk(\boldsymbol{x})}$$

Final remarks

- AMS F_2 sketch is a linear sketch

$$sk(\boldsymbol{x} + \boldsymbol{y}) = \Sigma\boldsymbol{x} + \Sigma\boldsymbol{y}$$

$$sk(\boldsymbol{x} - \boldsymbol{y}) = \Sigma\boldsymbol{x} - \Sigma\boldsymbol{y}$$

- CountMin is also a linear sketch. The pairwise independent hash functions $h_1, \ldots, h_d$, can be represented by a random matrix $H_{d\times n}$ with $\{0,1\}$ entries

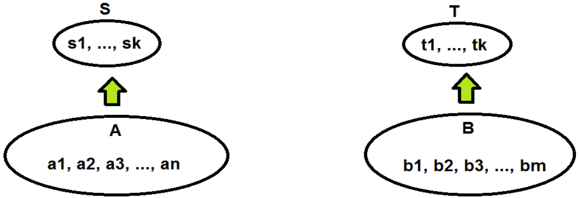
$$sk(\boldsymbol{x}) = H\boldsymbol{x}$$

- Misra-Gries summary is not a linear sketch. It is also not subtractable.

How to merge samples?

Let $S = s_1, \ldots, s_k$ be k uniform independent samples from the stream $A = a_1, \ldots, a_n$. For each $s \in S$, we have $Pr(s = a_i) = \frac{1}{n}$.

Similarly let $T = t_1, \ldots, t_k$ be k uniform independent samples from the stream $B = b_1, \ldots, b_m$. For each $t \in T$, we have $Pr(t = b_i) = \frac{1}{m}$.



How can we obtain samples from $A \cup B$ using S and T?