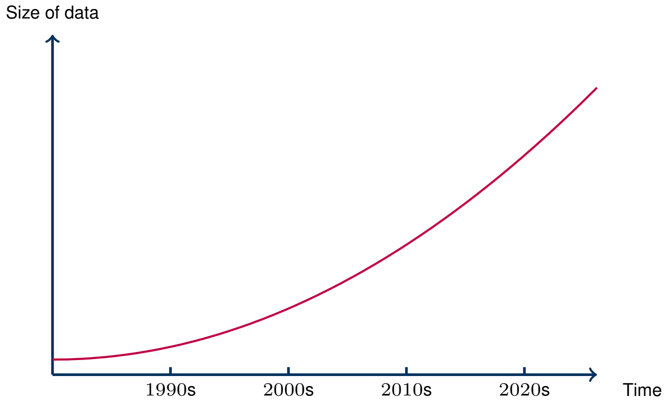
Data streaming algorithms (1)

He Sun

THE UNIVERSITY of EDINBURGH

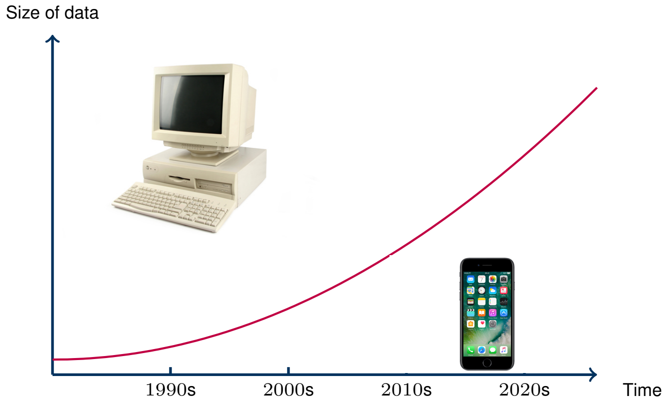
Background

- The amount of data has been increased exponentially in the past;



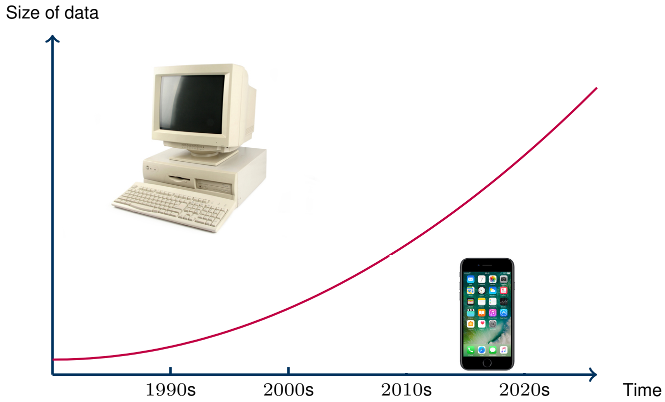
Background

- The amount of data has been increased exponentially in the past;
- For many applications computational devices' memories are limited;



Background

- The amount of data has been increased exponentially in the past;
- For many applications computational devices' memories are limited;
- We only need good approximate solutions!



Streaming algorithms

- The input of a streaming algorithm is given as a data stream, which is a sequence of data
$$\mathcal{S} = s_1, s_2, \cdots, s_m, \cdots,$$
and every s_i belongs to the universe U of size n.

Streaming algorithms

- The input of a streaming algorithm is given as a data stream, which is a sequence of data
$$\mathcal{S} = s_1, s_2, \cdots, s_m, \cdots,$$
and every s_i belongs to the universe U of size n.

- Constraints for streaming algorithms: the space complexity is sublinear in n, and is independent in the length of $\mathcal{S}$.

Streaming algorithms

- The input of a streaming algorithm is given as a data stream, which is a sequence of data
$$\mathcal{S} = s_1, s_2, \cdots, s_m, \cdots,$$
and every s_i belongs to the universe U of size n.

- Constraints for streaming algorithms: the space complexity is sublinear in n, and is independent in the length of $\mathcal{S}$.

- Quality of the output: The algorithm needs to give a good approximate value with high probability.

Streaming algorithms

- The input of a streaming algorithm is given as a data stream, which is a sequence of data
$$\mathcal{S} = s_1, s_2, \cdots, s_m, \cdots,$$
and every s_i belongs to the universe U of size n.

- Constraints for streaming algorithms: the space complexity is sublinear in n, and is independent in the length of $\mathcal{S}$.

- Quality of the output: The algorithm needs to give a good approximate value with high probability.

