Frequency Estimation

CS 5968/6968: Data Str & Alg for Scalable Computing Spring 2023

Benwei Shi

University of Utah

2023-02-13

Table of contents

Table of Contents

Frequency

Given

- Metadata: a universe $[u] := \{1, 2, \ldots, u\}$,
  - u is bigger than memory, $O(\lg u)$ is constant.

Frequency

Given

- Metadata: a universe $[u] := \{1, 2, \ldots, u\}$,
  - u is bigger than memory, $O(\lg u)$ is constant.
- Data: a sequence $X := \left[x_i \in [u]\right]_{i=1}^{n}$,
  - i.e. $[x_1, x_2, \ldots, x_n]$, $x_i \in [u]$ for all $i \in [n]$.

Frequency

Given

- Metadata: a universe $[u] := \{1, 2, \ldots, u\}$,
  - u is bigger than memory, $O(\lg u)$ is constant.
- Data: a sequence $X := \left[x_i \in [u]\right]_{i=1}^{n}$,
  - i.e. $[x_1, x_2, \ldots, x_n]$, $x_i \in [u]$ for all $i \in [n]$.

Goal

For any query $q \in [u]$, return the its count or frequency:

$$c(q) := \sum_{x \in X} \mathbb{1}(x, q), \quad f(q) := \frac{c(q)}{n}.$$

Frequency

Given

- Metadata: a universe $[u] := \{1, 2, \dots, u\}$,
  - u is bigger than memory, $O(\lg u)$ is constant.
- Data: a sequence $X := \left[x_i \in [u]\right]_{i=1}^{n}$,
  - i.e. $[x_1, x_2, \dots, x_n]$, $x_i \in [u]$ for all $i \in [n]$.

Goal

For any query $q \in [u]$, return the its count or frequency:

$$c(q) := \sum_{x \in X} \mathbb{1}(x, q), \quad f(q) := \frac{c(q)}{n}.$$

What if n is also too big for memory? Even bigger than external memory?

DDoS Attack Detection at Router

Detect high frequency IP addresses with limited memory.

Frequency Estimation in Stream

Given

- Metadata: a universe $[u] := \{1, 2, \ldots, u\}$,
  - u is bigger than memory, $O(\lg u)$ is constant.
- Data: a sequence $X := \left[x_i \in [u]\right]_{i=1}^{n}$,
  - n is bigger than memory.

Goal

For any query $q \in [u]$, return the ε-approximation of its frequency, $\hat{f}_\varepsilon(q)$, s.t.

$$f(q) - \varepsilon \leq \hat{f}_\varepsilon(q) \leq f(q) + \varepsilon$$

Heavy Hitters in Stream

ϕ-Heavy Hitter

$y \in [u]$ is a ϕ-heavy hitter iff $f(y) > \phi$.

Heavy Hitters in Stream

ϕ-Heavy Hitter

$y \in [u]$ is a ϕ-heavy hitter iff $f(y) > \phi$.

ε-approximation of ϕ-heavy hitters, $\hat{H}_\varepsilon^\phi$

- $y \in \hat{H}_\varepsilon^\phi$ if $f(y) > \phi$.
- $y \notin \hat{H}_\varepsilon^\phi$ if $f(y) < \phi - \varepsilon$.

Heavy Hitters in Stream

ϕ-Heavy Hitter

$y \in [u]$ is a ϕ-heavy hitter iff $f(y) > \phi$.

ε-approximation of ϕ-heavy hitters, $\hat{H}_\varepsilon^\phi$

- $y \in \hat{H}_\varepsilon^\phi$ if $f(y) > \phi$.
- $y \notin \hat{H}_\varepsilon^\phi$ if $f(y) < \phi - \varepsilon$.

- Given f, you can find all ϕ-Heavy Hitters.

Heavy Hitters in Stream

ϕ-Heavy Hitter

$y \in [u]$ is a ϕ-heavy hitter iff $f(y) > \phi$.

