Algorithm Foundations of Data Science and Engineering

Lecture 1: Probability Inequality and Its Applications

YANHAO WANG

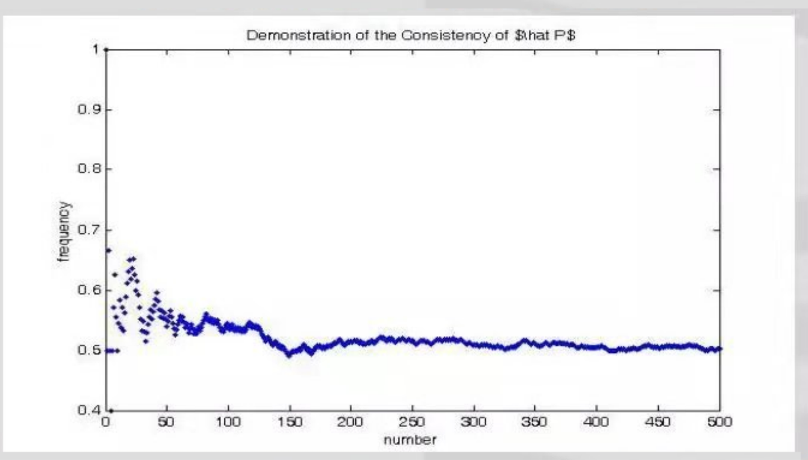
DaSE @ ECNU
(for course related communications)
yhwang@dase.ecnu.edu.cn

Sep. 13, 2021

Outline

Motivated example

Tossing a fair coin



Tail bounds

Question

Consider the experiment of tossing a fair coin n times. What is the probability that the number of heads exceeds $\frac{3n}{4}$.

Tail bounds

Question

Consider the experiment of tossing a fair coin n times. What is the probability that the number of heads exceeds $\frac{3n}{4}$.

Note

The tail bounds of a r.v. X are concerned with the probability that it deviates significantly from its expected value $E(X)$ on a run of the experiment

Markov inequality

Markov inequality

If X is any non-negative r.v. and $0 < a < +\infty$, then

$$P(X > a) \leq \frac{E(X)}{a} \text{ or } P(X > aE(X)) \leq \frac{1}{a}$$

Markov inequality

Markov inequality

If X is any non-negative r.v. and $0 < a < +\infty$, then

$$P(X > a) \leq \frac{E(X)}{a} \text{ or } P(X > aE(X)) \leq \frac{1}{a}$$

Proof

$$P(X > a) = \int_{X>a} dx \leq \int \frac{X}{a} dx = \frac{E(X)}{a} \tag{1}$$

For example,

$$P(X > \frac{3n}{4}) \leq \frac{n/2}{3n/4} = \frac{2}{3} \tag{2}$$

Chebyshev's inequality

Chebyshevs inequality

If r.v. X has mean and variance $\mu = E(X)$ and $\sigma^2 = E[(X-\mu)^2]$, then

$$P(|X-\mu| > a) \leq \frac{\sigma^2}{a^2} \text{ or } P(|X-\mu| > aE(X)) \leq \frac{\sigma^2}{a^2 E(X)^2}$$

Chebyshev's inequality

Chebyshevs inequality

If r.v. X has mean and variance $\mu = E(X)$ and $\sigma^2 = E[(X-\mu)^2]$, then

$$P(|X-\mu| > a) \leq \frac{\sigma^2}{a^2} \text{ or } P(|X-\mu| > aE(X)) \leq \frac{\sigma^2}{a^2 E(X)^2}$$

Proof

Let $Y = |X-\mu|^2$ in Markov's inequality, then

$$P(|X-\mu| > a) = P(Y > a^2) \leq \frac{E(Y)}{a^2} = \frac{\sigma^2}{a^2} \tag{3}$$

For Example,

$$P(X > \frac{3n}{4}) < P(|X - \frac{n}{2}| > \frac{n}{4}) \leq \frac{Var(X)}{(\frac{n}{4})^2} = \frac{4}{n}.$$

That is, if we toss the coin 1000 times, the probability is less than 0.004.

Chernoff bound

Theorem

Let X_i be a sequence of independent Bernoulli r.v.s with $P(X_i = 1) = p_i$. Assume that r.v. $X = \sum_{i=1}^n X_i$.

