

Algorithms for Big Data (FALL 25)

Lecture 3

MEDIAN ESTIMATION AND MORRIS COUNTER

ALI VAKILIAN (vakilian@vt.edu)

Mean and Median via Sampling

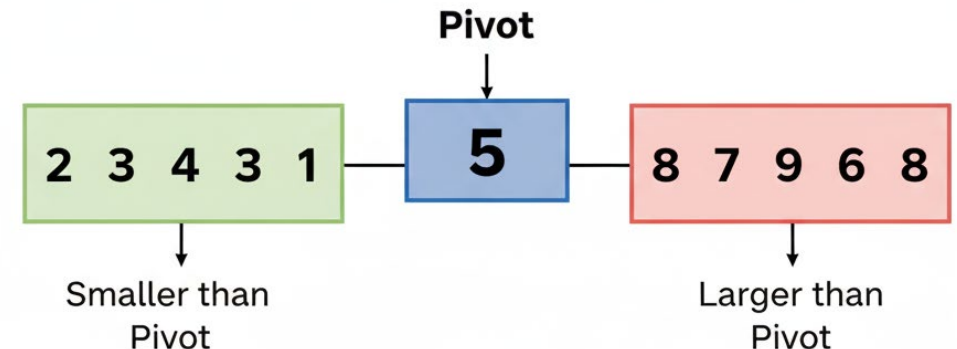
Mean and Median Statistics

- Given a list of n numbers x_1, \dots, x_n :
 - **Mean:** average value = $\sum_{i=1}^n x_i / n$
 - **Median:** the middle number after sorting (if n is even; the average of the two middle ones)

Mean can be computed easily in $O(n)$ time. Similarly, for **Median** (but much more involved).

 How to compute them in streaming setting?

- **Mean** is still easy! What about **Median**?



How to Compute Median in $O(n)$

- Median of Medians algorithm

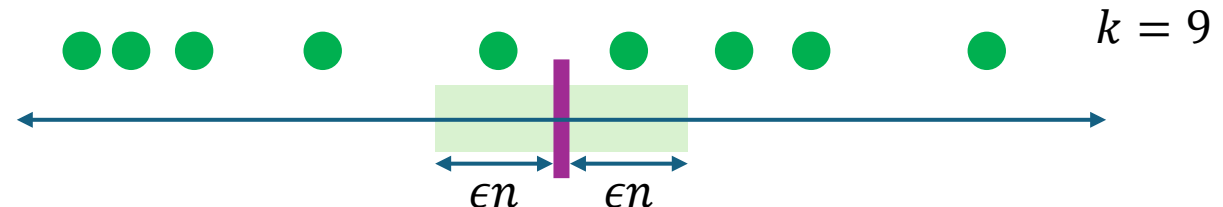
Median Estimation via Sampling

- Sample k elements from the stream (a_1, \dots, a_n) and Let S denote the sampled set.
- Compute the median of S and output it.

Question. How large should we set k to get a reasonable accuracy?

Theorem. If $k = \Omega(\frac{1}{\epsilon^2} \log \frac{1}{\delta})$, then the proposed algorithm outputs an ϵ -approximate median with probability at least $(1 - \delta)$.

When our algorithm fails?



Median Estimation via Sampling

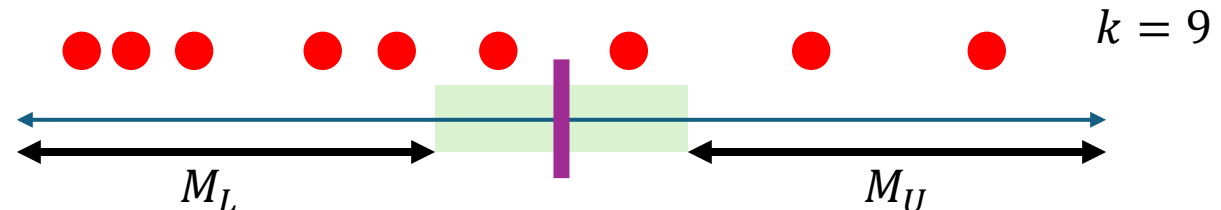
- Sample k elements from the stream (a_1, \dots, a_n) and Let S denote the sampled set.
- Compute the median of S and output it.