(ε, δ)-APPROXIMATION

For confidence parameter ε and approximation parameter δ, the algorithm's output Output and the exact answer Exact satisfies
$$\mathbb{P}\left[\mathsf{Output} \in (1-\varepsilon, 1+\varepsilon) \cdot \mathsf{Exact}\right] \geq 1 - \delta.$$

Two models of streaming algorithms

Cash register model: every item in stream $\mathcal{S}$ is an item in U.

Two models of streaming algorithms

Cash register model: every item in stream $\mathcal{S}$ is an item in U.

Turnstile model: every item s_i in $\mathcal{S}$ associates with " +" or "-", which indicates if s_i is added into or deleted from $\mathcal{S}$.

- "+" indicates that s_i is added into the dataset;
- "-" indicates that s_i is deleted from the dataset.

Why turnstile model?

- Data may be added or deleted over time, e.g. Facebook graph.
- We need *robust* algorithms to handle this situation.

Outline

- Recall: Pairwise independent hashing
- AMS algorithm
- BJKST algorithm
- Chernoff Bound

Recall: Pairwise independent hash functions

PAIRWISE INDEPENDENCE

A family of functions $H = \{h \mid h : U \mapsto [n]\}$ is pairwise independent if, for any h chosen uniformly at random from H, the following holds:

1. $h(x)$ is uniformly distributed in $[n]$ for any $x \in U$;
2. For any $x_1 \neq x_2 \in U$, $h(x_1)$ and $h(x_2)$ are independent.

Recall: Pairwise independent hash functions

PAIRWISE INDEPENDENCE

A family of functions $H = \{h \mid h : U \mapsto [n]\}$ is pairwise independent if, for any h chosen uniformly at random from H, the following holds:

1. $h(x)$ is uniformly distributed in $[n]$ for any $x \in U$;
2. For any $x_1 \neq x_2 \in U$, $h(x_1)$ and $h(x_2)$ are independent.

THEOREM

Let p be a prime number, and let $h_{a,b}(x) = (ax + b) \bmod p$. Define

$$H = \{h_{a,b} \mid 0 \leq a, b \leq p - 1\}.$$

Then H is a family of pairwise independent hash functions.

Outline

- Recall: Pairwise independent hashing
- AMS algorithm
- BJKST algorithm
- Chernoff Bound

Norm Estimation: the Alon-Matias-Szegedy algorithm

F_p-NORM

Let U with $|U| = n$ be a dataset, and m_j be the number of occurrences of j in a stream. The F_p-norm is defined by

$$F_p \triangleq \sum_{i \in U} |m_i|^p \,.$$

Norm Estimation: the Alon-Matias-Szegedy algorithm

F_p-NORM

Let U with $|U| = n$ be a dataset, and m_j be the number of occurrences of j in a stream. The F_p-norm is defined by

$$F_p \triangleq \sum_{i \in U} |m_i|^p .$$

- F_1 = total number of items in stream $\mathcal{S}$.
- F_0 = total number of distinct items in stream $\mathcal{S}$.

Norm Estimation: the Alon-Matias-Szegedy algorithm

F_p-NORM

Let U with $|U| = n$ be a dataset, and m_j be the number of occurrences of j in a stream. The F_p-norm is defined by

$$F_p \triangleq \sum_{i \in U} |m_i|^p .$$

- F_1 = total number of items in stream $\mathcal{S}$.
- F_0 = total number of distinct items in stream $\mathcal{S}$.

Alon, Matias, and Szegedy (1996) presented a systematical study for approximating the frequency moments.

- The numbers F_0, F_1, F_2 can be approximated in logarithmic space.
- Approximating F_k for $k \geq 6$ requires $n^{\Omega(1)}$ space.
- The paper won 2005 Gödel Award for “their foundational contribution to streaming algorithms”.

Intuitions behind the AMS algorithm

Assume that we have a random hash function h.

Intuitions behind the AMS algorithm

Assume that we have a random hash function h. Define

$$\rho(x) \triangleq \max_i \left\{ i : x \mod 2^i = 0 \right\},$$

which is the number of consecutive 0's at the right-side, in the binary expression of x.

Intuitions behind the AMS algorithm

Assume that we have a random hash function h. Define

$$\rho(x) \triangleq \max_i \left\{ i : x \mod 2^i = 0 \right\},$$

which is the number of consecutive 0's at the right-side, in the binary expression of x.

Example. $\rho(2) = 1, \rho(3) = 0, \rho(4) = 2, \rho(8) = 3, \rho(16) = 4, \rho(17) = 0.$

Intuitions behind the AMS algorithm

Assume that we have a random hash function h. Define

$$\rho(x) \triangleq \max_i \left\{ i : x \mod 2^i = 0 \right\},$$

which is the number of consecutive 0's at the right-side, in the binary expression of x.

