Advanced Algorithms

Hashing and Sketching

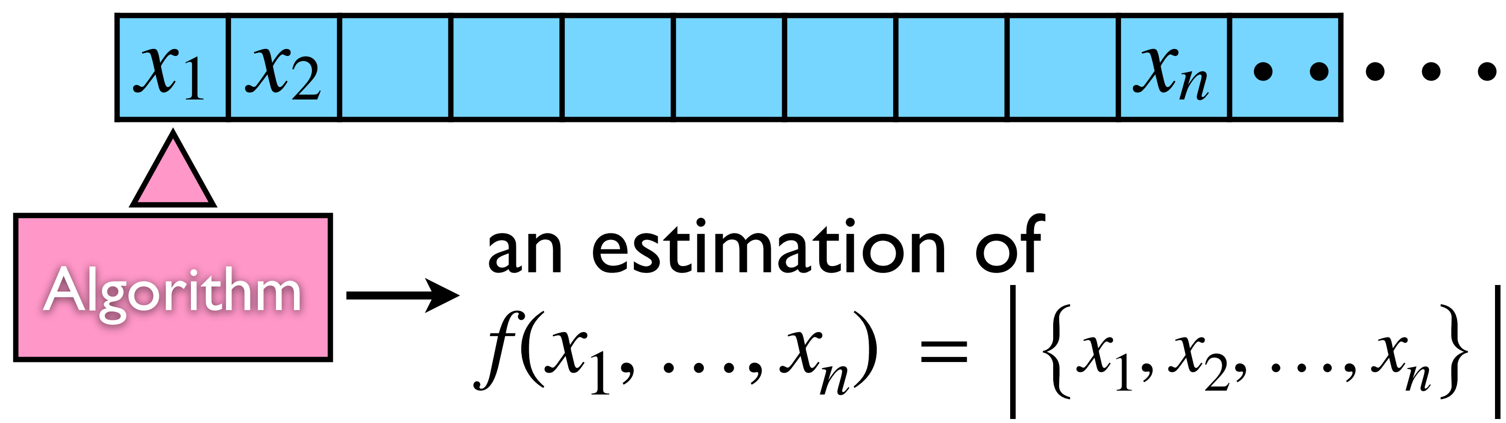
尹一通 Nanjing University, 2022 Fall

Count Distinct Elements

Input: a sequence $x_1, x_2, \ldots, x_n \in U = [N]$

Output: an estimation of $z = \left| \{x_1, x_2, \ldots, x_n\} \right|$

- **Data stream model**: input data item comes one at a time



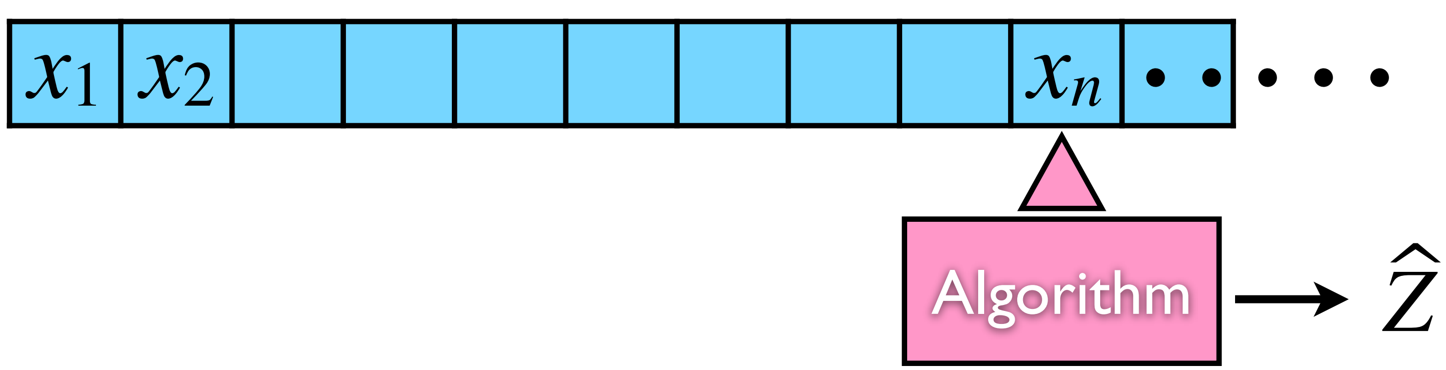
- Naïve algorithm: store all distinct data items using $\Omega(z \log N)$ bits
- **Sketch**: (lossy) representation of data using space $\ll z$
- **Lower bound** (Alon-Matias-Szegedy): any deterministic (exact or approx.) algorithm must use $\Omega(N)$ bits of space in the worst case

Count Distinct Elements

Input: a sequence $x_1, x_2, \ldots, x_n \in U = [N]$

Output: an estimation of $z = \left| \{x_1, x_2, \ldots, x_n\} \right|$

- **Data stream model**: input data item comes one at a time



- **(ϵ, δ)-estimator**: randomized variable $\hat{Z}$

$$\Pr\left[(1-\epsilon)z \le \hat{Z} \le (1+\epsilon)z\right] \ge 1-\delta$$

Using only memory equivalent to 5 lines of printed text, you can estimate with a typical accuracy of 5% and in a single pass the total vocabulary of Shakespeare.

——Durand and Flajolet 2003

Input: a sequence $x_1, x_2, \ldots, x_n \in U = [N]$

Output: an estimation of $z = \left| \{x_1, x_2, \ldots, x_n\} \right|$

Simple Uniform Hash Assumption (SUHA):

A uniform function is available, whose preprocessing, representation and evaluation are considered to be easy.

- (*idealized*) uniform hash function $h : U \to [0,1]$
  - $x_i = x_j \longrightarrow$ the same hash value $h(x_i) = h(x_j) \in_r [0,1]$
- $\{h(x_1), \ldots, h(x_n)\}$: $z \times$ uniform and independent values in $[0,1]$
  - partition $[0,1]$ into $z+1$ subintervals (with *identically distributed* lengths)

$$\mathbb{E}\left[\min_{1 \le i \le n} h(x_i)\right] = \Pr[\text{length of a subinterval}] = \frac{1}{z+1} \quad \text{(by symmetry)}$$

- estimator: $\widehat{Z} = \dfrac{1}{\min_i h(x_i)} - 1$? Variance is too large!

