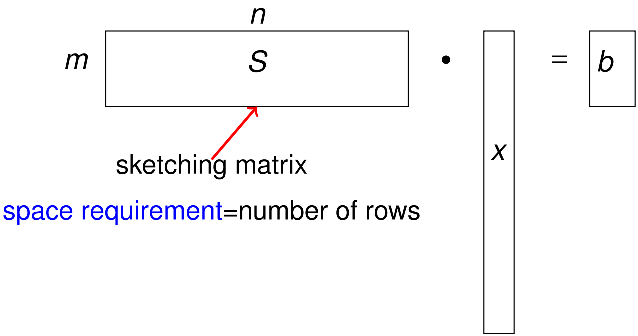
Lecture 2: Frequency Moments, Heavy Hitters

Michael Kapralov

EPFL

May 24, 2017

Linear sketching



Sketching algorithms for basic statistics, and then graph sketching

In this lecture:

- Frequency moments (AMS sketch)
- Heavy hitters (CountSketch)

In this lecture:

- **Frequency moments (AMS sketch)**
- Heavy hitters (CountSketch)

AMS sketch (Alon-Matias-Szegedy'96)

Goal: approximate

$$||x||_2 = \sqrt{\sum_{i \in [n]} x_i^2}$$

from a stream of increments/decrements to x_i.

Vector interpretation of a data stream

$x \in \mathbb{R}^n$ `1 2 3 4 5 6 7 8 9 10`

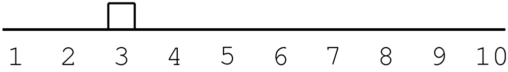
- Initially, $x = 0$
- Insertion of i interpreted as

$$x_i := x_i + 1$$

- Want to estimate $||x||_2^2$

Vector interpretation of a data stream

$x \in \mathbb{R}^n$



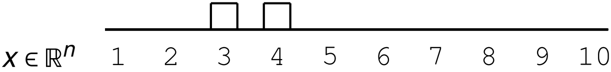
3

- Initially, $x = 0$
- Insertion of i interpreted as

$$x_i := x_i + 1$$

- Want to estimate $||x||_2^2$

Vector interpretation of a data stream



3 4

- Initially, $x = 0$
- Insertion of i interpreted as

$$x_i := x_i + 1$$

- Want to estimate $||x||_2^2$

Vector interpretation of a data stream

$x \in \mathbb{R}^n$ 1 2 3 4 5 6 7 8 9 10

3 4 3

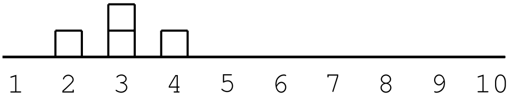
- Initially, $x = 0$
- Insertion of i interpreted as

$$x_i := x_i + 1$$

- Want to estimate $||x||_2^2$

Vector interpretation of a data stream

$x \in \mathbb{R}^n$



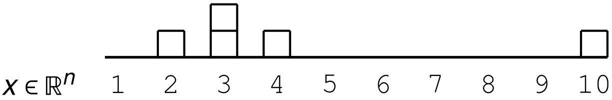
3 4 3 2

- Initially, $x = 0$
- Insertion of i interpreted as

$$x_i := x_i + 1$$

- Want to estimate $||x||_2^2$

Vector interpretation of a data stream



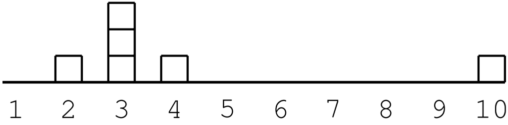
3 4 3 2 10

- Initially, $x = 0$
- Insertion of i interpreted as

$$x_i := x_i + 1$$

- Want to estimate $||x||_2^2$

Vector interpretation of a data stream



$x \in \mathbb{R}^n$

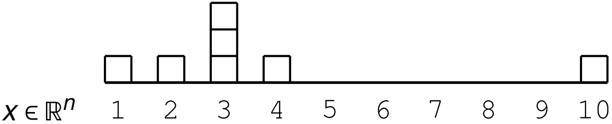
3 4 3 2 10 3

- Initially, $x = 0$
- Insertion of i interpreted as

$$x_i := x_i + 1$$

- Want to estimate $||x||_2^2$

Vector interpretation of a data stream



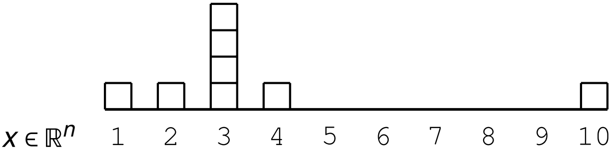
3 4 3 2 10 3 1

- Initially, $x = 0$
- Insertion of i interpreted as

$$x_i := x_i + 1$$

- Want to estimate $||x||_2^2$

Vector interpretation of a data stream



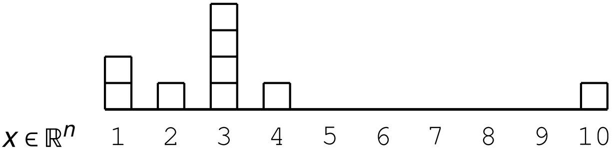
3 4 3 2 10 3 1 3

- Initially, $x = 0$
- Insertion of i interpreted as

$$x_i := x_i + 1$$

- Want to estimate $||x||_2^2$

Vector interpretation of a data stream



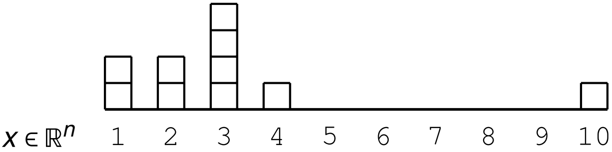
3 4 3 2 10 3 1 3 1

- Initially, $x = 0$
- Insertion of i interpreted as

$$x_i := x_i + 1$$

- Want to estimate $||x||_2^2$

Vector interpretation of a data stream



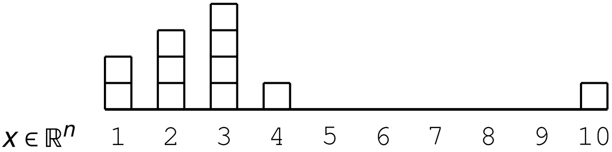
3 4 3 2 10 3 1 3 1 2

- Initially, $x = 0$
- Insertion of i interpreted as

$$x_i := x_i + 1$$

- Want to estimate $||x||_2^2$

Vector interpretation of a data stream



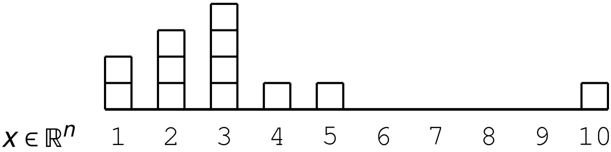
3 4 3 2 10 3 1 3 1 2 2

- Initially, $x = 0$
- Insertion of i interpreted as

$$x_i := x_i + 1$$

- Want to estimate $||x||_2^2$

Vector interpretation of a data stream



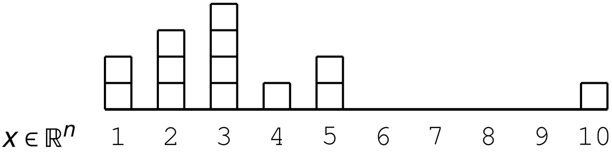
3 4 3 2 10 3 1 3 1 2 2 5

- Initially, $x = 0$
- Insertion of i interpreted as

$$x_i := x_i + 1$$

- Want to estimate $||x||_2^2$

Vector interpretation of a data stream



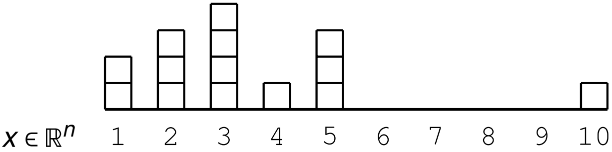
3 4 3 2 10 3 1 3 1 2 2 5 5

- Initially, $x = 0$
- Insertion of i interpreted as

$$x_i := x_i + 1$$

- Want to estimate $||x||_2^2$

Vector interpretation of a data stream



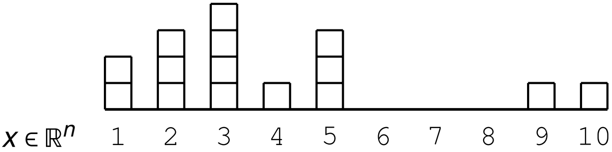
3 4 3 2 10 3 1 3 1 2 2 5 5 5

- Initially, $x = 0$
- Insertion of i interpreted as

$$x_i := x_i + 1$$

- Want to estimate $||x||_2^2$

Vector interpretation of a data stream



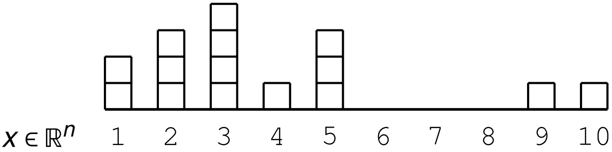
3 4 3 2 10 3 1 3 1 2 2 5 5 5 9

- Initially, $x = 0$
- Insertion of i interpreted as

$$x_i := x_i + 1$$

- Want to estimate $||x||_2^2$

Vector interpretation of a data stream



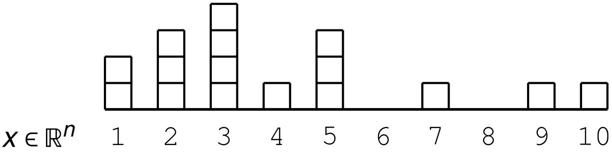
3 4 3 2 10 3 1 3 1 2 2 5 5 5 9

- Initially, $x = 0$
- Insertion of i interpreted as

$$x_i := x_i + 1$$

- Want to estimate $||x||_2^2$

Vector interpretation of a data stream



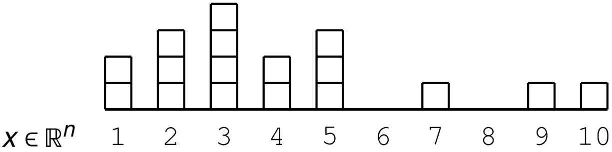
3 4 3 2 10 3 1 3 1 2 2 5 5 5 9 7

- Initially, $x = 0$
- Insertion of i interpreted as

$$x_i := x_i + 1$$

- Want to estimate $||x||_2^2$

Vector interpretation of a data stream



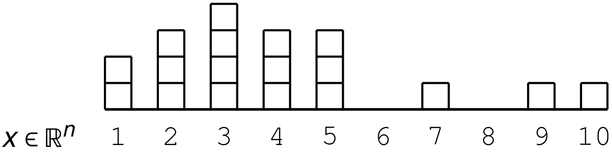
3 4 3 2 10 3 1 3 1 2 2 5 5 5 9 7 4

- Initially, $x = 0$
- Insertion of i interpreted as

$$x_i := x_i + 1$$

- Want to estimate $||x||_2^2$

Vector interpretation of a data stream



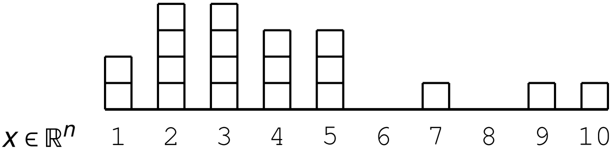
3 4 3 2 10 3 1 3 1 2 2 5 5 5 9 7 4 4

- Initially, $x = 0$
- Insertion of i interpreted as

$$x_i := x_i + 1$$

- Want to estimate $||x||_2^2$

Vector interpretation of a data stream



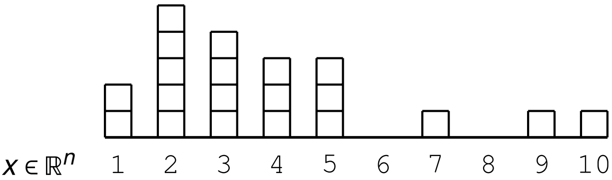
3 4 3 2 10 3 1 3 1 2 2 5 5 5 9 7 4 4 2

- Initially, $x = 0$
- Insertion of i interpreted as

$$x_i := x_i + 1$$

- Want to estimate $||x||_2^2$

Vector interpretation of a data stream



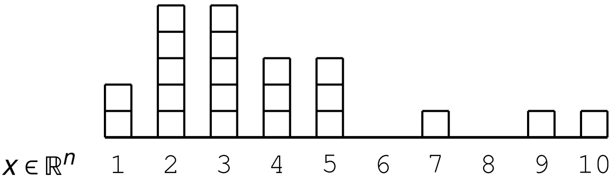
3 4 3 2 10 3 1 3 1 2 2 5 5 5 9 7 4 4 2 2

- Initially, $x = 0$
- Insertion of i interpreted as

$$x_i := x_i + 1$$

- Want to estimate $||x||_2^2$

Vector interpretation of a data stream



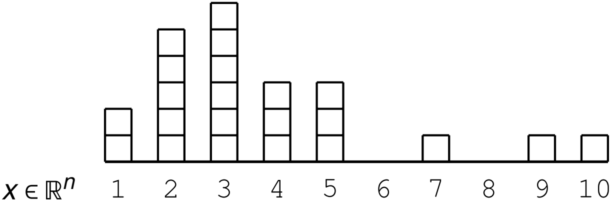
3 4 3 2 10 3 1 3 1 2 2 5 5 5 9 7 4 4 2 2 3

- Initially, $x = 0$
- Insertion of i interpreted as

$$x_i := x_i + 1$$

- Want to estimate $||x||_2^2$

Vector interpretation of a data stream



3 4 3 2 10 3 1 3 1 2 2 5 5 5 9 7 4 4 2 2 3 3

- Initially, $x = 0$
- Insertion of i interpreted as

$$x_i := x_i + 1$$

- Want to estimate $||x||_2^2$

AMS sketch (Alon-Matias-Szegedy'96)

Goal: approximate

$$||x||_2 = \sqrt{\sum_{i \in [n]} x_i^2}$$

from a stream of increments/decrements to x_i.

AMS sketch (Alon-Matias-Szegedy'96)

Goal: approximate

$$||x||_2 = \sqrt{\sum_{i \in [n]} x_i^2}$$

from a stream of increments/decrements to x_i.

Choose $r_1, \ldots, r_n$ to be i.i.d. r.v., with

$$\mathbf{Pr}[r_i = +1] = \mathbf{Pr}[r_i = -1] = 1/2.$$

AMS sketch (Alon-Matias-Szegedy'96)

Goal: approximate

$$||x||_2 = \sqrt{\sum_{i \in [n]} x_i^2}$$

from a stream of increments/decrements to x_i.

Choose $r_1, \ldots, r_n$ to be i.i.d. r.v., with

$$\mathbf{Pr}[r_i = +1] = \mathbf{Pr}[r_i = -1] = 1/2.$$

Maintain

$$Z = \sum_{i=1}^{n} r_i x_i$$

under increments/decrements of x.

AMS sketch (Alon-Matias-Szegedy'96)

Goal: approximate

$$||x||_2 = \sqrt{\sum_{i \in [n]} x_i^2}$$

from a stream of increments/decrements to x_i.