ε-approximation of ϕ-heavy hitters, $\hat{H}^{\phi}_{\varepsilon}$

- $y \in \hat{H}^{\phi}_{\varepsilon}$ if $f(y) > \phi$.
- $y \notin \hat{H}^{\phi}_{\varepsilon}$ if $f(y) < \phi - \varepsilon$.

- Given f, you can find all ϕ-Heavy Hitters.
- Given a $\hat{f}_{\varepsilon}$, for any $\phi \geq \varepsilon$, $\{y \in [u] \mid \hat{f}_{\varepsilon}(y) > \phi - \varepsilon\}$ is a $\hat{H}^{\phi}_{\varepsilon}$.

Table of Contents

Majority

Goal

Find y if $f(y) > \frac{1}{2}$.

Algorithm: Majority(X)

$y \leftarrow \text{NaN}, c \leftarrow 0$
forall $x \in X$ **do**
  if $y = x$ **then** $c \leftarrow c + 1$
  else if $c = 0$ **then** $y \leftarrow x, c \leftarrow 1$
  else $c \leftarrow c - 1$
return y

Boyer and Moore [1981].

Majority

Goal

Find y if $f(y) > \frac{1}{2}$.

Algorithm: Majority(X)

$y \leftarrow \mathrm{NaN}, c \leftarrow 0$
forall $x \in X$ **do**
  if $y = x$ **then** $c \leftarrow c + 1$
  else if $c = 0$ **then** $y \leftarrow x, c \leftarrow 1$
  else $c \leftarrow c - 1$
return y

Boyer and Moore [1981].

- If there is no m s.t. $f(m) > \frac{1}{2}$, then whatever y is correct.

Majority

Goal

Find y if $f(y) > \frac{1}{2}$.

Algorithm: Majority(X)

$y \leftarrow \text{NaN}, c \leftarrow 0$
forall $x \in X$ **do**
  if $y = x$ **then** $c \leftarrow c + 1$
  else if $c = 0$ **then** $y \leftarrow x, c \leftarrow 1$
  else $c \leftarrow c - 1$
return y

Boyer and Moore [1981].

- If there is no m s.t. $f(m) > \frac{1}{2}$, then whatever y is correct.
- Assume $f(m) > \frac{1}{2}$. Whenever c reaches 0:
  - The algorithm goes back to inital state and starts to process the rest of the sequence.
  - m must be the majority of the rest sequence as well.

Misra-Gries Sketch

Algorithm: Majority(X)

$y \leftarrow \text{NaN}, c \leftarrow 0$
forall $x \in X$ **do**
  if $y = x$ **then** $c \leftarrow c + 1$
  else if $c = 0$ **then** $y \leftarrow x, c \leftarrow 1$
  else
    $c \leftarrow c - 1$
return y

Algorithm: Misra-Gries(X, k)

$Y \leftarrow [\text{NaN}] * k, C \leftarrow [0] * k$
forall $x \in X$ **do**
  if $\exists i(Y[i] = x)$ **then** $C[i] \leftarrow C[i] + 1$
  else if $\exists i(C[i] = 0)$ **then** $Y[i] \leftarrow x, C[i] \leftarrow C[i] + 1$
  else
    forall i **do** $C[i] \leftarrow C[i] - 1$
return Y, C

Misra and Gries [1982].

Extends the majority algorithm by increasing the number of keys and coun- ters from 1 to k.

Misra-Gries Sketch

Algorithm: Majority(X)

```
y ← NaN, c ← 0
forall x ∈ X do
    if y = x then c ← c + 1
    else if c = 0 then y ← x, c ← 1
    else
        c ← c − 1
return y
```

Algorithm: Misra-Gries(X, k)

```
Y ← [NaN] ∗ k, C ← [0] ∗ k
forall x ∈ X do
    if ∃i(Y[i] = x) then C[i] ← C[i] + 1
    else if ∃i(C[i] = 0) then Y[i] ← x, C[i] ← C[i] + 1
    else
        forall i do C[i] ← C[i] − 1
return Y, C
```

Misra and Gries [1982].

Extends the majority algorithm by increasing the number of keys and coun- ters from 1 to k.