Example. $\rho(2) = 1, \rho(3) = 0, \rho(4) = 2, \rho(8) = 3, \rho(16) = 4, \rho(17) = 0.$

Observation. Since $h(x)$ is uniformly distributed, the following holds:

- with probability $1/2$, we have $\rho(h(x)) = 1$

Intuitions behind the AMS algorithm

Assume that we have a random hash function h. Define

$$\rho(x) \triangleq \max_i \left\{ i : x \mod 2^i = 0 \right\},$$

which is the number of consecutive 0's at the right-side, in the binary expression of x.

Example. $\rho(2) = 1, \rho(3) = 0, \rho(4) = 2, \rho(8) = 3, \rho(16) = 4, \rho(17) = 0.$

Observation. Since $h(x)$ is uniformly distributed, the following holds:

- with probability $1/2$, we have $\rho(h(x)) = 1$
- with probability $1/4$, we have $\rho(h(x)) = 2$

Intuitions behind the AMS algorithm

Assume that we have a random hash function h. Define

$$\rho(x) \triangleq \max_i \left\{ i : x \mod 2^i = 0 \right\},$$

which is the number of consecutive 0's at the right-side, in the binary expression of x.

Example. $\rho(2) = 1, \rho(3) = 0, \rho(4) = 2, \rho(8) = 3, \rho(16) = 4, \rho(17) = 0.$

Observation. Since $h(x)$ is uniformly distributed, the following holds:

- with probability $1/2$, we have $\rho(h(x)) = 1$
- with probability $1/4$, we have $\rho(h(x)) = 2$
- $\ldots$
- with probability $1/2^r$, we have $\rho(h(x)) = r$

The AMS algorithm

ALGORITHM: AMS

1: Choose a random hash function $h : [n] \to [n]$

The AMS algorithm

ALGORITHM: AMS

1: Choose a random hash function $h : [n] \to [n]$
2: $z = 0$

The AMS algorithm

ALGORITHM: AMS

1: Choose a random hash function $h : [n] \to [n]$
2: $z = 0$
3: **while** item x from stream $\mathcal{S}$ arrives

The AMS algorithm

ALGORITHM: AMS

1: Choose a random hash function $h : [n] \to [n]$
2: $z = 0$
3: **while** item x from stream $\mathcal{S}$ arrives
4: if $\rho(h(x)) > z$, then $z = \rho(h(x))$

The AMS algorithm

ALGORITHM: AMS

Choose a random hash function $h : [n] \rightarrow [n]$
$z = 0$
while item x from stream $\mathcal{S}$ arrives
  if $\rho(h(x)) > z$, then $z = \rho(h(x))$
return $2^{z+1/2}$

The AMS algorithm

Algorithm: AMS

1: Choose a random hash function $h : [n] \to [n]$
2: $z = 0$
3: **while** item x from stream $\mathcal{S}$ arrives
4: if $\rho(h(x)) > z$, then $z = \rho(h(x))$
5: **return** $2^{z+1/2}$

Theorem

With constant probability, the algorithm's output satisfies

$$2^{z+1/2} \in [F_0/3, 3 \cdot F_0].$$

The AMS algorithm

ALGORITHM: AMS

1: Choose a random hash function $h : [n] \to [n]$
2: $z = 0$
3: **while** item x from stream $\mathcal{S}$ arrives
4: if $\rho(h(x)) > z$, then $z = \rho(h(x))$
5: **return** $2^{z+1/2}$

THEOREM

With constant probability, the algorithm's output satisfies

$$2^{z+1/2} \in [F_0/3, 3 \cdot F_0].$$

We get an $(O(1), \delta)$-approximation of F_0 by running $\Theta(\log(1/\delta))$ independent copies of the algorithm and returning the medium.

The AMS algorithm

Algorithm: AMS

1: Choose a random hash function $h : [n] \to [n]$
2: $z = 0$
3: **while** item x from stream $\mathcal{S}$ arrives
4: if $\rho(h(x)) > z$, then $z = \rho(h(x))$
5: **return** $2^{z+1/2}$

Theorem

With constant probability, the algorithm's output satisfies

$$2^{z+1/2} \in [F_0/3, 3 \cdot F_0].$$

We get an $(O(1), \delta)$-approximation of F_0 by running $\Theta(\log(1/\delta))$ independent copies of the algorithm and returning the medium.

Recall (ε, δ)-approximation: $\mathbb{P}\left[\text{Output} \in (1-\varepsilon, 1+\varepsilon) \cdot \text{Exact}\right] \geq 1 - \delta$

Proof of the theorem (1/2)

Let $X_{r,j}$ be a 0/1 indicator random variable such that

$$X_{r,j} = 1 \Leftrightarrow \rho(h(j)) \geq r.$$

We say item j reaches level r if $X_{r,j} = 1$.

