Basic algorithmic techniques for data streams

Mariano Zelke

Goethe-Universität Frankfurt am Main
Institut für Informatik
Arbeitsgruppe Theorie Komplexer Systeme

DEIS'10, Dagstuhl, Nov. 2010

Contents

Data Stream Model

Data Stream Model

- Stream: m elements of some universe of size n

Data Stream Model

- Stream: m elements of some universe of size n

  $a_1, a_2, a_3, \ldots, a_m \qquad\qquad a_i \in \{1, \ldots, n\}$

Data Stream Model

- Stream: m elements of some universe of size n

  $a_1, a_2, a_3, \ldots, a_m \qquad\qquad a_i \in \{1, \ldots, n\}$
- Goal: gain information about stream

Data Stream Model

- Stream: m elements of some universe of size n

  $a_1, a_2, a_3, \ldots, a_m$ $\qquad\qquad$ $a_i \in \{1, \ldots, n\}$
- Goal: gain information about stream

  statistical information (median, frequency moments, ...), longest increasing subsequence,...

Data Stream Model

- Stream: m elements of some universe of size n

  $a_1, a_2, a_3, \ldots, a_m \qquad\qquad a_i \in \{1, \ldots, n\}$
- Goal: gain information about stream

  statistical information (median, frequency moments, ...), longest increasing subsequence,...
- But: algorithms are restricted to

Data Stream Model

- Stream: m elements of some universe of size n

  $a_1, a_2, a_3, \ldots, a_m \qquad\qquad a_i \in \{1, \ldots, n\}$
- Goal: gain information about stream

  statistical information (median, frequency moments, ...), longest increasing subsequence,...
- But: algorithms are restricted to
  - sequential access to items in stream

Data Stream Model

- Stream: m elements of some universe of size n

  $a_1, a_2, a_3, \ldots, a_m \qquad\qquad a_i \in \{1, \ldots, n\}$
- Goal: gain information about stream

  statistical information (median, frequency moments, ...), longest increasing subsequence,...
- But: algorithms are restricted to
  - sequential access to items in stream
  - limited memory, sublinear in m and n

Sampling

General idea:

Sampling

General idea:

- sample (= select) items from stream according to some rule

Sampling

General idea:

- sample (= select) items from stream according to some rule
- rule is randomized or deterministic

Sampling

General idea:

- sample (= select) items from stream according to some rule
- rule is randomized or deterministic

Sampling

General idea:

- sample (= select) items from stream according to some rule
- rule is randomized or deterministic
- use sampled items to get information about whole stream

Sampling

General idea:

- sample (= select) items from stream according to some rule
- rule is randomized or deterministic
- use sampled items to get information about whole stream
- only need to store sampled items
  $\Rightarrow$ good for streaming

Easy Starter

Sample out a single item uniformly at random from the stream

Easy Starter

Sample out a single item uniformly at random from the stream

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_r, \ldots, a_m$

Easy Starter

Sample out a single item uniformly at random from the stream

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_r, \ldots, a_m$

- Easy if m is known in advance:

Easy Starter

Sample out a single item uniformly at random from the stream

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_r, \ldots, a_m$

- Easy if m is known in advance:
- Pick a random number $r \in \{1, 2, \ldots, m\}$

Easy Starter

Sample out a single item uniformly at random from the stream

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_r, \ldots, a_m$

- Easy if m is known in advance:
- Pick a random number $r \in \{1, 2, \ldots, m\}$
- Go over the stream and snatch out sampled item

Easy Starter

Sample out a single item uniformly at random from the stream

$\downarrow$
$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_r, \ldots, a_m$

- Easy if m is known in advance:
- Pick a random number $r \in \{1, 2, \ldots, m\}$
- Go over the stream and snatch out sampled item

Easy Starter

Sample out a single item uniformly at random from the stream

$$\downarrow$$

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_r, \ldots, a_m$$

- Easy if m is known in advance:
- Pick a random number $r \in \{1, 2, \ldots, m\}$
- Go over the stream and snatch out sampled item

Easy Starter

Sample out a single item uniformly at random from the stream

$$\downarrow$$

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_r, \ldots, a_m$$

- Easy if m is known in advance:
- Pick a random number $r \in \{1, 2, \ldots, m\}$
- Go over the stream and snatch out sampled item

Easy Starter

Sample out a single item uniformly at random from the stream

$$\downarrow$$

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_r, \ldots, a_m$$

- Easy if m is known in advance:
- Pick a random number $r \in \{1, 2, \ldots, m\}$
- Go over the stream and snatch out sampled item

Easy Starter

Sample out a single item uniformly at random from the stream

$$\downarrow$$

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_r, \ldots, a_m$$

- Easy if m is known in advance:
- Pick a random number $r \in \{1, 2, \ldots, m\}$
- Go over the stream and snatch out sampled item

Easy Starter

Sample out a single item uniformly at random from the stream

$$\downarrow$$

$$a_1,\ a_2,\ a_3,\ a_4,\ a_5,\ a_6,\ a_7,\ a_8,\ a_9,\ a_{10},\ a_{11},\ a_{12},\ a_{13},\ \ldots,\ a_r,\ \ldots,\ a_m$$

- Easy if m is known in advance:
- Pick a random number $r \in \{1, 2, \ldots, m\}$
- Go over the stream and snatch out sampled item

Easy Starter

Sample out a single item uniformly at random from the stream

$$\downarrow$$

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_r, \ldots, a_m$$

- Easy if m is known in advance:
- Pick a random number $r \in \{1, 2, \ldots, m\}$
- Go over the stream and snatch out sampled item

Easy Starter

Sample out a single item uniformly at random from the stream

$$\downarrow$$

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_r, \ldots, a_m$$

- Easy if m is known in advance:
- Pick a random number $r \in \{1, 2, \ldots, m\}$
- Go over the stream and snatch out sampled item

Easy Starter

Sample out a single item uniformly at random from the stream

$$\downarrow$$

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_r, \ldots, a_m$$

- Easy if m is known in advance:
- Pick a random number $r \in \{1, 2, \ldots, m\}$
- Go over the stream and snatch out sampled item

Easy Starter

Sample out a single item uniformly at random from the stream

$$\downarrow$$

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_r, \ldots, a_m$$

- Easy if m is known in advance:
- Pick a random number $r \in \{1, 2, \ldots, m\}$
- Go over the stream and snatch out sampled item

Easy Starter

Sample out a single item uniformly at random from the stream

$$\downarrow$$

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_r, \ldots, a_m$$

- Easy if m is known in advance:
- Pick a random number $r \in \{1, 2, \ldots, m\}$
- Go over the stream and snatch out sampled item

Easy Starter

Sample out a single item uniformly at random from the stream

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, \overset{\downarrow}{a_{12}}, a_{13}, \ldots, a_r, \ldots, a_m$$

- Easy if m is known in advance:
- Pick a random number $r \in \{1, 2, \ldots, m\}$
- Go over the stream and snatch out sampled item

Easy Starter

Sample out a single item uniformly at random from the stream

$$\downarrow$$

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_r, \ldots, a_m$$

- Easy if m is known in advance:
- Pick a random number $r \in \{1, 2, \ldots, m\}$
- Go over the stream and snatch out sampled item

Easy Starter

Sample out a single item uniformly at random from the stream

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, \overset{\downarrow}{a_r}, \ldots, a_m$$

- Easy if m is known in advance:
- Pick a random number $r \in \{1, 2, \ldots, m\}$
- Go over the stream and snatch out sampled item

Easy Starter

Sample out a single item uniformly at random from the stream

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_r, \ldots, a_m$$

- Easy if m is known in advance:
- Pick a random number $r \in \{1, 2, \ldots, m\}$
- Go over the stream and snatch out sampled item a_r

Easy Starter

Sample out a single item uniformly at random from the stream

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, \overset{\downarrow}{a_r}, \ldots, a_m$$

- Easy if m is known in advance:
- Pick a random number $r \in \{1, 2, \ldots, m\}$
- Go over the stream and snatch out sampled item a_r

But what if m is not known in advance?

Reservoir Sampling

[J. Vitter '85]

Sample out a single item uniformly at random from the stream without knowing its length

Reservoir Sampling

[J. Vitter '85]

Sample out a single item uniformly at random from the stream without knowing its length

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, a_{i+1}, a_{i+2}, \ldots, a_m$$

Reservoir Sampling

[J. Vitter '85]

Sample out a single item uniformly at random from the stream without knowing its length

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, a_{i+1}, a_{i+2}, \ldots, a_m$

current sample:

Reservoir Sampling

[J. Vitter '85]

Sample out a single item uniformly at random from the stream without knowing its length

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, a_{i+1}, a_{i+2}, \ldots, a_m$

take as sample

current sample: a_1

Reservoir Sampling

[J. Vitter '85]

Sample out a single item uniformly at random from the stream without knowing its length

$$\downarrow$$

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, a_{i+1}, a_{i+2}, \ldots, a_m$

replace with probability $1/2$

current sample: a_r

Reservoir Sampling

[J. Vitter '85]

Sample out a single item uniformly at random from the stream without knowing its length

$$\downarrow$$

$$a_1,\, a_2,\, a_3,\, a_4,\, a_5,\, a_6,\, a_7,\, a_8,\, a_9,\, a_{10},\, a_{11},\, a_{12},\, a_{13},\, \ldots,\, a_i,\, a_{i+1},\, a_{i+2},\, \ldots,\, a_m$$

replace with probability $1/3$

current sample: a_r

Reservoir Sampling

[J. Vitter '85]

Sample out a single item uniformly at random from the stream without knowing its length

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, a_{i+1}, a_{i+2}, \ldots, a_m$$

replace with probability $1/4$

current sample: a_r

Reservoir Sampling

[J. Vitter '85]

Sample out a single item uniformly at random from the stream without knowing its length

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, a_{i+1}, a_{i+2}, \ldots, a_m$$

replace with probability $1/i$

current sample: a_r

Reservoir Sampling

[J. Vitter '85]

Sample out a single item uniformly at random from the stream without knowing its length

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, a_{i+1}, a_{i+2}, \ldots, a_m$

replace with probability $1/m$

current sample: a_r

Reservoir Sampling

[J. Vitter '85]

Sample out a single item uniformly at random from the stream without knowing its length

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, a_{i+1}, a_{i+2}, \ldots, a_m$$

final sample: a_r

Reservoir Sampling

[J. Vitter '85]

Sample out a single item uniformly at random from the stream without knowing its length

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, a_{i+1}, a_{i+2}, \ldots, a_m$$

current sample:

$Pr[\text{final sample } a_i] \ =$

Reservoir Sampling

[J. Vitter '85]

Sample out a single item uniformly at random from the stream without knowing its length

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, a_{i+1}, a_{i+2}, \ldots, a_m$$

replace

current sample: a_i

$$Pr[\text{final sample } a_i] = \frac{1}{i}$$

Reservoir Sampling

[J. Vitter '85]

Sample out a single item uniformly at random from the stream without knowing its length

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, a_{i+1}, a_{i+2}, \ldots, a_m$$

do not replace

current sample: a_i

$$Pr[\text{final sample } a_i] = \frac{1}{i} \times \left(1 - \frac{1}{i+1}\right)$$

Reservoir Sampling

[J. Vitter '85]

Sample out a single item uniformly at random from the stream without knowing its length

$a_1,\ a_2,\ a_3,\ a_4,\ a_5,\ a_6,\ a_7,\ a_8,\ a_9,\ a_{10},\ a_{11},\ a_{12},\ a_{13},\ \ldots,\ a_i,\ a_{i+1},\ a_{i+2},\ \ldots,\ a_m$

do not replace

current sample: a_i

$$Pr[\text{final sample } a_i] \ = \ \frac{1}{i} \times (1 - \frac{1}{i+1}) \times (1 - \frac{1}{i+2})$$

Reservoir Sampling

[J. Vitter '85]

Sample out a single item uniformly at random from the stream without knowing its length

$$a_1,\ a_2,\ a_3,\ a_4,\ a_5,\ a_6,\ a_7,\ a_8,\ a_9,\ a_{10},\ a_{11},\ a_{12},\ a_{13},\ \ldots,\ a_i,\ a_{i+1},\ a_{i+2},\ \ldots,\ a_m$$

$\ldots$

do not replace

current sample: a_i

$$Pr[\text{final sample } a_i] \;=\; \frac{1}{i} \times \left(1 - \frac{1}{i+1}\right) \times \left(1 - \frac{1}{i+2}\right) \times \;\ldots\; \times \left(1 - \frac{1}{m}\right)$$

Reservoir Sampling

[J. Vitter '85]

Sample out a single item uniformly at random from the stream without knowing its length

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, a_{i+1}, a_{i+2}, \ldots, a_m$$

final sample: a_i

$$\begin{aligned} Pr[\text{final sample } a_i] &= \tfrac{1}{i} \times (1 - \tfrac{1}{i+1}) \times (1 - \tfrac{1}{i+2}) \times \ \ldots \ \times (1 - \tfrac{1}{m}) \\ &= \tfrac{1}{i} \times \quad \tfrac{i}{i+1} \quad \times \quad \tfrac{i+1}{i+2} \quad \times \ \ldots \ \times \quad \tfrac{m-1}{m} \end{aligned}$$