To approximate $f(q)$ for any $q \in [u]$

$$\hat{f}_{MG}(q) := \begin{cases} \frac{C[i]}{n} & \exists i(Y[i] = q) \\ 0 & \text{otherwise} \end{cases}$$

Misra-Gries Sketch Analysis

Lemma

for all $q \in [u]$,

$$f(q) - \frac{1}{k+1} \leq \hat{f}_{MG(k)}(q) \leq f(q)$$

Misra-Gries Sketch Analysis

Lemma

for all $q \in [u]$,

$$f(q) - \frac{1}{k+1} \leq \hat{f}_{MG(k)}(q) \leq f(q)$$

Proof.

The upper bound is obvious.
When Line 6 executes, there must be $k+1$ distinct item are decremented. It can hapen at most $n/(k+1)$ times. $\square$

Misra-Gries Sketch Analysis

Lemma

for all $q \in [u]$,

$$f(q) - \frac{1}{k+1} \leq \hat{f}_{MG(k)}(q) \leq f(q)$$

Setting $\frac{1}{k+1} = \varepsilon$, or $k = \frac{1}{\varepsilon} - 1$,

$$f(q) - \varepsilon \leq \hat{f}_{MG(k)}(q) \leq f(q) \leq f(q) + \varepsilon$$

Misra-Gries Sketch Analysis

Lemma

for all $q \in [u]$,

$$f(q) - \frac{1}{k+1} \leq \hat{f}_{MG(k)}(q) \leq f(q)$$

Setting $\frac{1}{k+1} = \varepsilon$, or $k = \frac{1}{\varepsilon} - 1$,

$$f(q) - \varepsilon \leq \hat{f}_{MG(k)}(q) \leq f(q) \leq f(q) + \varepsilon$$

- $f_{MG(k)}(q)$ is an ε-approximation of $f(q)$.

Misra-Gries Sketch Analysis

Lemma

for all $q \in [u]$,

$$f(q) - \frac{1}{k+1} \leq \hat{f}_{MG(k)}(q) \leq f(q)$$

Setting $\frac{1}{k+1} = \varepsilon$, or $k = \frac{1}{\varepsilon} - 1$,

$$f(q) - \varepsilon \leq \hat{f}_{MG(k)}(q) \leq f(q) \leq f(q) + \varepsilon$$

- $f_{MG(k)}(q)$ is an ε-approximation of $f(q)$.
- Y is a $\hat{H}_{\varepsilon}^{\varepsilon}$

Table of Contents

Count Sketch Algorithm

Algorithm: Count-Sketch(X, t, k)

$C \leftarrow 0^{t\times k}, H \leftarrow (H_i : [u] \rightarrow [k])_{i=1}^{t}, S \leftarrow (S_i : [u] \rightarrow [\pm 1])_{i=1}^{t}$

forall $x \in X$ **do**

  forall i *in* $[t]$ **do**

    $C_{i,H_i(x)} \leftarrow C_{i,H_i(x)} + S_i(x)$

return C, H, S

Charikar et al. [2002].

Count Sketch Algorithm

Algorithm: Count-Sketch(X, t, k)

$C \leftarrow 0^{t\times k}, H \leftarrow (H_i : [u] \rightarrow [k])_{i=1}^{t}, S \leftarrow (S_i : [u] \rightarrow [\pm 1])_{i=1}^{t}$
forall $x \in X$ **do**
    forall i *in* $[t]$ **do**
        $C_{i,H_i(x)} \leftarrow C_{i,H_i(x)} + S_i(x)$
return C, H, S

Charikar et al. [2002].

- H_i, S_i are independent hash functions.

Count Sketch Algorithm

Algorithm: Count-Sketch(X, t, k)

$C \leftarrow 0^{t\times k}, H \leftarrow (H_i : [u] \rightarrow [k])_{i=1}^{t}, S \leftarrow (S_i : [u] \rightarrow [\pm 1])_{i=1}^{t}$
forall $x \in X$ **do**
    forall i *in* $[t]$ **do**
        $C_{i,H_i(x)} \leftarrow C_{i,H_i(x)} + S_i(x)$
return C, H, S

Charikar et al. [2002].