Proof of the theorem (1/2)

Let $X_{r,j}$ be a 0/1 indicator random variable such that

$$X_{r,j} = 1 \Leftrightarrow \rho(h(j)) \geq r.$$

We say item j reaches level r if $X_{r,j} = 1$.

Let $Y_r = \sum_{j \in \mathcal{S}} X_{r,j}$ be the number of items j reaching level r.

Proof of the theorem (1/2)

Let $X_{r,j}$ be a 0/1 indicator random variable such that

$$X_{r,j} = 1 \Leftrightarrow \rho(h(j)) \geq r.$$

We say item j reaches level r if $X_{r,j} = 1$.

Let $Y_r = \sum_{j \in \mathcal{S}} X_{r,j}$ be the number of items j reaching level r.

Since h is pairwise independent, $h(j)$ is uniformly distributed, and hence

$$\mathbb{E}[X_{r,j}] = \mathbb{P}[\rho(h(j)) \geq r] = \mathbb{P}[h(j) \mod 2^r = 0] = 1/2^r.$$

definition of function ρ

Proof of the theorem (1/2)

Let $X_{r,j}$ be a 0/1 indicator random variable such that

$$X_{r,j} = 1 \Leftrightarrow \rho(h(j)) \geq r.$$

We say item j reaches level r if $X_{r,j} = 1$.

Let $Y_r = \sum_{j \in \mathcal{S}} X_{r,j}$ be the number of items j reaching level r.

Since h is pairwise independent, $h(j)$ is uniformly distributed, and hence

$$\mathbb{E}[X_{r,j}] = \mathbb{P}[\rho(h(j)) \geq r] = \mathbb{P}[h(j) \mod 2^r = 0] = 1/2^r.$$

definition of function ρ

By linearity of expectation, we have

$$\mathbb{E}[Y_r] = \sum_{j \in \mathcal{S}} \mathbb{E}[X_{r,j}] = F_0/2^r,$$

Proof of the theorem (1/2)

Let $X_{r,j}$ be a 0/1 indicator random variable such that

$$X_{r,j} = 1 \Leftrightarrow \rho(h(j)) \geq r.$$

We say item j reaches level r if $X_{r,j} = 1$.

Let $Y_r = \sum_{j \in \mathcal{S}} X_{r,j}$ be the number of items j reaching level r.

Since h is pairwise independent, $h(j)$ is uniformly distributed, and hence

$$\mathbb{E}[X_{r,j}] = \mathbb{P}[\rho(h(j)) \geq r] = \mathbb{P}[h(j) \mod 2^r = 0] = 1/2^r.$$

definition of function ρ

By linearity of expectation, we have

$$\mathbb{E}[Y_r] = \sum_{j \in \mathcal{S}} \mathbb{E}[X_{r,j}] = F_0/2^r,$$

$$\mathbb{V}[Y_r] = \sum_{j \in \mathcal{S}} \mathbb{V}[X_{r,j}] \leq \sum_{j \in \mathcal{S}} \mathbb{E}[X_{r,j}^2] = \sum_{j \in \mathcal{S}} \mathbb{E}[X_{r,j}] = F_0/2^r$$

Proof of the theorem (2/2)

We have proved $\mathbb{E}[Y_r] = F_0/2^r$, and $\mathbb{V}[Y_r] \leq F_0/2^r$.

Proof of the theorem (2/2)

We have proved $\mathbb{E}[Y_r] = F_0/2^r$, and $\mathbb{V}[Y_r] \leq F_0/2^r$.

By Markov's inequality, we have

$$\mathbb{P}[Y_r > 0] = \mathbb{P}[Y_r \geq 1] \leq \frac{\mathbb{E}[Y_r]}{1} = \frac{F_0}{2^r}.$$

Proof of the theorem (2/2)

We have proved $\mathbb{E}[Y_r] = F_0/2^r$, and $\mathbb{V}[Y_r] \leq F_0/2^r$.

By Markov's inequality, we have

$$\mathbb{P}[Y_r > 0] = \mathbb{P}[Y_r \geq 1] \leq \frac{\mathbb{E}[Y_r]}{1} = \frac{F_0}{2^r}.$$

By Chebyshev's inequality, we have

$$\mathbb{P}[Y_r = 0] \leq \mathbb{P}\left[|Y_r - \mathbb{E}[Y_r]| \geq F_0/2^r\right] \leq \frac{\mathbb{V}[Y_r]}{(F_0/2^r)^2} \leq \frac{2^r}{F_0}.$$

Proof of the theorem (2/2)

We have proved $\mathbb{E}[Y_r] = F_0/2^r$, and $\mathbb{V}[Y_r] \leq F_0/2^r$.