Question. How large should we set k to get a reasonable accuracy?

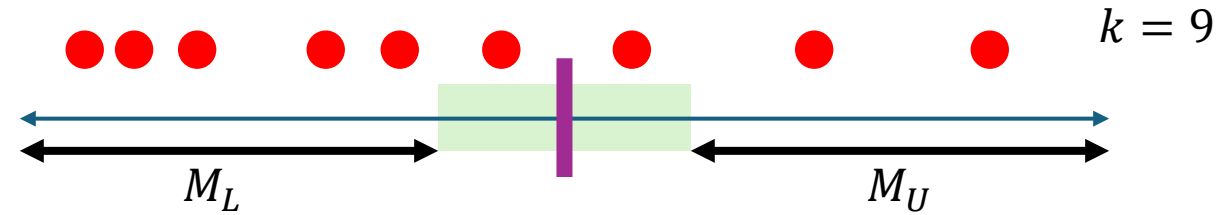
Theorem. If $k = \Omega(\frac{1}{\epsilon^2} \log \frac{1}{\delta})$, then the proposed algorithm outputs an ϵ -approximate median with probability at least $(1 - \delta)$.

When our algorithm fails?

One of M_U or M_L has more than $k/2$ in S



Proof



$$S_L = S \cap M_L \text{ and } S_U = S \cap M_U$$

- $\mathbb{E}[|S_L|] = k(\frac{1}{2} - \epsilon)$
- $\mathbb{E}[|S_U|] = k(\frac{1}{2} - \epsilon)$

Now we need to bound $\Pr[|S_L| > k]$ (and respectively $\Pr[|S_U| > k]$)

$$\begin{aligned} \text{Note that } \Pr\left[|S_L| > \frac{k}{2}\right] &= \Pr[|S_L| - \mathbb{E}[|S_L|] > k\epsilon] \\ &\leq \exp(-k\epsilon^2) \end{aligned}$$

Chernoff-Hoeffding bound



It suffices to set $k \geq \frac{1}{\epsilon^2} \ln(2/\delta)$ so that $\exp(-k\epsilon^2) \leq \delta/2$.

Probabilistic Counting

Estimate the **number of items** in a large dataset w/o storing all the items.

Use cases:

- **Network Traffic Monitoring.** A router or network switch needs to **count the total number of data packets** that pass through it in a specific time window.
- **Web Server Log Analysis.** A large web service like Google or Netflix needs to **count the total number of error log entries** (e.g., HTTP 500 errors) generated across its thousands of servers.
- **Financial Transaction.** A payment processing company like Visa or Stripe needs to **count the total number of transactions** occurring globally.

Probabilistic Counting in Streaming

Setting: monitoring a massive, continuous stream of data.

Input: A data stream $S = (e_1, e_2, e_3, \dots, e_N)$, where N is enormous (billions or trillions of items).

The Goal: Count the number of elements that have appeared in the stream.

Trivial approach requires $O(\log N)$ bits of space.

Questions. Can we do better?

Deterministically, no!

Morris Counter (1978)

- As X gets larger, X increased with a lower probability
- In some sense, we keep track of the $\log N$ (its binary representation) rather than N itself.

Morris Counter (stream):

$X \leftarrow 0$

while an item in stream arrives:

with probability $1/2^X$ **run**

$X \leftarrow X + 1$

return ?

Morris Counter (1978)

- As X gets larger, X increased with a lower probability
- In some sense, we keep track of the $\log N$ (its binary representation) rather than N itself.

What should we show about our designed estimator?

Morris Counter (stream):

$X \leftarrow 0$

while an item in stream arrives:

 with probability $1/2^X$ **run**

$X \leftarrow X + 1$

Return $2^X - 1$

Morris Counter (1978)

- As X gets larger, X increased with a lower probability
- In some sense, we keep track of the $\log N$ (its binary representation) rather than N itself.

What should we show about our designed estimator?

- $Y = 2^X$ has the correct expectation; $\mathbb{E}[Y - 1] = n$ (#events).

Morris Counter (stream):

$X \leftarrow 0$

while an item in stream arrives:

with probability $1/2^X$ **run**

$X \leftarrow X + 1$

Return $2^X - 1$

Motivations (Bell Labs)

- With 8 bits, possible to count to 256
- Managed to count to 130,000.