- H_i, S_i are independent hash functions.
- S_i are choosen from a **pairwise** independent family.

Count Sketch Algorithm

Algorithm: Count-Sketch(X, t, k)

$C \leftarrow 0^{t\times k}, H \leftarrow (H_i : [u] \to [k])_{i=1}^{t}, S \leftarrow (S_i : [u] \to [\pm 1])_{i=1}^{t}$
forall $x \in X$ **do**
  forall i *in* $[t]$ **do**
    $C_{i,H_i(x)} \leftarrow C_{i,H_i(x)} + S_i(x)$
return C, H, S

$$H = \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_t \end{bmatrix} \quad S = \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_t \end{bmatrix} \quad C = \begin{bmatrix} C_{1,1} & C_{1,2} & \dots & C_{1,k} \\ C_{2,1} & C_{2,2} & \dots & C_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ C_{t,1} & C_{t,2} & \dots & C_{t,k} \end{bmatrix}$$

Count Sketch Query

To approximate $f(q)$ for any $q \in [u]$

$$\hat{f}_{CS}(q) := \operatorname*{median}_{i \in [t]} \hat{f}_i(q), \quad \text{where } \hat{f}_i(q) := \frac{1}{n} S_i(q) C_{i,H_i(q)}.$$

$$H = \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_t \end{bmatrix} \quad S = \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_t \end{bmatrix} \quad C = \begin{bmatrix} C_{1,1} & C_{1,2} & \dots & C_{1,k} \\ C_{2,1} & C_{2,2} & \dots & C_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ C_{t,1} & C_{t,2} & \dots & C_{t,k} \end{bmatrix}$$

Count Sketch Randomness

Since the algorithm is not deterministic, it is randomized. We will analize it in a probabilistic way.

Count Sketch Randomness

Since the algorithm is not deterministic, it is randomized. We will analize it in a probabilistic way.

Question

Where is the randomness come from? Or what are the random variables?

Count Sketch Randomness

Since the algorithm is not deterministic, it is randomized. We will analize it in a probabilistic way.

Question

Where is the randomness come from? Or what are the random variables?

Answer

The random events are the choices of hash functions in H and S.

Count Sketch Randomness

Since the algorithm is not deterministic, it is randomized. We will analize it in a probabilistic way.

Question

Where is the randomness come from? Or what are the random variables?

Answer

The random events are the choices of hash functions in H and S. The random varialbes are H_i and S_i, or $H_i(q)$ and $S_i(q)$ for all $q \in [u]$.

Count Sketch Notations

$$
\begin{aligned}
C_{i,j} &:= \sum_{x \in X} S_i(x)\mathbb{1}(H_i(x), j) \\
&= \sum_{x \in [u]} nf(x)S_i(x)\mathbb{1}(H_i(x), j)
\end{aligned}
$$

Count Sketch Notations

$$\begin{aligned} C_{i,j} &:= \sum_{x \in X} S_i(x) \mathbb{1}(H_i(x), j) \\ &= \sum_{x \in [u]} nf(x) S_i(x) \mathbb{1}(H_i(x), j) \end{aligned}$$

- $\mathbb{1}(i, j)$: equal to 1 if $i = j$ and 0 otherwise.

Count Sketch Notations

$$
\begin{aligned}
C_{i,j} &:= \sum_{x \in X} S_i(x) \mathbb{1}(H_i(x), j) \\
&= \sum_{x \in [u]} nf(x) S_i(x) \mathbb{1}(H_i(x), j)
\end{aligned}
$$

- $\mathbb{1}(i,j)$: equal to 1 if $i = j$ and 0 otherwise.
- $C_{i,j}^x := nf(x) S_i(x) \mathbb{1}(H_i(x), j)$: the part of $C_{i,j}$ caused by $x \in [u]$.