- $P(X < (1-\delta)\mu) < \left(\frac{e^{-\delta}}{(1-\delta)^{(1-\delta)}}\right)^{\mu}$, where $\mu = \sum_{i=1}^n p_i$
- $P(X < (1-\delta)\mu) < \exp\left(-\mu\delta^2/2\right)$

Proof

For $t > 0$,

$$P(X < (1-\delta)\mu) = P\big(\exp(-tX) > \exp(-t(1-\delta)\mu)\big)$$
$$< \frac{\Pi_{i=1}^n E(\exp(-tX_i))}{\exp(-t(1-\delta)\mu)} (Markov\ inequality)$$

Proof of Chernoff bound Cont.

Proof Cont'd

Since $(1 - x < e^{-x})$, we have

$$E(\exp(-tX_i)) = p_i e^{-t} + (1 - p_i) = 1 - p_i(1 - e^{-t}) < \exp(p_i(e^{-t} - 1))$$

$$\Pi_{i=1}^{n} E(\exp(-tX_i)) < \Pi_{i=1}^{n} \exp(p_i(e^{-t} - 1)) = \exp(\mu(e^{-t} - 1))$$

Proof of Chernoff bound Cont.

Proof Cont'd

Since $(1 - x < e^{-x})$, we have

$$E(\exp(-tX_i)) = p_i e^{-t} + (1 - p_i) = 1 - p_i(1 - e^{-t}) < \exp(p_i(e^{-t} - 1))$$

$$\Pi_{i=1}^{n} E(\exp(-tX_i)) < \Pi_{i=1}^{n} \exp(p_i(e^{-t} - 1)) = \exp(\mu(e^{-t} - 1))$$

Hence,

$$\begin{aligned} P(X < (1 - \delta)\mu) &< \frac{\exp(\mu(e^{-t} - 1))}{\exp(-t(1 - \delta)\mu)} \\ &= \exp(\mu(e^{(-t)} + t - t\delta - 1)) \end{aligned}$$

Proof of Chernoff bound Cont.

Proof Cont'd

Since $(1 - x < e^{-x})$, we have

$$E(\exp(-tX_i)) = p_i e^{-t} + (1 - p_i) = 1 - p_i(1 - e^{-t}) < \exp(p_i(e^{-t} - 1))$$

$\Pi_{i=1}^{n} E(\exp(-tX_i)) < \Pi_{i=1}^{n} \exp(p_i(e^{-t} - 1)) = \exp(\mu(e^{-t} - 1))$

Hence,

$$\begin{aligned} P(X < (1-\delta)\mu) &< \frac{\exp(\mu(e^{-t} - 1))}{\exp(-t(1-\delta)\mu)} \\ &= \exp(\mu(e^{(-t)} + t - t\delta - 1)) \end{aligned}$$

Now its time to choose t to make the bound as tight as possible. Taking the derivative of $\mu(e^{(-t)} + t - t\delta - 1)$ and setting $-e^{(-t)} + 1 - \delta = 0$. We have $t = \ln(1/1 - \delta)$.

$$P(X < (1-\delta)\mu) < \left(\frac{e^{-\delta}}{(1-\delta)^{(1-\delta)}}\right)^{\mu}.$$

Proof of Chernoff bound Cont.

Proof of second statement

To get the simpler form of the bound, we need to get rid of the clumsy term $(1-\delta)^{(1-\delta)}$.

Proof of Chernoff bound Cont.

Proof of second statement

To get the simpler form of the bound, we need to get rid of the clumsy term $(1-\delta)^{(1-\delta)}$.

$$(1-\delta)\ln(1-\delta) = (1-\delta)(\sum_{i=1}^{\infty} -\frac{\delta^i}{i}) > -\delta + \frac{\delta^2}{2}$$

$$(1-\delta)^{(1-\delta)} > \exp(-\delta + \frac{\delta^2}{2})$$

Proof of Chernoff bound Cont.

Proof of second statement

To get the simpler form of the bound, we need to get rid of the clumsy term $(1-\delta)^{(1-\delta)}$.

$$(1-\delta)\ln(1-\delta) = (1-\delta)(\sum_{i=1}^{\infty} -\frac{\delta^i}{i}) > -\delta + \frac{\delta^2}{2}$$

$$(1-\delta)^{(1-\delta)} > \exp(-\delta + \frac{\delta^2}{2})$$

Furthermore,

$$\begin{aligned} P(X < (1-\delta)\mu) &< \left(\frac{e^{-\delta}}{(1-\delta)^{(1-\delta)}}\right)^{\mu} \\ &< \left(\frac{e^{-\delta}}{\exp(-\delta + \frac{\delta^2}{2})}\right)^{\mu} \\ &= \exp(-\mu\delta^2/2) \end{aligned}$$

Chernoff bound (Upper tail)

Theorem for upper tail

Let X_i be a sequence of independent and Bernoulli r.v.s with $P(X_i = 1) = p_i$. Assume that r.v. $X = \sum_{i=1}^{n} X_i$ and $\mu = \sum_{i=1}^{n} p_i$.