Choose $r_1, \ldots, r_n$ to be i.i.d. r.v., with

$$\mathbf{Pr}[r_i = +1] = \mathbf{Pr}[r_i = -1] = 1/2.$$

Maintain

$$Z = \sum_{i=1}^{n} r_i x_i$$

under increments/decrements of x.

Basic algorithm: output Z^2

AMS sketch (Alon-Matias-Szegedy'96)

Goal: approximate

$$||x||_2 = \sqrt{\sum_{i \in [n]} x_i^2}$$

from a stream of increments/decrements to x_i.

Choose $r_1, \ldots, r_n$ to be i.i.d. r.v., with

$$\mathbf{Pr}[r_i = +1] = \mathbf{Pr}[r_i = -1] = 1/2.$$

Maintain

$$Z = \sum_{i=1}^{n} r_i x_i$$

under increments/decrements of x.

Basic algorithm: output Z^2

Want to claim that Z^2 is 'close' to $||x||_2^2$ with 'high probability'

Alon-Matias-Szegedy – analysis (expectation)

Want to claim that Z^2 is 'close' to $||x||_2^2$ with 'high probability'

Alon-Matias-Szegedy – analysis (expectation)

Want to claim that Z^2 is 'close' to $||x||_2^2$ with 'high probability'

$\boxed{\text{Compute expectation of } Z^2\text{, then bound the variance}}$

Alon-Matias-Szegedy – analysis (expectation)

Want to claim that Z^2 is 'close' to $||x||_2^2$ with 'high probability'

Compute expectation of Z^2, then bound the variance

Expectation:

$$
\begin{aligned}
\mathbf{E}[Z^2] &= \mathbf{E}\left[(\sum_{i=1}^{n} r_i x_i)^2\right] \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbf{E}[r_i r_j x_i x_j] \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbf{E}[r_i r_j] x_i x_j \\
&= \sum_{i=1}^{n} x_i^2 + \sum_{i,j:i\neq j}^{n} \mathbf{E}[r_i]\mathbf{E}[r_j] x_i x_j \\
&= \sum_{i=1}^{n} x_i^2 \\
&= ||x||_2^2
\end{aligned}
$$

Alon-Matias-Szegedy – analysis (expectation)

Want to claim that Z^2 is 'close' to $||x||_2^2$ with 'high probability'

Compute expectation of Z^2, then bound the variance

Expectation:

$$\begin{aligned}
\mathbf{E}[Z^2] &= \mathbf{E}\left[(\sum_{i=1}^{n} r_i x_i)^2\right] \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbf{E}[r_i r_j x_i x_j] \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbf{E}[r_i r_j] x_i x_j \\
&= \sum_{i=1}^{n} x_i^2 + \sum_{i,j:i\neq j}^{n} \mathbf{E}[r_i]\mathbf{E}[r_j] x_i x_j \\
&= \sum_{i=1}^{n} x_i^2 \\
&= ||x||_2^2 \qquad \text{(our estimator is unbiased!)}
\end{aligned}$$

Alon-Matias-Szegedy – analysis (variance)

Want to claim that Z^2 is 'close' to $||x||_2^2$ with 'high probability'

Bound the variance $\mathbf{Var}[Z^2] = \mathbf{E}[Z^4] - (\mathbf{E}[Z^2])^2$?

Alon-Matias-Szegedy – analysis (variance)

Want to claim that Z^2 is ‘close’ to $||x||_2^2$ with ‘high probability’

Bound the variance $\mathbf{Var}[Z^2] = \mathbf{E}[Z^4] - (\mathbf{E}[Z^2])^2$?

Compute

$$\mathbf{E}[Z^4] = \mathbf{E}\left[(\sum_{i=1}^{n} r_i x_i)(\sum_{j=1}^{n} r_j x_j)(\sum_{k=1}^{n} r_k x_k)(\sum_{l=1}^{n} r_l x_l) \right]$$

Alon-Matias-Szegedy – analysis (variance)

Want to claim that Z^2 is 'close' to $||x||_2^2$ with 'high probability'

Bound the variance $\mathbf{Var}[Z^2] = \mathbf{E}[Z^4] - (\mathbf{E}[Z^2])^2$?

Compute

$$\mathbf{E}[Z^4] = \mathbf{E}\left[(\sum_{i=1}^{n} r_i x_i)(\sum_{j=1}^{n} r_j x_j)(\sum_{k=1}^{n} r_k x_k)(\sum_{l=1}^{n} r_l x_l) \right]$$

Can be decomposed as follows:

- $\sum_{i=1}^{n}(r_i x_i)^4$ – expectation $\sum_{i=1}^{n} x_i^4$
- $6\sum_{i<j}(r_i r_j x_i x_j)^2$ – expectation $6\sum_{i<j} x_i^2 x_j^2$
- Terms involving a single $r_i x_i$ – expectation zero.

In total: $\sum_{i=1}^{n} x_i^4 + 6\sum_{i<j} x_i^2 x_j^2$

Alon-Matias-Szegedy – analysis (variance)

Bound the variance $\mathbf{Var}[Z^2] = \mathbf{E}[Z^4] - (\mathbf{E}[Z^2])^2$?

Computed

$$\mathbf{E}[Z^4] = \sum_{i=1}^{n} x_i^4 + 6\sum_{i<j} x_i^2 x_j^2$$

Alon-Matias-Szegedy – analysis (variance)

Bound the variance $\mathbf{Var}[Z^2] = \mathbf{E}[Z^4] - (\mathbf{E}[Z^2])^2$?

Computed

$$\mathbf{E}[Z^4] = \sum_{i=1}^{n} x_i^4 + 6 \sum_{i<j} x_i^2 x_j^2$$

So

$$\begin{aligned}
\mathbf{Var}[Z^2] &= \mathbf{E}[Z^4] - (\mathbf{E}[Z^2])^2 \\
&= \sum_{i=1}^{n} x_i^4 + 6 \sum_{i<j} x_i^2 x_j^2 - (\sum_{i=1}^{n} x_i^2)^2 \\
&= \sum_{i=1}^{n} x_i^4 + 6 \sum_{i<j} x_i^2 x_j^2 - \sum_{i=1}^{n} x_i^4 - 2 \sum_{i<j} x_i^2 x_j^2 \\
&\leq 4 \sum_{i<j} x_i^2 x_j^2 \\
&\leq 2 (\sum_{i=1}^{n} x_i^2)^2
\end{aligned}$$

Analysis: putting it together

We showed that

- $\mathbf{E}[Z^2] = ||x||_2^2$
- $\sigma^2 = \mathbf{Var}[Z^2] \leq 2||x||_2^4$

Analysis: putting it together

We showed that

- $\mathbf{E}[Z^2] = ||x||_2^2$
- $\sigma^2 = \mathbf{Var}[Z^2] \leq 2||x||_2^4$

So by Chebyshev's inequality for $t \geq 1$

$$\mathbf{Pr}[|Z^2 - \mathbf{E}[Z^2]| \geq t\sigma] \leq 1/t^2$$

Analysis: putting it together

We showed that

- $\mathbf{E}[Z^2] = ||x||_2^2$
- $\sigma^2 = \mathbf{Var}[Z^2] \leq 2||x||_2^4$

So by Chebyshev's inequality for $t \geq 1$

$$\mathbf{Pr}[|Z^2 - \mathbf{E}[Z^2]| \geq t\sigma] \leq 1/t^2$$

and we get

$$\mathbf{Pr}[|Z^2 - ||x||_2^2| \geq \sqrt{2}t||x||_2^2] \leq 1/t^2$$

Analysis: putting it together

We showed that

- $\mathbf{E}[Z^2] = ||x||_2^2$
- $\sigma^2 = \mathbf{Var}[Z^2] \leq 2||x||_2^4$

So by Chebyshev's inequality for $t \geq 1$

$$\mathbf{Pr}[|Z^2 - \mathbf{E}[Z^2]| \geq t\sigma] \leq 1/t^2$$

and we get

$$\mathbf{Pr}[|Z^2 - ||x||_2^2| \geq \sqrt{2}t||x||_2^2] \leq 1/t^2$$

Not good...

Analysis: putting it together

We showed that

- $\mathbf{E}[Z^2] = ||x||_2^2$
- $\sigma^2 = \mathbf{Var}[Z^2] \leq 2||x||_2^4$

So by Chebyshev's inequality for $t \geq 1$

$$\mathbf{Pr}[|Z^2 - \mathbf{E}[Z^2]| \geq t\sigma] \leq 1/t^2$$

and we get

$$\mathbf{Pr}[|Z^2 - ||x||_2^2| \geq \sqrt{2}t||x||_2^2] \leq 1/t^2$$

Not good...but can reduce variance by averaging!

Analysis: putting it together

Actual algorithm:

- Maintain $Z_1, \ldots, Z_k$, $Z_i = \sum_{j=1}^{n} r_j^i x_j$
- Output $A := \frac{1}{k} \sum_{i=1}^{k} Z_i^2$

Now

$$\mathbf{Var}[A] = \mathbf{Var}\left[\frac{1}{k}\sum_{i=1}^{k} Z_i^2\right] = \frac{1}{k}\mathbf{Var}\left[Z^2\right] \leq (2/k)||x||_2^4,$$

Analysis: putting it together

Actual algorithm:

- Maintain $Z_1, \ldots, Z_k$, $Z_i = \sum_{j=1}^{n} r_j^i x_j$
- Output $A := \frac{1}{k} \sum_{i=1}^{k} Z_i^2$

Now

$$\mathbf{Var}[A] = \mathbf{Var}\left[\frac{1}{k}\sum_{i=1}^{k} Z_i^2\right] = \frac{1}{k}\mathbf{Var}\left[Z^2\right] \leq (2/k)||x||_2^4,$$

and by Chebyshev's inequality

$$\mathbf{Pr}\left[|A - ||x||_2^2| \geq t \cdot (2/k)^{1/2}||x||_2^2\right] \leq 1/t^2,$$

so setting $k = O(1/\varepsilon^2)$ and $t = 10$ suffices for a $(1 \pm \varepsilon)$-approximation with probability $\geq 99/100$!

Space complexity

How much space do we need to store r_i's?

4-wise independence suffices, hence $O(\log n)$ space

Some remarks

Can we reduce failure probability to $1 - \delta$?

Some remarks

Can we reduce failure probability to $1 - \delta$?

- Use $O(1/(\varepsilon^2\delta))$ repetitions – bad dependence on δ

Some remarks

Can we reduce failure probability to $1 - \delta$?

- Use $O(1/(\varepsilon^2\delta))$ repetitions – bad dependence on δ
- Median trick: keep $T = O(\log(1/\delta))$ copies of the estimator, output the median

Some remarks

Can we reduce failure probability to $1-\delta$?

- Use $O(1/(\varepsilon^2\delta))$ repetitions – bad dependence on δ
- Median trick: keep $T = O(\log(1/\delta))$ copies of the estimator, output the median

Let $Y_t = 1$ if t-th algorithm fails, and 0 otherwise.

We have $\mathbf{E}[Y_t] \le 1/100$, so by the Chernoff bound

$$
\begin{aligned}
&\mathbf{Pr}\left[|\text{median}_{i=1,\ldots,T}(A_i) - ||x||_2^2| > \varepsilon ||x||_2^2\right] \\
&\le \mathbf{Pr}[\text{at least half of } A_i \text{ fail}, i = 1,\ldots,T] \\
&\le \mathbf{Pr}[\sum_{t=1}^{T} Y_i \ge T/2] \\
&\le e^{-\Omega(T)}.
\end{aligned}
$$

So setting $T = O(\log(1/\delta))$ suffices.

Space complexity

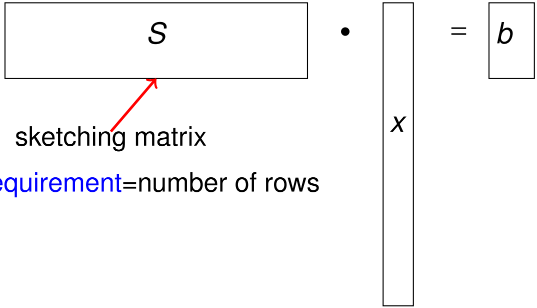
Downside of the median trick: nonlinear embedding

Median trick not needed if we have enough independence

Johnson-Lindenstrauss transform (see Ilya's lecture)

Some remarks

Take (randomized) linear measurements of the input



space requirement=number of rows

Some remarks

Take (randomized) linear measurements of the input

$$S \bullet x = b$$

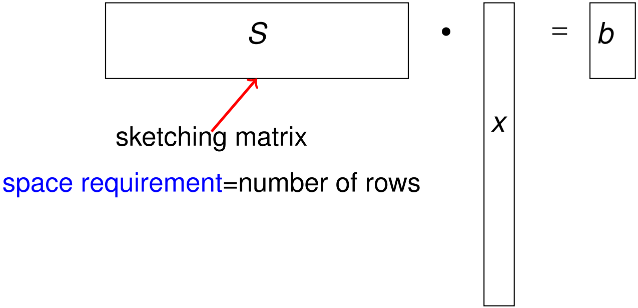
sketching matrix

space requirement=number of rows

Can get $(1 \pm \varepsilon)$-approximation to $||x||^2$ with $O(\frac{1}{\varepsilon^2}\log(1/\delta))$ rows

Some remarks

Take (randomized) linear measurements of the input



Can get $(1 \pm \varepsilon)$-approximation to $||x||^2$ with $O(\frac{1}{\varepsilon^2}\log(1/\delta))$ rows

Easy to maintain linear sketches in the (dynamic) streaming model

In this lecture:

- Frequency moments (AMS sketch)
- Heavy hitters (CountSketch)

In this lecture:

- Frequency moments (AMS sketch)
- **Heavy hitters (CountSketch)**

Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$

  Assume N is known

- Output k most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!

1 2 3 4 5 6 7 8 9 10

Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$

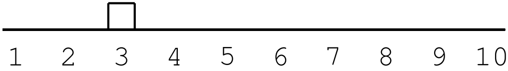
  Assume N is known

- Output k most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!



3

Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$

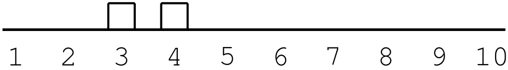
  Assume N is known

- Output k most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!



3 4

Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$

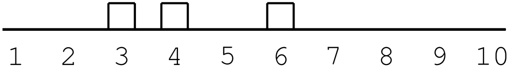
  Assume N is known

- Output k most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!



3 4 6

Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$

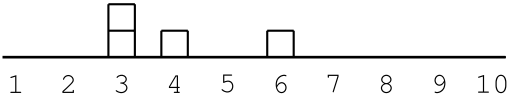
  Assume N is known

- Output k most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!



3 4 6 3

Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$

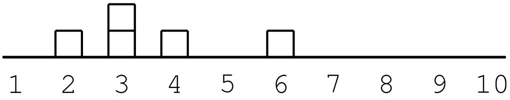
  Assume N is known

- Output k most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!