Count Sketch Notations

$$
\begin{aligned}
C_{i,j} &:= \sum_{x \in X} S_i(x) \mathbb{1}(H_i(x), j) \\
&= \sum_{x \in [u]} nf(x) S_i(x) \mathbb{1}(H_i(x), j)
\end{aligned}
$$

- $\mathbb{1}(i,j)$: equal to 1 if $i = j$ and 0 otherwise.
- $C_{i,j}^x := nf(x)S_i(x)\mathbb{1}(H_i(x), j)$: the part of $C_{i,j}$ caused by $x \in [u]$.

With these notations, we can simply write each $C_{i,j}$ as

$$
C_{i,j} = \sum_{x \in [u]} C_{i,j}^x
$$

Count Sketch Analysis - Mean

Lemma

For any $i \in [t], q \in [u]$,

$$\mathrm{E}[\hat{f}_i(q)] = f(q)$$

Count Sketch Analysis - Mean

Lemma

For any $i \in [t], q \in [u]$,

$$\mathrm{E}[\hat{f}_i(q)] = f(q)$$

$$\hat{f}_i(q) := \frac{1}{n} S_i(q) C_{i,H_i(q)} = f(q) + \frac{1}{n} \sum_{x \in [u], x \neq q} S_i(q) C^x_{i,H_i(q)}$$

Count Sketch Analysis - Mean

Lemma

For any $i \in [t], q \in [u]$,

$$\mathrm{E}[\hat{f}_i(q)] = f(q)$$

$$\hat{f}_i(q) := \frac{1}{n} S_i(q) C_{i,H_i(q)} = f(q) + \frac{1}{n} \sum_{x \in [u], x \neq q} S_i(q) C^x_{i,H_i(q)}$$

$$\begin{aligned}
& \mathrm{E}\left[S_i(q) C^x_{i,H_i(q)}\right] \\
= \; & nf(x)\, \mathrm{E}\left[S_i(q) S_i(x) \mathbb{1}(H_i(x), H_i(q))\right] \\
= \; & nf(x)\, \mathrm{E}\left[S_i(q) S_i(x)\right] \mathrm{E}\left[\mathbb{1}(H_i(x), H_i(q))\right] && S_i \text{ and } H_i \text{ are indep.} \\
= \; & nf(x)\, \mathrm{E}\left[S_i(q)\right] \mathrm{E}\left[S_i(x)\right] \mathrm{E}\left[\mathbb{1}(H_i(x), H_i(q))\right] && S_i \text{ is pairwise indep.} \\
= \; & 0 && \mathrm{E}\left[S_i(x)\right] = 0
\end{aligned}$$

Count Sketch Analysis - Variance

Lemma

For any $i \in [t], q \in [u]$,

$$\mathrm{V}[\hat{f}_i(q)] \leq \frac{1}{k} F_2^2$$

where $F_2^2 = \sum_{x \in [u]} f(x)^2$.

Count Sketch Analysis - Variance

Lemma

For any $i \in [t], q \in [u]$,

$$\mathrm{V}[\hat{f}_i(q)] \leq \frac{1}{k} F_2^2$$

where $F_2^2 = \sum_{x \in [u]} f(x)^2$.

$$\mathrm{V}[\hat{f}_i(q)] = \mathrm{V}\left[\frac{1}{n} S_i(q) C_{i,H_i(q)}\right] = \frac{1}{n^2} \mathrm{V}\left[\sum_{x \in [u]} S_i(q) C^x_{i,H_i(q)}\right]$$

Count Sketch Analysis - Variance - 2

$$\mathrm{V}[\hat{f}_i(q)] = \frac{1}{n^2}\mathrm{V}\left[\sum_{x\in[u]} S_i(q)C^x_{i,H_i(q)}\right] = \frac{1}{n^2}\sum_{x\in[u]}\mathrm{V}\left[S_i(q)C^x_{i,H_i(q)}\right]$$