- $P(X > (1+\delta)\mu) < \left(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right)^{\mu}$
- $P(X > (1+\delta)\mu) < \exp\left(-\mu\delta^2/4\right)$

Chernoff bound (Upper tail)

Theorem for upper tail

Let X_i be a sequence of independent and Bernoulli r.v.s with $P(X_i = 1) = p_i$. Assume that r.v. $X = \sum_{i=1}^{n} X_i$ and $\mu = \sum_{i=1}^{n} p_i$.

- $P(X > (1+\delta)\mu) < \left(\frac{e^\delta}{(1+\delta)^{(1+\delta)}}\right)^\mu$
- $P(X > (1+\delta)\mu) < \exp\left(-\mu\delta^2/4\right)$

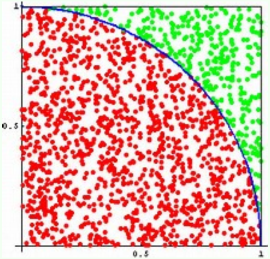
Example

Let X be the number of heads in n tosses of a fair coin, then $\mu = \frac{n}{2}$ and $\delta = \frac{1}{2}$, we have

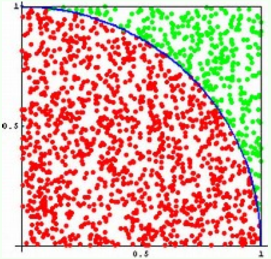
$$P(X > \frac{3n}{4}) = P(X > (1+\frac{1}{2})\frac{n}{2}) < \exp\left(-\frac{n}{2}\delta^2/4\right) = \exp\left(-n/32\right)$$

If we toss the coin 1000 times, the probability is less than $\exp\left(-125/4\right)$.

Why is this algorithm accurate?



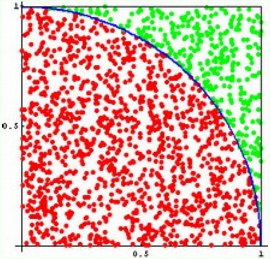
Why is this algorithm accurate?



For this case, sample space $\Omega = \{(x, y) | 0 \leq x, y \leq 1\}$, and $C = \{(x, y) | x^2 + y^2 \leq 1 \wedge x, y \geq 0\}$. Let E be an event that the point locates in the circle area C. Then we have

$$P(E) = \frac{S(C)}{S(\Omega)} = \frac{\pi}{4}.$$

Why is this algorithm accurate?



For this case, sample space $\Omega = \{(x, y)|0 \leq x, y \leq 1\}$, and $C = \{(x, y)|x^2 + y^2 \leq 1 \wedge x, y \geq 0\}$. Let E be an event that the point locates in the circle area C. Then we have

$$P(E) = \frac{S(C)}{S(\Omega)} = \frac{\pi}{4}.$$

Let X_i be a r.v., where $X_i = 1$ means a generated point p_i inside in the circle, otherwise 0, i.e., $X_i = I_C(P_i)$. Hence,

$$E(X_i) = \frac{\pi}{4}, E(\sum_{i=1}^{n} X_i) = \frac{n\pi}{4}, \text{ and } V(\sum_{i=1}^{n} X_i) = \frac{n\pi(4-\pi)}{16}.$$

Why is this algorithm accurate? Cont'd

Chebyshev bound

Hence, we have

$$Y = \frac{\sum_{i=1}^{n} X_i}{n} = \frac{\sum_{i=1}^{n} I_C(P_i)}{n} = \frac{\sum_{i=1}^{n} I_C(P_i)}{\sum_{i=1}^{n} I_C(P_i) + \sum_{i=1}^{n} I_{\Omega - C}(P_i)}.$$