3 4 6 3 2

Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$

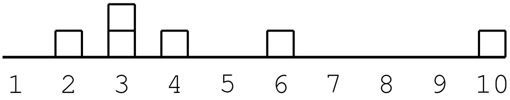
  Assume N is known

- Output k most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!



3 4 6 3 2 10

Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$

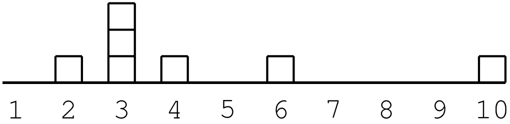
  Assume N is known

- Output k most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!



3 4 6 3 2 10 3

Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$

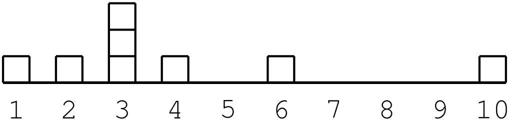
  Assume N is known

- Output k most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!



3 4 6 3 2 10 3 1

Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$

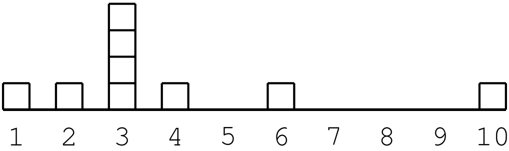
  Assume N is known

- Output k most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!



3 4 6 3 2 10 3 1 3

Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$

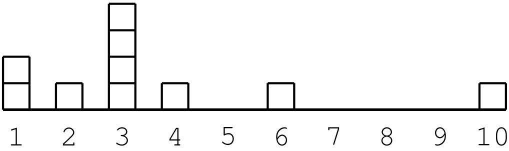
  Assume N is known

- Output k most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!



3 4 6 3 2 10 3 1 3 1

Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$

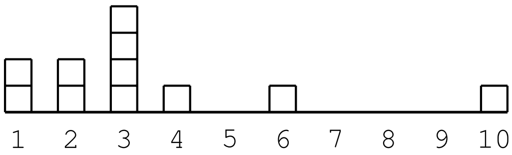
  Assume N is known

- Output k most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!



3 4 6 3 2 10 3 1 3 1 2

Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$

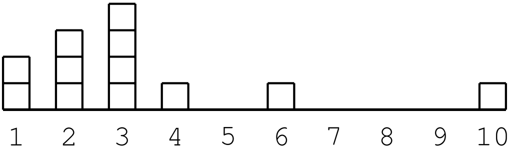
  Assume N is known

- Output k most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!



3 4 6 3 2 10 3 1 3 1 2 2

Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$

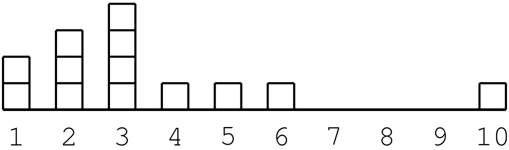
  Assume N is known

- Output k most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!



3 4 6 3 2 10 3 1 3 1 2 2 5

Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$

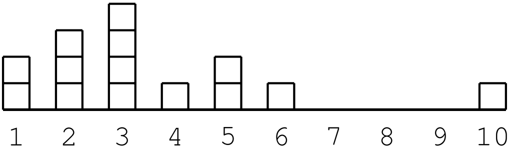
  Assume N is known

- Output k most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!



3 4 6 3 2 10 3 1 3 1 2 2 5 5

Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$

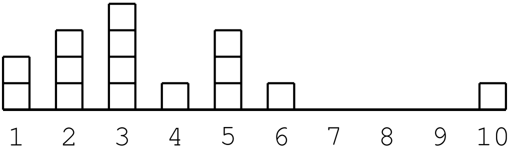
  Assume N is known

- Output k most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!



3 4 6 3 2 10 3 1 3 1 2 2 5 5 5

Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$

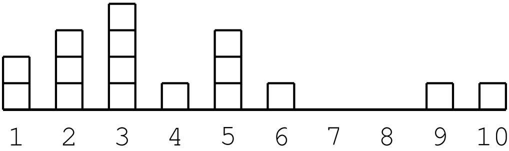
  Assume N is known

- Output k most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!



3 4 6 3 2 10 3 1 3 1 2 2 5 5 5 9

Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$

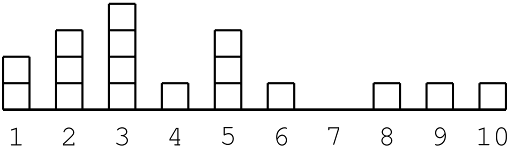
  Assume N is known

- Output k most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!



3 4 6 3 2 10 3 1 3 1 2 2 5 5 5 9 8

Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$

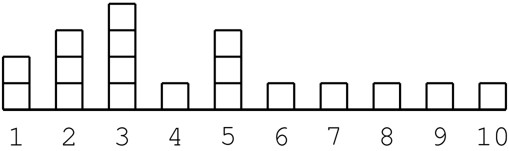
  Assume N is known

- Output k most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!



3 4 6 3 2 10 3 1 3 1 2 2 5 5 5 9 8 7

Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$

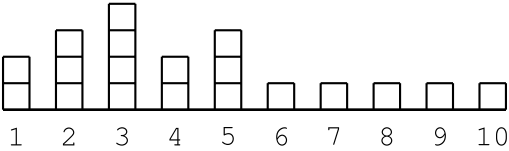
  Assume N is known

- Output k most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!



3 4 6 3 2 10 3 1 3 1 2 2 5 5 5 9 8 7 4

Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$

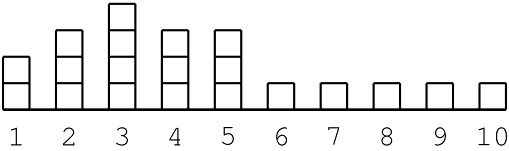
  Assume N is known

- Output k most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!



3 4 6 3 2 10 3 1 3 1 2 2 5 5 5 9 8 7 4 4

Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$

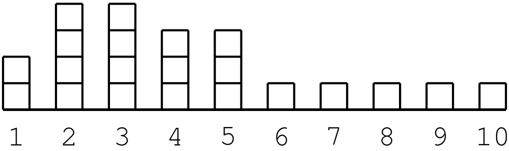
  Assume N is known

- Output k most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!



3 4 6 3 2 10 3 1 3 1 2 2 5 5 5 9 8 7 4 4 2

Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$

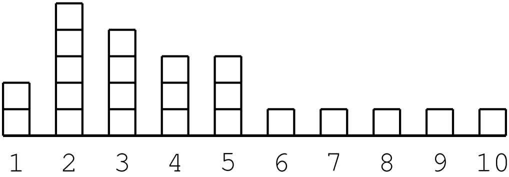
  Assume N is known

- Output k most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!



3 4 6 3 2 10 3 1 3 1 2 2 5 5 5 9 8 7 4 4 2 2

Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$

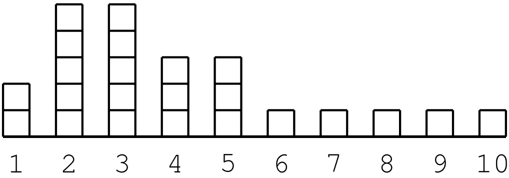
  Assume N is known

- Output k most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!



3 4 6 3 2 10 3 1 3 1 2 2 5 5 5 9 8 7 4 4 2 2 3

Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$

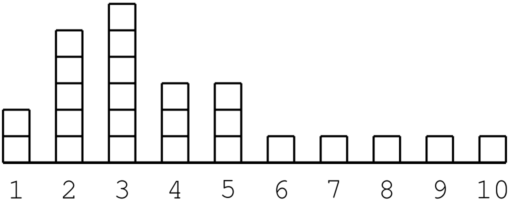
  Assume N is known

- Output k most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!



3 4 6 3 2 10 3 1 3 1 2 2 5 5 5 9 8 7 4 4 2 2 3 3

Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$

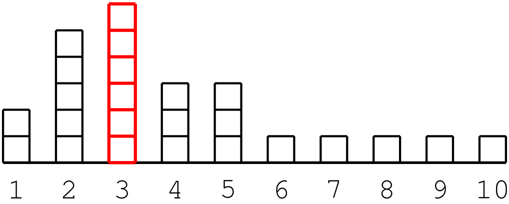
  Assume N is known

- Output k most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!



3 4 6 3 2 10 3 1 3 1 2 2 5 5 5 9 8 7 4 4 2 2 3 3

Estimating IP flows through a router

Estimate the dominant IP flows through a router

$$
\begin{array}{c|ccccccccc}
 & \multicolumn{9}{c}{\texttt{destination}} \\
\hline
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\texttt{source} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

Estimating IP flows through a router

$$\begin{array}{cc} & \text{destination} \\ \text{source} & \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix} \end{array}$$

Estimating IP flows through a router

$$
\begin{array}{c|ccccccccc}
 & \multicolumn{9}{c}{\texttt{destination}} \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\texttt{source} & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

Src	Dst
DATA	

Estimating IP flows through a router

$$
\begin{array}{c|ccccccccc}
 & \multicolumn{9}{c}{\texttt{destination}} \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\texttt{source} & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

Estimating IP flows through a router

$$
\begin{array}{c|c}
 & \texttt{destination} \\
\texttt{source} &
\begin{matrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{matrix}
\end{array}
$$

Src	Dst
DATA	

Estimating IP flows through a router

$$\begin{array}{cc} & \texttt{destination} \\ \texttt{source} & \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix} \end{array}$$

Estimating IP flows through a router

destination

source

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**2**	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	**1**	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Src	Dst
DATA	

Estimating IP flows through a router

destination

	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
source	0	0	0	**2**	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	**1**	0	0	0
	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0

Estimating IP flows through a router

$$
\begin{array}{c}
\text{destination} \\
\begin{array}{cc}
\text{source} &
\begin{matrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \mathbf{2} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{matrix}
\end{array}
\end{array}
$$

Src	Dst
DATA	

Estimating IP flows through a router

				destination					
	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
	0	0	0	**2**	0	0	0	0	0
source	0	0	0	0	0	0	0	0	0
	0	**1**	0	0	0	0	0	0	0
	0	0	0	0	0	**1**	0	0	0
	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0

Estimating IP flows through a router

$$
\begin{array}{cc}
 & \texttt{destination} \\
\texttt{source} & \begin{matrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \mathbf{2} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{matrix}
\end{array}
$$

Src	Dst
DATA	

Estimating IP flows through a router

$$\begin{array}{cc}
 & \texttt{destination} \\
\texttt{source} & \begin{matrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \mathbf{2} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\
0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{matrix}
\end{array}$$

Estimating IP flows through a router

$$
\begin{array}{c c}
 & \text{destination} \\
\text{source} & \begin{matrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \mathbf{3} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\
0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{matrix}
\end{array}
$$

<table>
<tr><td>Src</td><td>Dst</td></tr>
<tr><td colspan="2">DATA</td></tr>
</table>

Estimating IP flows through a router

$$\begin{array}{cc} & \texttt{destination} \\ \texttt{source} & \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{3} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix} \end{array}$$

Estimating IP flows through a router

$$
\begin{array}{c ccccccccc}
 & \multicolumn{9}{c}{\texttt{destination}} \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & \color{red}{1} & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & \mathbf{3} & 0 & 0 & 0 & 0 & 0 \\
\texttt{source} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\
 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

Src	Dst
DATA	

Estimating IP flows through a router

$$
\begin{array}{cc}
 & \texttt{destination} \\
\texttt{source} &
\begin{matrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \mathbf{3} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\
0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{matrix}
\end{array}
$$

Estimating IP flows through a router

$$
\begin{array}{cc}
 & \texttt{destination} \\
\texttt{source} &
\begin{matrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \mathbf{3} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \mathbf{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\
0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{matrix}
\end{array}
$$

Src	Dst
DATA	

Estimating IP flows through a router

$$
\begin{array}{cc}
 & \texttt{destination} \\
\texttt{source} &
\begin{array}{ccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \mathbf{3} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \mathbf{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\
0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
\end{array}
$$

Estimating IP flows through a router

destination

	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	**1**	0	0
	0	0	0	0	0	0	0	0	0
	0	0	0	**3**	0	0	0	0	0
source	0	0	0	0	0	0	0	0	0
	0	**2**	0	0	0	0	0	**1**	0
	0	0	0	0	0	**1**	0	0	0
	0	0	**1**	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0

Src	Dst
DATA	

Estimating IP flows through a router

$$
\begin{array}{c c}
 & \texttt{destination} \\
\texttt{source} & \begin{array}{ccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \mathbf{3} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \color{red}{\mathbf{2}} & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \\
0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\
0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
\end{array}
$$

Estimating IP flows through a router

$$
\begin{array}{c|c}
 & \texttt{destination} \\
\hline
\texttt{source} & \begin{matrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \mathbf{4} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \mathbf{2} & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \\
0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\
0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{matrix}
\end{array}
$$

Src	Dst
DATA	

Estimating IP flows through a router

$$\begin{array}{c|c} & \texttt{destination} \\ \hline \texttt{source} & \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{4} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{2} & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix} \end{array}$$

Estimating IP flows through a router

$$
\begin{array}{c|ccccccccc}
 & \multicolumn{9}{c}{\texttt{destination}} \\
\hline
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & \mathbf{4} & 0 & 0 & 0 & 0 & 0 \\
\texttt{source} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & \color{red}{\mathbf{3}} & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \\
 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\
 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

<table>
<tr><td>Src</td><td>Dst</td></tr>
<tr><td colspan="2">DATA</td></tr>
</table>

Estimating IP flows through a router

$$
\begin{array}{c|ccccccccc}
 & \multicolumn{9}{c}{\texttt{destination}} \\
\hline
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & \mathbf{4} & 0 & 0 & 0 & 0 & 0 \\
\texttt{source} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & \mathbf{3} & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \\
 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\
 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

Estimating IP flows through a router

$$
\begin{array}{c|ccccccccc}
 & \multicolumn{9}{c}{\texttt{destination}} \\
\hline
 & \color{red}{\mathbf{1}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\texttt{source} & 0 & 0 & 0 & \mathbf{4} & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & \mathbf{3} & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \\
 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\
 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

Src	Dst
DATA	

Estimating IP flows through a router

$$
\begin{array}{c|ccccccccc}
 & \multicolumn{9}{c}{\texttt{destination}} \\
\hline
 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & \mathbf{4} & 0 & 0 & 0 & 0 & 0 \\
\texttt{source} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & \mathbf{3} & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \\
 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\
 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

Estimating IP flows through a router

$$\begin{array}{cc}
 & \texttt{destination} \\
\texttt{source} & \begin{matrix}
\mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \mathbf{4} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \mathbf{3} & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \\
0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\
0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{matrix}
\end{array}$$

Estimating IP flows through a router

$$
\begin{array}{c|ccccccccc}
 & \multicolumn{9}{c}{\texttt{destination}} \\
\hline
 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\
\texttt{source} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & \color{red}{4} & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

Src	Dst
DATA	

Estimating IP flows through a router

$$
\begin{array}{cc}
 & \texttt{destination} \\
\texttt{source} &
\begin{array}{ccccccccc}
\mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \mathbf{4} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \mathbf{4} & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \\
0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\
0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
\end{array}
$$