Reservoir Sampling

[J. Vitter '85]

Sample out a single item uniformly at random from the stream without knowing its length

$$a_1,\ a_2,\ a_3,\ a_4,\ a_5,\ a_6,\ a_7,\ a_8,\ a_9,\ a_{10},\ a_{11},\ a_{12},\ a_{13},\ \ldots,\ a_i,\ a_{i+1},\ a_{i+2},\ \ldots,\ a_m$$

final sample: a_i

$$\begin{aligned} Pr[\text{final sample } a_i] &= \tfrac{1}{i} \times (1 - \tfrac{1}{i+1}) \times (1 - \tfrac{1}{i+2}) \times \ \ldots\ \times (1 - \tfrac{1}{m}) \\ &= \tfrac{1}{i} \times \quad \tfrac{i}{i+1} \quad \times \quad \tfrac{i+1}{i+2} \quad \times \ \ldots\ \times \quad \tfrac{m-1}{m} \end{aligned}$$

Reservoir Sampling

[J. Vitter '85]

Sample out a single item uniformly at random from the stream without knowing its length

$$a_1,\ a_2,\ a_3,\ a_4,\ a_5,\ a_6,\ a_7,\ a_8,\ a_9,\ a_{10},\ a_{11},\ a_{12},\ a_{13},\ \ldots,\ a_i,\ a_{i+1},\ a_{i+2},\ \ldots,\ a_m$$

final sample: a_i

$$\begin{aligned} Pr[\text{final sample } a_i] &= \tfrac{1}{i} \times (1 - \tfrac{1}{i+1}) \times (1 - \tfrac{1}{i+2}) \times \ \ldots \ \times (1 - \tfrac{1}{m}) \\ &= \frac{1}{m} \end{aligned}$$

Reservoir Sampling

[J. Vitter '85]

Sample out a single item uniformly at random from the stream without knowing its length

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, a_{i+1}, a_{i+2}, \ldots, a_m$$

final sample: a_i

$$\begin{aligned} Pr[\text{final sample } a_i] &= \frac{1}{i} \times (1 - \frac{1}{i+1}) \times (1 - \frac{1}{i+2}) \times \ \ldots \ \times (1 - \frac{1}{m}) \\ &= \frac{1}{m} \end{aligned}$$

Reservoir Sampling [J. Vitter '85]

Sample out several items uniformly at random from the stream without knowing its length

Reservoir Sampling

[J. Vitter '85]

Sample out several items uniformly at random from the stream without knowing its length

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$

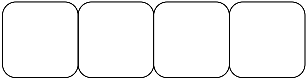
Reservoir Sampling

[J. Vitter '85]

Sample out several items uniformly at random from the stream without knowing its length

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$

current samples

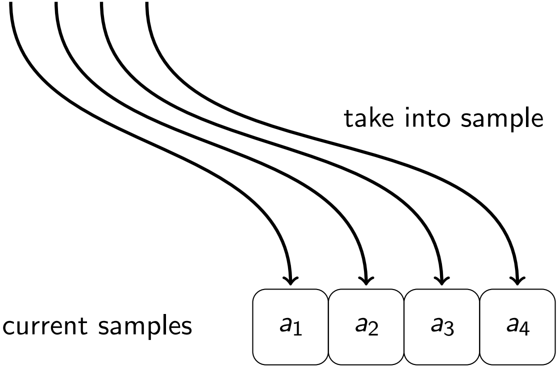


Reservoir Sampling

[J. Vitter '85]

Sample out several items uniformly at random from the stream without knowing its length

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$$



Reservoir Sampling

[J. Vitter '85]

Sample out several items uniformly at random from the stream without knowing its length

$$\overset{\downarrow}{}$$

$$a_1,\ a_2,\ a_3,\ a_4,\ \overset{\downarrow}{a_5},\ a_6,\ a_7,\ a_8,\ a_9,\ a_{10},\ a_{11},\ a_{12},\ a_{13},\ \ldots,\ a_i,\ \ldots,\ a_m$$

current samples $\boxed{a_1}\boxed{a_2}\boxed{a_3}\boxed{a_4}$

Reservoir Sampling

[J. Vitter '85]

Sample out several items uniformly at random from the stream without knowing its length

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$$

sample with probability 1/5

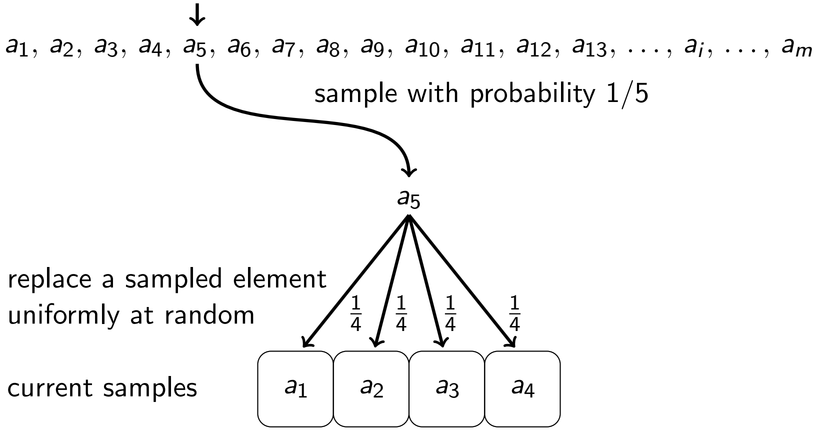
a_5

current samples $\boxed{a_1}\boxed{a_2}\boxed{a_3}\boxed{a_4}$

Reservoir Sampling

[J. Vitter '85]

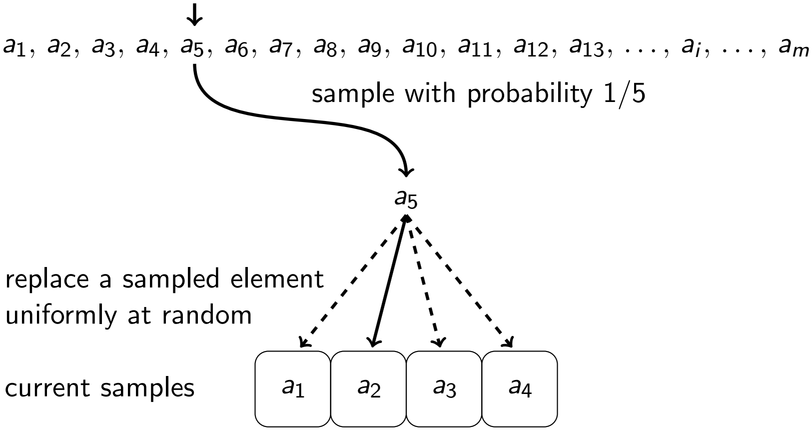
Sample out several items uniformly at random from the stream without knowing its length



Reservoir Sampling

[J. Vitter '85]

Sample out several items uniformly at random from the stream without knowing its length



Reservoir Sampling

[J. Vitter '85]

Sample out several items uniformly at random from the stream without knowing its length

$$\overset{\downarrow}{}$$

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$

current samples $\boxed{a_1}\boxed{a_5}\boxed{a_3}\boxed{a_4}$

Reservoir Sampling

[J. Vitter '85]

Sample out several items uniformly at random from the stream without knowing its length

$$\downarrow$$

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$$

current samples $\boxed{a_1}\boxed{a_5}\boxed{a_3}\boxed{a_4}$

Reservoir Sampling

[J. Vitter '85]

Sample out several items uniformly at random from the stream without knowing its length

$$a_1,\ a_2,\ a_3,\ a_4,\ a_5,\ a_6,\ a_7,\ a_8,\ a_9,\ a_{10},\ a_{11},\ a_{12},\ a_{13},\ \ldots,\ a_i,\ \ldots,\ a_m$$

sample with probability $1/6$

a_6

current samples $\boxed{a_1}\boxed{a_5}\boxed{a_3}\boxed{a_4}$

Reservoir Sampling

[J. Vitter '85]

Sample out several items uniformly at random from the stream without knowing its length

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$

a_6

current samples $\boxed{a_1}\boxed{a_5}\boxed{a_3}\boxed{a_4}$

Reservoir Sampling

[J. Vitter '85]

Sample out several items uniformly at random from the stream without knowing its length

$$\downarrow$$

$$a_1,\ a_2,\ a_3,\ a_4,\ a_5,\ a_6,\ a_7,\ a_8,\ a_9,\ a_{10},\ a_{11},\ a_{12},\ a_{13},\ \ldots,\ a_i,\ \ldots,\ a_m$$

current samples $\boxed{a_1}\boxed{a_5}\boxed{a_3}\boxed{a_4}$

Reservoir Sampling

[J. Vitter '85]

Sample out several items uniformly at random from the stream without knowing its length

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, \overset{\downarrow}{a_i}, \ldots, a_m$$

current samples $\boxed{a_{r_1}}\boxed{a_{r_2}}\boxed{a_{r_3}}\boxed{a_{r_4}}$

Reservoir Sampling

[J. Vitter '85]

Sample out several items uniformly at random from the stream without knowing its length

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \dots, a_i, \dots, a_m$$

sample with probability 1/i

$$a_i$$

current samples $\boxed{a_{r_1}}\boxed{a_{r_2}}\boxed{a_{r_3}}\boxed{a_{r_4}}$

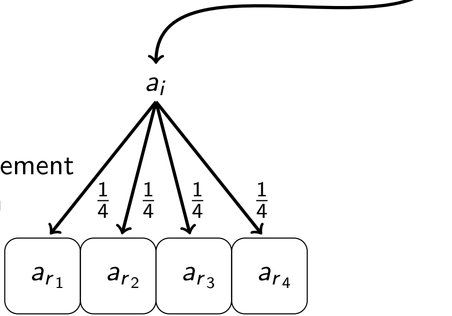
Reservoir Sampling

[J. Vitter '85]

Sample out several items uniformly at random from the stream without knowing its length

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$$

sample with probability 1/i



replace a sampled element uniformly at random

current samples

Reservoir Sampling

[J. Vitter '85]

Sample out several items uniformly at random from the stream without knowing its length

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$

sample with probability 1/i

a_i

replace a sampled element uniformly at random

$\frac{1}{4}$ $\frac{1}{4}$ $\frac{1}{4}$ $\frac{1}{4}$

current samples

a_{r_1}	a_{r_2}	a_{r_3}	a_{r_4}

Memory usage for sampling k items:

Reservoir Sampling

[J. Vitter '85]

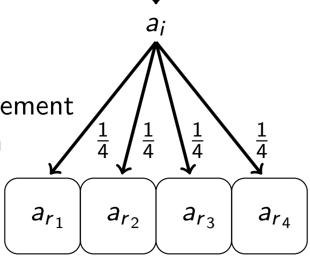
Sample out several items uniformly at random from the stream without knowing its length

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \dots, a_i, \dots, a_m$

sample with probability 1/i

replace a sampled element uniformly at random

current samples

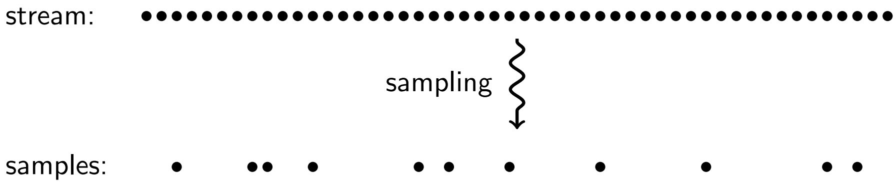


Memory usage for sampling k items: $k \cdot \log n$

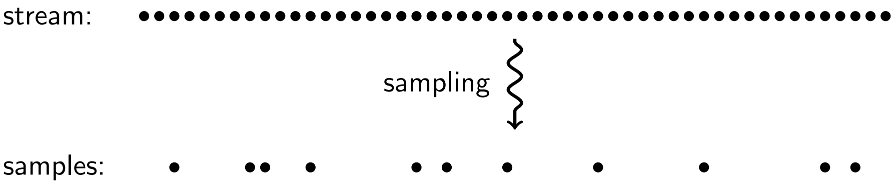
Sampling Applications

stream: 

Sampling Applications



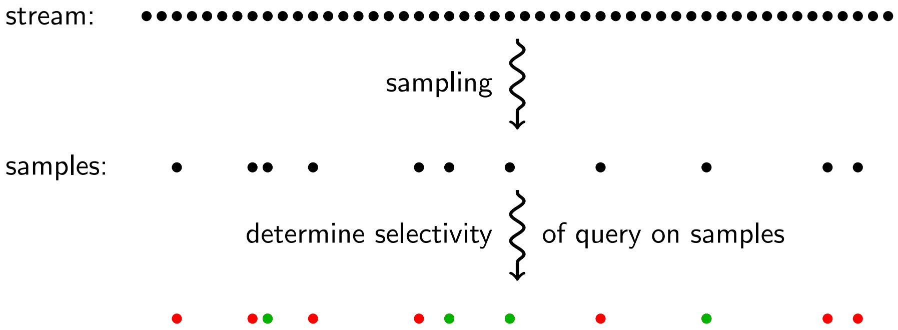
Sampling Applications



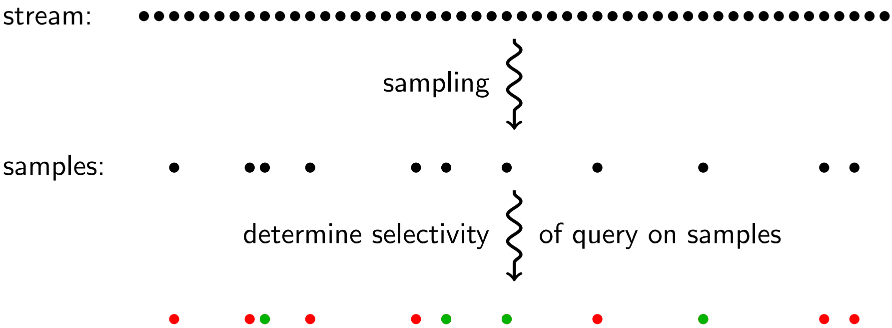
Goal: determine query selectivity on items of stream

(assume query to be invariant of stream order)

Sampling Applications

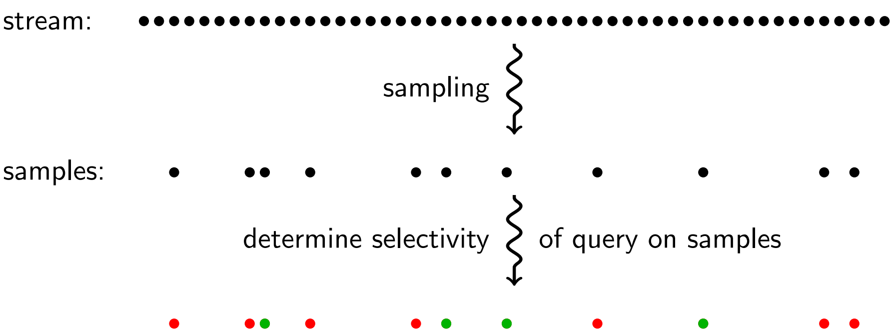


Sampling Applications



To get $(1 \pm \varepsilon)$-estimate with probability $1 - \delta$: What sample size s ?