Because

$$\begin{aligned}
&\mathrm{cov}\left[S_i(q)C^x_{i,H_i(q)}, S_i(q)C^y_{i,H_i(q)}\right] \\
= \ &\mathrm{E}\left[\left(S_i(q)C^x_{i,H_i(q)} - \mathrm{E}[S_i(q)C^x_{i,H_i(q)}]\right)\left(S_i(q)C^y_{i,H_i(q)} - \mathrm{E}[S_i(q)C^y_{i,H_i(q)}]\right)\right] \\
= \ &\mathrm{E}\left[\left(S_i(q)C^x_{i,H_i(q)}\right)\left(S_i(q)C^y_{i,H_i(q)}\right)\right] = 0
\end{aligned}$$

for all $x \neq y$, if S_i and H_i are indep., S_i is pairwise indep.,

Count Sketch Analysis - Variance - 3

$$
\begin{aligned}
\mathrm{V}[\hat{f}_i(q)] &= \frac{1}{n^2} \sum_{x \in [u]} \mathrm{V}\left[S_i(q) C^x_{i,H_i(q)} \right] \\
&\leq \frac{1}{n^2} \sum_{x \in [u]} \mathrm{E}\left[\left(S_i(q) C^x_{i,H_i(q)} \right)^2 \right] \\
&= \frac{1}{n^2} \sum_{x \in [u]} \mathrm{E}\left[\left(n f(x) S_i(x) \mathbb{1}(H_i(x), H_i(q)) \right)^2 \right] \\
&= \sum_{x \in [u]} f^2(x)\, \mathrm{E}\left[\left(\mathbb{1}(H_i(x), H_i(q)) \right)^2 \right] \\
&= \sum_{x \in [u]} f^2(x) \frac{1}{k} = \frac{1}{k} F_2^2
\end{aligned}
$$

Count Sketch Analysis - Failure Probability

Lemma

For any $q \in [u], i \in [t]$,

$$\Pr\left[\left|\hat{f}_i(q) - f(q)\right| \geq \varepsilon\right] \leq \frac{F_2^2}{k\varepsilon^2}$$

Count Sketch Analysis - Failure Probability

Lemma

For any $q \in [u], i \in [t]$,

$$\Pr\left[\left|\hat{f}_i(q) - f(q)\right| \geq \varepsilon\right] \leq \frac{F_2^2}{k\varepsilon^2}$$

The Chebyshev's inequality: $\Pr[|R - \mathrm{E}[R]| \geq \varepsilon] \leq \dfrac{\mathrm{V}[R]}{\varepsilon^2}$

Count Sketch Analysis - Failure Probability

Lemma

For any $q \in [u], i \in [t]$,

$$\Pr\left[\left|\hat{f}_i(q) - f(q)\right| \geq \varepsilon\right] \leq \frac{F_2^2}{k\varepsilon^2}$$

The Chebyshev's inequality: $\Pr\left[|R - \mathrm{E}[R]| \geq \varepsilon\right] \leq \dfrac{\mathrm{V}[R]}{\varepsilon^2}$

$$\Pr\left[\left|\hat{f}_i(q) - f(q)\right| \geq \varepsilon\right] = \Pr\left[\left|\hat{f}_i(q) - \mathrm{E}[\hat{f}_i(q)]\right| \geq \varepsilon\right] \leq \frac{\mathrm{V}[\hat{f}_i(q)]}{\varepsilon^2} \leq \frac{F_2^2}{k\varepsilon^2}$$

Count Sketch Analysis - Confidence Boosting

Now we know $\Pr\left[\left|\hat{f}_i(q) - f(q)\right| \geq \varepsilon\right] \leq \frac{F_2^2}{k\varepsilon^2}$ for each $i \in [t]$.

Count Sketch Analysis - Confidence Boosting

Now we know $\Pr\left[\left|\hat{f}_i(q) - f(q)\right| \geq \varepsilon\right] \leq \frac{F_2^2}{k\varepsilon^2}$ for each $i \in [t]$.
At the end, we will return

$$\hat{f}_{CS}(q) := \operatorname*{median}_{i \in [t]} \hat{f}_i(q)$$

Count Sketch Analysis - Confidence Boosting

Now we know $\Pr\left[\left|\hat{f}_i(q) - f(q)\right| \geq \varepsilon\right] \leq \frac{F_2^2}{k\varepsilon^2}$ for each $i \in [t]$.
At the end, we will return

$$\hat{f}_{CS}(q) := \operatorname*{median}_{i \in [t]} \hat{f}_i(q)$$

Whay the median?