Estimating IP flows through a router

$$
\begin{array}{c|ccccccccc}
 & \multicolumn{9}{c}{\texttt{destination}} \\
 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\texttt{source} & 0 & 0 & 0 & \mathbf{5} & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & \mathbf{4} & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \\
 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\
 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

Src	Dst
DATA	

Estimating IP flows through a router

$$
\begin{array}{cc}
 & \texttt{destination} \\
\texttt{source} &
\begin{array}{ccccccccc}
\mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \mathbf{5} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \mathbf{4} & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \\
0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\
0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
\end{array}
$$

Estimating IP flows through a router

$$
\begin{array}{cc}
 & \texttt{destination} \\
\texttt{source} &
\begin{array}{ccccccccc}
\mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \mathbf{5} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \mathbf{4} & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \\
0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\
0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
\end{array}
$$

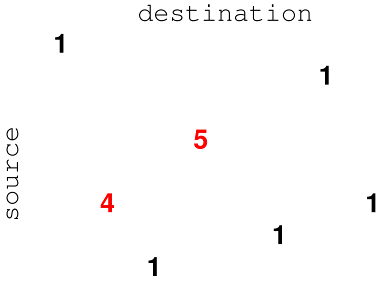
Estimating IP flows through a router

Estimate the dominant IP flows through a router

$$
\begin{array}{c|ccccccccc}
 & \multicolumn{9}{c}{\texttt{destination}} \\
\hline
 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\texttt{source} & 0 & 0 & 0 & 5 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

Estimating IP flows through a router

Estimate the dominant IP flows through a router



Estimating IP flows through a router

Estimate the dominant IP flows through a router

destination

source

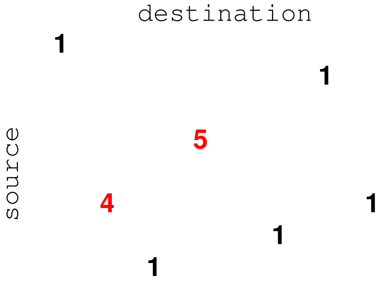
1
1
5
4 1
1
1

Trivial: store all distinct `IP` pairs

Space complexity: $\Theta(N)$

Estimating IP flows through a router

Estimate the dominant IP flows through a router



Trivial: store all distinct `IP` pairs

Space complexity: $\Theta(N)$

This lecture: solve in space $O(\log N)$

Exponential improvement!

Estimating search statistics

Given a set of items as a stream (e.g. queries on `google.com` over a period of time)

`Geneva to NYC, coffee in Geneva, Geneva to NYC`

Estimating search statistics

Given a set of items as a stream (e.g. queries on `google.com` over a period of time)

`Geneva to NYC, coffee in Geneva, Geneva to NYC`

Find the most frequent items in the set

`Geneva to NYC, coffee in Geneva`

Estimating search statistics

Given a set of items as a stream (e.g. queries on `google.com` over a period of time)

`Geneva to NYC, coffee in Geneva, Geneva to NYC`

Find the most frequent items in the set

`Geneva to NYC, coffee in Geneva`

Estimating search statistics

Given a set of items as a stream (e.g. queries on `google.com` over a period of time)

`Geneva to NYC, coffee in Geneva, Geneva to NYC`

Find the most frequent items in the set

`Geneva to NYC, coffee in Geneva`

	Trivial	This lecture
Solution	`hash<string> h;`	COUNTSKETCH
Space	`# of distinct items`	$O(\log N)$

Heavy hitters problem

- Single pass over the data: $i_1, i_2, \ldots, i_N$

  Assume N is known

- Output k most frequent items

  (Heavy hitters)

- Small storage: will get $O(k \log N)$

  Much better than storing all items!

Goal: design a small space data structure

FINDTOP(S, k): returns top k most frequent items seen so far

Goal: design a small space data structure

FINDTOP(S, k): returns top k most frequent items seen so far

Useful to first design

POINTQUERY(S, i): processes stream, then for any query item i can return f_i=number of times item i appeared

Denote the number of times item i appears in the stream by f_i (frequency of i)

Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$

Denote the number of times item i appears in the stream by f_i (frequency of i)

Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$

$\boxed{\text{PointQuery}(S, i) \text{ in space } O(k \log N)?}$

Denote the number of times item i appears in the stream by f_i (frequency of i)

Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$

$\boxed{\textsc{PointQuery}(S, i) \text{ in space } O(k \log N)?}$

Impossible in general...

Imagine a stream where all elements occur with about the same frequency

FINDAPPROXTOP(S, k, ε): returns set of k items such that $f_i \geq (1-\varepsilon) f_k$ for all reported i

APPROXPOINTQUERY(S, i, ε): processes stream, then for any query item i can return approximation $\widehat{f_i} \in [f_i - \varepsilon f_k, f_i + \varepsilon f_k]$

FindApproxTop(S, k, ε): returns set of k items such that $f_i \geq (1-\varepsilon) f_k$ for all reported i

ApproxPointQuery(S, i, ε): processes stream, then for any query item i can return approximation $\widehat{f_i} \in [f_i - \varepsilon f_k, f_i + \varepsilon f_k]$

In this lecture: find most frequent (head) items **if they contribute the bulk of the stream** under some measure

1. Finding top k elements via (APPROX)POINTQUERY
2. Basic version of APPROXPOINTQUERY
3. APPROXPOINTQUERY and the COUNTSKETCH algorithm

1. **Finding top k elements via (APPROX)POINTQUERY**

2. Basic version of APPROXPOINTQUERY

3. APPROXPOINTQUERY and the COUNTSKETCH algorithm

POINTQUERY implies FINDTOP, $k = 1$

Given: stream $S = (i_1, \ldots, i_N)$

POINTQUERY implies FINDTOP, $k = 1$

Given: stream $S = (i_1, \ldots, i_N)$

Maintain: data structure for POINTQUERY

```
currMax=NULL; currFreq=0;
```

POINTQUERY implies FINDTOP, $k = 1$

Given: stream $S = (i_1, \ldots, i_N)$

Maintain: data structure for POINTQUERY

```
currMax=NULL; currFreq=0;
```

For $p = 1, \ldots, N$

Compute frequency $f \leftarrow$ POINTQUERY(i_p)

PointQuery implies FindTop, $k = 1$

Given: stream $S = (i_1, \ldots, i_N)$

Maintain: data structure for PointQuery

```
currMax=NULL; currFreq=0;
```

For $p = 1, \ldots, N$

Compute frequency $f \leftarrow$ PointQuery(i_p)

If `currMax==`i_p

`currFreq=`f`; continue;`

end if

POINTQUERY implies FINDTOP, $k = 1$

Given: stream $S = (i_1, \ldots, i_N)$

Maintain: data structure for POINTQUERY

`currMax=NULL; currFreq=0;`

For $p = 1, \ldots, N$

Compute frequency $f \leftarrow$ POINTQUERY(i_p)

If `currMax==`i_p

`currFreq=`f`; continue;`

end if

If `currMax==NULL`

`currMax=`i_p`; currFreq=1;`

else

If `currFreq<`f

`currMax=`i_p`; currFreq=`f`;`

end if

end if

end for

Why does this work?

At each point in the stream `currMax` is either `NULL` or the most frequent element so far...

Why does this work?

At each point in the stream `currMax` is either `NULL` or the most frequent element so far...

What about finding k most frequent elements for $k > 1$?

PointQuery implies FindTop

Given: stream $S = (i_1, \ldots, i_N)$

PointQuery implies FindTop

Given: stream $S = (i_1, \ldots, i_N)$

Maintain: data structure for PointQuery
a heap H of at most k items, by count

PointQuery implies FindTop

Given: stream $S = (i_1, \ldots, i_N)$

Maintain: data structure for PointQuery
a heap H of at most k items, by count

For $p = 1, \ldots, N$
Compute frequency $f \leftarrow \text{PointQuery}(i_p)$

PointQuery implies FindTop

Given: stream $S = (i_1, \ldots, i_N)$

Maintain: data structure for PointQuery
a heap H of at most k items, by count

For $p = 1, \ldots, N$
Compute frequency $f \leftarrow$ PointQuery(i_p)

If H contains i_p
update i_p's key to f; **continue;**
end if

PointQuery implies FindTop

Given: stream $S = (i_1, \ldots, i_N)$

Maintain: data structure for PointQuery
a heap H of at most k items, by count

For $p = 1, \ldots, N$
  Compute frequency $f \leftarrow$ PointQuery(i_p)
  If H contains i_p
    update i_p's key to f; **continue;**
  end if
  If H contains $< k$ elements,
    add $< f, i_p >$ to H
  else
    $< f_{min}, i_{min} > \leftarrow$ element in H with smallest key
    If $f_{min} < f$, insert $< f, i_p >$ and evict $< f_{min}, i_{min} >$
  end if
end for

Why does this work?

(Assume the set of top k items is unique for simplicity)

For every item i in top k let p_i denote the last position where i occurs

Why does this work?

(Assume the set of top k items is unique for simplicity)

For every item i in top k let p_i denote the last position where i occurs

Note that

1. at position p_i element i is inserted into heap H if it was not in H at that time
2. element i is never evicted after p_i

PointQuery implies FindTop

Given: stream $S = (i_1, \ldots, i_N)$

Maintain: data structure for PointQuery
a heap H of at most k items, by count

For $p = 1, \ldots, N$
Compute frequency $f \leftarrow \text{PointQuery}(i_p)$
If H contains i_p
update i_p's key to f; **continue;**
end if
If H contains $< k$ elements,
add $< f, i_p >$ to H
else
$< f_{min}, i_{min} > \leftarrow$ element in H with smallest key
If $f_{min} < f$, insert $< f, i_p >$ and evict $< f_{min}, i_{min} >$
end if
end for

Why is this useful? We know that PointQuery requires essentially storing the entire stream...

Why is this useful? We know that PointQuery requires essentially storing the entire stream...

A similar reduction shows that ApproxPointQuery implies FindApproxTop!

APPROXPOINTQUERY implies FINDAPPROXTOP

Given: stream $S = (i_1, \ldots, i_N)$

Maintain: data structure for APPROXPOINTQUERY
a heap H of at most k items, by count

For $p = 1, \ldots, N$
Compute frequency $\widehat{f} \leftarrow$ APPROXPOINTQUERY(i_p)
If H contains i_p
update i_p's key to $\widehat{f}$; **continue;**
end if
If H contains $< k$ elements,
add $< \widehat{f}, i_p >$ to H
else
$< \widehat{f}_{min}, i_{min} > \leftarrow$ element in H with smallest key
If $\widehat{f}_{min} < \widehat{f}$, insert $< \widehat{f}, i_p >$ and evict $< \widehat{f}_{min}, i_{min} >$
end if
end for

FINDAPPROXTOP(S, k, ε): returns set S of k items such that $f_i \geq (1-\varepsilon) f_k$ for all $i \in S$

APPROXPOINTQUERY(S, i, ε): returns $\widehat{f_i} \in [f_i - \varepsilon f_k, f_i + \varepsilon f_k]$

In what follows: APPROXPOINTQUERY in small space

Observe a stream of updates, maintain small space data structure

Task: after observing the stream, given $i \in \{1, 2, \ldots, m\}$, compute estimate $\widehat{f}_i$ of f_i

In what follows: APPROXPOINTQUERY in small space

Observe a stream of updates, maintain small space data structure

Task: after observing the stream, given $i \in \{1, 2, \ldots, m\}$, compute estimate $\widehat{f_i}$ of f_i

To be specified:

- space complexity?
- quality of approximation?
- success probability?

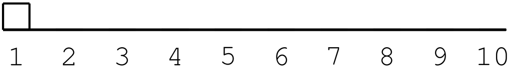
1. Finding top k elements via (APPROX)POINTQUERY
2. Basic version of APPROXPOINTQUERY
3. APPROXPOINTQUERY and the COUNTSKETCH algorithm

1. Finding top k elements via (APPROX)POINTQUERY
2. **Basic version of APPROXPOINTQUERY**
3. APPROXPOINTQUERY and the COUNTSKETCH algorithm

Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$

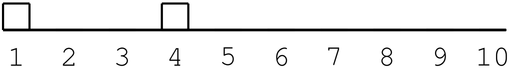
1 2 3 4 5 6 7 8 9 10

Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$



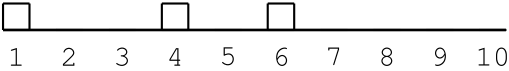
1

Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$



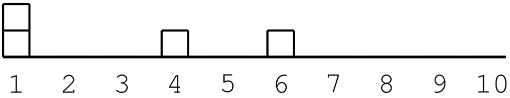
1 4

Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$



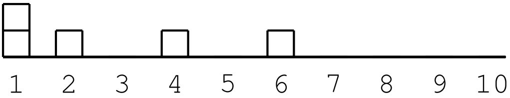
1 4 6

Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$



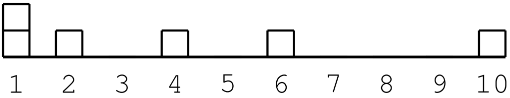
1 4 6 1

Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$



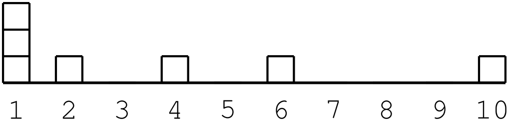
1 4 6 1 2

Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$



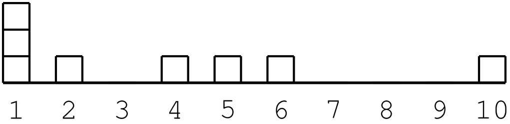
1 4 6 1 2 10

Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$



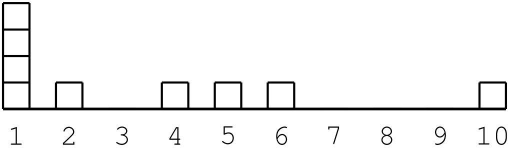
1 4 6 1 2 10 1

Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$



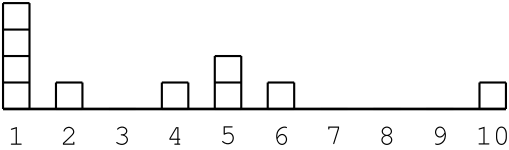
1 4 6 1 2 10 1 5

Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$



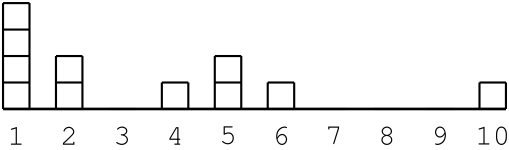
1 4 6 1 2 10 1 5 1

Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$



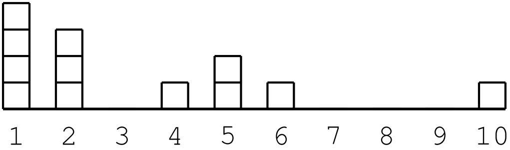
1 4 6 1 2 10 1 5 1 5

Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$



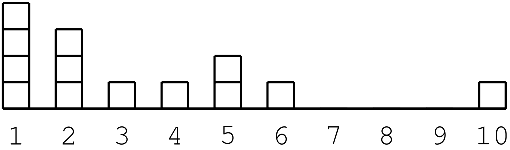
1 4 6 1 2 10 1 5 1 5 2

Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$



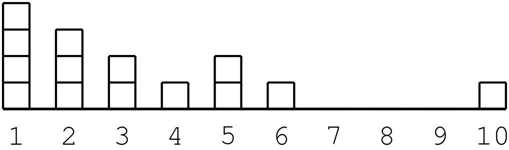
1 4 6 1 2 10 1 5 1 5 2 2

Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$



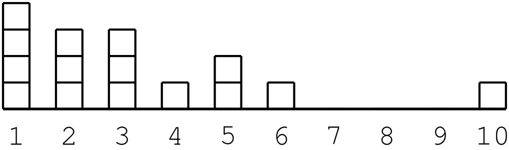
1 4 6 1 2 10 1 5 1 5 2 2 3

Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$



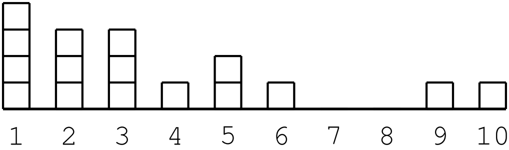
1 4 6 1 2 10 1 5 1 5 2 2 3 3

Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$



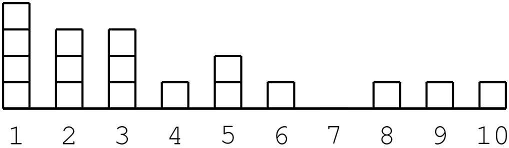
1 4 6 1 2 10 1 5 1 5 2 2 3 3 3

Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$



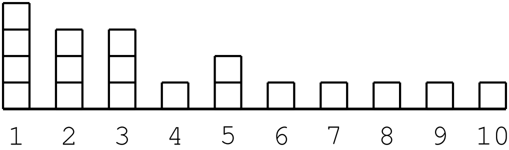
1 4 6 1 2 10 1 5 1 5 2 2 3 3 3 9

Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$



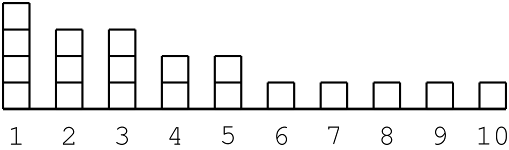
1 4 6 1 2 10 1 5 1 5 2 2 3 3 3 9 8

Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$



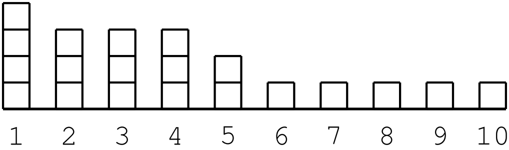
1 4 6 1 2 10 1 5 1 5 2 2 3 3 3 9 8 7

Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$



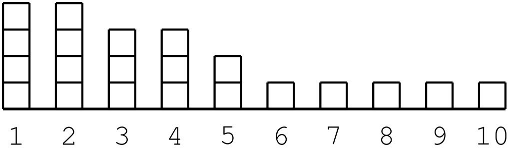
1 4 6 1 2 10 1 5 1 5 2 2 3 3 3 9 8 7 4

Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$



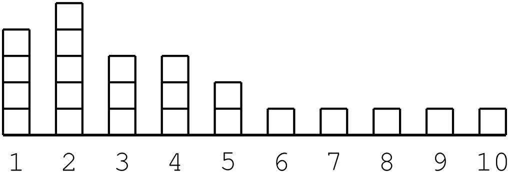
1 4 6 1 2 10 1 5 1 5 2 2 3 3 3 9 8 7 4 4

Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$



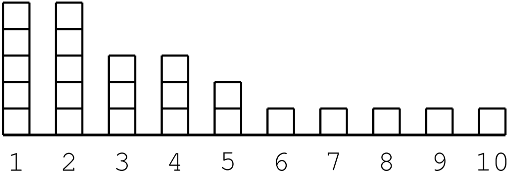
1 4 6 1 2 10 1 5 1 5 2 2 3 3 3 9 8 7 4 4 2

Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$



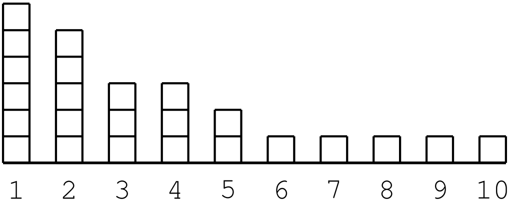
1 4 6 1 2 10 1 5 1 5 2 2 3 3 3 9 8 7 4 4 2 2

Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$



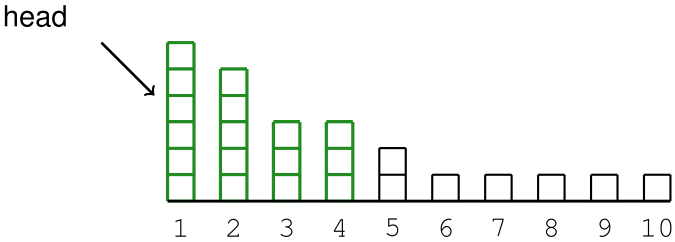
1 4 6 1 2 10 1 5 1 5 2 2 3 3 3 9 8 7 4 4 2 2 1

Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$



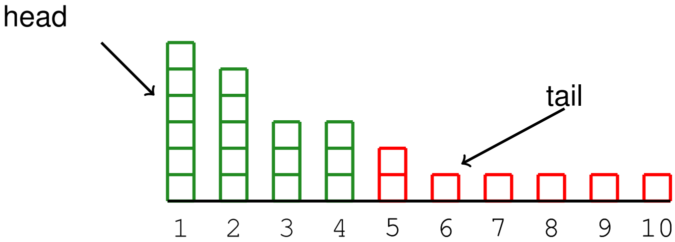
1 4 6 1 2 10 1 5 1 1 2 2 3 3 3 9 8 7 4 4 2 2 1 5

Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$



1 4 6 1 2 10 1 5 1 1 2 2 3 3 3 9 8 7 4 4 2 2 1 5

Assume elements are ordered by frequency: $f_1 \geq f_2 \geq \ldots \geq f_m$



1 4 6 1 2 10 1 5 1 1 2 2 3 3 3 9 8 7 4 4 2 2 1 5

Basic estimate

Will design a basic estimate with $O(1)$ space complexity, analyze precision

Basic estimate

Will design a basic estimate with $O(1)$ space complexity, analyze precision

Choose a hash function $s : [m] \to \{-1, +1\}$ uniformly at random

INITIALIZE
$C \leftarrow 0$

UPDATE(C, i)
$C \leftarrow C + s(i)$

Basic estimate

Will design a basic estimate with $O(1)$ space complexity, analyze precision

Choose a hash function $s : [m] \to \{-1, +1\}$ uniformly at random

INITIALIZE
$C \leftarrow 0$

UPDATE(C, i)
$C \leftarrow C + s(i)$

for every $p = 1, \ldots, N$ (every element in the stream)
UPDATE(C, i_p)
end for

Basic estimate

Will design a basic estimate with $O(1)$ space complexity, analyze precision

Choose a hash function $s : [m] \to \{-1, +1\}$ uniformly at random

INITIALIZE
  $C \leftarrow 0$

UPDATE(C, i)
  $C \leftarrow C + s(i)$

for every $p = 1, \ldots, N$ (every element in the stream)
  UPDATE(C, i_p)
end for

ESTIMATE(C, i)
  return $C \cdot s(i)$

How does one argue that a randomized estimate works?

UPDATE(C, i)

$C \leftarrow C + s(i)$

ESTIMATE(C, i)

return $C \cdot s(i)$

$\boxed{\text{How does one argue that a randomized estimate works?}}$

UPDATE(C, i)
$C \leftarrow C + s(i)$

ESTIMATE(C, i)
return $C \cdot s(i)$

How does one argue that a randomized estimate works?

Want to show that $C \cdot s(i)$ is close to f_i ‘with high probability’

UPDATE(C, i)
$C \leftarrow C + s(i)$

ESTIMATE(C, i)
return $C \cdot s(i)$

How does one argue that a randomized estimate works?

Want to show that $C \cdot s(i)$ is close to f_i 'with high probability'

Typically show this in two steps:

- show that $\mathbf{E}_s[C \cdot s(i)] = f_i$
  (so $C \cdot s(i)$ is an unbiased estimate of f_i)
- show that $\mathbf{Var}_s[C \cdot s(i)]$ is 'small'

UPDATE(C, i)
$C \leftarrow C + s(i)$

ESTIMATE(C, i)
return $C \cdot s(i)$

How does one argue that a randomized estimate works?

Want to show that $C \cdot s(i)$ is close to f_i 'with high probability'

Typically show this in two steps:

- show that $\mathbf{E}_s[C \cdot s(i)] = f_i$
  (so $C \cdot s(i)$ is an unbiased estimate of f_i)
- show that $\mathbf{Var}_s[C \cdot s(i)]$ is 'small'

It then follows that $|C \cdot s(i) - f_i|$ is 'small' with high probability
(essentially law of large numbers)

Basic estimate:mean

UPDATE(C, i)
$C \leftarrow C + s(i)$

ESTIMATE(C, i)
return $C \cdot s(i)$

$$C \cdot s(i) = \sum_{p=1}^{N} s(i_p) s(i)$$

Basic estimate:mean

UPDATE(C, i)

$C \leftarrow C + s(i)$

ESTIMATE(C, i)

return $C \cdot s(i)$

$$C \cdot s(i) = \sum_{p=1}^{N} s(i_p) s(i) = \sum_{j \in [m]} f_j \cdot s(j) s(i)$$

Basic estimate:mean

UPDATE(C, i)

$C \leftarrow C + s(i)$

ESTIMATE(C, i)

return $C \cdot s(i)$

$$C \cdot s(i) = \sum_{p=1}^{N} s(i_p)s(i) = \sum_{j \in [m]} f_j \cdot s(j)s(i)$$

$$= f_i s(i)^2 + \sum_{j \in [m] \setminus i} f_j \cdot s(j)s(i)$$

Basic estimate:mean

UPDATE(C, i)
$C \leftarrow C + s(i)$

ESTIMATE(C, i)
return $C \cdot s(i)$

$$
\begin{aligned}
C \cdot s(i) = \sum_{p=1}^{N} s(i_p)s(i) &= \sum_{j \in [m]} f_j \cdot s(j)s(i) \\
&= f_i s(i)^2 + \sum_{j \in [m] \setminus i} f_j \cdot s(j)s(i) \\
&= f_i + \sum_{j \in [m] \setminus i} f_j \cdot s(j)s(i) \quad \longleftarrow \text{random } \pm 1\text{'s}
\end{aligned}
$$

Basic estimate:mean

UPDATE(C, i)
$C \leftarrow C + s(i)$

ESTIMATE(C, i)
return $C \cdot s(i)$

$$C \cdot s(i) = f_i + \sum_{j \in [m] \setminus i} f_j \cdot s(j) s(i)$$

Basic estimate:mean

UPDATE(C, i)
$C \leftarrow C + s(i)$

ESTIMATE(C, i)
return $C \cdot s(i)$

$$\mathbf{E}[C \cdot s(i)] = f_i + \mathbf{E}[\sum_{j \in [m] \setminus i} f_j \cdot s(j) s(i)]$$

Basic estimate:mean

UPDATE(C, i)
$C \leftarrow C + s(i)$

ESTIMATE(C, i)
return $C \cdot s(i)$

$$
\begin{aligned}
\mathbf{E}[C \cdot s(i)] &= f_i + \mathbf{E}[\sum_{j \in [m] \setminus i} f_j \cdot s(j) s(i)] \\
&= f_i + \sum_{j \in [m] \setminus i} f_j \cdot \mathbf{E}[s(j)] \mathbf{E}[s(i)] \quad \text{(by independence of } s(i)\text{)}
\end{aligned}
$$

Basic estimate:mean

UPDATE(C, i)
$C \leftarrow C + s(i)$

ESTIMATE(C, i)
return $C \cdot s(i)$

$$\begin{aligned}
\mathbf{E}[C \cdot s(i)] &= f_i + \mathbf{E}[\sum_{j \in [m] \setminus i} f_j \cdot s(j)s(i)] \\
&= f_i + \sum_{j \in [m] \setminus i} f_j \cdot \mathbf{E}[s(j)]\mathbf{E}[s(i)] \quad \text{(by independence of } s(i)\text{)} \\
&= f_i
\end{aligned}$$

Basic estimate:mean

UPDATE(C, i)
$C \leftarrow C + s(i)$

ESTIMATE(C, i)
return $C \cdot s(i)$

$$
\begin{aligned}
\mathbf{E}[C \cdot s(i)] &= f_i + \mathbf{E}[\sum_{j \in [m] \setminus i} f_j \cdot s(j) s(i)] \\
&= f_i + \sum_{j \in [m] \setminus i} f_j \cdot \mathbf{E}[s(j)] \mathbf{E}[s(i)] \quad \text{(by independence of } s(i)\text{)} \\
&= f_i
\end{aligned}
$$

The mean is correct: our estimator is unbiased!

Basic estimate:mean

UPDATE(C, i)
$C \leftarrow C + s(i)$

ESTIMATE(C, i)
return $C \cdot s(i)$

$$
\begin{aligned}
\mathbf{E}[C \cdot s(i)] &= f_i + \mathbf{E}[\sum_{j \in [m] \setminus i} f_j \cdot s(j) s(i)] \\
&= f_i + \sum_{j \in [m] \setminus i} f_j \cdot \mathbf{E}[s(j)] \mathbf{E}[s(i)] \quad \text{(by independence of } s(i)\text{)} \\
&= f_i
\end{aligned}
$$

The mean is correct: our estimator is unbiased!

Is the estimate $C \cdot s(i)$ close to f_i with high probability?

Chebyshev's inequality

Theorem

For every random variable X with mean μ and variance σ^2, and every $t \geq 1$ one has

$$\mathbf{Pr}[|X - \mu| \geq t \cdot \sigma] \leq 1/t^2$$

Chebyshev's inequality

Theorem

For every random variable X with mean μ and variance σ^2, and every $t \geq 1$ one has

$$\mathbf{Pr}[|X - \mu| \geq t \cdot \sigma] \leq 1/t^2$$

Apply Chebyshev's inequality with $X = C \cdot s(i)$ and $\mu = \mathbf{E}[C \cdot s(i)]$?

Chebyshev's inequality

Theorem

For every random variable X with mean μ and variance σ^2, and every $t \geq 1$ one has

$$\mathbf{Pr}[|X - \mu| \geq t \cdot \sigma] \leq 1/t^2$$

Apply Chebyshev's inequality with $X = C \cdot s(i)$ and $\mu = \mathbf{E}[C \cdot s(i)]$?