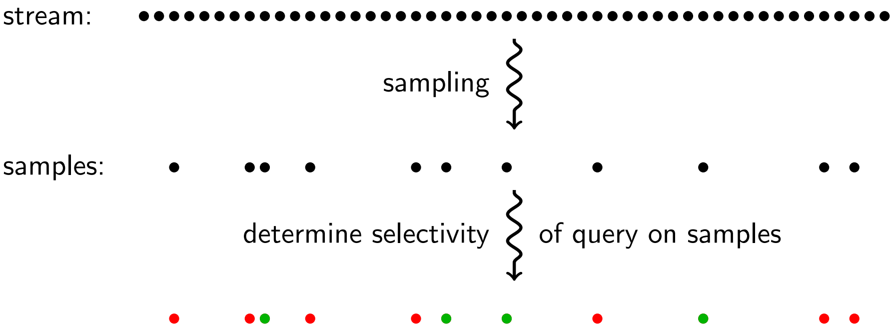
Sampling Applications



To get $(1 \pm \varepsilon)$-estimate with probability $1 - \delta$: What sample size s ?

Query selects $\frac{m}{c}$ stream items $\Rightarrow$ $Exp[\, s^+ \,] = \frac{s}{c}$

Sampling Applications

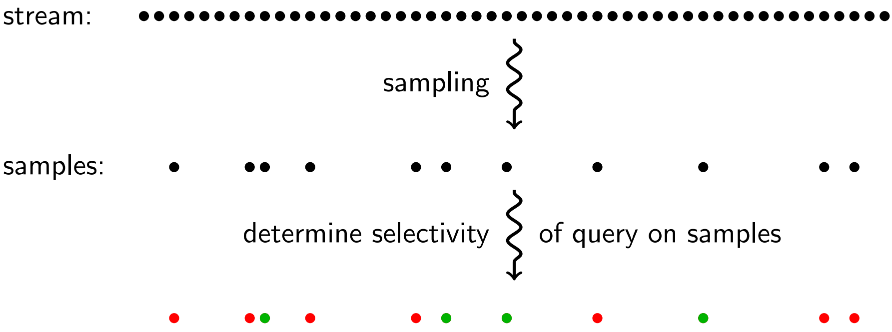


To get $(1 \pm \varepsilon)$-estimate with probability $1 - \delta$: What sample size s ?

Query selects $\frac{m}{c}$ stream items $\Rightarrow$ $Exp[\, s^+ \,] = \frac{s}{c}$

Chernoff-Hoeffding-Ineq.: $Pr[\, |s^+ - Exp[\, s^+ \,]| > \varepsilon \cdot s^+ \,] \;\leq\; e^{-\Theta(\varepsilon^2 s)}$

Sampling Applications



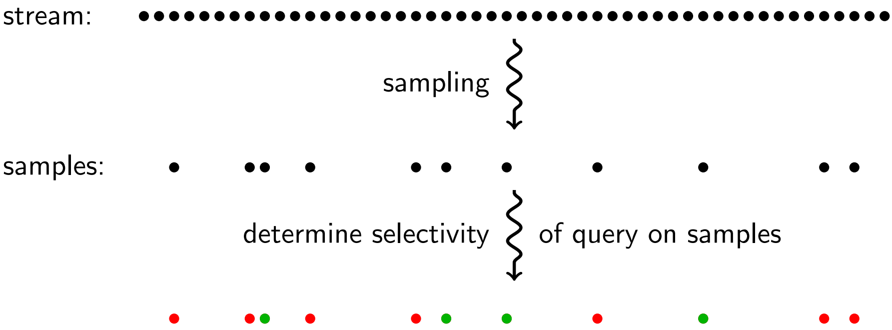
To get $(1 \pm \varepsilon)$-estimate with probability $1 - \delta$: What sample size s ?

Query selects $\frac{m}{c}$ stream items $\Rightarrow$ $Exp[\, s^+ \,] = \frac{s}{c}$

Chernoff-Hoeffding-Ineq.: $Pr[\, |s^+ - Exp[\, s^+ \,]| > \varepsilon \cdot s^+ \,] \;\leq\; e^{-\Theta(\varepsilon^2 s)}$

$$\text{sample } \; \mathcal{O}\left(\frac{1}{\varepsilon^2} \cdot \log \frac{1}{\delta}\right) \; \text{ items}$$

Sampling Applications



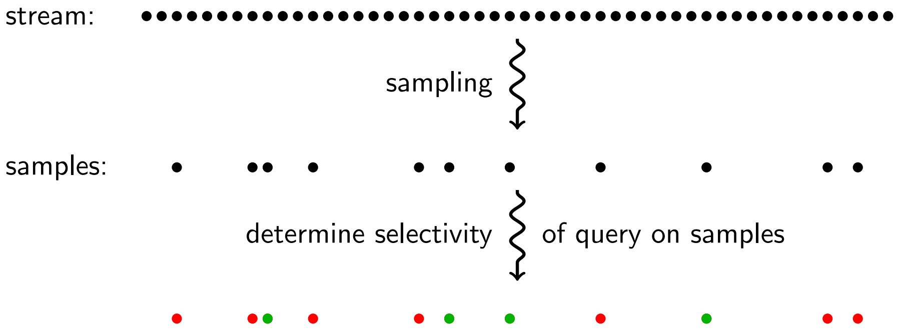
To get $(1 \pm \varepsilon)$-estimate with probability $1 - \delta$: What sample size s ?

Query selects $\frac{m}{c}$ stream items $\Rightarrow$ $Exp[\, s^+ \,] = \frac{s}{c}$

Chernoff-Hoeffding-Ineq.: $Pr[\,|s^+ - Exp[\, s^+ \,]| > \varepsilon \cdot s^+ \,] \;\leq\; e^{-\Theta(\varepsilon^2 s)}$

Memory usage: $\mathcal{O}\left(\frac{1}{\varepsilon^2} \cdot \log \frac{1}{\delta}\right)$ items $\times$ $\log n$ bits

Sampling Applications



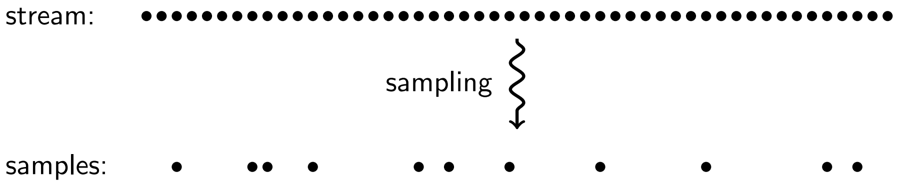
To get $(1 \pm \varepsilon)$-estimate with probability $1 - \delta$: What sample size s ?

Query selects $\frac{m}{c}$ stream items $\Rightarrow$ $Exp[\, s^+ \,] = \frac{s}{c}$

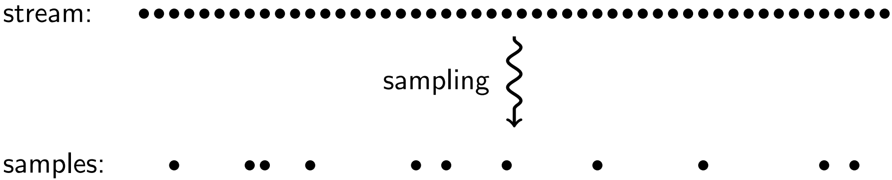
Chernoff-Hoeffding-Ineq.: $Pr[\, |s^+ - Exp[\, s^+ \,]| > \varepsilon \cdot s^+ \,] \;\leq\; e^{-\Theta(\varepsilon^2 s)}$

Memory usage: $\mathcal{O}\left(\frac{1}{\varepsilon^2} \cdot \log \frac{1}{\delta} \cdot \log n\right)$ bits

Sampling Applications



Sampling Applications



Goal: find the median of the stream

Sampling Applications

stream: $\bullet$

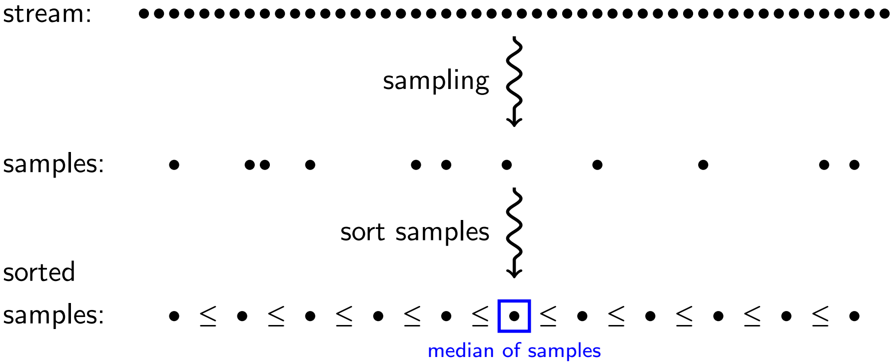
sampling $\rightsquigarrow$

samples: $\bullet \quad \bullet\bullet \quad \bullet \quad \bullet \; \bullet \quad \bullet \quad \bullet \quad \bullet \quad \bullet \; \bullet$

sort samples $\rightsquigarrow$

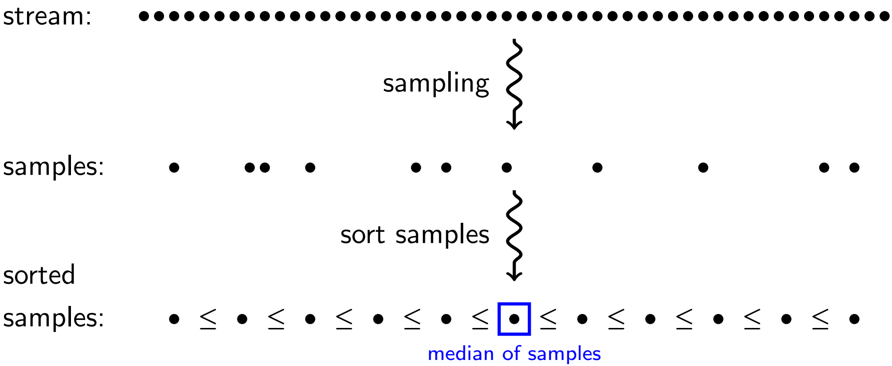
sorted samples: $\bullet \leq \bullet \leq \bullet \leq \bullet \leq \bullet \leq \bullet \leq \bullet \leq \bullet \leq \bullet \leq \bullet \leq \bullet$

Sampling Applications



Pick median of samples as an estimate of stream's median

Sampling Applications



Pick median of samples as an estimate of stream's median

To get $(1 \pm \varepsilon)$-estimate with probability $1 - \delta$:

$$\text{sample } \ \mathcal{O}\left(\frac{1}{\varepsilon^2} \cdot \log \frac{1}{\delta}\right) \ \text{ items}$$

AMS Sampling

[Alon, Matias, Szegedy '96]

AMS Sampling

[Alon, Matias, Szegedy '96]

Stream $a_1, a_2, a_3, \ldots, a_m$

AMS Sampling

[Alon, Matias, Szegedy '96]

Stream $a_1, a_2, a_3, \ldots, a_m$

Frequency of an item: $f_i = \big|\{j : a_j = i\}\big|$

AMS Sampling

[Alon, Matias, Szegedy '96]