Expectation Analysis of Morris Counter

- Let X_i be the value of counter after i events, and $Y_i = 2^{X_i}$
- Both are random variables

By Induction (Goal: $Y_n = n + 1$).

- Base case: $n = 0, 1 \implies Y_0 = 1$ and $Y_1 = 2$

$$\begin{aligned}\mathbb{E}[Y_n] &= \mathbb{E}[2^{X_n}] \\ &= \sum_{j=0}^{\infty} 2^j \cdot \Pr[X_n = j] \\ &= \sum_{j=0}^{\infty} 2^j \cdot (\Pr[X_{n-1} = j] \cdot (1 - 2^{-j}) + \Pr[X_{n-1} = j - 1] \cdot 2^{-(j-1)}) \\ &= \sum_{j=0}^{\infty} 2^j \cdot \Pr[X_{n-1} = j] \\ &\quad + \sum_{j=0}^{\infty} (2 \cdot \Pr[X_{n-1} = j - 1] - \Pr[X_{n-1} = j])\end{aligned}$$

Expectation Analysis of Morris Counter (contd.)

$$\begin{aligned}\mathbb{E}[Y_n] &= \mathbb{E}[2^{X_n}] \\&= \sum_{j=0}^{\infty} 2^j \cdot \Pr[X_n = j] \\&= \sum_{j=0}^{\infty} 2^j \cdot (\Pr[X_{n-1} = j] \cdot (1 - 2^{-j}) + \Pr[X_{n-1} = j - 1] \cdot 2^{-(j-1)}) \\&= \sum_{j=0}^{\infty} 2^j \cdot \Pr[X_{n-1} = j] \\&\quad + \sum_{j=0}^{\infty} (2 \cdot \Pr[X_{n-1} = j - 1] - \Pr[X_{n-1} = j]) \\&= \mathbb{E}[Y_{n-1}] \\&\quad + \sum_{j=0}^{\infty} \Pr[X_{n-1} = j] \\&= \mathbb{E}[Y_{n-1}] + 1 = n + 1\end{aligned}$$

Induction
hypothesis

telescoping
series

So far,...

- We showed that output of the Morris counter in **expectation** is equal to n (#events).
- What about the space complexity?

In other words, how large X gets?

We know,

- $Y_n = 2^{X_n}$
- $\mathbb{E}[Y_n] = n + 1$

Morris Counter (stream):

$X \leftarrow 0$

while an item in stream arrives:
with probability $1/2^X$ **run**

$X \leftarrow X + 1$

Return $2^X - 1$

Ideally, we would like to say

$$2^{\mathbb{E}[X_n]} \leq \mathbb{E}[Y_n]$$

So far,...

- We showed that output of the Morris counter in **expectation** is equal to n (#events).
- What about the space complexity?

In other words, how large X gets?

We know,

- $Y_n = 2^{X_n}$
- $\mathbb{E}[Y_n] = n + 1$

Morris Counter (stream):

$X \leftarrow 0$

while an item in stream arrives:
with probability $1/2^X$ **run**

$X \leftarrow X + 1$

Return $2^X - 1$

Ideally, we would like to say

$$2^{\mathbb{E}[X_n]} \leq \mathbb{E}[Y_n]$$

Then,

$$\mathbb{E}[X_n] \leq \log(\mathbb{E}[Y_n]) = \log(n + 1)$$

So far,...

- We showed that output of the Morris counter in **expectation** is equal to n (#events).
- What about the space complexity?

In other words, how large X gets?

We know,

- $Y_n = 2^{X_n}$
- $\mathbb{E}[Y_n] = n + 1$

Morris Counter (stream):