A quantitative form of the 'law of large numbers'

Chebyshev's inequality

Theorem

For every random variable X with mean μ and variance σ^2, and every $t \geq 1$ one has

$$\mathbf{Pr}[|X - \mu| \geq t \cdot \sigma] \leq 1/t^2$$

Apply Chebyshev's inequality with $X = C \cdot s(i)$ and $\mu = \mathbf{E}[C \cdot s(i)]$?

A quantitative form of the 'law of large numbers'

Need to compute the variance $\sigma^2 = \mathbf{E}[(C \cdot s(i) - \mu)^2]$

Basic estimate: variance

UPDATE(C, i)
$C \leftarrow C + s(i)$

ESTIMATE(C, i)
return $C \cdot s(i)$

We have

$$C \cdot s(i) = f_i + \sum_{j \in [m] \setminus i} f_j \cdot s(j) s(i)$$

and

$$\mathbf{E}[C \cdot s(i)] = f_i.$$

Basic estimate: variance

UPDATE(C, i)
$C \leftarrow C + s(i)$

ESTIMATE(C, i)
return $C \cdot s(i)$

We have

$$C \cdot s(i) = f_i + \sum_{j \in [m] \setminus i} f_j \cdot s(j) s(i)$$

and

$$\mathbf{E}[C \cdot s(i)] = f_i.$$

We need to bound

$$\begin{aligned} \mathbf{Var}(C \cdot s(i)) &= \mathbf{E}[(C \cdot s(i) - \mathbf{E}[C \cdot s(i)])^2] \\ &= \mathbf{E}[(C \cdot s(i) - f_i)^2] \\ &= \mathbf{E}\left[\left(\sum_{j \in [m] \setminus i} f_j \cdot s(j) s(i)\right)^2\right] \end{aligned}$$

Basic estimate: variance

UPDATE(C, i)
$C \leftarrow C + s(i)$

ESTIMATE(C, i)
return $C \cdot s(i)$

$$
\begin{aligned}
(C \cdot s(i) - f_i)^2 &= \left(\sum_{j \in [m] \setminus i} f_j \cdot s(j) s(i) \right)^2 \\
&= \sum_{j \in [m] \setminus i} \sum_{j' \in [m] \setminus i} f_j f_{j'} \cdot s(j) s(j') \cdot s^2(i) \\
&= \sum_{j \in [m] \setminus i} \sum_{j' \in [m] \setminus i} f_j f_{j'} \cdot s(j) s(j')
\end{aligned}
$$

Basic estimate: variance

UPDATE(C, i)
$C \leftarrow C + s(i)$

ESTIMATE(C, i)
return $C \cdot s(i)$

$$
\begin{aligned}
\mathbf{E}[(C \cdot s(i) - f_i)^2] &= \mathbf{E}[\sum_{j \in [m] \setminus i} \sum_{j' \in [m] \setminus i} f_j f_{j'} \cdot s(j) s(j')] \\
&= \sum_{j \in [m] \setminus i} \sum_{j' \in [m] \setminus i} f_j f_{j'} \cdot \mathbf{E}[s(j) s(j')] \\
&= \sum_{j \in [m] \setminus i} f_j^2
\end{aligned}
$$

since

- $s(j)^2 = 1$ for all j
- $\mathbf{E}[s(j)s(j')] = \mathbf{E}[s(j)]\mathbf{E}[s(j')] = 0$ for $j \neq j'$.

Basic estimate: variance

UPDATE(C, i)
$C \leftarrow C + s(i)$

ESTIMATE(C, i)
return $C \cdot s(i)$

Basic estimate: variance

UPDATE(C, i)
$C \leftarrow C + s(i)$

ESTIMATE(C, i)
return $C \cdot s(i)$

We have proved that

$$\mathbf{Var}(C \cdot s(i)) = \mathbf{E}[(C \cdot s(i) - f_i)^2] = \sum_{j \in [m] \setminus i} f_j^2$$

Basic estimate: variance

UPDATE(C, i)
$C \leftarrow C + s(i)$

ESTIMATE(C, i)
return $C \cdot s(i)$

We have proved that

$$\mathbf{Var}(C \cdot s(i)) = \mathbf{E}[(C \cdot s(i) - f_i)^2] = \sum_{j \in [m] \setminus i} f_j^2$$

By Chebyshev's inequality

$$\mathbf{Pr}\left[|C \cdot s(i) - f_i| \geq 8 \cdot \sqrt{\sum_{j \in [m] \setminus i} f_j^2} \right] \leq 1/64$$

Basic estimate: variance

UPDATE(C, i)
$C \leftarrow C + s(i)$

ESTIMATE(C, i)
return $C \cdot s(i)$

We have proved that

$$\mathbf{Var}(C \cdot s(i)) = \mathbf{E}[(C \cdot s(i) - f_i)^2] = \sum_{j \in [m] \setminus i} f_j^2$$

By Chebyshev's inequality

$$\mathbf{Pr}\left[|C \cdot s(i) - f_i| \geq 8 \cdot \sqrt{\sum_{j \in [m] \setminus i} f_j^2} \right] \leq 1/64$$

$\boxed{\text{So } C \cdot s(i) \text{ is close (?) to } f_i \text{ with high probability}}$

Basic estimate: variance

UPDATE(C, i)
$C \leftarrow C + s(i)$

ESTIMATE(C, i)
return $C \cdot s(i)$

We have proved that

$$\mathbf{Var}(C \cdot s(i)) = \mathbf{E}[(C \cdot s(i) - f_i)^2] = \sum_{j \in [m] \setminus i} f_j^2$$

By Chebyshev's inequality

$$\mathbf{Pr}\left[|C \cdot s(i) - f_i| > \mathbf{8} \cdot \sqrt{\sum_{\mathbf{j} \in [\mathbf{m}] \setminus \mathbf{i}} \mathbf{f_j^2}} \right] \leq 1/64$$

So $C \cdot s(i)$ is close **(?)** to f_i with high probability

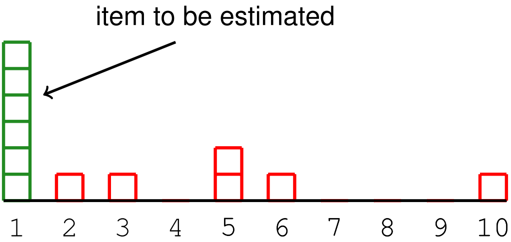
Basic estimate: summary

UPDATE(C, i)
$C \leftarrow C + s(i)$

ESTIMATE(C, i)
return $C \cdot s(i)$

Estimate f_i up to

$$8 \cdot \sqrt{\sum_{j \in [m] \setminus i} f_j^2}$$



Pro: works well for most frequent item, if other items are small

Basic estimate: summary

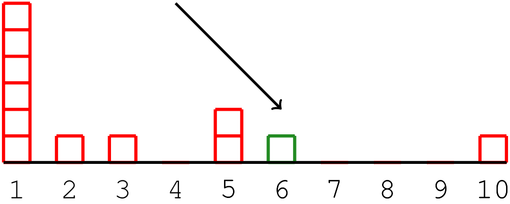
UPDATE(C, i)
$C \leftarrow C + s(i)$

ESTIMATE(C, i)
return $C \cdot s(i)$

Estimate f_i up to

$$8 \cdot \sqrt{\sum_{j \in [m] \setminus i} f_j^2}$$

item to be estimated



Pro: works well for most frequent item, if other items are small

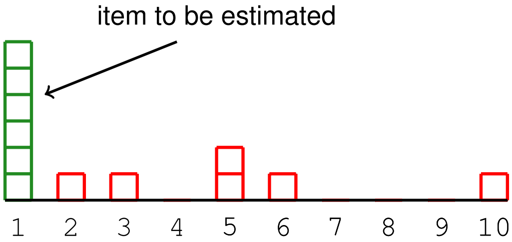
Con: estimate for a small items contaminated by large items

Next: final APPROXPOINTQUERY and the COUNTSKETCH algorithm

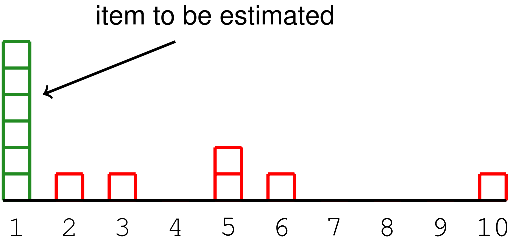
COUNTSKETCH algorithm: find top k elements (approximately)

- hash items into $O(k)$ buckets (i.e. substreams)
- run simple estimate on every bucket
- repeat $O(\log N)$ times independently, take median as answer

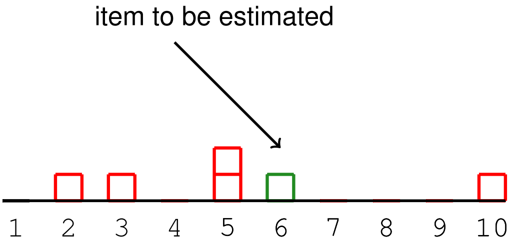
Main intuition: estimate large items from substreams like



Main intuition: estimate large items from substreams like



and small items from substreams like



1. Finding top k elements via (APPROX)POINTQUERY
2. Basic version of APPROXPOINTQUERY
3. APPROXPOINTQUERY and the COUNTSKETCH algorithm

1. Finding top k elements via (APPROX)POINTQUERY
2. Basic version of APPROXPOINTQUERY
3. **APPROXPOINTQUERY and the COUNTSKETCH algorithm**

ApproxPointQuery and CountSketch

Main ideas:

1. run basic estimate on subsampled/hashed stream (reduces variance)

ApproxPointQuery and CountSketch

Main ideas:

1. run basic estimate on subsampled/hashed stream (reduces variance)

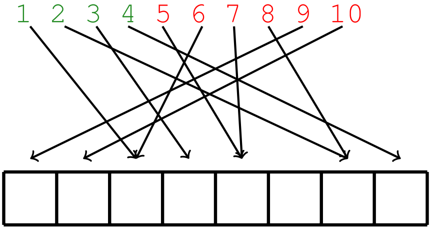
2. aggregate independent estimates to boost confidence (take medians)

APPROXPOINTQUERY and COUNTSKETCH

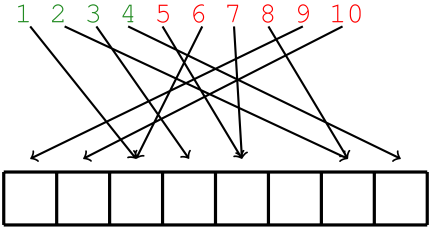
Main ideas:

1. run basic estimate on subsampled/hashed stream
   (reduces variance)

2. aggregate independent estimates to boost confidence
   (take medians)



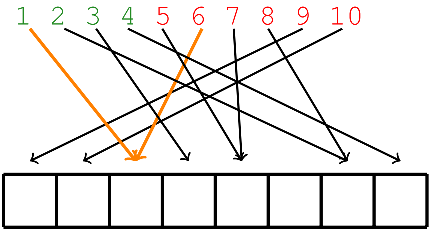
Hashing the items



Hashed into $b = 8$ buckets, get 8 subsampled streams

For item i its stream consists of $j \in [m]$ such that $h(j) = h(i)$

Hashing the items



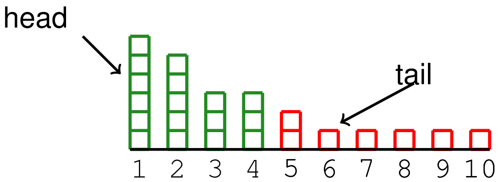
Hashed into $b = 8$ buckets, get 8 subsampled streams

For item i its stream consists of $j \in [m]$ such that $h(j) = h(i)$

For example,

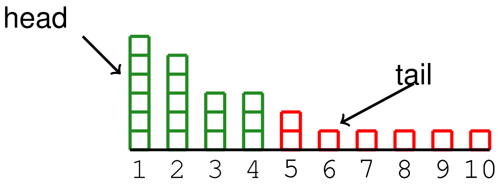
- subsampled stream of item 1 is {1, 6}
- subsampled stream of item 5 is {5, 7}

Note: hashing the universe [m], not positions in the stream



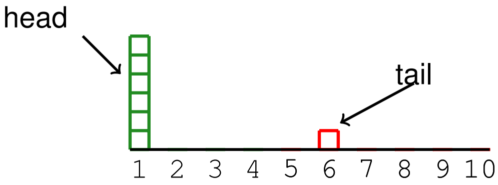
1 4 6 1 2 10 1 5 1 1 2 2 3 3 3 9 8 7 4 4 2 2 1 5

Note: hashing the universe $[m]$, not positions in the stream



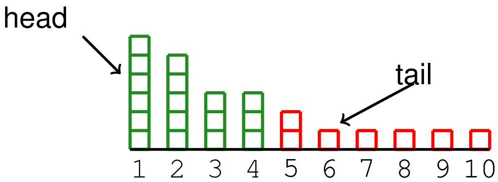
1 4 6 1 2 10 1 5 1 1 2 2 3 3 3 9 8 7 4 4 2 2 1 5

E.x. the subsampled stream of item 1 is {1, 6}



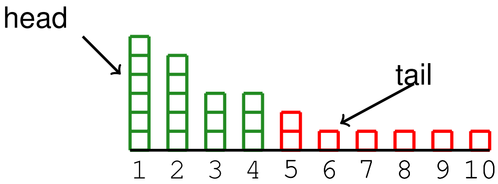
1 6 1 1 1 1 1

Note: hashing the universe [m], not positions in the stream



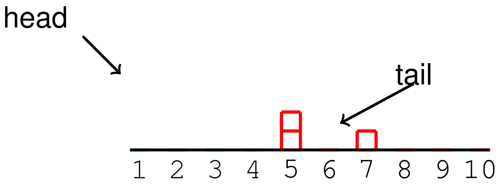
1 4 6 1 2 10 1 5 1 1 2 2 3 3 3 9 8 7 4 4 2 2 1 5

Note: hashing the universe $[m]$, not positions in the stream



1 4 6 1 2 10 1 5 1 1 2 2 3 3 3 9 8 7 4 4 2 2 1 5

E.x. the subsampled stream of item 5 is {5, 7}

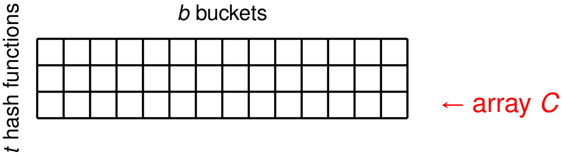


5 7 5

Final ApproxPointQuery

Choose

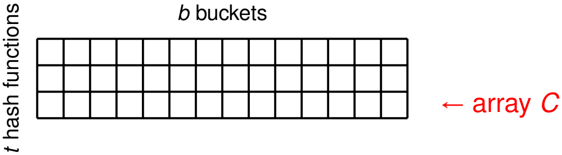
- t random hash functions $h_1, h_2, \ldots, h_t$ from items $[m]$ to $b \approx k$ buckets $\{1, 2, \ldots, b\}$
- t random hash functions $s_1, s_2, \ldots, s_t$ from items $[m]$ to $\{-1, +1\}$