Markov's Inequality

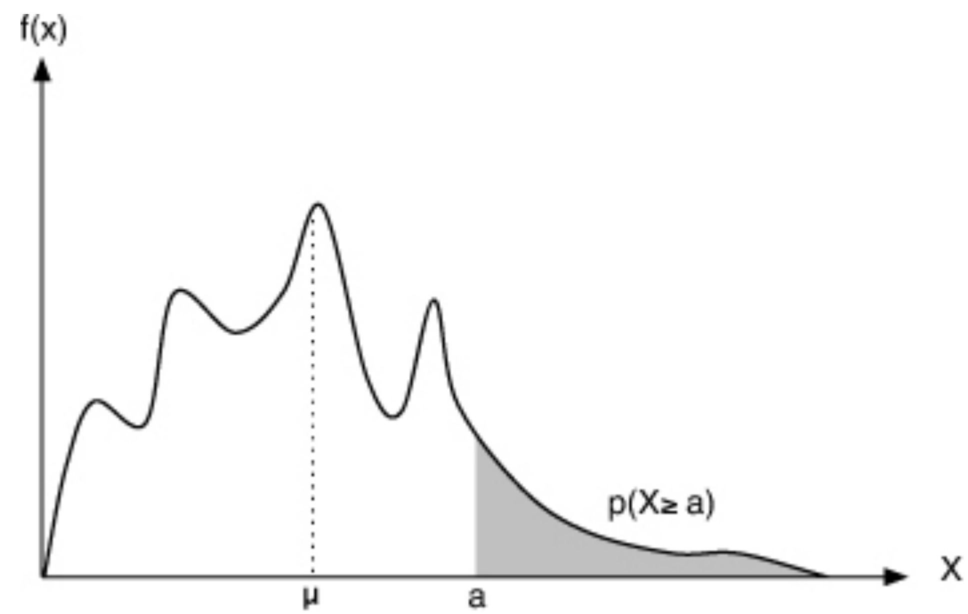
Markov's Inequality

For *nonnegative* random variable X, for any $t > 0$,

$$\Pr[X \geq t] \leq \frac{\mathbb{E}[X]}{t}$$

Let $Y = \begin{cases} 1 & X \geq t \\ 0 & \text{o.w.} \end{cases} \implies Y \leq \left\lfloor \frac{X}{t} \right\rfloor \leq \frac{X}{t}$

$$\Pr[X \geq t] = \mathbb{E}[Y] \leq \mathbb{E}\left[\frac{X}{t}\right] = \frac{\mathbb{E}[X]}{t}$$



tight if only knowing the expectation of X

Markov's Inequality

Markov's Inequality

For *nonnegative* random variable X, for any $t > 0$,

$$\Pr[X \geq t] \leq \frac{\mathbb{E}[X]}{t}$$

Corollary

For random variable X and *nonnegative-valued* function f, for any $t > 0$,

$$\Pr[f(X) \geq t] \leq \frac{\mathbb{E}[f(X)]}{t}$$

Chebyshev's Inequality

Chebyshev's Inequality

For random variable X, for any $t > 0$,

$$\Pr\left[\,|X - \mathbb{E}[X]| \geq t\,\right] \leq \frac{\mathbf{Var}[X]}{t^2}$$

- Variance:

$$\mathbf{Var}[X] = \mathbb{E}[(X - \mathbb{E}[X])^2] = \mathbb{E}[X^2] - (\mathbb{E}[X])^2$$

Apply Markov's inequality to $Y = (X - \mathbb{E}[X])^2$:

$$\Pr\left[\,|X - \mathbb{E}[X]| \geq t\right] = \Pr[Y \geq t^2] \leq \frac{\mathbb{E}\left[Y\right]}{t^2} \leq \frac{\mathbf{Var}[X]}{t^2}$$

Input: a sequence $x_1, x_2, \ldots, x_n \in U = [N]$

Output: an estimation of $z = \left| \{x_1, x_2, \ldots, x_n\} \right|$

- (*idealized*) uniform hash function $h : U \to [0,1]$

Min Sketch:

let $Y = \min_{1 \le i \le n} h(x_i)$;

return $\widehat{Z} = \frac{1}{Y} - 1$;

- By symmetry:

$$\mathbb{E}[Y] = \frac{1}{n+1}$$

- Goal:

$$\Pr\left[\widehat{Z} < (1-\epsilon)z \text{ or } \widehat{Z} > (1+\epsilon)z\right] \le \delta$$

assuming $\epsilon \le 1/2$

$$\left|Y - \mathbb{E}[Y]\right| > \frac{\epsilon/2}{z+1} \iff \left|Y - \frac{1}{z+1}\right| > \frac{\epsilon/2}{z+1}$$

Input: a sequence $x_1, x_2, \ldots, x_n \in U = [N]$

Output: an estimation of $z = \left| \{x_1, x_2, \ldots, x_n\} \right|$

- (*idealized*) uniform hash function $h : U \to [0,1]$

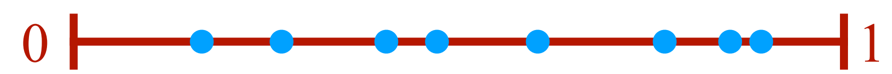
Min Sketch:

let $Y = \min_{1 \le i \le n} h(x_i)$;

return $\widehat{Z} = \frac{1}{Y} - 1$;

- Uniform independent hash values:

$$H_1, \ldots, H_z \in [0,1]$$



- $Y = \min_{1 \le i \le z} H_i$

geometry probability: $\Pr[Y > y] = (1-y)^z$ ⇒ pdf: $p(y) = z(1-y)^{z-1}$

$$\mathbb{E}[Y^2] = \int_0^1 y^2 p(y)\, \mathrm{d}y = \int_0^1 y^2 z(1-y)^{z-1}\, \mathrm{d}y = \frac{2}{(z+1)(z+2)}$$

$$\mathbf{Var}[Y] = \mathbb{E}[Y^2] - \mathbb{E}[Y]^2 = \frac{z}{(z+1)^2(z+2)} \le \frac{1}{(z+1)^2}$$

Input: a sequence $x_1, x_2, \ldots, x_n \in U = [N]$

Output: an estimation of $z = \left| \{x_1, x_2, \ldots, x_n\} \right|$

- (*idealized*) uniform hash function $h : U \to [0,1]$

Min Sketch:

let $Y = \min_{1 \le i \le n} h(x_i)$;

return $\widehat{Z} = \frac{1}{Y} - 1$;

- By symmetry:

$$\mathbb{E}[Y] = \frac{1}{z+1}$$

- Goal:

$$\Pr\left[\widehat{Z} < (1-\epsilon)z \text{ or } \widehat{Z} > (1+\epsilon)z\right] \le \delta$$

assuming $\epsilon \le 1/2$

$$\mathbf{Var}[Y] \le \frac{1}{(z+1)^2}$$

(Chebyshev)

$$\Pr\left[\left|Y - \mathbb{E}[Y]\right| > \frac{\epsilon/2}{z+1}\right] \le \frac{4}{\epsilon^2}$$

The Mean Trick (for Variance Reduction)

- Variance and covariance:

$$\mathbf{Var}[X] = \mathbb{E}[(X - \mathbb{E}[X])^2] = \mathbb{E}[X^2] - (\mathbb{E}[X])^2$$

$$\mathbf{Cov}(X, Y) = \mathbb{E}\left[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])\right]$$

- Useful properties:

$$\mathbf{Var}[X + a] = \mathbf{Var}[X]$$

$$\mathbf{Var}[aX] = a^2\mathbf{Var}[X]$$

$$\mathbf{Var}\left[\sum_i X_i\right] = \sum_i \mathbf{Var}[X_i] + \sum_{i \neq j} \mathbf{Cov}(X_i, X_j)$$

- For pairwise independent identically distributed X_i's:

$$\mathbf{Var}\left[\frac{1}{k}\sum_{i=1}^{k} X_i\right] = \frac{1}{k^2}\sum_{i=1}^{k} \mathbf{Var}[X_i] = \frac{1}{k}\mathbf{Var}[X_1]$$

Input: a sequence $x_1, x_2, \ldots, x_n \in U = [N]$

Output: an estimation of $z = \left| \left\{ x_1, x_2, \ldots, x_n \right\} \right|$