Stream $a_1, a_2, a_3, \ldots, a_m$

Frequency of an item: $f_i = \left|\{j : a_j = i\}\right|$

Example: stream 2, 3, 3, 2, 3, 1, 2, 3

$$f_1 = 1, \quad f_2 = 3, \quad f_3 = 4$$

AMS Sampling

[Alon, Matias, Szegedy '96]

Stream $a_1, a_2, a_3, \ldots, a_m$

Frequency of an item: $f_i = \left|\{j : a_j = i\}\right|$

kth frequency moment $F_k = \sum_{i=1}^{n} f_i^k$

AMS Sampling

[Alon, Matias, Szegedy '96]

Stream $a_1, a_2, a_3, \ldots, a_m$

Frequency of an item: $f_i = \left|\{j : a_j = i\}\right|$

kth frequency moment $\quad F_k = \sum_{i=1}^{n} f_i^k$

Frequency moments provide useful statistics:

- F_0: number of distinct elements in stream
- F_1: length of stream, m
- F_2: size of self join
- F_k, $k \geq 2$: skew of distribution

AMS Sampling

[Alon, Matias, Szegedy '96]

Stream $a_1, a_2, a_3, \ldots, a_m$

Frequency of an item: $f_i = \left|\{j : a_j = i\}\right|$

kth frequency moment $F_k = \sum_{i=1}^{n} f_i^k$

Frequency moments provide useful statistics:

- F_0: number of distinct elements in stream
- F_1: length of stream, m
- F_2: size of self join
- F_k, $k \geq 2$: skew of distribution

Trivial determination of F_k: maintain counters for each f_i

AMS Sampling

[Alon, Matias, Szegedy '96]

Stream $a_1, a_2, a_3, \ldots, a_m$

Frequency of an item: $f_i = \left|\{j : a_j = i\}\right|$

*k*th frequency moment $F_k = \sum_{i=1}^{n} f_i^k$

Frequency moments provide useful statistics:

- F_0: number of distinct elements in stream
- F_1: length of stream, m
- F_2: size of self join
- F_k, $k \geq 2$: skew of distribution

Trivial determination of F_k: maintain counters for each f_i

$\Rightarrow \Omega(n)$ bits needed

AMS Sampling

[Alon, Matias, Szegedy '96]

AMS Sampling (for estimating F_k)

AMS Sampling

[Alon, Matias, Szegedy '96]

AMS Sampling (for estimating F_k)

1. Pick random item a_j from stream

AMS Sampling

[Alon, Matias, Szegedy '96]

AMS Sampling (for estimating F_k)

1. Pick random item a_j from stream
2. Compute $r = \left|\{j' : j' \geq j, a_{j'} = a_j\}\right|$

AMS Sampling

[Alon, Matias, Szegedy '96]

AMS Sampling (for estimating F_k)

1. Pick random item a_j from stream
2. Compute $r = \left|\{j' : j' \geq j, a_{j'} = a_j\}\right|$

Example: stream 2, 3, 3, 2, 3, 1, 2, 3

$a_j = 3$, $r = 3$

AMS Sampling

[Alon, Matias, Szegedy '96]

AMS Sampling (for estimating F_k)

1. Pick random item a_j from stream
2. Compute $r = \left|\{j' : j' \geq j, a_{j'} = a_j\}\right|$
3. At the end of the stream calculate

$$m(r^k - (r-1)^k)$$

AMS Sampling

[Alon, Matias, Szegedy '96]

AMS Sampling (for estimating F_k)

1. Pick random item a_j from stream
2. Compute $r = \left|\{j' : j' \geq j, a_{j'} = a_j\}\right|$
3. At the end of the stream calculate

$$m(r^k - (r-1)^k)$$

for $k \geq 1$: $$Exp[\, m(r^k - (r-1)^k)\,]$$

AMS Sampling

[Alon, Matias, Szegedy '96]

AMS Sampling (for estimating F_k)

1. Pick random item a_j from stream
2. Compute $r = \left|\{j' : j' \geq j, a_{j'} = a_j\}\right|$
3. At the end of the stream calculate

$$m(r^k - (r-1)^k)$$

for $k \geq 1$: $$Exp[\, m(r^k - (r-1)^k)\,]$$

$$= \Big[\ m(f_1^k - (f_1 - 1)^k)$$

AMS Sampling

[Alon, Matias, Szegedy '96]

AMS Sampling (for estimating F_k)

1. Pick random item a_j from stream
2. Compute $r = \left|\{j' : j' \geq j, a_{j'} = a_j\}\right|$
3. At the end of the stream calculate

$$m(r^k - (r-1)^k)$$

for $k \geq 1$: $\quad Exp[\, m(r^k - (r-1)^k)\,]$

$$= \Big[\; m(f_1^k - (f_1 - 1)^k) + m((f_1 - 1)^k - (f_1 - 2)^k)$$

AMS Sampling

[Alon, Matias, Szegedy '96]

AMS Sampling (for estimating F_k)

1. Pick random item a_j from stream
2. Compute $r = \left|\{j' : j' \geq j, a_{j'} = a_j\}\right|$
3. At the end of the stream calculate

$$m(r^k - (r-1)^k)$$

for $k \geq 1$: $$Exp[\, m(r^k - (r-1)^k)\,]$$

$$= \Big[\; m(f_1^k - (f_1-1)^k) + m((f_1-1)^k - (f_1-2)^k) + \ldots + m(2^k - 1^k)$$

AMS Sampling

[Alon, Matias, Szegedy '96]

AMS Sampling (for estimating F_k)

1. Pick random item a_j from stream
2. Compute $r = \left|\{j' : j' \geq j, a_{j'} = a_j\}\right|$
3. At the end of the stream calculate

$$m(r^k - (r-1)^k)$$

for $k \geq 1$: $$Exp[\, m(r^k - (r-1)^k)\,]$$

$$= \Big[\ m(f_1^k - (f_1 - 1)^k) + m((f_1-1)^k - (f_1-2)^k) + \ldots + m(2^k - 1^k) + m \cdot 1^k$$

AMS Sampling

[Alon, Matias, Szegedy '96]

AMS Sampling (for estimating F_k)

1. Pick random item a_j from stream
2. Compute $r = \left|\{j' : j' \geq j, a_{j'} = a_j\}\right|$
3. At the end of the stream calculate

$$m(r^k - (r-1)^k)$$

for $k \geq 1$: $\quad Exp[\, m(r^k - (r-1)^k)\,]$

$$= \Big[\ m(f_1^k - (f_1-1)^k) + m((f_1-1)^k - (f_1-2)^k) + \ldots + m(2^k - 1^k) + m \cdot 1^k$$

$$+ m(f_2^k - (f_2-1)^k) + m((f_2-1)^k - (f_2-2)^k) + \ldots + m(2^k - 1^k) + m \cdot 1^k$$

AMS Sampling

[Alon, Matias, Szegedy '96]

AMS Sampling (for estimating F_k)

1. Pick random item a_j from stream
2. Compute $r = \left|\{j' : j' \geq j, a_{j'} = a_j\}\right|$
3. At the end of the stream calculate

$$m(r^k - (r-1)^k)$$

for $k \geq 1$: $\qquad Exp[\, m(r^k - (r-1)^k)\,]$

$$\begin{aligned}
= \Big[\; & m(f_1^k - (f_1-1)^k) + m((f_1-1)^k - (f_1-2)^k) + \ldots + m(2^k - 1^k) + m \cdot 1^k \\
& + m(f_2^k - (f_2-1)^k) + m((f_2-1)^k - (f_2-2)^k) + \ldots + m(2^k - 1^k) + m \cdot 1^k \\
& \ldots \\
& + m(f_n^k - (f_n-1)^k) + m((f_n-1)^k - (f_n-2)^k) + \ldots + m(2^k - 1^k) + m \cdot 1^k \Big]
\end{aligned}$$

AMS Sampling

[Alon, Matias, Szegedy '96]

AMS Sampling (for estimating F_k)

1. Pick random item a_j from stream
2. Compute $r = \left|\{j' : j' \geq j, a_{j'} = a_j\}\right|$
3. At the end of the stream calculate

$$m(r^k - (r-1)^k)$$

for $k \geq 1$: $\qquad Exp[\, m(r^k - (r-1)^k)\,]$

$$= \Big[\ m(f_1^k-(f_1-1)^k) + m((f_1-1)^k-(f_1-2)^k) + \ldots + m(2^k-1^k) + m\cdot 1^k$$

$$+m(f_2^k-(f_2-1)^k) + m((f_2-1)^k-(f_2-2)^k) + \ldots + m(2^k-1^k) + m\cdot 1^k$$

$$\ldots$$

$$+m(f_n^k-(f_n-1)^k) + m((f_n-1)^k-(f_n-2)^k) + \ldots + m(2^k-1^k) + m\cdot 1^k\Big]\frac{1}{m}$$

AMS Sampling

[Alon, Matias, Szegedy '96]

AMS Sampling (for estimating F_k)

1. Pick random item a_j from stream
2. Compute $r = \left|\{j' : j' \geq j, a_{j'} = a_j\}\right|$
3. At the end of the stream calculate

$$m(r^k - (r-1)^k)$$

for $k \geq 1$:

$$Exp[\, m(r^k - (r-1)^k)\,]$$
$$= f_1^k + f_2^k + f_3^k + \ldots + f_n^k$$

AMS Sampling (for estimating F_k)

1. Pick random item a_j from stream
2. Compute $r = \left|\{j' : j' \geq j, a_{j'} = a_j\}\right|$
3. At the end of the stream calculate

$$m(r^k - (r-1)^k)$$

for $k \geq 1$:

$$Exp[\, m(r^k - (r-1)^k)\,]$$

$$= \; f_1^k \; + \; f_2^k \; + \; f_3^k \; + \; \ldots \; + \; f_n^k \; = \; F_k$$

AMS Sampling

[Alon, Matias, Szegedy '96]

AMS Sampling (for estimating F_k)

1. Pick random item a_j from stream
2. Compute $r = \left|\{j' : j' \geq j, a_{j'} = a_j\}\right|$
3. At the end of the stream calculate

$$m(r^k - (r-1)^k)$$

To get $(1 \pm \varepsilon)$-estimate of F_k with probability $1 - \delta$:

AMS Sampling

[Alon, Matias, Szegedy '96]

AMS Sampling (for estimating F_k)

1. Pick random item a_j from stream
2. Compute $r = \left|\{j' : j' \geq j, a_{j'} = a_j\}\right|$
3. At the end of the stream calculate

$$m(r^k - (r-1)^k)$$

To get $(1 \pm \varepsilon)$-estimate of F_k with probability $1 - \delta$:

Run $\mathcal{O}\left(\frac{n^{1-1/k}}{\varepsilon^2}\right)$ parallel instances, take average A

AMS Sampling

[Alon, Matias, Szegedy '96]

AMS Sampling (for estimating F_k)

1. Pick random item a_j from stream
2. Compute $r = \left|\{j' : j' \geq j, a_{j'} = a_j\}\right|$
3. At the end of the stream calculate

$$m(r^k - (r-1)^k)$$

To get $(1 \pm \varepsilon)$-estimate of F_k with probability $1 - \delta$:

Run $\mathcal{O}\left(\frac{n^{1-1/k}}{\varepsilon^2}\right)$ parallel instances, take average A

$\Rightarrow$ Chebyshev-Ineq.: $Pr[\,|A - F_k| > \varepsilon \cdot F_k\,] \leq p < \frac{1}{2}$

AMS Sampling

[Alon, Matias, Szegedy '96]

AMS Sampling (for estimating F_k)

1. Pick random item a_j from stream
2. Compute $r = \left|\{j' : j' \geq j, a_{j'} = a_j\}\right|$
3. At the end of the stream calculate

$$m(r^k - (r-1)^k)$$

To get $(1 \pm \varepsilon)$-estimate of F_k with probability $1 - \delta$:

Run $\mathcal{O}\left(\frac{n^{1-1/k}}{\varepsilon^2}\right)$ parallel instances, take average A

$\Rightarrow$ Chebyshev-Ineq.: $Pr[\,|A - F_k| \;>\; \varepsilon \cdot F_k\,] \;\leq\; p \;<\; \frac{1}{2}$

Take median M over $\mathcal{O}\left(\log \frac{1}{\delta}\right)$ such averages

AMS Sampling

[Alon, Matias, Szegedy '96]

AMS Sampling (for estimating F_k)

1. Pick random item a_j from stream
2. Compute $r = \left|\{j' : j' \geq j, a_{j'} = a_j\}\right|$
3. At the end of the stream calculate

$$m(r^k - (r-1)^k)$$

To get $(1 \pm \varepsilon)$-estimate of F_k with probability $1 - \delta$:

Run $\mathcal{O}\left(\frac{n^{1-1/k}}{\varepsilon^2}\right)$ parallel instances, take average A

$\Rightarrow$ Chebyshev-Ineq.: $Pr[\,|A - F_k| > \varepsilon \cdot F_k\,] \leq p < \frac{1}{2}$

Take median M over $\mathcal{O}\left(\log \frac{1}{\delta}\right)$ such averages

$\Rightarrow$ Chernoff-Ineq.: $Pr[\,|M - F_k| > \varepsilon \cdot F_k\,] \leq \delta$