Final ApproxPointQuery

Choose

- t random hash functions $h_1, h_2, \ldots, h_t$ from items $[m]$ to $b \approx k$ buckets $\{1, 2, \ldots, b\}$
- t random hash functions $s_1, s_2, \ldots, s_t$ from items $[m]$ to $\{-1, +1\}$



The algorithm runs t independent copies of basic estimate:

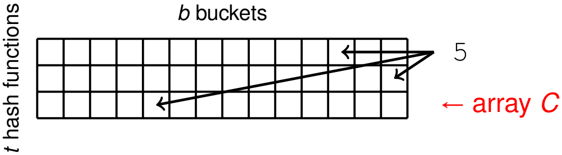
UPDATE(C, i)
for $r \in [1:t]$
$\quad C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i)$
end for

ESTIMATE(C, i)
return $\text{median}_r \{C[r, h_r(i)] \cdot s_r(i)\}$

Final ApproxPointQuery

Choose

- t random hash functions $h_1, h_2, \ldots, h_t$ from items $[m]$ to $b \approx k$ buckets $\{1, 2, \ldots, b\}$
- t random hash functions $s_1, s_2, \ldots, s_t$ from items $[m]$ to $\{-1, +1\}$



The algorithm runs t independent copies of basic estimate:

UPDATE(C, i)
for $r \in [1:t]$
$\quad C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i)$
end for

ESTIMATE(C, i)
return $\text{median}_r \{C[r, h_r(i)] \cdot s_r(i)\}$

UPDATE(C, i)
for $r \in [1:t]$
$\quad C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i)$
end for

ESTIMATE(C, i)
return $\text{median}_r \{C[r, h_r(i)] \cdot s_r(i)\}$

By basic estimate analysis for every $r \in [1:t]$

$$\mathbf{E}[C[r, h_r(i)] \cdot s_r(i)] = f_i$$

UPDATE(C, i)
for $r \in [1:t]$
$C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i)$
end for

ESTIMATE(C, i)
return $\text{median}_r \{C[r, h_r(i)] \cdot s_r(i)\}$

By basic estimate analysis for every $r \in [1:t]$

$$\mathbf{E}[C[r, h_r(i)] \cdot s_r(i)] = f_i$$

and

$$\mathbf{E}_s[(C[r, h_r(i)] \cdot s_r(i) - f_i)^2] = \sum_{j \neq i : \mathbf{h_r(j) = h_r(i)}} f_j^2$$

UPDATE(C, i)
for $r \in [1:t]$
$C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i)$
end for

ESTIMATE(C, i)
return $\text{median}_r \{C[r, h_r(i)] \cdot s_r(i)\}$

By basic estimate analysis for every $r \in [1:t]$

$$\mathbf{E}[C[r, h_r(i)] \cdot s_r(i)] = f_i$$

and

$$\mathbf{E}_s[(C[r, h_r(i)] \cdot s_r(i) - f_i)^2] = \sum_{j \neq i : \mathbf{h_r(j) = h_r(i)}} f_j^2$$

How large can the variance be? Can be be reduced by making number of buckets b large?

UPDATE(C, i)
for $r \in [1:t]$
$$C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i)$$
end for

ESTIMATE(C, i)
return $\text{median}_r \{C[r, h_r(i)] \cdot s_r(i)\}$

By basic estimate analysis for every $r \in [1:t]$

$$\mathbf{E}[C[r, h_r(i)] \cdot s_r(i)] = f_i$$

and

$$\mathbf{E}_s[(C[r, h_r(i)] \cdot s_r(i) - f_i)^2] = \sum_{j \neq i : \mathbf{h_r(j) = h_r(i)}} f_j^2$$

How large can the variance be? Can be be reduced by making number of buckets b large?

YES: hashing into a sufficiently large number of buckets reduces estimation error to below $\varepsilon \cdot f_k$

UPDATE(C, i)
for $r \in [1:t]$
$\quad C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i)$
end for

ESTIMATE(C, i)
return $\text{median}_r \{C[r, h_r(i)] \cdot s_r(i)\}$

By basic estimate analysis for every $r \in [1:t]$

$$\mathbf{E}[C[r, h_r(i)] \cdot s_r(i)] = f_i$$

and

$$\mathbf{E}_s[(C[r, h_r(i)] \cdot s_r(i) - f_i)^2] = \sum_{j \neq i : \mathbf{h_r(j) = h_r(i)}} f_j^2$$

How large can the variance be? Can be be reduced by making number of buckets b large?

YES: hashing into a sufficiently large number of buckets reduces estimation error to below $\varepsilon \cdot f_k$

$O(\log N)$ repetitions ensure estimates are correct **for all** i with high probability

UPDATE(C, i)
for $r \in [1:t]$
$C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i)$
end for

ESTIMATE(C, i)
return $\text{median}_r \{C[r, h_r(i)] \cdot s_r(i)\}$

UPDATE(C, i)
for $r \in [1 : t]$
$\quad C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i)$
end for

ESTIMATE(C, i)
return $\text{median}_r \{C[r, h_r(i)] \cdot s_r(i)\}$

Lemma

If $b \geq 8 \max\left\{k, \frac{32 \sum_{j \in TAIL} f_j^2}{(\varepsilon f_k)^2}\right\}$ *and* $t = O(\log N)$*, then for every* $i \in [m]$

$$|\mathit{median}_r \{C[r, h_r(i)] \cdot s_r(i)\} - f_i(p)| \leq \varepsilon f_k$$

at every point $p \in [1 : N]$ *in the stream.*

($f_i(p)$ is the frequency of i up to position p)

UPDATE(C, i)
for $r \in [1:t]$
  $C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i)$
end for

ESTIMATE(C, i)
return $\text{median}_r \{C[r, h_r(i)] \cdot s_r(i)\}$

Lemma

If $b \geq 8 \max\left\{k, \frac{32 \sum_{j \in TAIL} f_j^2}{(\varepsilon f_k)^2}\right\}$ *and* $t = O(\log N)$, *then for every* $i \in [m]$

$$|\mathit{median}_r \{C[r, h_r(i)] \cdot s_r(i)\} - f_i(p)| \leq \varepsilon f_k$$

at every point $p \in [1:N]$ *in the stream.*

($f_i(p)$ is the frequency of i up to position p)

Space complexity is $O(b \log N)$

UPDATE(C, i)
for $r \in [1:t]$
$C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i)$
end for

ESTIMATE(C, i)
return $\text{median}_r \{C[r, h_r(i)] \cdot s_r(i)\}$

Lemma

If $b \geq 8 \max\left\{k, \frac{32 \sum_{j \in TAIL} f_j^2}{(\varepsilon f_k)^2}\right\}$ *and* $t = O(\log N)$, *then for every* $i \in [m]$

$$|median_r \{C[r, h_r(i)] \cdot s_r(i)\} - f_i(p)| \leq \varepsilon f_k$$

at every point $p \in [1:N]$ *in the stream.*

($f_i(p)$ is the frequency of i up to position p)

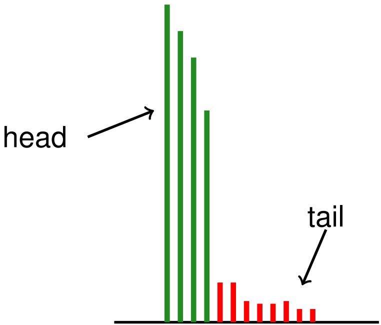
Space complexity is $O(b \log N)$

How large is b?

Space complexity

Set $b = 8\max\left\{k, \frac{32\sum_{j\in TAIL} f_j^2}{(\varepsilon f_k)^2}\right\}$

Note that $b = O(k/\varepsilon^2)$ if $\frac{1}{k}\sum_{j\in TAIL} f_j^2 = O(f_k^2)$



In practice, choose b subject to space constraints, detect elements with counts above $O\left(\varepsilon\sqrt{\frac{1}{k}\sum_{j\in TAIL} f_j^2}\right)$

Space complexity

Set $k = 1$. Suppose that 1 appears $\sqrt{N}$ times in the stream, and other $N - \sqrt{N}$ elements are distinct

Space complexity

Set $k = 1$. Suppose that `1` appears $\sqrt{N}$ times in the stream, and other $N - \sqrt{N}$ elements are distinct

Then $f_1 = \sqrt{N}$, $f_i = 1$ for $i = 2, N - \sqrt{N}$.

$$\text{Set } b = 8\max\left\{1, \frac{32\sum_{j\in TAIL} f_j^2}{(\varepsilon f_1)^2}\right\}$$

Space complexity

Set $k = 1$. Suppose that `1` appears $\sqrt{N}$ times in the stream, and other $N - \sqrt{N}$ elements are distinct

Then $f_1 = \sqrt{N}$, $f_i = 1$ for $i = 2, N - \sqrt{N}$.

$$\text{Set } b = 8 \max\left\{1, \frac{32 \sum_{j \in TAIL} f_j^2}{(\varepsilon f_1)^2}\right\}$$

$$\boxed{\text{We have } \sum_{j \in TAIL} f_j^2 = N - \sqrt{N} \leq N, \text{ and } f_1^2 = N}$$

Space complexity

Set $k = 1$. Suppose that 1 appears $\sqrt{N}$ times in the stream, and other $N - \sqrt{N}$ elements are distinct

Then $f_1 = \sqrt{N}$, $f_i = 1$ for $i = 2, N - \sqrt{N}$.

$$\text{Set } b = 8 \max \left\{ 1, \frac{32 \sum_{j \in TAIL} f_j^2}{(\varepsilon f_1)^2} \right\}$$

$$\boxed{\text{We have } \sum_{j \in TAIL} f_j^2 = N - \sqrt{N} \leq N, \text{ and } f_1^2 = N}$$

$$\text{So } b = 8 \max \left\{ 1, \frac{32 \sum_{j \in TAIL} f_j^2}{(\varepsilon f_1)^2} \right\} = O(1/\varepsilon^2) \text{ suffices}$$

Space complexity

Set $k = 1$. Suppose that `1` appears $\sqrt{N}$ times in the stream, and other $N - \sqrt{N}$ elements are distinct

Then $f_1 = \sqrt{N}$, $f_i = 1$ for $i = 2, N - \sqrt{N}$.

Set $b = 8 \max\left\{1, \frac{32 \sum_{j \in TAIL} f_j^2}{(\varepsilon f_1)^2}\right\}$

$\boxed{\text{We have } \sum_{j \in TAIL} f_j^2 = N - \sqrt{N} \leq N, \text{ and } f_1^2 = N}$

So $b = 8 \max\left\{1, \frac{32 \sum_{j \in TAIL} f_j^2}{(\varepsilon f_1)^2}\right\} = O(1/\varepsilon^2)$ suffices

Remarkable, as `1` appears only in $\sqrt{N}$ positions out of N: a vanishingly small fraction of positions!

Final algorithm: COUNTSKETCH

FINDAPPROXTOP(S, k, ε): returns set of k items such that $f_i \geq (1-\varepsilon) f_k$ for all returned i

(In fact also every i with $f_i \geq (1-\varepsilon) f_k$ is reported)

APPROXPOINTQUERY(S, i, ε): returns $\widehat{f_i} \in [f_i - \varepsilon f_k, f_i + \varepsilon f_k]$

Find head items if they contribute the bulk of the stream in ℓ_2 sense

CountSketch: proof details

UPDATE(C, i)
for $r \in [1:t]$
$\quad C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i)$
end for

ESTIMATE(C, i)
return $\text{median}_r \{C[r, h_r(i)] \cdot s_r(i)\}$

Lemma

If $b \geq 8 \max\left\{k, \frac{32 \sum_{j \in TAIL} f_j^2}{(\varepsilon f_k)^2}\right\}$ *and* $t \geq A \log N$ *for an absolute constant* $A > 0$*, then for every* $i \in [m]$

$$\left| \text{median}_r \{C[r, h_r(i)] \cdot s_r(i)\} - f_i(p) \right| \leq \varepsilon f_k$$

at every point $p \in [1:N]$ *in the stream with high probability.*

($f_i(p)$ is the frequency of i up to position p)

UPDATE(C, i)
for $r \in [1:t]$
$\quad C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i)$
end for

ESTIMATE(C, i)
return $\text{median}_r \{C[r, h_r(i)] \cdot s_r(i)\}$

Lemma

If $b \geq 8 \max \left\{ k, \frac{32 \sum_{j \in TAIL} f_j^2}{(\varepsilon f_k)^2} \right\}$ *and* $t \geq A \log N$ *for an absolute constant* $A > 0$*, then for every* $i \in [m]$

$$\left| \textit{median}_r \{C[r, h_r(i)] \cdot s_r(i)\} - f_i \right| \leq \varepsilon f_k$$

with high probability.

(f_i is the frequency of i)

UPDATE(C, i)

for $r \in [1:t]$

$C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i)$

end for

ESTIMATE(C, i)

return median$_r \{C[r, h_r(i)] \cdot s_r(i)\}$

By basic estimate analysis for every $r \in [1:t]$

$$\mathbf{E}[C[r, h_r(i)] \cdot s_r(i)] = f_i$$

UPDATE(C, i)
for $r \in [1:t]$
$C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i)$
end for

ESTIMATE(C, i)
return $\text{median}_r \{C[r, h_r(i)] \cdot s_r(i)\}$

By basic estimate analysis for every $r \in [1:t]$

$$\mathbf{E}[C[r, h_r(i)] \cdot s_r(i)] = f_i$$

and

$$\mathbf{E}_s[(C[r, h_r(i)] \cdot s_r(i) - f_i)^2] = \sum_{j \neq i : \mathbf{h_r(j) = h_r(i)}} f_j^2$$

UPDATE(C, i)
for $r \in [1:t]$
    $C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i)$
end for

ESTIMATE(C, i)
return $\text{median}_r \{C[r, h_r(i)] \cdot s_r(i)\}$

By basic estimate analysis for every $r \in [1:t]$

$$\mathbf{E}[C[r, h_r(i)] \cdot s_r(i)] = f_i$$

and

$$\mathbf{E}_s[(C[r, h_r(i)] \cdot s_r(i) - f_i)^2] = \sum_{j \neq i : \mathbf{h_r(j) = h_r(i)}} f_j^2$$

How large can the variance be? Does it reduce by about a factor of b?

UPDATE(C, i)
for $r \in [1:t]$
$\quad C[r, h_r(i)] \leftarrow C[r, h_r(i)] + s_r(i)$
end for

ESTIMATE(C, i)
return $\text{median}_r \{C[r, h_r(i)] \cdot s_r(i)\}$

By basic estimate analysis for every $r \in [1:t]$

$$\mathbf{E}[C[r, h_r(i)] \cdot s_r(i)] = f_i$$

and

$$\mathbf{E}_s[(C[r, h_r(i)] \cdot s_r(i) - f_i)^2] = \sum_{j \neq i : \mathbf{h_r(j) = h_r(i)}} f_j^2$$

How large can the variance be? Does it reduce by about a factor of b?