- uniform & independent hash functions $h_1, \ldots, h_k : U \to [0,1]$

Min Sketch:

for each $1 \leq j \leq k$, let $Y_j = \min_{1 \leq i \leq n} h_j(x_i)$;

return $\widehat{Z} = \frac{1}{\overline{Y}} - 1$ where $\overline{Y} = \frac{1}{k} \sum_{j=1}^{k} Y_j$;

- For every $1 \leq j \leq k$:

$$\mathbb{E}\left[Y_j\right] = \frac{1}{z+1} \quad \xrightarrow{\text{linearity of expectation}} \quad \mathbb{E}\left[\overline{Y}\right] = \frac{1}{z+1}$$

$$\mathbf{Var}[Y_j] \leq \frac{1}{(z+1)^2} \quad \xrightarrow{\text{independence}} \quad \mathbf{Var}\left[\overline{Y}\right] \leq \frac{1}{k(z+1)^2}$$

Input: a sequence $x_1, x_2, \ldots, x_n \in U = [N]$

Output: an estimation of $z = \left| \{x_1, x_2, \ldots, x_n\} \right|$

- uniform & independent hash functions $h_1, \ldots, h_k : U \to [0,1]$

Min Sketch:

for each $1 \le j \le k$, let $Y_j = \min_{1 \le i \le n} h_j(x_i)$;

return $\widehat{Z} = \frac{1}{\overline{Y}} - 1$ where $\overline{Y} = \frac{1}{k} \sum_{j=1}^{k} Y_j$;

$$\mathbb{E}\left[\overline{Y}\right] = \frac{1}{z+1}$$

$$\mathbf{Var}\left[\overline{Y}\right] \le \frac{1}{k(z+1)^2}$$

- Goal: $\Pr\left[\widehat{Z} < (1-\epsilon)z \text{ or } \widehat{Z} > (1+\epsilon)z\right] \le \delta$

assuming $\epsilon \le 1/2$

$$\Pr\left[\left|\overline{Y} - \mathbb{E}\left[\overline{Y}\right]\right| > \frac{\epsilon/2}{z+1}\right] \le \frac{4}{k\epsilon^2} \le \delta$$

(Chebyshev)

Set $k = \left\lceil \frac{4}{\epsilon^2 \delta} \right\rceil$

Input: a sequence $x_1, x_2, \ldots, x_n \in U = [N]$

Output: an estimation of $z = \left| \{x_1, x_2, \ldots, x_n\} \right|$

- uniform & independent hash functions $h_1, \ldots, h_k : U \to [0,1]$

Min Sketch: set $k = \lceil 4/(\epsilon^2 \delta) \rceil$

for each $1 \le j \le k$, let $Y_j = \min_{1 \le i \le n} h_j(x_i)$;

return $\widehat{Z} = \frac{1}{\overline{Y}} - 1$ where $\overline{Y} = \frac{1}{k} \sum_{j=1}^{k} Y_j$;

$$\Pr\left[(1-\epsilon)z \le \widehat{Z} \le (1+\epsilon)z \right] \ge 1 - \delta$$

- **Space cost**: $k = O\left(\frac{1}{\epsilon^2 \delta}\right)$ ***real numbers*** in $[0,1]$
- Storing k ***idealized*** hash functions.

Universal Hashing

Universal Hash Family (Carter and Wegman 1979):

A family $\mathscr{H}$ of hash functions in $U \to [m]$ is **k-universal** if for any distinct $x_1, \ldots, x_k \in U$,

$$\Pr_{h \in \mathscr{H}} \left[h(x_1) = \cdots = h(x_k) \right] \le \frac{1}{m^{k-1}}.$$

Moreover, $\mathscr{H}$ is **strongly k-universal** (k-wise independent) if for any distinct $x_1, \ldots, x_k \in U$ and any $y_1, \ldots, y_k \in [m]$,

$$\Pr_{h \in \mathscr{H}} \left[\bigwedge_{i=1}^{k} h(x_i) = y_i \right] = \frac{1}{m^k}.$$

k-Universal Hash Family

hash functions $h : U \to [m]$

- Linear congruential hashing:
  - Represent $U \subseteq \mathbb{Z}_p$ for sufficiently large prime p
  - $h_{a,b}(x) = ((ax + b) \bmod p) \bmod m$
  - $\mathscr{H} = \left\{ h_{a,b} \mid a \in \mathbb{Z}_p \setminus \{0\}, b \in \mathbb{Z}_p \right\}$

Theorem:
The linear congruential family $\mathscr{H}$ is 2-wise independent.

- Degree-k polynomial in finite field with random coefficients
- Hashing between binary fields: $GF(2^w) \to GF(2^l)$

$$h_{a,b}(x) = \texttt{(a*x+b)>>(w-l)}$$

Flajolet-Martin Algorithm

Input: a sequence $x_1, x_2, \ldots, x_n \in [N] \subseteq [2^w]$

Output: an estimation of $z = \left| \{x_1, x_2, \ldots, x_n\} \right|$

- 2-wise independent hash function $h : [2^w] \to [2^w]$
- For $y \in [2^w]$, let $\mathrm{zeros}(y) = \max\{i : 2^i \mid y\}$ denote # of trailing 0's

Flajolet-Martin Algorithm:

let $R = \max_{1 \le i \le n} \mathrm{zeros}(h(x_i))$;

return $\widehat{Z} = 2^R$;

$$\Pr\left[\widehat{Z} < \frac{z}{C} \text{ or } \widehat{Z} > C \cdot z\right] \le \frac{3}{C}$$

Input: a sequence $x_1, x_2, \ldots, x_n \in [N] \subseteq [2^w]$

Output: an estimation of $z = \left| \{x_1, x_2, \ldots, x_n\} \right|$

- 2-wise independent hash function $h : [2^w] \to [2^w]$
- For $y \in [2^w]$, let $\text{zeros}(y) = \max\{i : 2^i \mid y\}$ denote # of trailing 0's

Flajolet-Martin Algorithm:

let $R = \max_{1 \le i \le n} \text{zeros}(h(x_i))$;

return $\widehat{Z} = 2^R$;

Let

$$Y_r = \sum_{x \in \{x_1, \ldots, x_n\}} I\left[\text{zeros}\left(h(x)\right) \ge r\right]$$

(linearity of expectation)

$$\mathbb{E}[Y_r] = \sum_{x \in \{x_1, \ldots, x_n\}} \Pr\left[\text{zeros}\left(h(x)\right) \ge r\right] = z2^{-r}$$

(pairwise independence)

$$\mathbf{Var}[Y_r] = \sum_{x \in \{x_1, \ldots, x_n\}} \mathbf{Var}\left[I[\text{zeros}\left(h(x)\right) \ge r]\right] = z2^{-r}(1 - 2^{-r}) \le z2^{-r}$$

Pairwise Independent Trials

Proposition:

If X is a sum of pairwise independent random variables taking values in $\{0,1\}$, then $\mathbf{Var}[X] \leq \mathbb{E}[X]$.

$$\mathbf{Var}[X] = \sum_i \mathbf{Var}[X_i] = \sum_i (\mathbb{E}[X_i^2] - \mathbb{E}[X_i]^2) = \sum_i (\mathbb{E}[X_i] - \mathbb{E}[X_i]^2)$$

$$= \mathbb{E}[X] - \sum_i \mathbb{E}[X_i]^2 \leq \mathbb{E}[X]$$

Corollary (Chebyshev's Inequality):