AMS Sampling

[Alon, Matias, Szegedy '96]

AMS Sampling (for estimating F_k)

1. Pick random item a_j from stream
2. Compute $r = \left|\{j' : j' \geq j, a_{j'} = a_j\}\right|$
3. At the end of the stream calculate

$$m(r^k - (r-1)^k)$$

To get $(1 \pm \varepsilon)$-estimate of F_k with probability $1 - \delta$:

Run $\mathcal{O}\left(\frac{n^{1-1/k}}{\varepsilon^2}\right)$ parallel instances, take average A

$\Rightarrow$ Chebyshev-Ineq.: $Pr[\,|A - F_k| > \varepsilon \cdot F_k\,] \leq p < \frac{1}{2}$

Take median M over $\mathcal{O}\left(\log \frac{1}{\delta}\right)$ such averages

$\Rightarrow$ Chernoff-Ineq.: $Pr[\,|M - F_k| > \varepsilon \cdot F_k\,] \leq \delta$

Memory consumption: $\mathcal{O}\left(\frac{n^{1-1/k}}{\varepsilon^2} \cdot \log \frac{1}{\delta} \cdot (\log n + \log m)\right)$ bits

Sliding Window Sampling

Sliding Window Sampling

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$

Sliding Window Sampling

$$\downarrow$$

$$a_1,\ a_2,\ a_3,\ a_4,\ a_5,\ a_6,\ a_7,\ a_8,\ a_9,\ a_{10},\ a_{11},\ a_{12},\ a_{13},\ \ldots,\ a_i,\ \ldots,\ a_m$$

Sliding Window Sampling

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, \overset{\downarrow}{a_i}, \ldots, a_m$$

current sample: a_2

Sliding Window Sampling

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$$

current sample: a_2

Current sample might be "far away" from actual item

Sliding Window Sampling

$$\downarrow$$

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$$

current sample: a_2

Current sample might be "far away" from actual item

- fine for some applications
- for others only w recent items matter

Sliding Window Sampling

$$\downarrow$$

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$$

current sample: a_2

Current sample might be "far away" from actual item

- fine for some applications
- for others only w recent items matter

  $\Rightarrow$ only consider items in a sliding window of size w

Sliding Window Sampling

$$\downarrow$$
$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$$

Current sample might be "far away" from actual item

- fine for some applications
- for others only w recent items matter

  $\Rightarrow$ only consider items in a sliding window of size w

Sliding Window Sampling

$$\downarrow$$
$$a_1,\ a_2,\ a_3,\ a_4,\ a_5,\ a_6,\ a_7,\ a_8,\ a_9,\ a_{10},\ a_{11},\ a_{12},\ a_{13},\ \ldots,\ a_i,\ \ldots,\ a_m$$

Current sample might be "far away" from actual item

- fine for some applications
- for others only w recent items matter

  $\Rightarrow$ only consider items in a sliding window of size w

Sliding Window Sampling

$$\downarrow$$

$$a_1,\ a_2,\ a_3,\ a_4,\ a_5,\ a_6,\ a_7,\ a_8,\ a_9,\ a_{10},\ a_{11},\ a_{12},\ a_{13},\ \ldots,\ a_i,\ \ldots,\ a_m$$

Current sample might be "far away" from actual item

- fine for some applications
- for others only w recent items matter

$\Rightarrow$ only consider items in a sliding window of size w

Sliding Window Sampling

$$\downarrow$$

$$a_1,\ a_2,\ a_3,\ a_4,\ a_5,\ a_6,\ a_7,\ a_8,\ a_9,\ a_{10},\ a_{11},\ a_{12},\ a_{13},\ \ldots,\ a_i,\ \ldots,\ a_m$$

Current sample might be "far away" from actual item

- fine for some applications
- for others only w recent items matter

$\Rightarrow$ only consider items in a sliding window of size w

Sliding Window Sampling

$$\downarrow$$

$$\boxed{a_1,\ a_2,\ a_3,\ a_4,\ a_5,}\ a_6,\ a_7,\ a_8,\ a_9,\ a_{10},\ a_{11},\ a_{12},\ a_{13},\ \ldots,\ a_i,\ \ldots,\ a_m$$

Current sample might be "far away" from actual item

- fine for some applications
- for others only w recent items matter

$\Rightarrow$ only consider items in a sliding window of size w

Sliding Window Sampling

$$a_1, \boxed{a_2,\ a_3,\ a_4,\ a_5,\ \overset{\downarrow}{a_6},}\ a_7,\ a_8,\ a_9,\ a_{10},\ a_{11},\ a_{12},\ a_{13},\ \dots,\ a_i,\ \dots,\ a_m$$

Current sample might be "far away" from actual item

- fine for some applications
- for others only w recent items matter

  $\Rightarrow$ only consider items in a sliding window of size w

Sliding Window Sampling

$$a_1, a_2, \boxed{a_3, a_4, a_5, a_6, \overset{\downarrow}{a_7},} a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$$

Current sample might be "far away" from actual item

- fine for some applications
- for others only w recent items matter

  $\Rightarrow$ only consider items in a sliding window of size w

Sliding Window Sampling

$$a_1, \; a_2, \; a_3, \boxed{a_4, \; a_5, \; a_6, \; a_7, \; a_8,} \; a_9, \; a_{10}, \; a_{11}, \; a_{12}, \; a_{13}, \; \ldots, \; a_i, \; \ldots, \; a_m$$

Current sample might be "far away" from actual item

- fine for some applications
- for others only w recent items matter

$\Rightarrow$ only consider items in a sliding window of size w

Sliding Window Sampling

$$a_1, a_2, a_3, a_4, \boxed{a_5, a_6, a_7, a_8, a_9,} \, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$$

Current sample might be "far away" from actual item

- fine for some applications
- for others only w recent items matter

$\Rightarrow$ only consider items in a sliding window of size w

Sliding Window Sampling

$$a_1, a_2, a_3, a_4, a_5, \boxed{a_6, a_7, a_8, a_9, a_{10},} a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$$

Current sample might be "far away" from actual item

- fine for some applications
- for others only w recent items matter

  $\Rightarrow$ only consider items in a sliding window of size w

Sliding Window Sampling

$$a_1,\ a_2,\ a_3,\ a_4,\ a_5,\ \boxed{a_6,\ a_7,\ a_8,\ a_9,\ a_{10},}\ a_{11},\ a_{12},\ a_{13},\ \ldots,\ a_i,\ \ldots,\ a_m$$

Goal: sample an item uniformly from sliding window of size w

Sliding Window Sampling

$$a_1, a_2, a_3, a_4, a_5, \boxed{a_6, a_7, a_8, a_9, a_{10},} a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$$

Goal: sample an item uniformly from sliding window of size w

Trivial: - memorize whole window content $\Rightarrow w \cdot \log n$ bits

- impractical if w is large

Sliding Window Sampling

$$a_1, a_2, a_3, a_4, a_5, \boxed{a_6, a_7, a_8, a_9, a_{10},} \, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$$

Goal: sample an item uniformly from sliding window of size w

Trivial: - memorize whole window content $\Rightarrow w \cdot \log n$ bits

- impractical if w is large

Better idea:

Algorithm:

Sliding Window Sampling

$$a_1, a_2, a_3, a_4, a_5, \boxed{a_6, a_7, a_8, a_9, a_{10},} a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$$

Goal: sample an item uniformly from sliding window of size w

Trivial: - memorize whole window content $\Rightarrow$ $w \cdot \log n$ bits

- impractical if w is large

Better idea:

Algorithm: 1: For each a_i pick random value $r_i \in (0, 1)$

Sliding Window Sampling

$$a_1,\ a_2,\ a_3,\ a_4,\ a_5,\ \boxed{a_6,\ a_7,\ a_8,\ a_9,\ a_{10},}\ a_{11},\ a_{12},\ a_{13},\ \ldots,\ a_i,\ \ldots,\ a_m$$

Goal: sample an item uniformly from sliding window of size w

Trivial: - memorize whole window content $\Rightarrow w \cdot \log n$ bits

- impractical if w is large

Better idea:

Algorithm: 1: For each a_i pick random value $r_i \in (0,1)$

2: In window $(a_{i-w+1}, \ldots, a_i)$ choose a_j with smallest r_j

Sliding Window Sampling

$$a_1,\ a_2,\ a_3,\ a_4,\ a_5,\boxed{a_6,\ a_7,\ a_8,\ a_9,\ a_{10},}\ a_{11},\ a_{12},\ a_{13},\ \ldots,\ a_i,\ \ldots,\ a_m$$

Goal: sample an item uniformly from sliding window of size w

Trivial: - memorize whole window content $\Rightarrow w \cdot \log n$ bits

- impractical if w is large

Better idea:

Algorithm: 1: For each a_i pick random value $r_i \in (0, 1)$

2: In window $(a_{i-w+1}, \ldots, a_i)$ choose a_j with smallest r_j

3: Only maintain items in window whose r-value is minimal among subsequent r-values

Sliding Window Sampling

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$

Algorithm: 1: For each a_i pick random value $r_i \in (0, 1)$

2: In window $(a_{i-w+1}, \ldots, a_i)$ choose a_j with smallest r_j

3: Only maintain items in window whose r-value is minimal among subsequent r-values

Sliding Window Sampling

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$$

Algorithm: 1: For each a_i pick random value $r_i \in (0,1)$

2: In window $(a_{i-w+1}, \ldots, a_i)$ choose a_j with smallest r_j

3: Only maintain items in window whose r-value is minimal among subsequent r-values

Sliding Window Sampling

$\downarrow$

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$

random value $r_1 = 0.2$

Algorithm: 1: For each a_i pick random value $r_i \in (0, 1)$

2: In window $(a_{i-w+1}, \ldots, a_i)$ choose a_j with smallest r_j

3: Only maintain items in window whose r-value is minimal among subsequent r-values

Sliding Window Sampling

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$

random value $r_1 = 0.2$

a_1

0.2

Algorithm: 1: For each a_i pick random value $r_i \in (0, 1)$

2: In window $(a_{i-w+1}, \ldots, a_i)$ choose a_j with smallest r_j

3: Only maintain items in window whose r-value is minimal among subsequent r-values

Sliding Window Sampling

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$

random value $r_1 = 0.2$

current sample

a_1

0.2

Algorithm: 1: For each a_i pick random value $r_i \in (0, 1)$

2: In window $(a_{i-w+1}, \ldots, a_i)$ choose a_j with smallest r_j

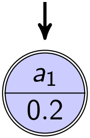
3: Only maintain items in window whose r-value is minimal among subsequent r-values

Sliding Window Sampling

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$

random value $r_2 = 0.4$

current sample

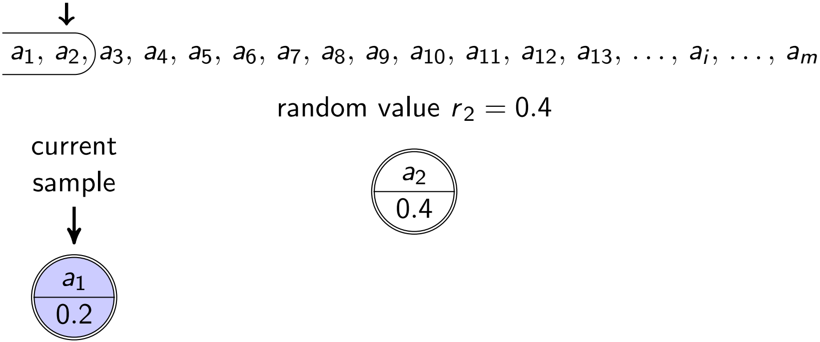


Algorithm: 1: For each a_i pick random value $r_i \in (0, 1)$

2: In window $(a_{i-w+1}, \ldots, a_i)$ choose a_j with smallest r_j

3: Only maintain items in window whose r-value is minimal among subsequent r-values

Sliding Window Sampling



Algorithm: 1: For each a_i pick random value $r_i \in (0, 1)$

2: In window $(a_{i-w+1}, \ldots, a_i)$ choose a_j with smallest r_j

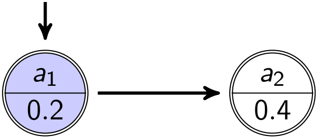
3: Only maintain items in window whose r-value is minimal among subsequent r-values

Sliding Window Sampling

$a_1,\ a_2,\ a_3,\ a_4,\ a_5,\ a_6,\ a_7,\ a_8,\ a_9,\ a_{10},\ a_{11},\ a_{12},\ a_{13},\ \ldots,\ a_i,\ \ldots,\ a_m$

random value $r_2 = 0.4$

current sample



Algorithm: 1: For each a_i pick random value $r_i \in (0,1)$

2: In window $(a_{i-w+1}, \ldots, a_i)$ choose a_j with smallest r_j

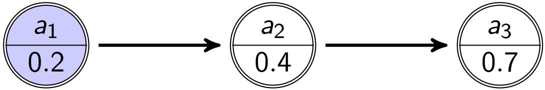
3: Only maintain items in window whose r-value is minimal among subsequent r-values