By Markov's inequality, we have

$$\mathbb{P}[Y_r > 0] = \mathbb{P}[Y_r \geq 1] \leq \frac{\mathbb{E}[Y_r]}{1} = \frac{F_0}{2^r}.$$

By Chebyshev's inequality, we have

$$\mathbb{P}[Y_r = 0] \leq \mathbb{P}\left[|Y_r - \mathbb{E}[Y_r]| \geq F_0/2^r\right] \leq \frac{\mathbb{V}[Y_r]}{(F_0/2^r)^2} \leq \frac{2^r}{F_0}.$$

Let z be the final index the algo. keeps. So the algo. returns $Z = 2^{z+1/2}$.

Proof of the theorem (2/2)

We have proved $\mathbb{E}[Y_r] = F_0/2^r$, and $\mathbb{V}[Y_r] \leq F_0/2^r$.

By Markov's inequality, we have

$$\mathbb{P}[Y_r > 0] = \mathbb{P}[Y_r \geq 1] \leq \frac{\mathbb{E}[Y_r]}{1} = \frac{F_0}{2^r}.$$

By Chebyshev's inequality, we have

$$\mathbb{P}[Y_r = 0] \leq \mathbb{P}\left[|Y_r - \mathbb{E}[Y_r]| \geq F_0/2^r\right] \leq \frac{\mathbb{V}[Y_r]}{(F_0/2^r)^2} \leq \frac{2^r}{F_0}.$$

Let z be the final index the algo. keeps. So the algo. returns $Z = 2^{z+1/2}$.

Let p be the smallest index such that $2^{p+1/2} \geq 3F_0$. Then

$$\mathbb{P}[Z \geq 3F_0] = \mathbb{P}[z \geq p] = \mathbb{P}[Y_p > 0] \leq \frac{F_0}{2^p} \leq \frac{\sqrt{2}}{3}.$$

Proof of the theorem (2/2)

We have proved $\mathbb{E}[Y_r] = F_0/2^r$, and $\mathbb{V}[Y_r] \leq F_0/2^r$.

By Markov's inequality, we have

$$\mathbb{P}[Y_r > 0] = \mathbb{P}[Y_r \geq 1] \leq \frac{\mathbb{E}[Y_r]}{1} = \frac{F_0}{2^r}.$$

By Chebyshev's inequality, we have

$$\mathbb{P}[Y_r = 0] \leq \mathbb{P}\left[|Y_r - \mathbb{E}[Y_r]| \geq F_0/2^r\right] \leq \frac{\mathbb{V}[Y_r]}{(F_0/2^r)^2} \leq \frac{2^r}{F_0}.$$

Let z be the final index the algo. keeps. So the algo. returns $Z = 2^{z+1/2}$.

Let p be the smallest index such that $2^{p+1/2} \geq 3F_0$. Then

$$\mathbb{P}[Z \geq 3F_0] = \mathbb{P}[z \geq p] = \mathbb{P}[Y_p > 0] \leq \frac{F_0}{2^p} \leq \frac{\sqrt{2}}{3}.$$

Let q be the largest index such that $2^{q+1/2} \leq F_0/3$. Then

$$\mathbb{P}[Z \leq F_0/3] = \mathbb{P}[z \leq q] = \mathbb{P}[Y_{q+1} = 0] \leq \frac{2^{q+1}}{F_0} \leq \frac{\sqrt{2}}{3}.$$

Outline

- Recall: Pairwise independent hashing
- AMS algorithm
- BJKST algorithm
- Chernoff Bound

Another algorithm to approximate F_0

Another algorithm to approximate F_0

1: Choose a random pairwise independent hash function $h : [n] \to [n]$

Another algorithm to approximate F_0

Another algorithm to approximate F_0

1: Choose a random pairwise independent hash function $h : [n] \to [n]$
2: $z = 0$

Another algorithm to approximate F_0

Another algorithm to approximate F_0

1: Choose a random pairwise independent hash function $h : [n] \to [n]$
2: $z = 0$
3: $B = \emptyset$

Another algorithm to approximate F_0

Another algorithm to approximate F_0

1: Choose a random pairwise independent hash function $h : [n] \to [n]$
2: $z = 0$
3: $B = \emptyset$
4: **while** item x from stream $\mathcal{S}$ arrives

Another algorithm to approximate F_0

Another algorithm to approximate F_0

Choose a random pairwise independent hash function $h : [n] \to [n]$
$z = 0$
$B = \emptyset$
while item x from stream $\mathcal{S}$ arrives
    if $\rho(h(x)) \geq z$, then

Another algorithm to approximate F_0

ANOTHER ALGORITHM TO APPROXIMATE F_0

Choose a random pairwise independent hash function $h : [n] \to [n]$
$z = 0$
$B = \emptyset$
while item x from stream $\mathcal{S}$ arrives
    if $\rho(h(x)) \geq z$, then
        $B = B \cup \{(x, \rho(h(x)))\}$