$X \leftarrow 0$

while an item in stream arrives:
 with probability $1/2^X$ **run**

$X \leftarrow X + 1$

Return $2^X - 1$

Ideally, we would like to say

$$2^{\mathbb{E}[X_n]} \leq \mathbb{E}[Y_n]$$

Then,

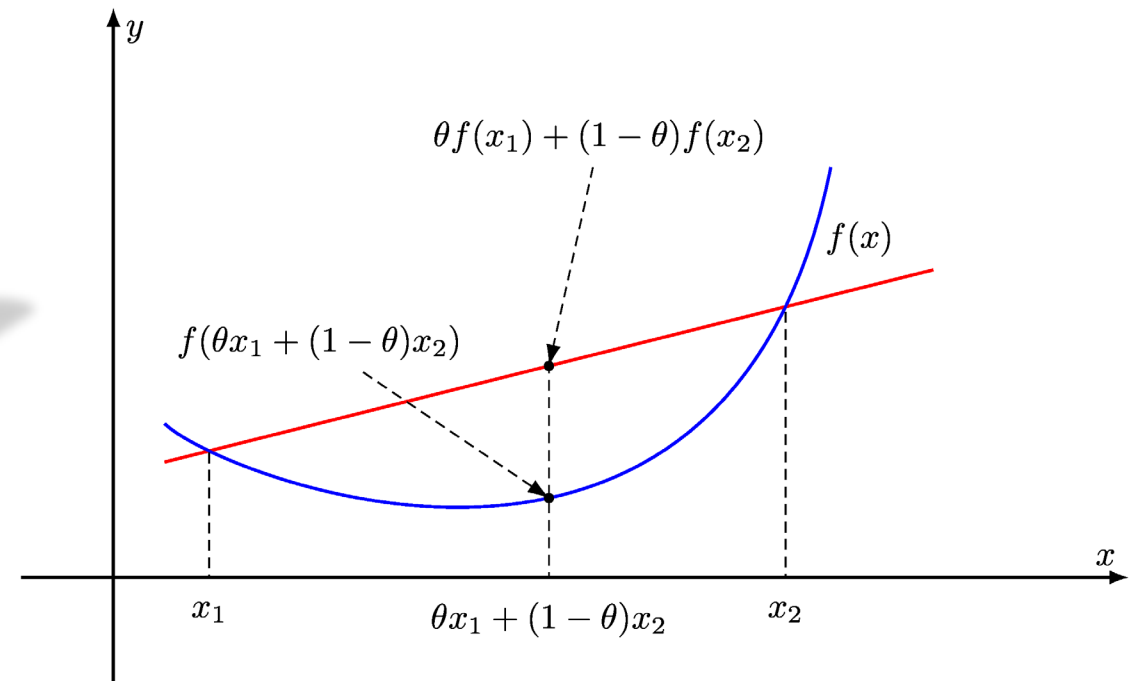
$$\mathbb{E}[X_n] \leq \log(\mathbb{E}[Y_n]) = \log(n + 1)$$

Jensen's Inequality

- $f: \mathbb{R} \rightarrow \mathbb{R}$ is convex if and only if,
 - $f\left(\frac{x_1+x_2}{2}\right) \leq \frac{f(x_1)+f(x_2)}{2}$, for all $x_1, x_2 \in \mathbb{R}$, or equivalently
 - $f(\theta x_1 + (1 - \theta)x_2) \leq \theta f(x_1) + (1 - \theta)f(x_2)$, for all $x_1, x_2 \in \mathbb{R}, 0 \leq \theta \leq 1$

Jensen's Inequality. Let Z be a random variable with $\mathbb{E}[Z] < \infty$. For **convex** f ,