Sliding Window Sampling

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$

random value $r_3 = 0.7$

current sample



Algorithm: 1: For each a_i pick random value $r_i \in (0, 1)$

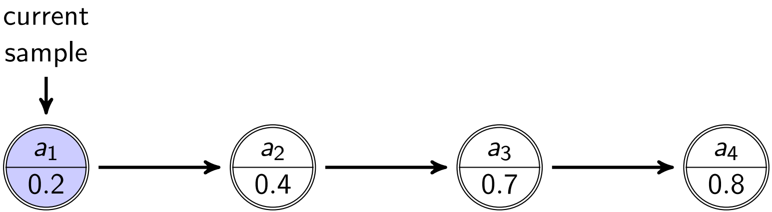
2: In window $(a_{i-w+1}, \ldots, a_i)$ choose a_j with smallest r_j

3: Only maintain items in window whose r-value is minimal among subsequent r-values

Sliding Window Sampling

$$a_1,\ a_2,\ a_3,\ a_4,\ a_5,\ a_6,\ a_7,\ a_8,\ a_9,\ a_{10},\ a_{11},\ a_{12},\ a_{13},\ \ldots,\ a_i,\ \ldots,\ a_m$$

random value $r_4 = 0.8$



Algorithm: 1: For each a_i pick random value $r_i \in (0, 1)$

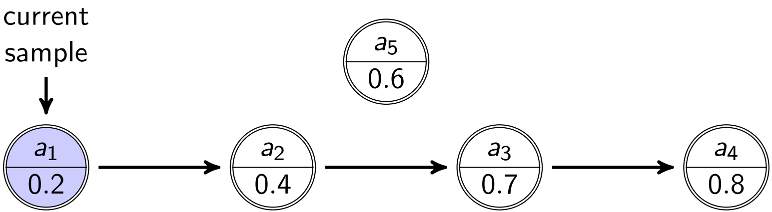
2: In window $(a_{i-w+1}, \ldots, a_i)$ choose a_j with smallest r_j

3: Only maintain items in window whose r-value is minimal among subsequent r-values

Sliding Window Sampling

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$

random value $r_5 = 0.6$



Algorithm: 1: For each a_i pick random value $r_i \in (0, 1)$

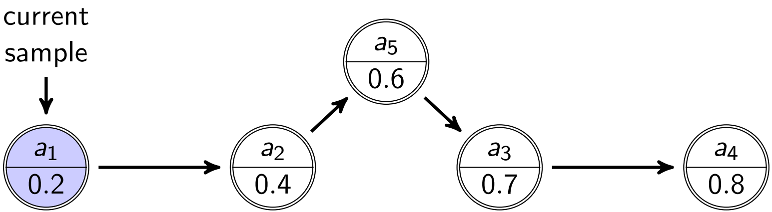
2: In window $(a_{i-w+1}, \ldots, a_i)$ choose a_j with smallest r_j

3: Only maintain items in window whose r-value is minimal among subsequent r-values

Sliding Window Sampling

$\boxed{a_1, a_2, a_3, a_4, a_5,} a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$

random value $r_5 = 0.6$



Algorithm: 1: For each a_i pick random value $r_i \in (0,1)$

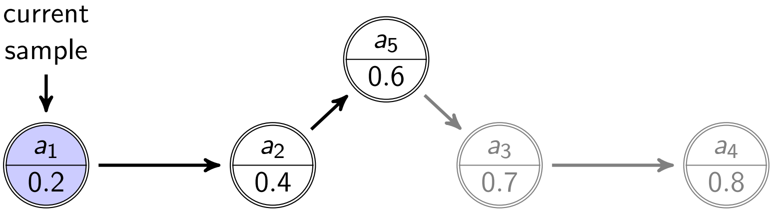
2: In window $(a_{i-w+1}, \ldots, a_i)$ choose a_j with smallest r_j

3: Only maintain items in window whose r-value is minimal among subsequent r-values

Sliding Window Sampling

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$

random value $r_5 = 0.6$



Algorithm: 1: For each a_i pick random value $r_i \in (0, 1)$

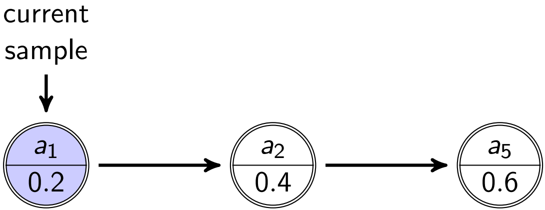
2: In window $(a_{i-w+1}, \ldots, a_i)$ choose a_j with smallest r_j

3: Only maintain items in window whose r-value is minimal among subsequent r-values

Sliding Window Sampling

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$

random value $r_5 = 0.6$



Algorithm: 1: For each a_i pick random value $r_i \in (0,1)$

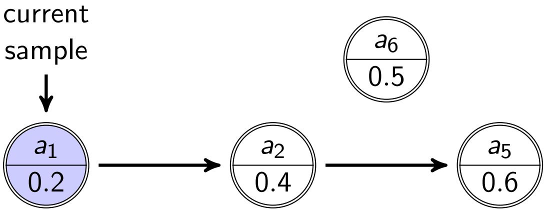
2: In window $(a_{i-w+1}, \ldots, a_i)$ choose a_j with smallest r_j

3: Only maintain items in window whose r-value is minimal among subsequent r-values

Sliding Window Sampling

$$a_1, \boxed{a_2,\ a_3,\ a_4,\ a_5,\ a_6,}\ a_7,\ a_8,\ a_9,\ a_{10},\ a_{11},\ a_{12},\ a_{13},\ \ldots,\ a_i,\ \ldots,\ a_m$$

random value $r_6 = 0.5$



Algorithm: 1: For each a_i pick random value $r_i \in (0, 1)$

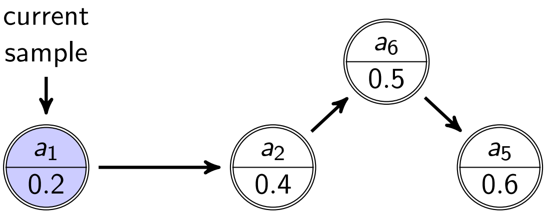
2: In window $(a_{i-w+1}, \ldots, a_i)$ choose a_j with smallest r_j

3: Only maintain items in window whose r-value is minimal among subsequent r-values

Sliding Window Sampling

$$a_1, \boxed{a_2,\ a_3,\ a_4,\ a_5,\ a_6,}\ a_7,\ a_8,\ a_9,\ a_{10},\ a_{11},\ a_{12},\ a_{13},\ \ldots,\ a_i,\ \ldots,\ a_m$$

random value $r_6 = 0.5$



Algorithm: 1: For each a_i pick random value $r_i \in (0, 1)$

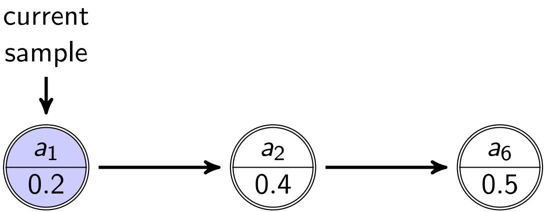
2: In window $(a_{i-w+1}, \ldots, a_i)$ choose a_j with smallest r_j

3: Only maintain items in window whose r-value is minimal among subsequent r-values

Sliding Window Sampling

$$a_1, \boxed{a_2,\ a_3,\ a_4,\ a_5,\ a_6,}\ a_7,\ a_8,\ a_9,\ a_{10},\ a_{11},\ a_{12},\ a_{13},\ \ldots,\ a_i,\ \ldots,\ a_m$$

random value $r_6 = 0.5$



Algorithm: 1: For each a_i pick random value $r_i \in (0,1)$

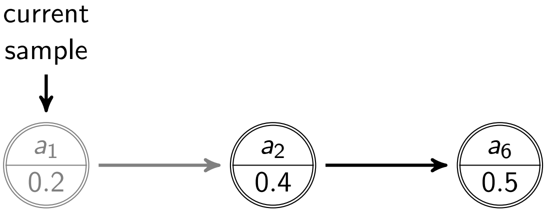
2: In window $(a_{i-w+1}, \ldots, a_i)$ choose a_j with smallest r_j

3: Only maintain items in window whose r-value is minimal among subsequent r-values

Sliding Window Sampling

$a_1, \boxed{a_2,\ a_3,\ a_4,\ a_5,\ a_6,}\ a_7,\ a_8,\ a_9,\ a_{10},\ a_{11},\ a_{12},\ a_{13},\ \ldots,\ a_i,\ \ldots,\ a_m$

random value $r_6 = 0.5$



Algorithm: 1: For each a_i pick random value $r_i \in (0, 1)$

2: In window $(a_{i-w+1}, \ldots, a_i)$ choose a_j with smallest r_j

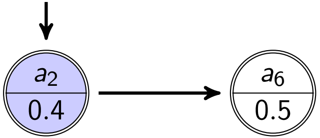
3: Only maintain items in window whose r-value is minimal among subsequent r-values

Sliding Window Sampling

$a_1, \boxed{a_2,\ a_3,\ a_4,\ a_5,\ a_6,}\ a_7,\ a_8,\ a_9,\ a_{10},\ a_{11},\ a_{12},\ a_{13},\ \ldots,\ a_i,\ \ldots,\ a_m$

random value $r_6 = 0.5$

current sample



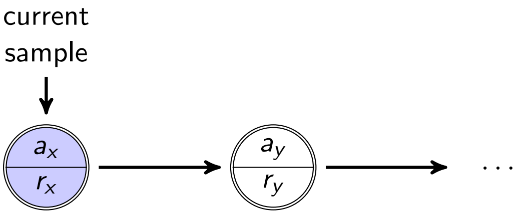
Algorithm: 1: For each a_i pick random value $r_i \in (0, 1)$

2: In window $(a_{i-w+1}, \ldots, a_i)$ choose a_j with smallest r_j

3: Only maintain items in window whose r-value is minimal among subsequent r-values

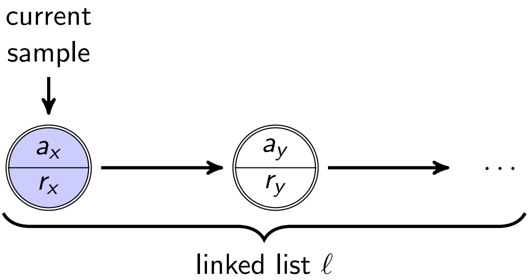
Sliding Window Sampling

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$



Sliding Window Sampling

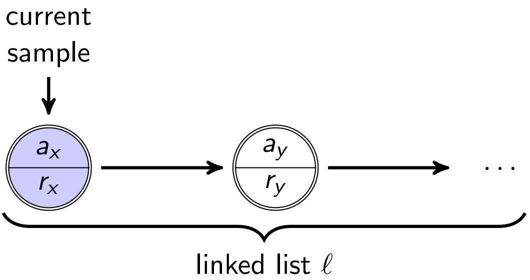
$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$



Analysis: What is the length of ℓ ?

Sliding Window Sampling

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$

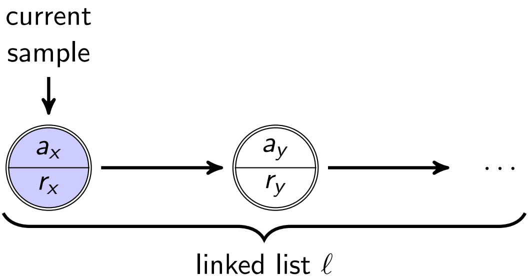


Analysis: What is the length of ℓ ?

Worst case: $|\ell| = w$ But that is very unlikely.