Another algorithm to approximate F_0

```
Choose a random pairwise independent hash function h : [n] → [n]
z = 0
B = ∅
while item x from stream S arrives
    if ρ(h(x)) ≥ z, then
        B = B ∪ {(x, ρ(h(x)))}
        while |B| ≥ 100/ε^2
```

Another algorithm to approximate F_0

Another algorithm to approximate F_0

Choose a random pairwise independent hash function $h : [n] \to [n]$
$z = 0$
$B = \emptyset$
while item x from stream $\mathcal{S}$ arrives
    if $\rho(h(x)) \geq z$, then
        $B = B \cup \{(x, \rho(h(x)))\}$
        while $|B| \geq 100/\varepsilon^2$
            $z = z + 1$

Another algorithm to approximate F_0

ANOTHER ALGORITHM TO APPROXIMATE F_0

Choose a random pairwise independent hash function $h : [n] \to [n]$

$z = 0$

$B = \emptyset$

while item x from stream $\mathcal{S}$ arrives

  if $\rho(h(x)) \geq z$, then

    $B = B \cup \{(x, \rho(h(x)))\}$

    while $|B| \geq 100/\varepsilon^2$

      $z = z + 1$

      Shrink B by removing all $(x, \rho(x))$ with $\rho(h(x)) < z$

Another algorithm to approximate F_0

Another algorithm to approximate F_0

Choose a random pairwise independent hash function $h : [n] \to [n]$

$z = 0$

$B = \emptyset$

while item x from stream $\mathcal{S}$ arrives

  if $\rho(h(x)) \geq z$, then

    $B = B \cup \{(x, \rho(h(x)))\}$

    while $|B| \geq 100/\varepsilon^2$

      $z = z + 1$

      Shrink B by removing all $(x, \rho(x))$ with $\rho(h(x)) < z$

return $|B| \cdot 2^z$

Another algorithm to approximate F_0

ANOTHER ALGORITHM TO APPROXIMATE F_0

```
Choose a random pairwise independent hash function h : [n] → [n]
z = 0
B = ∅
while item x from stream S arrives
    if ρ(h(x)) ≥ z, then
        B = B ∪ {(x, ρ(h(x)))}
        while |B| ≥ 100/ε^2
            z = z + 1
            Shrink B by removing all (x, ρ(x)) with ρ(h(x)) < z
return |B| · 2^z
```

THEOREM

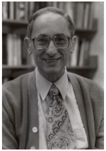
The medium of the returned values from $\Theta(\log(1/\delta))$ independent copies of the algorithm above gives an (ε, δ)-approximation of F_0.

Outline

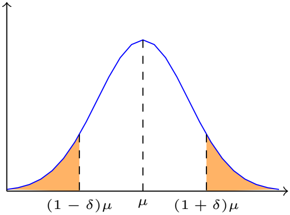
- Recall: Pairwise independent hashing
- AMS algorithm
- BJKST algorithm
- Chernoff Bound

Chernoff Bounds

- Chernoffs bounds are "strong" bounds on the tail probabilities of sums of independent random variables (random variables can be discrete or continuous)
- usually these bounds decrease exponentially as opposed to a polynomial decrease in Markov's or Chebysheff's inequality (see example later)
- have found various applications in:
  - Approximation and Sampling Algorithms
  - Learning Theory (e.g., PAC-learning)
  - Statistics



Hermann Chernoff (1923-)



A simple Chernoff Bound for uniform coin flips

Uniform Chernoff Bound

Let $X_1, \ldots, X_n$ be independent random variables with $\mathbb{P}[X_i = 1] = \mathbb{P}[X_i = -1] = 1/2$. Let $X := \sum_{i=1}^{n} X_i$. Then for any $\lambda > 0$,

$$\mathbb{P}[X \geq \lambda] \leq \mathrm{e}^{-\lambda^2/(2n)}.$$

A simple Chernoff Bound for uniform coin flips

Uniform Chernoff Bound

Let $X_1, \ldots, X_n$ be independent random variables with $\mathbb{P}[X_i = 1] = \mathbb{P}[X_i = -1] = 1/2$. Let $X := \sum_{i=1}^{n} X_i$. Then for any $\lambda > 0$,

$$\mathbb{P}[X \geq \lambda] \leq \mathrm{e}^{-\lambda^2/(2n)}.$$

- This is a simple yet important setting, since r.v.'s are identical and symmetric.
- Bound on $\mathbb{P}[X \leq -\lambda]$ follows by symmetry.
- Bounds for the case $\mathbb{P}[X_i = 1] = \mathbb{P}[X_i = 0] = 1/2$ through substitution, see below.