Consider contribution of head and tail items separately:

$$\sum_{j \neq i : h_r(j) = h_r(i)} f_j^2 = \sum_{\substack{j \in HEAD, j \neq i \\ \mathbf{h_r(j) = h_r(i)}}} f_j^2 + \sum_{\substack{j \in TAIL, j \neq i \\ \mathbf{h_r(j) = h_r(i)}}} f_j^2$$

Consider contribution of head and tail items separately:

$$\sum_{j \neq i : h_r(j) = h_r(i)} f_j^2 = \sum_{\substack{j \in HEAD, j \neq i \\ \mathbf{h_r(j) = h_r(i)}}} f_j^2 + \sum_{\substack{j \in TAIL, j \neq i \\ \mathbf{h_r(j) = h_r(i)}}} f_j^2$$

Consider contribution of head and tail items separately:

$$\sum_{j \neq i : h_r(j) = h_r(i)} f_j^2 = \sum_{\substack{j \in HEAD, j \neq i \\ \mathbf{h_r(j) = h_r(i)}}} f_j^2 + \sum_{\substack{j \in TAIL, j \neq i \\ \mathbf{h_r(j) = h_r(i)}}} f_j^2$$

For each $r \in [1 : t]$ and each item $i \in [m]$ define three events:

- $\textsc{No-Collisions}_r(i)$ – i does not collide with any of the head items under hashing r

Consider contribution of head and tail items separately:

$$\sum_{j \neq i : h_r(j) = h_r(i)} f_j^2 = \sum_{\substack{j \in HEAD, j \neq i \\ \mathbf{h_r(j) = h_r(i)}}} f_j^2 + \sum_{\substack{j \in TAIL, j \neq i \\ \mathbf{h_r(j) = h_r(i)}}} f_j^2$$

For each $r \in [1:t]$ and each item $i \in [m]$ define three events:

- $\text{NO-COLLISIONS}_r(i)$ – i does not collide with any of the head items under hashing r
- $\text{SMALL-VARIANCE}_r(i)$ – i does not collide with too many of tail items under hashing r

Consider contribution of head and tail items separately:

$$\sum_{j \neq i : h_r(j) = h_r(i)} f_j^2 = \sum_{\substack{j \in HEAD, j \neq i \\ \mathbf{h_r(j) = h_r(i)}}} f_j^2 + \sum_{\substack{j \in TAIL, j \neq i \\ \mathbf{h_r(j) = h_r(i)}}} f_j^2$$

For each $r \in [1:t]$ and each item $i \in [m]$ define three events:

- $\textsc{No-Collisions}_r(i)$ – i does not collide with any of the head items under hashing r
- $\textsc{Small-Variance}_r(i)$ – i does not collide with too many of tail items under hashing r
- $\textsc{Small-Deviation}_r(i)$ – success event from basic analysis

Consider contribution of head and tail items separately:

$$\sum_{j \neq i : h_r(j) = h_r(i)} f_j^2 = \sum_{\substack{j \in HEAD, j \neq i \\ \mathbf{h_r(j) = h_r(i)}}} f_j^2 + \sum_{\substack{j \in TAIL, j \neq i \\ \mathbf{h_r(j) = h_r(i)}}} f_j^2$$

For each $r \in [1:t]$ and each item $i \in [m]$ define three events:

- NO-COLLISIONS$_r(i)$ – i does not collide with any of the head items under hashing r
- SMALL-VARIANCE$_r(i)$ – i does not collide with too many of tail items under hashing r
- SMALL-DEVIATION$_r(i)$ – success event from basic analysis

Show that all three events hold simultaneously with probability strictly bigger than $1/2$ – so median gives good estimate

(No) collisions with head items

NO-COLLISIONS$_r(i)$:=event that

$$\{j \in HEAD \setminus i : h_r(j) = h_r(i)\} = \emptyset,$$

i.e. that i collides with none of top k elements under h_r.

(No) collisions with head items

$\text{NO-COLLISIONS}_r(i)$:=event that

$$\{j \in HEAD \setminus i : h_r(j) = h_r(i)\} = \emptyset,$$

i.e. that i collides with none of top k elements under h_r.

For every $j \neq i$ and every $r \in [1:t]$

$$\mathbf{Pr}[h_r(i) = h_r(j)] \leq 1/b$$

(No) collisions with head items

$\textsc{No-Collisions}_r(i)$:=event that

$$\{j \in HEAD \setminus i : h_r(j) = h_r(i)\} = \emptyset,$$

i.e. that i collides with none of top k elements under h_r.

For every $j \neq i$ and every $r \in [1:t]$

$$\mathbf{Pr}[h_r(i) = h_r(j)] \leq 1/b$$

Suppose that $b \geq 8k$. Then by the union bound

$$\begin{aligned}\mathbf{Pr}[\textsc{No-Collisions}_r(i)] &\geq 1 - k/b \\ &\geq 1 - 1/8\end{aligned}$$

Consider contribution of head and tail items separately:

$$\sum_{j \neq i : h_r(j) = h_r(i)} f_j^2 = \sum_{\substack{j \in HEAD, j \neq i \\ \mathbf{h_r(j) = h_r(i)}}} f_j^2 + \sum_{\substack{j \in TAIL, j \neq i \\ \mathbf{h_r(j) = h_r(i)}}} f_j^2$$

For each $r \in [1:t]$ and each item $i \in [m]$ define three events:

- NO-COLLISIONS$_r(i)$ – i does not collide with any of the head items under hashing r
- SMALL-VARIANCE$_r(i)$ – i does not collide with too many of tail items under hashing r
- SMALL-DEVIATION$_r(i)$ – success event from basic analysis

Show that all three events hold simulaneously with probability strictly bigger than $1/2$ – so median gives good estimate

Consider contribution of head and tail items separately:

$$\sum_{j \neq i : h_r(j) = h_r(i)} f_j^2 = \sum_{\substack{j \in HEAD, j \neq i \\ \mathbf{h_r(j) = h_r(i)}}} f_j^2 + \sum_{\substack{j \in TAIL, j \neq i \\ \mathbf{h_r(j) = h_r(i)}}} f_j^2$$

For each $r \in [1 : t]$ and each item $i \in [m]$ define three events:

- NO-COLLISIONS$_r(i)$ – i does not collide with any of the head items under hashing r
- **SMALL-VARIANCE$_r(i)$ – i does not collide with too many of tail items under hashing r**
- SMALL-DEVIATION$_r(i)$ – success event from basic analysis

Show that all three events hold simulaneously with probability strictly bigger than $1/2$ – so median gives good estimate

Small variance from tail elements

SMALL-VARIANCE$_r(i)$:=event that

$$\sum_{\substack{j \in TAIL, j \neq i \\ h_r(j)=h_r(i)}} f_j^2 \leq \frac{8}{b} \sum_{j \in TAIL} f_j^2$$

Small variance from tail elements

SMALL-VARIANCE$_r(i)$:=event that

$$\sum_{\substack{j \in TAIL, j \neq i \\ h_r(j) = h_r(i)}} f_j^2 \leq \frac{8}{b} \sum_{j \in TAIL} f_j^2$$

For every $i, j \in [m], i \neq j$ and $r \in [1:t]$

$$\mathbf{Pr}_{h_r}[h_r(i) = h_r(j)] = 1/b \quad (b \text{ is the number of buckets})$$

Small variance from tail elements

SMALL-VARIANCE$_r(i)$:=event that

$$\sum_{\substack{j \in TAIL, j \neq i \\ h_r(j)=h_r(i)}} f_j^2 \leq \frac{8}{b} \sum_{j \in TAIL} f_j^2$$

For every $i, j \in [m], i \neq j$ and $r \in [1:t]$

$$\mathbf{Pr}_{h_r}[h_r(i) = h_r(j)] = 1/b \quad (b \text{ is the number of buckets})$$

So by linearity of expectation

$$\mathbf{E}\left[\sum_{\substack{j \in TAIL, j \neq i \\ h_r(j)=h_r(i)}} f_j^2\right] = \sum_{j \in TAIL, j \neq i} f_j^2 \cdot \mathbf{Pr}_{h_r}[h_r(i) = h_r(j)]$$

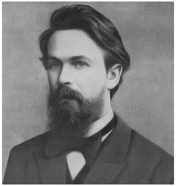
$$\leq \frac{1}{b} \sum_{j \in TAIL} f_j^2$$

Markov's inequality

Theorem

For every non-negative random variable X with mean $\mu \geq 0$, and every $k \geq 1$ one has

$$\mathbf{Pr}[X \geq k \cdot \mu] \leq 1/k$$



We proved that

$$\mathbf{E}\left[\sum_{\substack{j\in TAIL, j\neq i\\ h_r(j)=h_r(i)}} f_j^2\right] \le \frac{1}{b}\sum_{j\in TAIL} f_j^2$$

By Markov's inequality one has, for every i and every r,

$$\mathbf{Pr}[\textsc{Small-Variance}_r(i)] \ge 1 - 1/8$$

$\textsc{No-Collisions}_r(i)$ and $\textsc{Small-Variance}_r(i)$: recap

Consider contribution of head and tail items separately:

$$\sum_{j \neq i : h_r(j) = h_r(i)} f_j^2 = \sum_{\substack{j \in HEAD, j \neq i \\ h_r(j) = h_r(i)}} f_j^2 + \sum_{\substack{j \in TAIL, j \neq i \\ h_r(j) = h_r(i)}} f_j^2$$

Conditioned on $\textsc{No-Collisions}_r(i)$ and $\textsc{Small-Variance}_r(i)$

$\textsc{No-Collisions}_r(i)$ and $\textsc{Small-Variance}_r(i)$: recap

Consider contribution of head and tail items separately:

$$\sum_{j \neq i : h_r(j) = h_r(i)} f_j^2 = \sum_{\substack{j \in HEAD, j \neq i \\ h_r(j) = h_r(i)}} f_j^2 + \sum_{\substack{j \in TAIL, j \neq i \\ h_r(j) = h_r(i)}} f_j^2$$

Conditioned on $\textsc{No-Collisions}_r(i)$ and $\textsc{Small-Variance}_r(i)$

- first term is zero

$\text{No-Collisions}_r(i)$ and $\text{Small-Variance}_r(i)$: recap

Consider contribution of head and tail items separately:

$$\sum_{j \neq i : h_r(j) = h_r(i)} f_j^2 = \sum_{\substack{j \in HEAD, j \neq i \\ h_r(j) = h_r(i)}} f_j^2 + \sum_{\substack{j \in TAIL, j \neq i \\ h_r(j) = h_r(i)}} f_j^2$$

Conditioned on $\text{No-Collisions}_r(i)$ and $\text{Small-Variance}_r(i)$

- first term is zero
- second term is at most

$$\frac{8}{b} \sum_{j \in TAIL} f_j^2$$

Small deviation event

SMALL-DEVIATION$_r(i)$=event that

$$(C[r, h_r(i)] \cdot s_r(i) - f_i)^2 \leq 8\mathbf{Var}(C[r, h_r(i)] \cdot s_r(i)).$$

Small deviation event

$\textsc{Small-Deviation}_r(i)$=event that

$$(C[r, h_r(i)] \cdot s_r(i) - f_i)^2 \leq 8\mathbf{Var}(C[r, h_r(i)] \cdot s_r(i)).$$

By Chebyshev's inequality one has, for every i and every r,

$$\mathbf{Pr}[\textsc{Small-Deviation}_r(i)] \geq 1 - 1/8$$

$$\mathbf{Pr}[\textsc{Small-Variance}_r(i)] \geq 1 - 1/8$$

$$\mathbf{Pr}[\textsc{No-Collisions}_r(i)] \geq 1 - 1/8$$

$$\mathbf{Pr}[\textsc{Small-Deviation}_r(i)] \geq 1 - 1/8$$

So by the union bound

$$\begin{aligned}\mathbf{Pr}[&\textsc{Small-Variance}_r(i) \text{ and } \textsc{No-Collisions}_r(i) \\ &\text{and } \textsc{Small-Deviation}_r(i)] \geq 5/8.\end{aligned}$$

Let

$$\gamma := \sqrt{\frac{1}{b}\sum_{j\in TAIL} f_j^2}$$

For every $p \in [1:N]$ let $f_i(p) :=$ frequency of i up to position p

Lemma

If $b \geq 8k$, then for every i, every $r \in [1:t]$,

$$\mathbf{Pr}\left[|C[r,h_r(i)]\cdot s_r(i) - f_i| \leq 8\gamma\right] \geq 5/8$$

Let

$$\gamma := \sqrt{\frac{1}{b}\sum_{j\in TAIL} f_j^2}$$

For every $p \in [1:N]$ let $f_i(p)$:=frequency of i up to position p

Lemma

If $b \geq 8k$ and $t \geq A \log N$ for an absolute constant $A > 0$, then for every i, with probability $\geq 1 - 1/N^4$

$$\left| median_r \{C[r, h_r(i)] \cdot s_r(i)\} - f_i \right| \leq 8\gamma$$

at the end of the stream.

Proof.

Chernoff bounds. □

Let

$$\gamma := \sqrt{\frac{1}{b}\sum_{j\in TAIL} f_j^2}$$

For every $p \in [1:N]$ let $f_i(p) :=$frequency of i up to position p

Lemma

If $b \geq 8k$ and $t \geq A \log N$ for an absolute constant $A > 0$, then with probability $\geq 1 - 1/N^3$ for every $i \in [m]$

$$\left| median_r \{C[r, h_r(i)] \cdot s_r(i)\} - f_i(p) \right| \leq 8\gamma$$

at the end of the stream.

Proof.

Chernoff bounds. $\square$

Let

$$\gamma := \sqrt{\frac{1}{b} \sum_{j \in TAIL} f_i^2}$$

For every $p \in [1:N]$ let $f_i(p)$:=frequency of i up to position p

Lemma

If $b \geq 8 \max\left\{k, \frac{32 \sum_{j \in TAIL} f_j^2}{(\varepsilon f_k)^2}\right\}$ *and* $t \geq A \log N$ *for an absolute constant* $A > 0$*, then with probability* $\geq 1 - 1/N^3$ *for every* $i \in [m]$

$$\left| median_r \{C[r, h_r(i)] \cdot s_r(i)\} - f_i(p) \right| \leq \varepsilon f_k$$

at the end of the stream.

Let

$$\gamma := \sqrt{\frac{1}{b}\sum_{j\in TAIL} f_i^2}$$

For every $p \in [1:N]$ let $f_i(p)$:=frequency of i up to position p

Lemma

If $b \geq 8\max\left\{k, \frac{32\sum_{j\in TAIL} f_j^2}{(\varepsilon f_k)^2}\right\}$ and $t \geq A\log N$ for an absolute constant $A > 0$, then with probability $\geq 1 - 1/N^3$ for every $i \in [m]$

$$\left|median_r\{C[r, h_r(i)] \cdot s_r(i)\} - f_i(p)\right| \leq \varepsilon f_k$$

at the end of the stream.

Proof.

Substitute value of b into definition of γ:

$$\gamma = \sqrt{\frac{1}{b}\sum_{j\in TAIL} f_i^2} \leq \varepsilon f_k/8$$