If X is a sum of pairwise independent random variables taking values in $\{0,1\}$, for any $t > 0$,

$$\Pr\Big[\,|X - \mathbb{E}[X]| \geq t\Big] \leq \frac{\mathbb{E}[X]}{t^2}$$

Input: a sequence $x_1, x_2, \ldots, x_n \in [N] \subseteq [2^w]$

Output: an estimation of $z = \left| \left\{ x_1, x_2, \ldots, x_n \right\} \right|$

- 2-wise independent hash function $h : [2^w] \to [2^w]$
- For $y \in [2^w]$, let $\mathrm{zeros}(y) = \max\{i : 2^i \mid y\}$ denote # of trailing 0's

Flajolet-Martin Algorithm:

let $R = \max_{1 \le i \le n} \mathrm{zeros}(h(x_i))$;

return $\widehat{Z} = 2^R$;

Let

$$Y_r = \sum_{x \in \{x_1, \ldots, x_n\}} I\left[\mathrm{zeros}\left(h(x)\right) \ge r\right]$$

(linearity of expectation)

$$\mathbb{E}[Y_r] = \sum_{x \in \{x_1, \ldots, x_n\}} \Pr\left[\mathrm{zeros}\left(h(x)\right) \ge r\right] = z2^{-r}$$

(pairwise independence) $\quad \mathbf{Var}[Y_r] \le \mathbb{E}[Y_r] \le z2^{-r}$

Input: a sequence $x_1, x_2, \ldots, x_n \in [N] \subseteq [2^w]$

Output: an estimation of $z = \left| \{x_1, x_2, \ldots, x_n\} \right|$

- 2-wise independent hash function $h : [2^w] \to [2^w]$
- For $y \in [2^w]$, let $\mathrm{zeros}(y) = \max\{i : 2^i \mid y\}$ denote # of trailing 0's

Flajolet-Martin Algorithm:

let $R = \max_{1 \le i \le n} \mathrm{zeros}(h(x_i))$;

return $\widehat{Z} = 2^R$;

Let

$$Y_r = \sum_{x \in \{x_1, \ldots, x_n\}} I\left[\mathrm{zeros}\left(h(x)\right) \ge r\right]$$

$$\mathbb{E}[Y_r] = z2^{-r} \qquad \mathbf{Var}[Y_r] \le z2^{-r}$$

(denote $r^* = \lceil \log_2 Cz \rceil$) $\quad \Pr\left[\widehat{Z} > Cz\right] \le \Pr[R \ge r^*]$

(observe $R = \max\{r : Y_r > 0\}$) $\quad \le \Pr[Y_{r^*} > 0] = \Pr[Y_{r^*} \ge 1]$

(Markov's inequality) $\quad \le \mathbb{E}[Y_{r^*}] = z/2^{r^*} \le 1/C$

Input: a sequence $x_1, x_2, \ldots, x_n \in [N] \subseteq [2^w]$

Output: an estimation of $z = \left| \{x_1, x_2, \ldots, x_n\} \right|$

- 2-wise independent hash function $h : [2^w] \to [2^w]$
- For $y \in [2^w]$, let $\mathrm{zeros}(y) = \max\{i : 2^i \,|\, y\}$ denote # of trailing 0's

Flajolet-Martin Algorithm:

let $R = \max_{1 \le i \le n} \mathrm{zeros}(h(x_i))$;

return $\widehat{Z} = 2^R$;

Let

$$Y_r = \sum_{x \in \{x_1, \ldots, x_n\}} I\left[\mathrm{zeros}\left(h(x)\right) \ge r\right]$$

$$\mathbb{E}[Y_r] = z2^{-r} \qquad \mathbf{Var}[Y_r] \le z2^{-r}$$

(denote $r^{**} = \lceil \log_2(z/C) \rceil$) $\quad \Pr\left[\widehat{Z} < z/C\right] \le \Pr[R < r^{**}]$

(observe $R = \max\{r : Y_r > 0\}$) $\quad \le \Pr[Y_{r^{**}} = 0]$

(Chebyshev's inequality) $\quad \le \mathbf{Var}[Y_{r^{**}}]/\mathbb{E}[Y_{r^{**}}]^2 \le 2^{r^{**}}/z$

$$\le 2/C$$

Input: a sequence $x_1, x_2, \ldots, x_n \in [N] \subseteq [2^w]$

Output: an estimation of $z = \left| \{x_1, x_2, \ldots, x_n\} \right|$

- 2-wise independent hash function $h : [2^w] \to [2^w]$
- For $y \in [2^w]$, let $\text{zeros}(y) = \max\{i : 2^i \mid y\}$ denote # of trailing 0's

Flajolet-Martin Algorithm:

let $R = \max_{1 \le i \le n} \text{zeros}(h(x_i))$;

return $\hat{Z} = 2^R$;

$$\Pr\left[\hat{Z} < \frac{z}{C} \text{ or } \hat{Z} > C \cdot z\right] \le \frac{3}{C}$$

- **Space cost:** $O(\log \log N)$ bits for maintaining R
- $O(\log N)$ bits for storing 2-wise independent hash function

BJKST Algorithm

Input: a sequence $x_1, x_2, \ldots, x_n \in [N]$

Output: an estimation of $z = \left| \left\{ x_1, x_2, \ldots, x_n \right\} \right|$

- 2-wise independent hash function $h : [N] \to [M] = \{1, \ldots, M\}$

BJKST Algorithm:

let $Y_1, \ldots, Y_k$ be the k smallest hash values among $\left\{ h(x_1), h(x_2) \ldots, h(x_n) \right\}$;

return $\widehat{Z} = \dfrac{kM}{Y_k}$;

(Bar-Yossef, Jayram, Kumar, Sivakumar and Trevisan, 2002)

Input: a sequence $x_1, x_2, \ldots, x_n \in [N]$

Output: an estimation of $z = \left| \left\{ x_1, x_2, \ldots, x_n \right\} \right|$

- 2-wise independent hash function $h : [N] \to [M] = \{1, \ldots, M\}$

BJKST Algorithm:

let $Y_1, \ldots, Y_k$ be the k smallest hash values among $\left\{ h(x_1), h(x_2) \ldots, h(x_n) \right\}$;

return $\widehat{Z} = \dfrac{kM}{Y_k}$;

- Goal: $\Pr\left[\widehat{Z} < (1-\epsilon)z \text{ or } \widehat{Z} > (1+\epsilon)z \right] \le \delta$

assuming $\epsilon \le 1$

$$\left| Y_k - \frac{kM}{z} \right| > \frac{\epsilon}{2} \cdot \frac{kM}{z}$$

- uniform and 2-wise independent $X_1, \ldots, X_n \in [N^3]$
- let $Y_1, \ldots, Y_z$ be these elements in non-decreasing order

Let $V = \sum_{i=1}^{z} I\left[X_i \le \left(1 - \frac{\epsilon}{2}\right)\frac{kM}{z}\right]$ $\quad W = \sum_{i=1}^{z} I\left[X_i \le \left(1 + \frac{\epsilon}{2}\right)\frac{kM}{z}\right]$

$$\mathbb{E}[V] = \left(1 - \frac{\epsilon}{2} \pm o(1)\right)k \qquad \mathbb{E}[W] = \left(1 + \frac{\epsilon}{2} \pm o(1)\right)k$$

$$Y_k < \left(1 - \frac{\epsilon}{2}\right)\frac{k(M+1)}{z} \implies V \ge k \qquad Y_k > \left(1 + \frac{\epsilon}{2}\right)\frac{k(M+1)}{z} \implies W \le k$$