A simple Chernoff Bound for uniform coin flips

Uniform Chernoff Bound

Let $X_1, \ldots, X_n$ be independent random variables with $\mathbb{P}[X_i = 1] = \mathbb{P}[X_i = -1] = 1/2$. Let $X := \sum_{i=1}^{n} X_i$. Then for any $\lambda > 0$,

$$\mathbb{P}[X \geq \lambda] \leq \mathrm{e}^{-\lambda^2/(2n)}.$$

- This is a simple yet important setting, since r.v.'s are identical and symmetric.
- Bound on $\mathbb{P}[X \leq -\lambda]$ follows by symmetry.
- Bounds for the case $\mathbb{P}[X_i = 1] = \mathbb{P}[X_i = 0] = 1/2$ through substitution, see below.

Corollary (Homework)

Let $Y_1, \ldots, Y_n$ be independent random variables with $\mathbb{P}[Y_i = 0] = \mathbb{P}[Y_i = 1] = 1/2$. Let $Y := \sum_{i=1}^{n} Y_i$ and $\mu := \mathbb{E}[Y] = n/2$. Then for any $0 < \lambda < \mu$,

$$\mathbb{P}[Y \geq \mu + \lambda] \leq \mathrm{e}^{-2\lambda^2/n}.$$

Example: Repeated uniform coin flips

Consider 100 independent coin flips. We wish to find an upper bound on the probability that the number of heads is greater or equal than 75.

Example: Repeated uniform coin flips

Consider 100 independent coin flips. We wish to find an upper bound on the probability that the number of heads is greater or equal than 75.

- Markov's inequality: $X = \sum_{i=1}^{100} X_i$, $X_i \in \{0,1\}$ and $\mathbb{E}[X] = 100 \cdot \frac{1}{2} = 50$.

$$\mathbb{P}[X \geq 3/2 \cdot \mathbb{E}[X]] \leq 2/3 = 0.666.$$

Example: Repeated uniform coin flips

Consider 100 independent coin flips. We wish to find an upper bound on the probability that the number of heads is greater or equal than 75.

- Markov's inequality: $X = \sum_{i=1}^{100} X_i$, $X_i \in \{0, 1\}$ and $\mathbb{E}[X] = 100 \cdot \frac{1}{2} = 50$.

$$\mathbb{P}[X \geq 3/2 \cdot \mathbb{E}[X]] \leq 2/3 = 0.666.$$

- Chebyshev's inequality: $\mathbb{V}[X] = \sum_{i=1}^{100} \mathbb{V}[X_i] = 100 \cdot (1/4) = 25$.

$$\mathbb{P}[|X - \mu| \geq t] \leq \frac{\mathbb{V}[X]}{t^2},$$

  and plugging in $t = 25$ gives an upper bound of $25/25^2 = 1/25 = 0.04$, much better than what we obtained by Markov's inequality.

Example: Repeated uniform coin flips

Consider 100 independent coin flips. We wish to find an upper bound on the probability that the number of heads is greater or equal than 75.

- Markov's inequality: $X = \sum_{i=1}^{100} X_i$, $X_i \in \{0,1\}$ and $\mathbb{E}[X] = 100 \cdot \frac{1}{2} = 50$.

$$\mathbb{P}[X \geq 3/2 \cdot \mathbb{E}[X]] \leq 2/3 = 0.666.$$

- Chebyshev's inequality: $\mathbb{V}[X] = \sum_{i=1}^{100} \mathbb{V}[X_i] = 100 \cdot (1/4) = 25$.

$$\mathbb{P}[|X - \mu| \geq t] \leq \frac{\mathbb{V}[X]}{t^2},$$

  and plugging in $t = 25$ gives an upper bound of $25/25^2 = 1/25 = 0.04$, much better than what we obtained by Markov's inequality.

- The uniform Chernoff bound (Corollary) with $\mu = 50, \lambda = 25$ gives:

$$\mathbb{P}[X \geq \mu + \lambda] \leq \mathrm{e}^{-2\lambda^2/100} = \mathrm{e}^{-625/50} = \mathrm{e}^{-12.5} = 0.00000372\ldots.$$

Example: Repeated uniform coin flips

Consider 100 independent coin flips. We wish to find an upper bound on the probability that the number of heads is greater or equal than 75.

- Markov's inequality: $X = \sum_{i=1}^{100} X_i$, $X_i \in \{0,1\}$ and $\mathbb{E}[X] = 100 \cdot \frac{1}{2} = 50$.

$$\mathbb{P}[X \geq 3/2 \cdot \mathbb{E}[X]] \leq 2/3 = 0.666.$$

- Chebyshev's inequality: $\mathbb{V}[X] = \sum_{i=1}^{100} \mathbb{V}[X_i] = 100 \cdot (1/4) = 25$.

$$\mathbb{P}[|X-\mu| \geq t] \leq \frac{\mathbb{V}[X]}{t^2},$$

  and plugging in $t = 25$ gives an upper bound of $25/25^2 = 1/25 = 0.04$, much better than what we obtained by Markov's inequality.