Sliding Window Sampling

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$

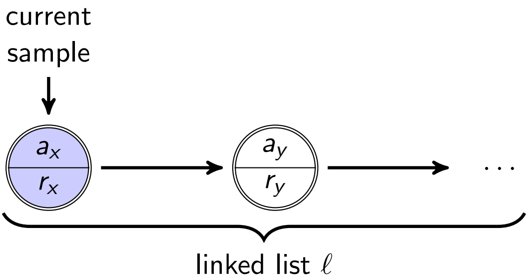


Analysis: What is the length of ℓ ?

$$Exp[\,|\ell|\,] =$$

Sliding Window Sampling

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$

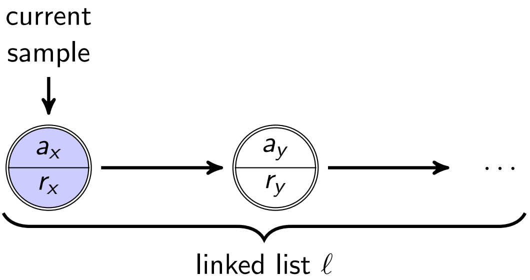


Analysis: What is the length of ℓ ?

$$Exp[\,|\ell|\,] = Pr[a_i \in \ell] +$$

Sliding Window Sampling

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$

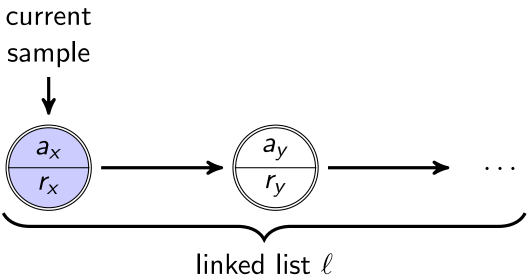


Analysis: What is the length of ℓ ?

$$Exp[\,|\ell|\,] = Pr[a_i \in \ell] + Pr[a_{i-1} \in \ell] +$$

Sliding Window Sampling

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$

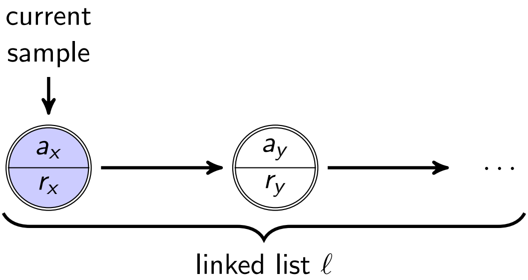


Analysis: What is the length of ℓ ?

$$Exp[\,|\ell|\,] = Pr[a_i \in \ell\,] + Pr[a_{i-1} \in \ell\,] + \ldots + Pr[a_{i-w+1} \in \ell\,]$$

Sliding Window Sampling

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$

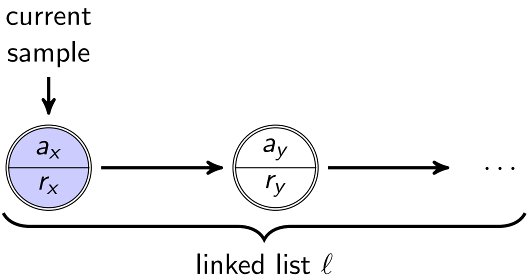


Analysis: What is the length of ℓ ?

$$\begin{aligned} Exp[\,|\ell|\,] &= Pr[a_i \in \ell\,] + Pr[a_{i-1} \in \ell\,] + \ldots + Pr[a_{i-w+1} \in \ell\,] \\ &= 1 + \end{aligned}$$

Sliding Window Sampling

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$

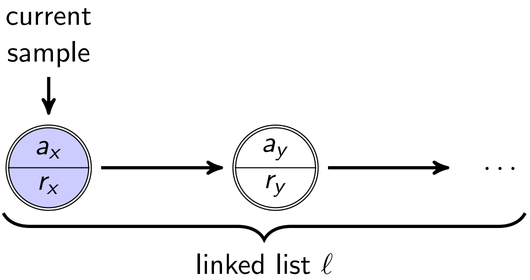


Analysis: What is the length of ℓ ?

$$
\begin{aligned}
Exp[\,|\ell|\,] &= Pr[a_i \in \ell\,] + Pr[a_{i-1} \in \ell\,] + \ldots + Pr[a_{i-w+1} \in \ell\,] \\
&= 1 + \tfrac{1}{2} +
\end{aligned}
$$

Sliding Window Sampling

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$



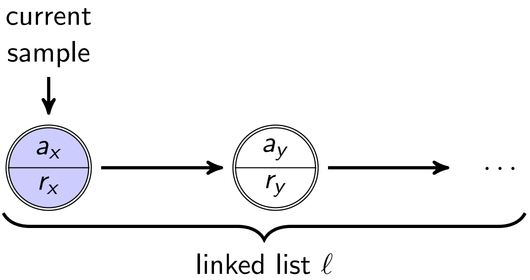
Analysis: What is the length of ℓ ?

$$Exp[\,|\ell|\,] = Pr[a_i \in \ell] + Pr[a_{i-1} \in \ell] + \ldots + Pr[a_{i-w+1} \in \ell]$$

$$= 1 + \frac{1}{2} + \frac{1}{3} +$$

Sliding Window Sampling

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$



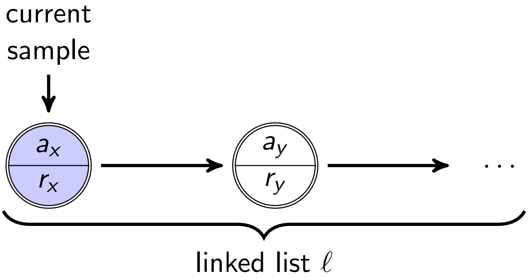
Analysis: What is the length of ℓ ?

$$Exp[\,|\ell|\,] = Pr[a_i \in \ell\,] + Pr[a_{i-1} \in \ell\,] + \ldots + Pr[a_{i-w+1} \in \ell\,]$$

$$= 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \ldots + \frac{1}{w}$$

Sliding Window Sampling

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$



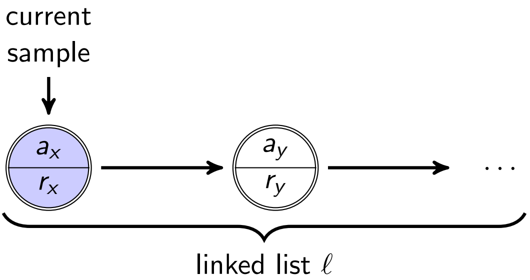
Analysis: What is the length of ℓ ?

$$Exp[\,|\ell|\,] = Pr[a_i \in \ell] + Pr[a_{i-1} \in \ell] + \ldots + Pr[a_{i-w+1} \in \ell]$$

$$= 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \ldots + \frac{1}{w} \; = \; \mathcal{O}(\log w)$$

Sliding Window Sampling

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_i, \ldots, a_m$



Analysis:

$$Exp[\,\text{memory usage}\,] = \mathcal{O}(\log w \cdot \log n)$$

Count-Min Sketch

[Cormode, Muthukrishnan '04]

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_j, \ldots, a_m$$

Count-Min Sketch

[Cormode, Muthukrishnan '04]

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_j, \ldots, a_m$

Point query: $f_i = ?$

Count-Min Sketch

[Cormode, Muthukrishnan '04]

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_j, \ldots, a_m$

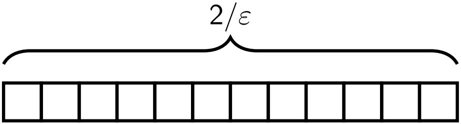
Point query: $f_i = ?$

Trivial solution: maintain n counters

Count-Min Sketch

[Cormode, Muthukrishnan '04]

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_j, \ldots, a_m$

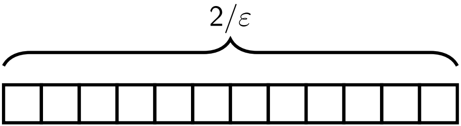


Count-Min Sketch

[Cormode, Muthukrishnan '04]

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_j, \ldots, a_m$

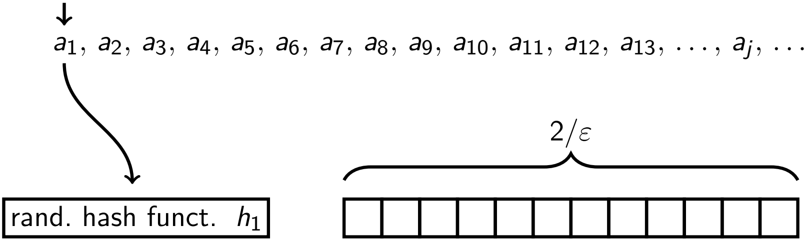
rand. hash funct. h_1



Count-Min Sketch

[Cormode, Muthukrishnan '04]

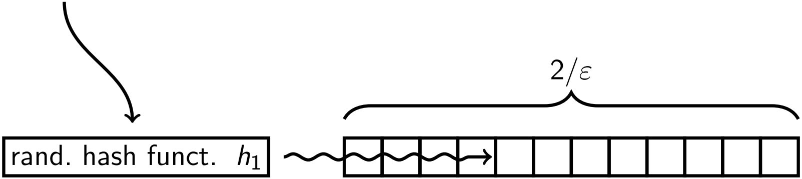
$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_j, \ldots, a_m$



Count-Min Sketch

[Cormode, Muthukrishnan '04]

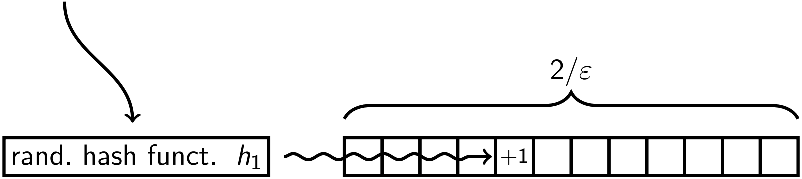
$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_j, \ldots, a_m$



Count-Min Sketch

[Cormode, Muthukrishnan '04]

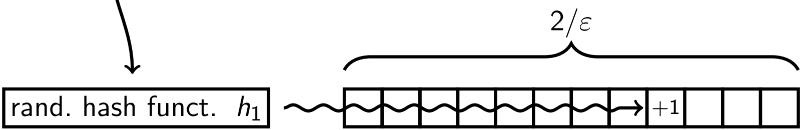
$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_j, \ldots, a_m$$



Count-Min Sketch

[Cormode, Muthukrishnan '04]

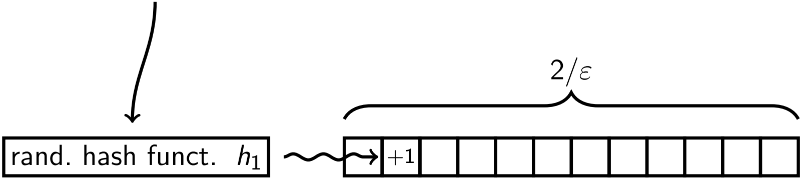
$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_j, \ldots, a_m$



Count-Min Sketch

[Cormode, Muthukrishnan '04]

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_j, \ldots, a_m$



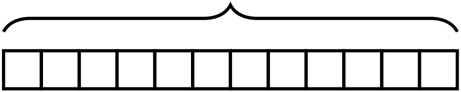
Count-Min Sketch

[Cormode, Muthukrishnan '04]

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_j, \ldots, a_m$

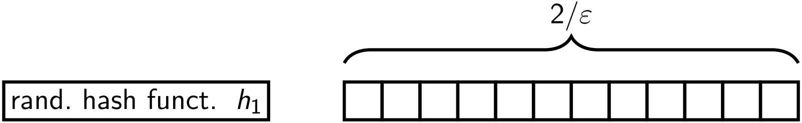
$2/\varepsilon$

rand. hash funct. h_1



Count-Min Sketch

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_j, \ldots, a_m$

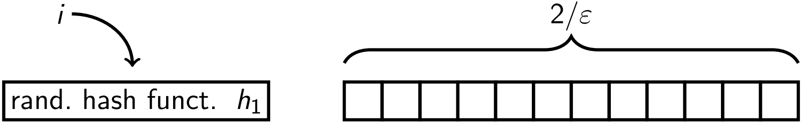


Point query: $f_i = ?$

Count-Min Sketch

[Cormode, Muthukrishnan '04]

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_j, \ldots, a_m$

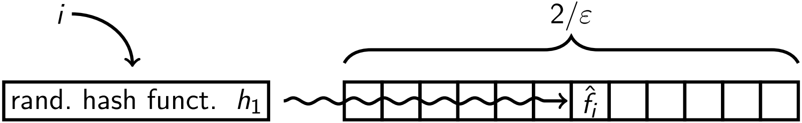


Point query: $f_i = ?$

Count-Min Sketch

[Cormode, Muthukrishnan '04]

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_j, \ldots, a_m$

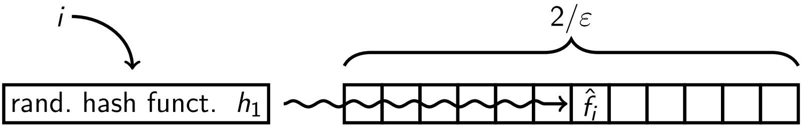


Point query: $f_i = ?$

Count-Min Sketch

[Cormode, Muthukrishnan '04]

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_j, \ldots, a_m$



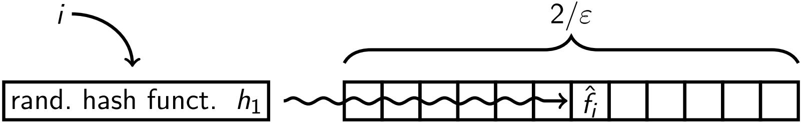
Point query: $f_i = ?$

Give $\hat{f}_i$ as estimate for f_i

Count-Min Sketch

[Cormode, Muthukrishnan '04]

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_j, \ldots, a_m$



Point query: $f_i = ?$

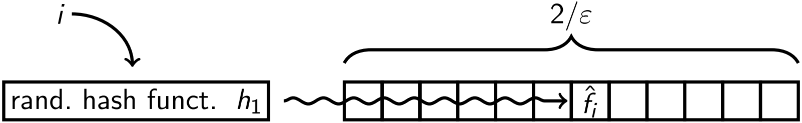
Give $\hat{f}_i$ as estimate for f_i

Analysis:

Count-Min Sketch

[Cormode, Muthukrishnan '04]

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_j, \ldots, a_m$



Point query: $f_i = ?$

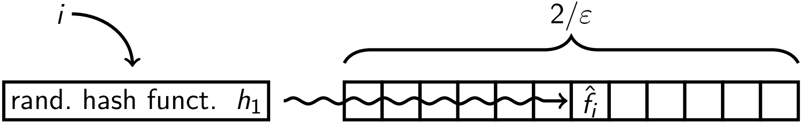
Give $\hat{f}_i$ as estimate for f_i

Analysis: $\hat{f}_i \geq f_i$

Count-Min Sketch

[Cormode, Muthukrishnan '04]