(Chebyshev's inequality for sum of pairwise independent trials)

$$\Pr[V \ge k] \le \frac{8}{k\epsilon^2} \qquad \Pr[W \le k] \le \frac{8}{k\epsilon^2}$$

- Goal: $\Pr\left[\left|Y_k - \frac{kM}{z}\right| > \frac{\epsilon}{2} \cdot \frac{kM}{z}\right] \le \delta$ $\quad$ Set $k = \left\lceil \frac{16}{\epsilon^2 \delta} \right\rceil$

Input: a sequence $x_1, x_2, \ldots, x_n \in [N]$

Output: an estimation of $z = \left| \left\{ x_1, x_2, \ldots, x_n \right\} \right|$

- 2-wise independent hash function $h : [N] \to [N^3]$

BJKST Algorithm: Set $k = \lceil 16/(\epsilon^2 \delta) \rceil$

let $Y_1, \ldots, Y_k$ be the k smallest hash values among $\left\{ h(x_1), h(x_2) \ldots, h(x_n) \right\}$;

return $\widehat{Z} = \frac{kM}{Y_k}$;

$$\Pr\left[(1-\epsilon)z \leq \widehat{Z} \leq (1+\epsilon)z \right] \geq 1 - \delta$$

- **Space cost:** $O(k \log N) = O(\epsilon^{-2} \log N)$ bits when $\delta = \Omega(1)$

Frequency Moments

- **Data stream:** $x_1, x_2, \ldots, x_n \in U$
- for each $x \in U$, define ***frequency*** of x as $f_x = |\{i : x_i = x\}|$

  k-th ***frequency moments***: $F_k = \sum_{x \in U} f_x^k$

- **Space complexity** for (ϵ, δ)-estimation: constant ϵ, δ
  - for $k \leq 2$: $\text{polylog}(N)$ [Alon-Matias-Szegedy '96]
  - for $k > 2$: $\tilde{\Theta}(N^{1-2/k})$ [Indyk-Woodruff '05]
- **Count distinct elements:** F_0
  - optimal algorithm [Kane-Nelson-Woodruff '10]: $O(\epsilon^{-2} + \log N)$ bits

Frequency Estimation

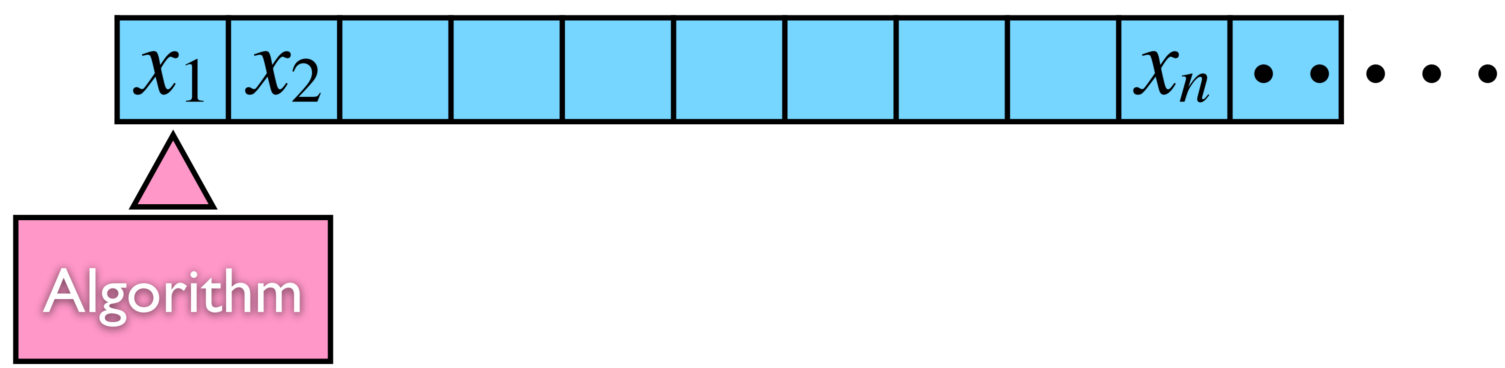
Frequency Estimation

Data: a sequence $x_1, x_2, \ldots, x_n \in U = [N]$

Query: an item $x \in U$

Estimate the *frequency* $f_x = |\{i : x_i = x\}|$ of x.

- **Data stream model**: input data item comes one at a time



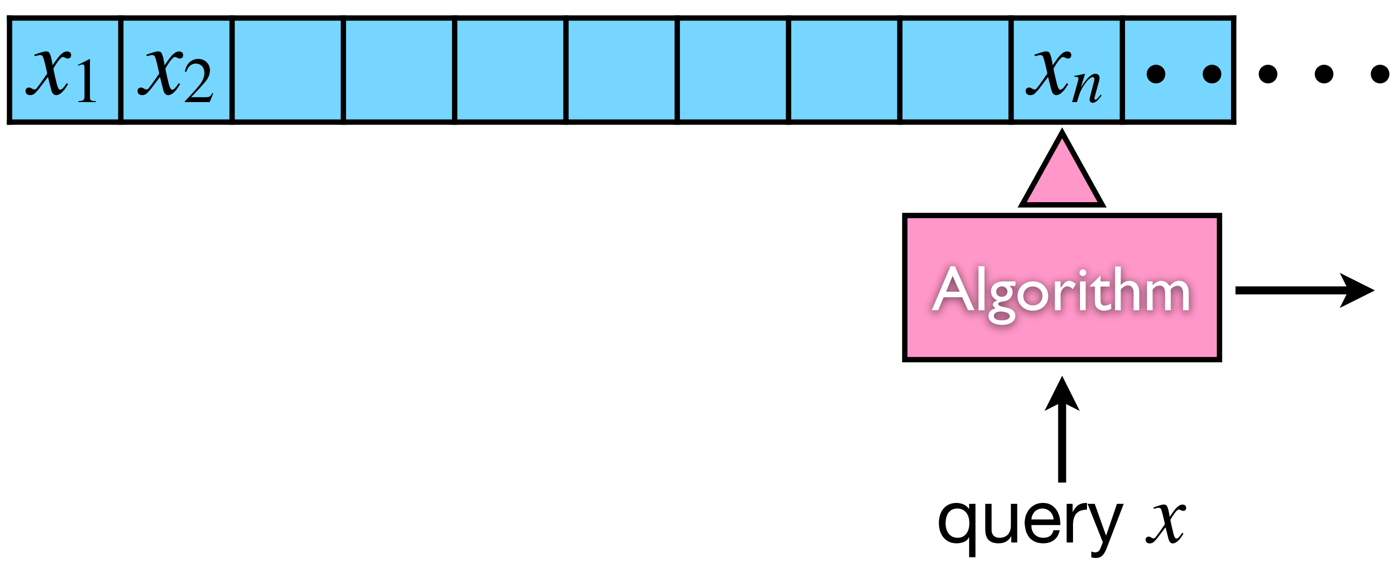
Frequency Estimation

Data: a sequence $x_1, x_2, \ldots, x_n \in U = [N]$

Query: an item $x \in U$

Estimate the ***frequency*** $f_x = |\{i : x_i = x\}|$ of x.