Count Sketch Analysis - Confidence Boosting

Now we know $\Pr\left[\left|\hat{f}_i(q) - f(q)\right| \geq \varepsilon\right] \leq \frac{F_2^2}{k\varepsilon^2}$ for each $i \in [t]$.
At the end, we will return

$$\hat{f}_{CS}(q) := \underset{i \in [t]}{\text{median}}\ \hat{f}_i(q)$$

Whay the median?
If the median has error $\geq \varepsilon$, then at least half of the $\hat{f}_i(q)$ have error $\geq \varepsilon$.

Count Sketch Analysis - Confidence Boosting 2

Let the failure probability of each $\hat{f}_i(q)$ is $p = \frac{F_2^2}{k\varepsilon^2}$.
Repeat it t times independently, what is the probability of at least half failures?

Count Sketch Analysis - Confidence Boosting 2

Let the failure probability of each $\hat{f}_i(q)$ is $p = \frac{F_2^2}{k\varepsilon^2}$.
Repeat it t times independently, what is the probability of at least half failures?
Binomial distribution! The number of failure is $B(t, p)$.

Count Sketch Analysis - Confidence Boosting 2

Let the failure probability of each $\hat{f}_i(q)$ is $p = \frac{F_2^2}{k\varepsilon^2}$.
Repeat it t times independently, what is the probability of at least half failures?
Binomial distribution! The number of failure is $B(t, p)$.
Chernoff bound:

$$\Pr\left[B(t,p) \geq \frac{t}{2}\right] \leq \exp\big(-t(1/2-p)^2/(2p)\big)$$

Count Sketch Analysis - Confidence Boosting 3

Set $p = \frac{F_2^2}{k\varepsilon^2} = \frac{1}{4}$, or $k = \frac{F_2^2}{4\varepsilon^2}$:

$$\Pr\left[B(t,p) \geq \frac{t}{2}\right] \leq \exp\Big(-t(1/2-p)^2/(2p)\Big) \leq \exp(-t/8)$$

Count Sketch Analysis - Confidence Boosting 3

Set $p = \frac{F_2^2}{k\varepsilon^2} = \frac{1}{4}$, or $k = \frac{F_2^2}{4\varepsilon^2}$:

$$\Pr\left[B(t,p) \geq \frac{t}{2}\right] \leq \exp\left(-t(1/2-p)^2/(2p)\right) \leq \exp(-t/8)$$

Set $\exp(-t/8) = \delta$, or $t = 8\log\frac{1}{\delta}$:

$$\Pr\left[\left|\hat{f}_{CS}(q) - f(q)\right| \geq \varepsilon\right] \leq \Pr[B(t,1/4) \geq t/2] \leq \delta$$

Count Sketch Analysis - Confidence Boosting 3

Set $p = \frac{F_2^2}{k\varepsilon^2} = \frac{1}{4}$, or $k = \frac{F_2^2}{4\varepsilon^2}$:

$$\Pr\left[B(t,p) \geq \frac{t}{2}\right] \leq \exp\left(-t(1/2-p)^2/(2p)\right) \leq \exp(-t/8)$$

Set $\exp(-t/8) = \delta$, or $t = 8\log\frac{1}{\delta}$:

$$\Pr\left[\left|\hat{f}_{CS}(q) - f(q)\right| \geq \varepsilon\right] \leq \Pr[B(t,1/4) \geq t/2] \leq \delta$$

Theorem

If $k = \frac{F_2^2}{4\varepsilon^2}$ and $t = 8\log\frac{1}{\delta}$, $\hat{f}_{CS}(q)$ is an $\hat{f}_{\varepsilon}(q)$ with probability at least $1-\delta$ for any $q \in [u]$.