- The uniform Chernoff bound (Corollary) with $\mu = 50, \lambda = 25$ gives:

$$\mathbb{P}[X \geq \mu + \lambda] \leq \mathrm{e}^{-2\lambda^2/100} = \mathrm{e}^{-625/50} = \mathrm{e}^{-12.5} = 0.00000372\ldots.$$

- the exact probability is $0.00000028\ldots$, so the Chernoff bound overestimates the actual probability by a factor of ≈ 10.

Example: Repeated uniform coin flips

Consider 100 independent coin flips. We wish to find an upper bound on the probability that the number of heads is greater or equal than 75.

- Markov's inequality: $X = \sum_{i=1}^{100} X_i$, $X_i \in \{0,1\}$ and $\mathbb{E}[X] = 100 \cdot \frac{1}{2} = 50$.

$$\mathbb{P}[X \geq 3/2 \cdot \mathbb{E}[X]] \leq 2/3 = 0.666.$$

- Chebyshev's inequality: $\mathbb{V}[X] = \sum_{i=1}^{100} \mathbb{V}[X_i] = 100 \cdot (1/4) = 25$.

$$\mathbb{P}[|X - \mu| \geq t] \leq \frac{\mathbb{V}[X]}{t^2},$$

  and plugging in $t = 25$ gives an upper bound of $25/25^2 = 1/25 = 0.04$, much better than what we obtained by Markov's inequality.

- The uniform Chernoff bound (Corollary) with $\mu = 50, \lambda = 25$ gives:

$$\mathbb{P}[X \geq \mu + \lambda] \leq \mathrm{e}^{-2\lambda^2/100} = \mathrm{e}^{-625/50} = \mathrm{e}^{-12.5} = 0.00000372\ldots.$$

- the exact probability is $0.00000028\ldots$, so the Chernoff bound overestimates the actual probability by a factor of ≈ 10.

Chernoff bound yields a more accurate result but needs independence!

Chernoff Bounds

Chernoff Bound (Multiplicative Version)

Let $X_1, \ldots, X_n$ be independent random variables with $\mathbb{P}[X_i = 1] = p_i$ and $\mathbb{P}[X_i = 0] = 1 - p_i$ for each i. Let $X := \sum_{i=1}^{n} X_i$ and $\mu := \mathbb{E}[X] = \sum_{i=1}^{n} p_i$. Then:

Chernoff Bounds

Chernoff Bound (Multiplicative Version)

Let $X_1, \ldots, X_n$ be independent random variables with $\mathbb{P}[X_i = 1] = p_i$ and $\mathbb{P}[X_i = 0] = 1 - p_i$ for each i. Let $X := \sum_{i=1}^{n} X_i$ and $\mu := \mathbb{E}[X] = \sum_{i=1}^{n} p_i$. Then:

- For any $\varepsilon \geq 0$,

$$\mathbb{P}[X \geq (1+\varepsilon)\mu] \leq \left(\frac{e^{\varepsilon}}{(1+\varepsilon)^{(1+\varepsilon)}} \right)^{\mu}.$$

$$\mathbb{P}[X \leq (1-\varepsilon)\mu] \leq \left(\frac{e^{1-\varepsilon}}{(1-\varepsilon)^{(1-\varepsilon)}} \right)^{\mu}.$$

Chernoff Bounds

Chernoff Bound (Multiplicative Version)

Let $X_1, \ldots, X_n$ be independent random variables with $\mathbb{P}[X_i = 1] = p_i$ and $\mathbb{P}[X_i = 0] = 1 - p_i$ for each i. Let $X := \sum_{i=1}^{n} X_i$ and $\mu := \mathbb{E}[X] = \sum_{i=1}^{n} p_i$. Then:

- For any $\varepsilon \geq 0$,

$$\mathbb{P}[X \geq (1+\varepsilon)\mu] \leq \left(\frac{e^{\varepsilon}}{(1+\varepsilon)^{(1+\varepsilon)}} \right)^{\mu}.$$

$$\mathbb{P}[X \leq (1-\varepsilon)\mu] \leq \left(\frac{e^{1-\varepsilon}}{(1-\varepsilon)^{(1-\varepsilon)}} \right)^{\mu}.$$

- For any $\varepsilon \in [0,1]$, the inequality can be simplified to

$$\mathbb{P}[|X - \mu| \geq \varepsilon\mu] \leq 2e^{-\mu\varepsilon^2/3}.$$