- **Data stream model**: input data item comes one at a time



$\hat{f}_x$: estimation of f_x within additive error

$$\Pr\left[\,|\hat{f}_x - f_x| \geq \epsilon n\right] \leq \delta$$

- Heavy hitters: items that appears $> \epsilon n$ times

Data Structure for Set

Data: a set S of n items $x_1, x_2, \ldots, x_n \in U = [N]$

Query: an item $x \in U$

Determine whether $x \in S$.

- **Space cost**: size of data structure (in bits)
  - entropy of a set: $\mathrm{O}(n \log N)$ bits (when $N \gg n$)
- **Time cost**: time to answer a query (in memory accesses)
- Balanced tree: $\mathrm{O}(n \log N)$ space, $\mathrm{O}(\log n)$ time
- Perfect hashing: $\mathrm{O}(n \log N)$ space, $\mathrm{O}(1)$ time
- **Sketch**: lossy representation of S using < entropy space

Approximate Set

Data: a set S of n items $x_1, x_2, \ldots, x_n \in U = [N]$

Query: an item $x \in U$

Determine whether $x \in S$.

- uniform hash function $h : U \to [m]$ (m to be fixed)

Data Structure: bit vector $v \in \{0,1\}^m$

v is initialized to all 0's;

for each $x_i \in S$: set $v[h(x_i)] = 1$;

Query x: answer "yes" iff $v[h(x)] = 1$

- $x \in S$: always correct
- $x \notin S$: false positive $\Pr\left[v[h(x)] = 1\right] = 1 - (1 - 1/m)^n \approx 1 - e^{-n/m}$

Bloom Filters (Bloom 1970)

Data: a set S of n items $x_1, x_2, \ldots, x_n \in U = [N]$

Query: an item $x \in U$

Determine whether $x \in S$.

- uniform & independent hash function $h_1, \ldots, h_k : U \to [m]$

(k and m to be fixed)

Data Structure: bit vector $v \in \{0,1\}^m$

v is initialized to all 0's;

for each $x_i \in S$: set $v[h_j(x_i)] = 1$ for all $1 \le j \le k$;

Query x: "yes" iff $v[h_j(x)] = 1$ for all $1 \le j \le k$

Bloom Filters (Bloom 1970)

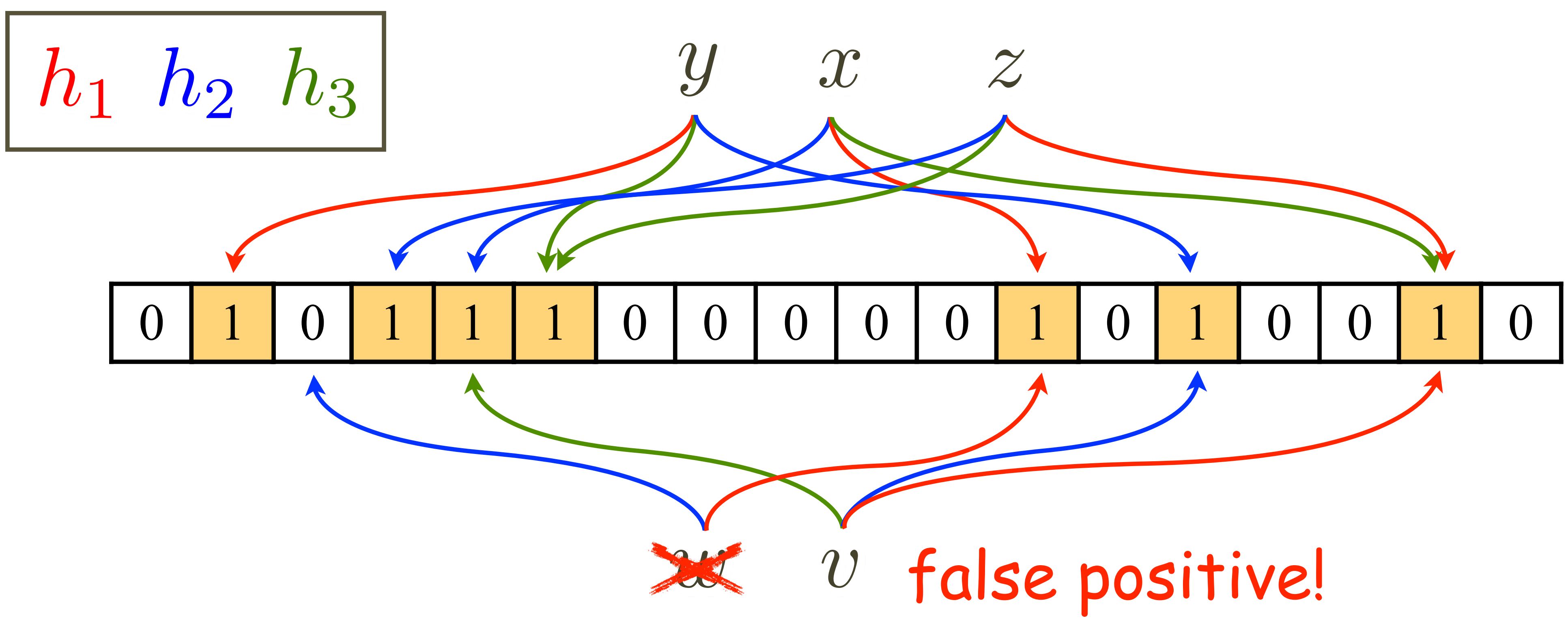
- uniform & independent hash function $h_1, \ldots, h_k : U \to [m]$

Data Structure: bit vector $v \in \{0,1\}^m$

v is initialized to all 0's;

for each $x_i \in S$: set $v[h_j(x_i)] = 1$ for all $1 \le j \le k$;

Query x: "yes" iff $v[h_j(x)] = 1$ for all $1 \le j \le k$



Data: set $S \subseteq U$ of size n **Query**: $x \in U$

- uniform & independent hash function $h_1, \ldots, h_k : U \to [m]$

Data Structure: bit vector $v \in \{0,1\}^m$

v is initialized to all 0's;

for each $x_i \in S$: set $v[h_j(x_i)] = 1$ for all $1 \le j \le k$;