$$f(\mathbb{E}[Z]) \leq \mathbb{E}[f(Z)]$$



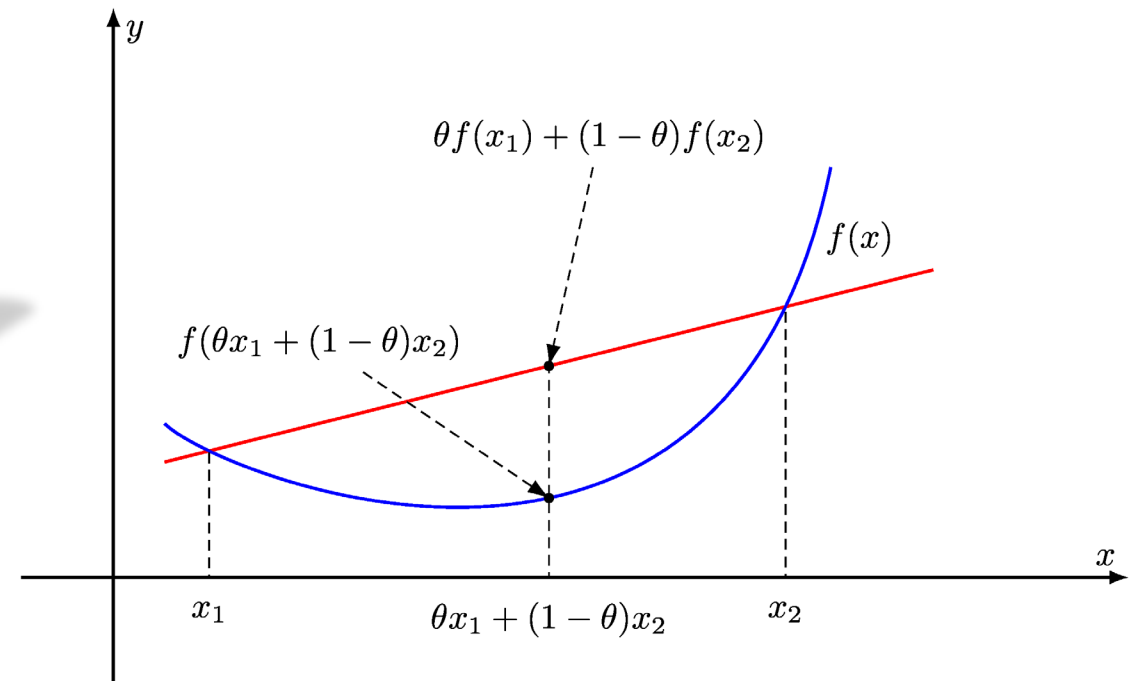
Jensen's Inequality

- $f: \mathbb{R} \rightarrow \mathbb{R}$ is convex if and only if,
 - $f\left(\frac{x_1+x_2}{2}\right) \leq \frac{f(x_1)+f(x_2)}{2}$, for all $x_1, x_2 \in \mathbb{R}$, **or equivalently**
 - $f(\theta x_1 + (1 - \theta)x_2) \leq \theta f(x_1) + (1 - \theta)f(x_2)$, for all $x_1, x_2 \in \mathbb{R}, 0 \leq \theta \leq 1$

Jensen's Inequality. Let Z be a random variable with $\mathbb{E}[Z] < \infty$. For **convex** f ,

$$f(\mathbb{E}[Z]) \leq \mathbb{E}[f(Z)]$$

- $f(x) = 2^x$ is convex.



Jensen's Inequality

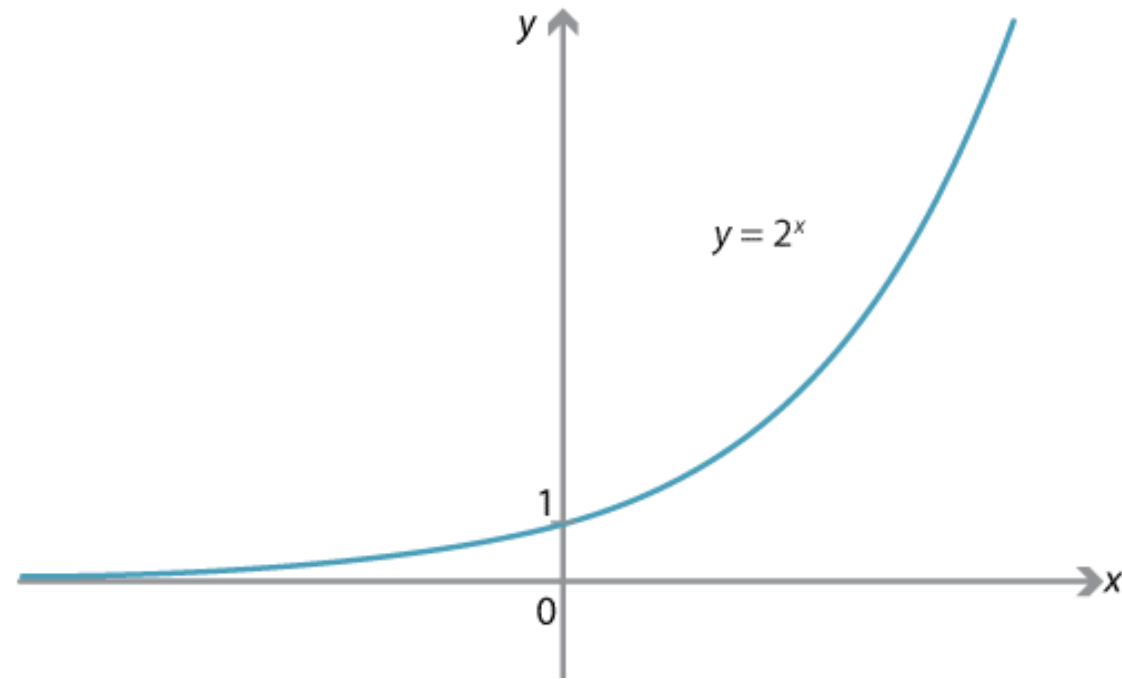
- $f: \mathbb{R} \rightarrow \mathbb{R}$ is convex if and only if,
 - $f\left(\frac{x_1+x_2}{2}\right) \leq \frac{f(x_1)+f(x_2)}{2}$, for all $x_1, x_2 \in \mathbb{R}$, or equivalently
 - $f(\theta x_1 + (1 - \theta)x_2) \leq \theta f(x_1) + (1 - \theta)f(x_2)$, for all $x_1, x_2 \in \mathbb{R}, 0 \leq \theta \leq 1$