Table of Contents

Count-Min Sketch - Algorithm

Algorithm: Count-Min(X, t, k)

$C \leftarrow 0^{t\times k}$, $H \leftarrow (H_i : [u] \rightarrow [k])_{i=1}^{t}$
forall $x \in X$ **do**
  forall i *in* $[t]$ **do**
    $C_{i,H_i(x)} \leftarrow C_{i,H_i(x)} + 1$
return C, H

Cormode and Muthukrishnan [2005]

To approximate $f(q)$ for any $q \in [u]$

$$\hat{f}_{CMS}(q) := \min_{i\in[t]} \hat{f}_i(q), \quad \text{where } \hat{f}_i(q) := \frac{1}{n} C_{i,H_i(q)}.$$

Count-Min Sketch - Bounds

Lemma

For any $q \in [u], i \in [t]$,

$$f(q) \leq \hat{f}_i(q).$$

If H_i *is drawn from a pairwise independent hash family, then*

$$\mathrm{E}[\hat{f}_i(q) - f(q)] \leq \frac{1}{k}.$$

If $k = \frac{1}{\delta\varepsilon}$, *then*

$$\Pr[\hat{f}_i(q) - f(q) \geq \varepsilon] \leq \delta$$

Count-Min Sketch - Confidence Boosting

Lemma

For any $q \in [u], i \in [t]$, if H_i is drawn from a pairwise independent hash family, and $k = \frac{2}{\varepsilon}$, then $\hat{f}_i(q)$ is a $\hat{f}_\varepsilon(q)$ with probability at least $1/2$.

Theorem

If $t = \lg \frac{1}{\delta}, k = \frac{2}{\varepsilon}$, H_is are independently drawn from a pairwise independent hash family, then for any $q \in [u]$, $\hat{f}_{CMS}(q) := \min_{i \in [t]} \hat{f}_i(q)$ is a $\hat{f}_\varepsilon(q)$ with probability at least $1 - \delta$.

Table of Contents

Summary

Sketch	Space	Technique	Deterministic
Misra-Gries	$O(1/\varepsilon)$	Counter	Yes
Count Sketch	$O\left(\frac{F_2^2}{\varepsilon^2}\log\frac{1}{\delta}\right)$	Hashing	No
Count-Min Sketch	$O\left(\frac{1}{\varepsilon}\log\frac{1}{\delta}\right)$	Hashing	No

Table: Studied Sketches to obtain $\hat{f}_\varepsilon$ (with probability at least $1-\delta$ if aplicable).

Summary

Sketch	Space	Technique	Deterministic
Misra-Gries	$O(1/\varepsilon)$	Counter	Yes
Count Sketch	$O\left(\frac{F_2^2}{\varepsilon^2}\log\frac{1}{\delta}\right)$	Hashing	No
Count-Min Sketch	$O\left(\frac{1}{\varepsilon}\log\frac{1}{\delta}\right)$	Hashing	No

Table: Studied Sketches to obtain $\hat{f}_\varepsilon$ (with probability at least $1-\delta$ if aplicable).

It seems like the Count-Min sketch is better than the Count sketch in the error-space tradeoff, but the bound is based on F_2^2 which is usually much smaller than 1. The Count sketch is also more versatile than Count-Min sketch and works very well in practice.

Biased vs Unbiased

Definition (biased, unbiased, under-estimated, over-estimated approximation)

To estimate a ground truth value f, a random variable $\hat{f}$ (the output of any estimation method) is

- unbiased approximation if $E[\hat{f}] = f$;
- biased approximation if $E[\hat{f}] \neq f$;
- under-estimated approximation if $E[\hat{f}] < f$;
- over-estimated approximation if $E[\hat{f}] > f$;

question

Does unbiased approximation always better than biased approximation?

Questions

question

Can Count/Count-Min sketch solve heavy hitters? What is the query time?

Questions

question

Can Count/Count-Min sketch solve heavy hitters? What is the query time?

question

What is the failure probability of Count/Count-Min sketch actually is? For one q or for all $q \in [u]$?

Questions

question

Can Count/Count-Min sketch solve heavy hitters? What is the query time?

question

What is the failure probability of Count/Count-Min sketch actually is? For one q or for all $q \in [u]$?

question

What about weighted data? Real value weights? Negative weights?