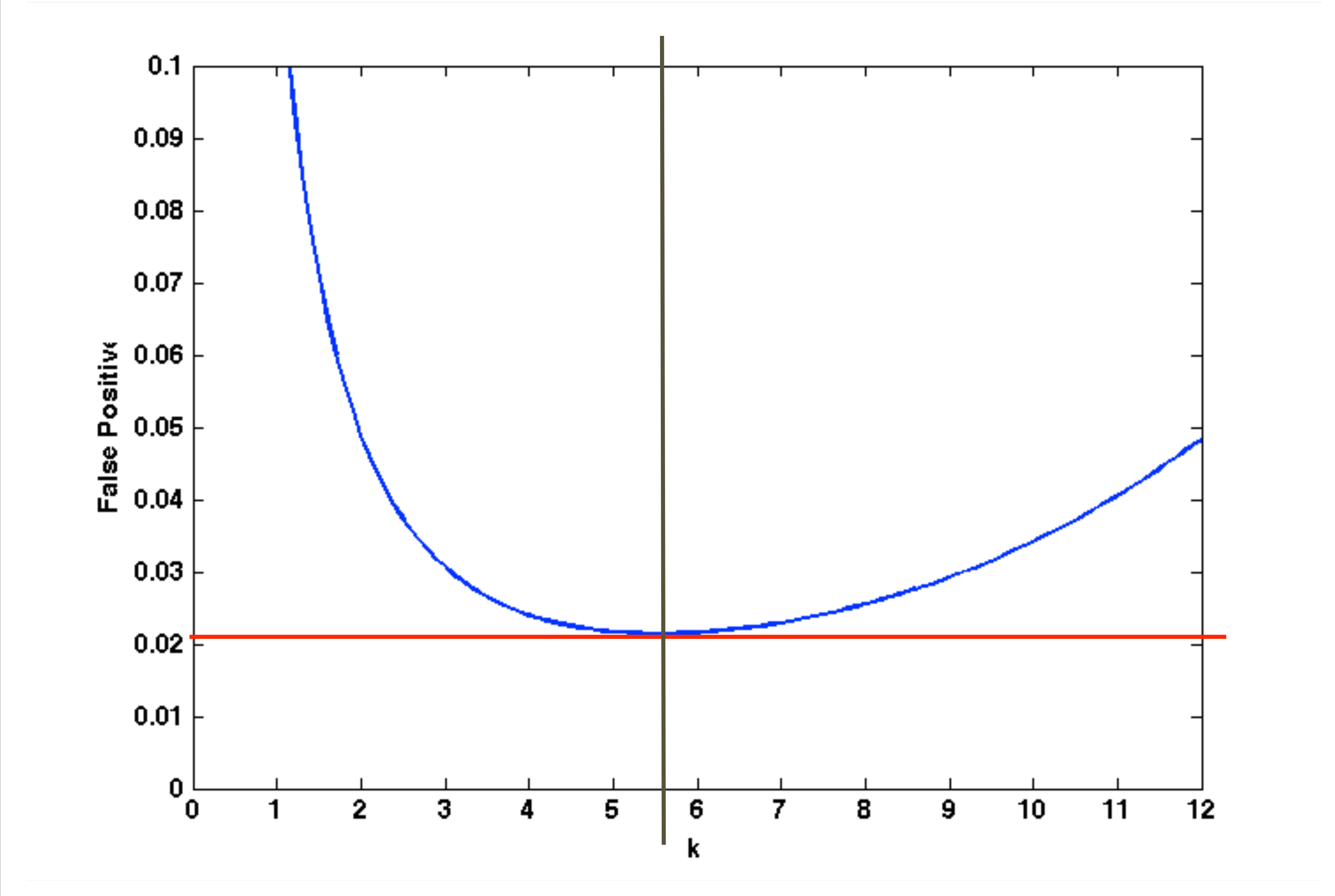
Query x: "yes" iff $v[h_j(x)] = 1$ for all $1 \le j \le k$

- $x \in S$: always correct
- $x \notin S$: false positive

$$\Pr\left[\forall 1 \le j \le k:\ \ v[h_j(x)] = 1\right]$$

$$= \left(\Pr\left[v[h_j(x)] = 1\right]\right)^k = \left(1 - \Pr\left[v[h_j(x)] = 0\right]\right)^k$$

$$\le (1-(1-1/m)^{kn})^k \ \approx \left(1 - \mathrm{e}^{-kn/m}\right)^k$$



y: $x \in U$

$\ldots, h_k : U \to [m]$

ll $1 \le j \le k$;

$\le j \le k$

- $x \notin S$: false positive

$$\Pr\left[\forall 1 \le j \le k:\ v[h_j(x)] = 1\right]$$

choose $k = c \ln 2$

$m = cn$

$$= \left(\Pr\left[v[h_j(x)] = 1\right]\right)^k = \left(1 - \Pr\left[v[h_j(x)] = 0\right]\right)^k$$

$$\le (1 - (1 - 1/m)^{kn})^k \approx \left(1 - e^{-kn/m}\right)^k = 2^{-c \ln 2} \le (0.6185)^c$$

Bloom Filters (Bloom 1970)

Data: a set S of n items $x_1, x_2, \ldots, x_n \in U = [N]$

Query: an item $x \in U$

Determine whether $x \in S$.

- uniform & independent hash function $h_1, \ldots, h_k : U \to [m]$

Data Structure: bit vector $v \in \{0,1\}^m$

v is initialized to all 0's;

for each $x_i \in S$: set $v[h_j(x_i)] = 1$ for all $1 \le j \le k$;

Query x: "yes" iff $v[h_j(x)] = 1$ for all $1 \le j \le k$

- choose $k = c \ln 2$ and $m = cn$
  - **space cost:** $m = cn$ bits, **time cost:** $k = c \ln 2$
  - **false positive** $\le (0.6185)^c$

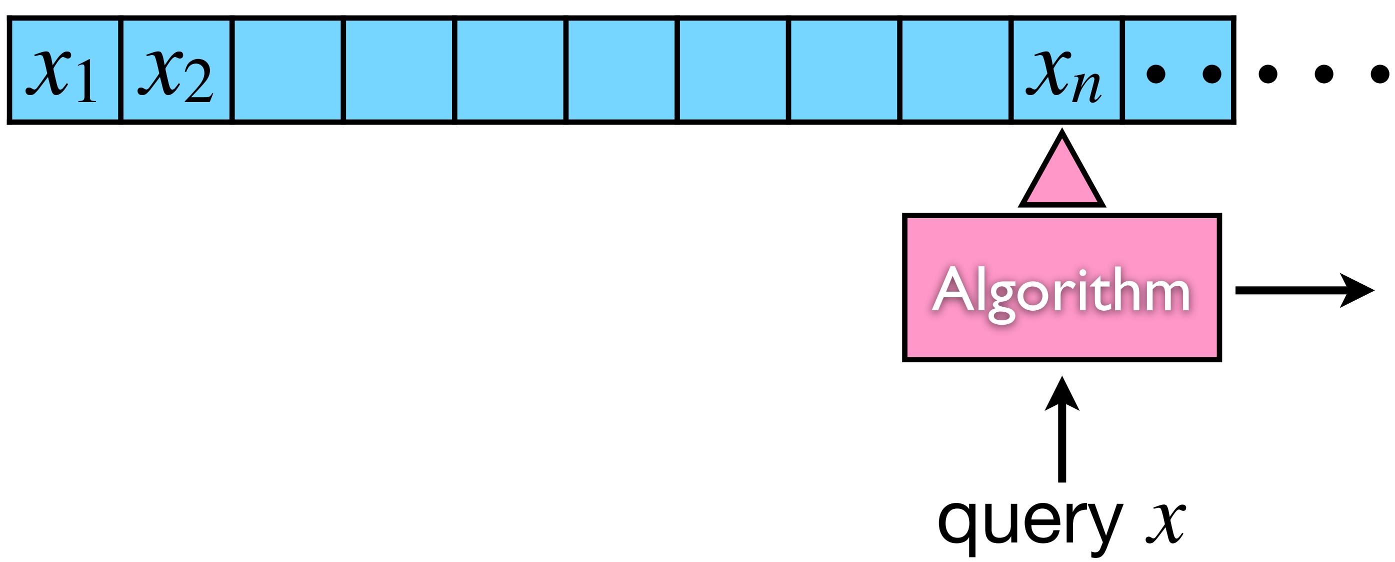
Frequency Estimation

Data: a sequence $x_1, x_2, \ldots, x_n \in U = [N]$

Query: an item $x \in U$

Estimate the *frequency* $f_x = |\{i : x_i = x\}|$ of x.