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_j, \ldots, a_m$



Point query: $f_i = ?$

Give $\hat{f}_i$ as estimate for f_i

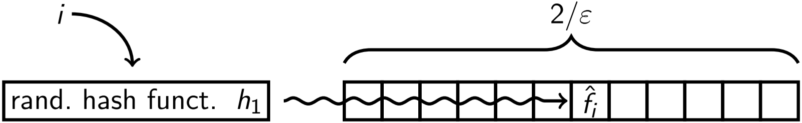
Analysis: $\hat{f}_i \geq f_i$

$$Exp[\,\hat{f}_i\,] \leq f_i + \varepsilon \cdot m/2$$

Count-Min Sketch

[Cormode, Muthukrishnan '04]

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_j, \ldots, a_m$



Point query: $f_i = ?$

Give $\hat{f}_i$ as estimate for f_i

Analysis: $\hat{f}_i \geq f_i$

$Exp[\,\hat{f}_i\,] \leq f_i + \varepsilon \cdot m/2$

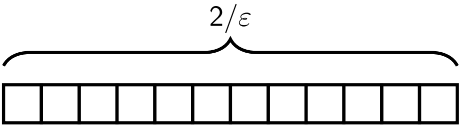
Markov-Ineq.: $Pr[\,\hat{f}_i > f_i + \varepsilon \cdot m\,] \leq \frac{1}{2}$

Count-Min Sketch

[Cormode, Muthukrishnan '04]

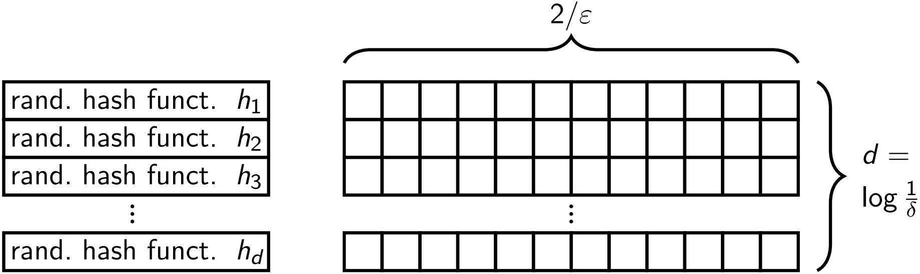
$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_j, \ldots, a_m$

rand. hash funct. h_1



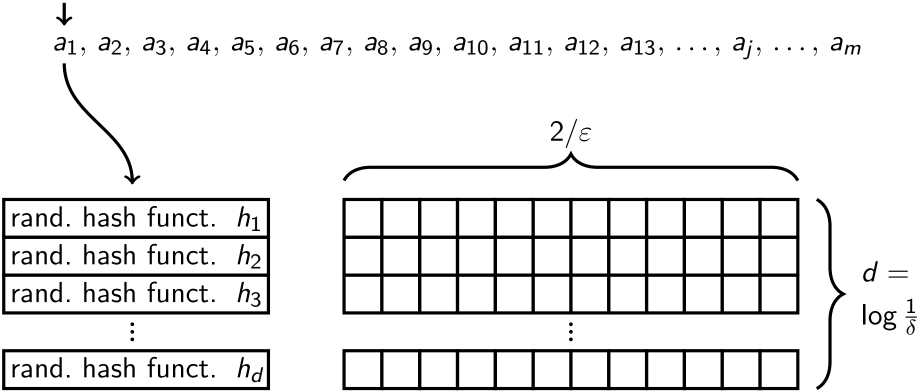
Count-Min Sketch

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_j, \ldots, a_m$



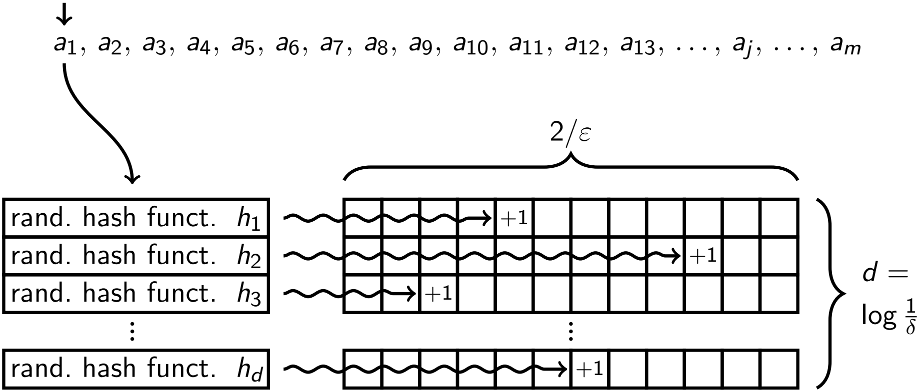
Count-Min Sketch

[Cormode, Muthukrishnan '04]



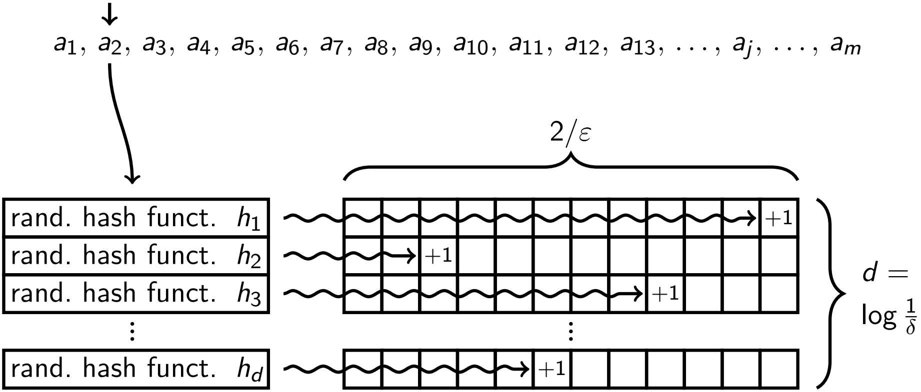
Count-Min Sketch

[Cormode, Muthukrishnan '04]



Count-Min Sketch

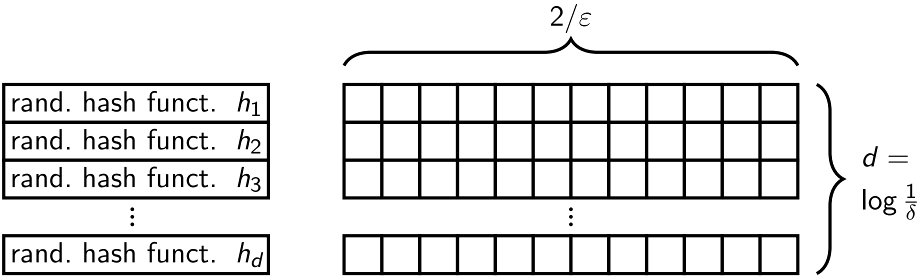
[Cormode, Muthukrishnan '04]



Count-Min Sketch

[Cormode, Muthukrishnan '04]

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_j, \ldots, a_m$

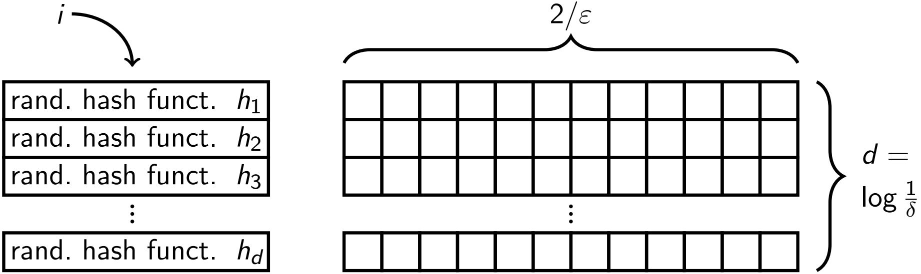


Point query: $f_i = ?$

Count-Min Sketch

[Cormode, Muthukrishnan '04]

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_j, \ldots, a_m$

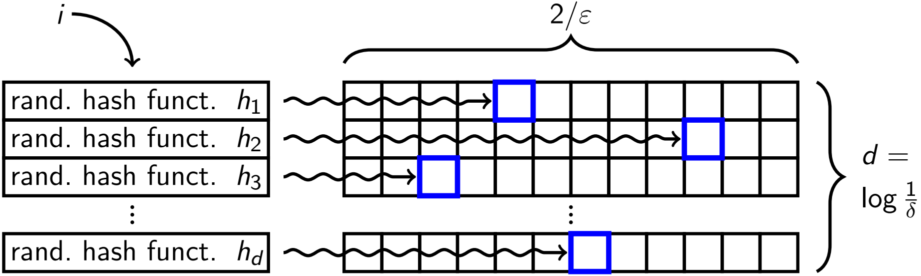


Point query: $f_i = ?$

Count-Min Sketch

[Cormode, Muthukrishnan '04]

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_j, \ldots, a_m$

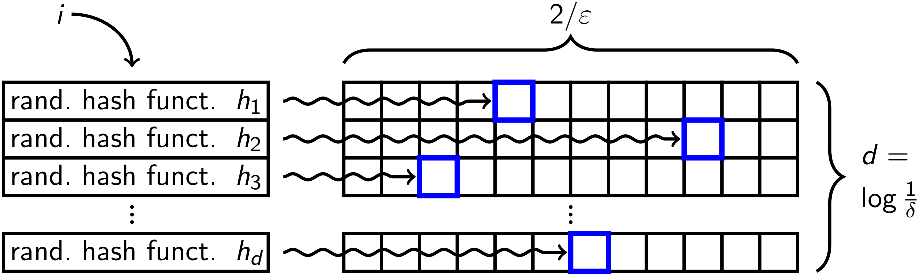


Point query: $f_i = ?$

Count-Min Sketch

[Cormode, Muthukrishnan '04]

$$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_j, \ldots, a_m$$



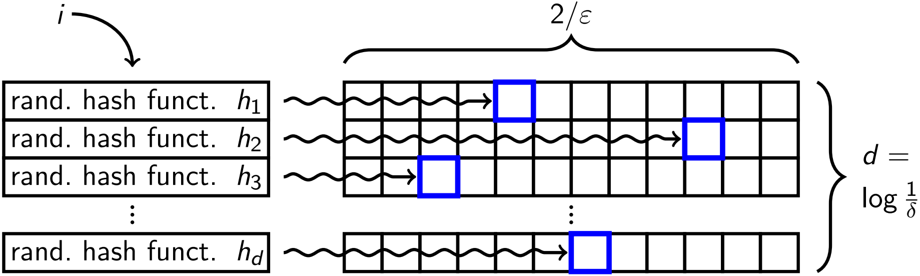
Point query: $f_i = ?$

Give $\hat{f}_i :=$ minimum of $\boxed{}$-values as estimate for f_i

Count-Min Sketch

[Cormode, Muthukrishnan '04]

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_j, \ldots, a_m$

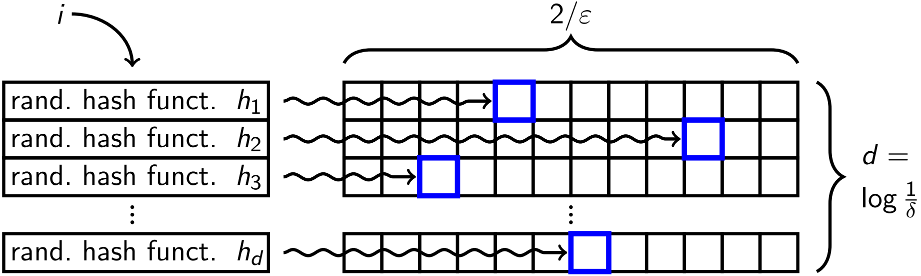


Analysis: $Pr[\,\hat{f}_i > f_i + \varepsilon \cdot m\,] \leq \left(\frac{1}{2}\right)^d = \delta$

Count-Min Sketch

[Cormode, Muthukrishnan '04]

$a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \ldots, a_j, \ldots, a_m$



Analysis: $Pr[\,\hat{f}_i > f_i + \varepsilon \cdot m\,] \leq \left(\frac{1}{2}\right)^d = \delta$

Memory consumption: $\mathcal{O}\left(\frac{1}{\varepsilon} \cdot \log\frac{1}{\delta} \cdot \log m + \log\frac{1}{\delta} \cdot \log n\right)$

Recap

- Reservoir sampling for uniform selection
- AMS sampling for frequency moments
- Sliding window sampling
- Count-Min sketch for point queries

References

- N. Alon, Y. Matias, and M. Szegedy: The space complexity of approximating the frequency moments. J. Comput. Syst. Sci. 58, pp. 137-147, 1999.
- G. Cormode and S. Muthukrishnan: An Improved Data Stream Summary: The Count-Min Sketch and its Applications. Journal of Algorithms 55(1), pp. 58-75, 2005.
- B. Lahiri and S. Tirthapura: Stream Sampling. Pages 2838-2842 in: Ling Liu, M. Tamer zsu (Eds.): Encyclopedia of Database Systems. Springer US, 2009.
- S. Muthukrishnan: Data Streams: Algorithms and Applications. Foundations and Trends in Theoretical Computer Science, 1(2), 2005.
- J. Vitter: Random Sampling with a Reservoir. ACM Transactions on Mathematical Software, 11(1), pp. 37-57, 1985.