Jensen's Inequality. Let Z be a random variable with $\mathbb{E}[Z] < \infty$. For **convex** f ,

$$f(\mathbb{E}[Z]) \leq \mathbb{E}[f(Z)]$$

- $f(x) = 2^x$ is convex.

$$2^{\mathbb{E}[X]} \leq \mathbb{E}[2^X] = \mathbb{E}[Y] \text{ (Morris Counter)}$$



So far,...

- We showed that output of the Morris counter in **expectation** is equal to n (#events).
- What about the space complexity?
In other words, how large X gets?

We know,

- $Y_n = 2^{X_n}$
- $\mathbb{E}[Y_n] = n + 1$
- $2^{\mathbb{E}[X_n]} \leq \mathbb{E}[Y_n] = n + 1$

Morris Counter (stream):

$$X \leftarrow 0$$

Jensen's Inequality. Let Z be a random variable with $\mathbb{E}[Z] < \infty$. For **convex** f ,

$$f(\mathbb{E}[Z]) \leq \mathbb{E}[f(Z)]$$

- Expected number of bits to represent X_n is

$$\begin{aligned} \mathbb{E}[\log X_n] &\leq \log(\mathbb{E}[X_n]) \\ &\leq \log \log(n + 1) \end{aligned}$$

So far,...

- We showed that output of the Morris counter in **expectation** is equal to n (#events). $\mathbb{E}[Y_n] = n + 1$
- We showed that the space complexity of the counter is $\mathbb{E}[X] = O(\log \log n)$

Morris Counter (stream):

$X \leftarrow 0$

while an item in stream arrives:
with probability $1/2^X$ **run**

$X \leftarrow X + 1$

Return $2^X - 1$

How well are they concentrated around their expectation?

What're their variances?

Variance Analysis of Morris Counter

$$\text{Var}[Y_n] = \mathbb{E}[Y_n^2] - \mathbb{E}[Y_n]^2.$$

- We've computed the second term. What about the first term?

Claim) $\mathbb{E}[Y_n^2] = 1.5 n^2 + 1.5 n + 1$

Proof is similar to the analysis of $\mathbb{E}[Y_n]$, via induction.

$$\text{Var}[Y_n] = \mathbb{E}[Y_n^2] - \mathbb{E}[Y_n]^2 = 1.5(n^2 + n) + 1 - (n + 1)^2 = .5(n^2 - n).$$

In particular, $\sigma_{Y_n} = \sqrt{n(n-1)/2} \leq n.$

By Chebyshev's inequality, $\Pr[|Y_n - \mathbb{E}[Y_n]| \geq tn] \leq 1/(2t^2)$

- So, by setting $t = 3/4$, with constant probability, $Y_n = O(n).$

How to get a sharper estimate?

Tighter Estimate for Morris Counter

Goal. For any given $\epsilon > 0$, output a $(1 \pm \epsilon)$ -approximation with probability $(1 - \delta)$ for a given $\delta > 0$.