- Data stream **model**: input data item comes one at a time



$\hat{f}_x$: estimation of f_x within additive error

$$\Pr\left[|\hat{f}_x - f_x| \geq \epsilon n \right] \leq \delta$$

- Heavy hitters: items that appears $> \epsilon n$ times

Count-Min Sketch

Data: a sequence $x_1, x_2, \ldots, x_n \in U = [N]$

Query: an item $x \in U$

Estimate the ***frequency*** $f_x = |\{i : x_i = x\}|$ of x.

- k independent 2-universal hash functions $h_1, \ldots, h_k : [N] \to [m]$

Count-Min Sketch: $\mathrm{CMS}[k][m]$ (initialized to all 0's)

Upon each x_i: $\mathrm{CMS}[j][h_j(x_i)]++$ for all $1 \le j \le k$;

Query x: return $\hat{f}_x = \min_{1 \le j \le k} \mathrm{CMS}[j][h_j(x)]$

Observation: $\mathrm{CMS}[j][h_j(x)] \ge f_x$ for all $1 \le j \le k$

$$f_x \le \hat{f}_x \le ?$$

Universal Hashing

Universal Hash Family (Carter and Wegman 1979):

A family $\mathscr{H}$ of hash functions in $U \to [m]$ is **k-universal** if for any distinct $x_1, \ldots, x_k \in U$,

$$\Pr_{h \in \mathscr{H}} \left[h(x_1) = \cdots = h(x_k) \right] \leq \frac{1}{m^{k-1}}.$$

Moreover, $\mathscr{H}$ is **strongly k-universal** (k-wise independent) if for any distinct $x_1, \ldots, x_k \in U$ and any $y_1, \ldots, y_k \in [m]$,

$$\Pr_{h \in \mathscr{H}} \left[\bigwedge_{i=1}^{k} h(x_i) = y_i \right] = \frac{1}{m^k}.$$

k-Universal Hash Family

hash functions $h : U \to [m]$

- **Linear congruential hashing**:
  - Represent $U \subseteq \mathbb{Z}_p$ for sufficiently large prime p
  - $h_{a,b}(x) = ((ax + b) \bmod p) \bmod m$
  - $\mathscr{H} = \left\{ h_{a,b} \mid a \in \mathbb{Z}_p \setminus \{0\}, b \in \mathbb{Z}_p \right\}$

> **Theorem**:
> The linear congruential family $\mathscr{H}$ is 2-wise independent.

- **Degree-k polynomial in finite field with random coefficients**
- Hashing between binary fields: $GF(2^w) \to GF(2^l)$

$$h_{a,b}(x) = \texttt{(a*x+b)>>(w-l)}$$

Data: sequence $x_1, \ldots, x_n \in [N]$ **Query**: $x \in [N]$

frequency $f_x = |\{i : x_i = x\}|$ of x

- k independent 2-universal hash functions $h_1, \ldots, h_k : [N] \to [m]$

Count-Min Sketch: $\mathrm{CMS}[k][m]$ (initialized to all 0's)

Upon each x_i: $\mathrm{CMS}[j][h_j(x_i)]++$ for all $1 \le j \le k$;

Query x: return $\hat{f}_x = \min_{1\le j\le k} \mathrm{CMS}[j][h_j(x)]$

- for any $x \in [N]$ and any $1 \le j \le k$:

$$\mathrm{CMS}[j][h_j(x)] = f_x + \sum_{\substack{y \in \{x_1, \ldots, x_n\}\setminus\{x\} \\ h_j(y) = h_j(x)}} f_y$$

$$\mathbb{E}\left[\mathrm{CMS}[j][h_j(x)]\right] = f_x + \sum_{y\in\{x_1,\ldots,x_n\}\setminus\{x\}} f_y \Pr[h_j(y) = h_j(x)]$$

Data: sequence $x_1, \ldots, x_n \in [N]$ **Query**: $x \in [N]$

frequency $f_x = |\{i : x_i = x\}|$ of x

- k independent 2-universal hash functions $h_1, \ldots, h_k : [N] \to [m]$

Count-Min Sketch: $\mathrm{CMS}[k][m]$ (initialized to all 0's)

Upon each x_i: $\mathrm{CMS}[j][h_j(x_i)] ++$ for all $1 \le j \le k$;

Query x: return $\hat{f}_x = \min_{1 \le j \le k} \mathrm{CMS}[j][h_j(x)]$

- for any $x \in [N]$ and any $1 \le j \le k$:

$$\mathbb{E}\left[\mathrm{CMS}[j][h_j(x)]\right] = f_x + \sum_{y \in \{x_1, \ldots, x_n\} \setminus \{x\}} f_y \Pr[h_j(y) = h_j(x)]$$

$$\le f_x + \frac{1}{m} \sum_{y \in \{x_1, \ldots, x_n\} \setminus \{x\}} f_y \le f_x + \frac{1}{m} \sum_{y \in \{x_1, \ldots, x_n\}} f_y = f_x + \frac{n}{m}$$

Data: sequence $x_1, \ldots, x_n \in [N]$ **Query**: $x \in [N]$

frequency $f_x = |\{i : x_i = x\}|$ of x

- k independent 2-universal hash functions $h_1, \ldots, h_k : [N] \to [m]$

Count-Min Sketch: $\mathrm{CMS}[k][m]$ (initialized to all 0's)

Upon each x_i: $\mathrm{CMS}[j][h_j(x_i)]++$ for all $1 \le j \le k$;

Query x: return $\hat{f}_x = \min_{1 \le j \le k} \mathrm{CMS}[j][h_j(x)]$

$\forall x, \forall j$: $\mathrm{CMS}[j][h_j(x)] \ge f_x$

$$\mathbb{E}\left[\mathrm{CMS}[j][h_j(x)]\right] \le f_x + \frac{n}{m}$$

(Markov's inequality) $\Pr\left[\mathrm{CMS}[j][h_j(x)] - f_x \ge \epsilon n\right] \le \frac{1}{\epsilon m}$

$$\Pr\left[|\hat{f}_x - f_x| \ge \epsilon n\right] = \Pr\left[\forall 1 \le j \le k : \mathrm{CMS}[j][h_j(x)] - f_x \ge \epsilon n\right] \le \left(\frac{1}{\epsilon m}\right)^k$$

Data: a sequence $x_1, x_2, \ldots, x_n \in U = [N]$

Query: an item $x \in U$

Estimate the ***frequency*** $f_x = |\{i : x_i = x\}|$ of x.

- k independent 2-universal hash functions $h_1, \ldots, h_k : [N] \to [m]$

Count-Min Sketch: $\mathrm{CMS}[k][m]$ (initialized to all 0's)

Upon each x_i: $\mathrm{CMS}[j][h_j(x_i)] + +$ for all $1 \le j \le k$;

Query x: return $\hat{f}_x = \min_{1 \le j \le k} \mathrm{CMS}[j][h_j(x)]$

$$\Pr\left[|\hat{f}_x - f_x| \ge \epsilon n\right] \le \left(\frac{1}{\epsilon m}\right)^k \le \delta$$

- choose $m = \lceil e/\epsilon \rceil$ and $k = \lceil \ln(1/\delta) \rceil$
  - **space cost:** $O\left(\frac{1}{\epsilon}\log(1/\delta)\log n\right)$ bits
  - $O\left(\log(1/\delta)\log N\right)$ bits for hash functions
  - **time cost for query:** $O\left(\log(1/\delta)\right)$