Estimators, like Morris counter, give expectation. We can further bound their variance to apply Chebyshev. **How to improve these estimators?**

A common technique: **Variance reduction via averaging**

- Run k **independent** copies of the algorithm (Morris counter) in parallel.
 - *Each run uses its own independent random bits.*
- Let $Y^{(1)}, \dots, Y^{(k)}$ be estimators from these k **independent** runs.
- Output $Y_{\text{avg}} = (\sum_{i=1}^k Y^{(i)})/k$

$$\mathbb{E}[Y_{\text{avg}}] = n$$

$$\begin{aligned}\text{Var}[Y_{\text{avg}}] &= \frac{\text{Var}[Y]}{k} \\ &= \frac{n^2 - n}{2k}\end{aligned}$$

Tighter Estimate for Morris Counter (contd.)

A common technique: **Variance reduction via averaging**

- Run k **independent** copies of the algorithm (Morris counter) in parallel.
 - *Each run uses its own independent random bits.*
- Let $Y^{(1)}, \dots, Y^{(k)}$ be estimators from these k **independent** runs.
- Output $Y_{\text{avg}} = (\sum_{i=1}^k Y^{(i)})/k$

$$\mathbb{E}[Y_{\text{avg}}] = n$$

$$\begin{aligned}\text{Var}[Y_{\text{avg}}] &= \frac{\text{Var}[Y]}{k} \\ &= \frac{n^2 - n}{2k}\end{aligned}$$

Set $k = 2/\epsilon^2$ and apply Chebyshev's inequality. Then,

$$\Pr[|Y_{\text{avg}} - \mathbb{E}[Y_{\text{avg}}]| \geq \epsilon n] \leq 1/4$$

How much the space complexity change?

- To run k copies need $O(\frac{1}{\epsilon^2} \log \log n)$ bits for all these counters.

compare to

$$\Pr[|Y_n - \mathbb{E}[Y_n]| \geq tn] \leq 1/(2t^2)$$

Let's Recap

Goal. For any given $\epsilon > 0$, output a $(1 \pm \epsilon)$ -approximation with probability $(1 - \delta)$ for a given $\delta > 0$.

However, what we know is,

$$\Pr[|Y_{\text{avg}} - \mathbb{E}[Y_{\text{avg}}]| \geq \epsilon n] \leq 1/4$$

Simple fix. Set $k = 1/(2\epsilon^2\delta)$, then $\Pr[|Y_{\text{avg}} - \mathbb{E}[Y_{\text{avg}}]| \geq \epsilon n] \leq \delta$

- Now, the space complexity is $O(\frac{1}{\epsilon^2\delta} \log \log n)$

Question. Can we use smaller number of counters and still get δ failure probability.

- Specifically, better dependence on $1/\delta$

Another Technique: Median Trick

Error reduction via median trick

- Run $\ell \times k$ independent copies of the algorithm (Morris counter) in parallel.
 - Each run uses its own independent random bits.
- Let $Y_{\text{avg}}^1, \dots, Y_{\text{avg}}^\ell$ be estimators from these k independent runs.
- Output $Y_{\text{med}} = \text{Median}(Y_{\text{avg}}^1, \dots, Y_{\text{avg}}^\ell)$

This is helpful because enables us to apply Chernoff bound.

Let A_i be the event that estimate (i.e., counter) Y_{avg}^i is bad; i.e.,
$$|Y_{\text{avg}}^i - (n + 1)| > \epsilon n$$

We showed that $\Pr[A_i] \leq 1/4$. Hence, the expected number of bad estimators is $\ell/4$.
When Y_{med} is a bad estimate? There are $\geq \ell/2$ bad estimators.


Using Chernoff, $\Pr[\text{bad median}] \leq 2^{-c\ell}$, for some constant c .

Altogether, ...

- Using **variance reduction** and **median trick**:

Goal. For any given $\epsilon > 0$, output a $(1 \pm \epsilon)$ -approximation with probability $(1 - \delta)$ for a given $\delta > 0$.

Using $O(\frac{1}{\epsilon^2} \log \frac{1}{\delta} \log \log n)$ bits, one can maintain a $(1 \pm \epsilon)$ w.p. at least $1 - \delta$.

 generic scheme
we repeatedly see.



In **HW 1**: how to set k and ℓ to achieve the stated bound?