In terms of the Chebyshev bound, we have

Why is this algorithm accurate? Cont'd

Chebyshev bound

Hence, we have

$$Y = \frac{\sum_{i=1}^{n} X_i}{n} = \frac{\sum_{i=1}^{n} I_C(P_i)}{n} = \frac{\sum_{i=1}^{n} I_C(P_i)}{\sum_{i=1}^{n} I_C(P_i) + \sum_{i=1}^{n} I_{\Omega - C}(P_i)}.$$

In terms of the Chebyshev bound, we have

$$P(Y - \frac{\pi}{4} > \frac{\pi}{4}) < P(|X - \frac{n\pi}{4}| > \frac{n\pi}{4})$$
$$\leq \frac{V(X)}{(\frac{n\pi}{4})^2} = \frac{n\pi(4-\pi)}{16}\frac{16}{n^2\pi^2} \approx \frac{1}{4n}$$

Why is this algorithm accurate? Cont'd

Chebyshev bound

Hence, we have

$$Y = \frac{\sum_{i=1}^{n} X_i}{n} = \frac{\sum_{i=1}^{n} I_C(P_i)}{n} = \frac{\sum_{i=1}^{n} I_C(P_i)}{\sum_{i=1}^{n} I_C(P_i) + \sum_{i=1}^{n} I_{\Omega - C}(P_i)}.$$

In terms of the Chebyshev bound, we have

$$P(Y - \frac{\pi}{4} > \frac{\pi}{4}) < P(|X - \frac{n\pi}{4}| > \frac{n\pi}{4})$$
$$\leq \frac{V(X)}{(\frac{n\pi}{4})^2} = \frac{n\pi(4-\pi)}{16}\frac{16}{n^2\pi^2} \approx \frac{1}{4n}$$

When $n = 1000$, the probability of large deviation is less than 0.00025.

Why is this algorithm accurate? Cont'd

Chernoff bound

In terms of the Chernoff bound, we have

Why is this algorithm accurate? Cont'd

Chernoff bound

In terms of the Chernoff bound, we have

$$\begin{aligned} P(Y - \frac{\pi}{4} > \frac{\pi}{4}) &= P(X - \frac{n\pi}{4} > \frac{n\pi}{4}) \\ &= P(X > (1+1)\frac{n\pi}{4}) \\ &< \exp\left(-\frac{n\pi}{4} 1^2/4\right) \\ &= \exp\left(-n\pi/16\right). \end{aligned}$$

Why is this algorithm accurate? Cont'd

Chernoff bound

In terms of the Chernoff bound, we have

$$\begin{aligned} P(Y - \frac{\pi}{4} > \frac{\pi}{4}) &= P(X - \frac{n\pi}{4} > \frac{n\pi}{4}) \\ &= P(X > (1+1)\frac{n\pi}{4}) \\ &< \exp(-\frac{n\pi}{4} 1^2/4) \\ &= \exp(-n\pi/16). \end{aligned}$$

When $n = 1000$, the probability of large deviation is less than $\exp(-750/4)$.

Why is this algorithm accurate? Cont'd

Chernoff bound

In terms of the Chernoff bound, we have

$$\begin{aligned} P(Y - \frac{\pi}{4} > \frac{\pi}{4}) &= P(X - \frac{n\pi}{4} > \frac{n\pi}{4}) \\ &= P(X > (1+1)\frac{n\pi}{4}) \\ &< \exp\left(-\frac{n\pi}{4}1^2/4\right) \\ &= \exp\left(-n\pi/16\right). \end{aligned}$$

When $n = 1000$, the probability of large deviation is less than $\exp(-750/4)$.
Please explain which inequalities give better tail bounds? **Why?**

Counting problem

Problem

Suppose we have a router with internet traffic passing through it. The router receives some information, along with a source and an IP address, and sends it to the corresponding machine.

Counting problem

Problem

Suppose we have a router with internet traffic passing through it. The router receives some information, along with a source and an IP address, and sends it to the corresponding machine.We would like to compute certain statistics with the information passing through the router. For example, we would like to know the number of times a certain IP address makes a request.

Counting problem

Problem

Suppose we have a router with internet traffic passing through it. The router receives some information, along with a source and an IP address, and sends it to the corresponding machine.We would like to compute certain statistics with the information passing through the router. For example, we would like to know the number of times a certain IP address makes a request.

Counting problem

Problem

Suppose we have a router with internet traffic passing through it. The router receives some information, along with a source and an IP address, and sends it to the corresponding machine.We would like to compute certain statistics with the information passing through the router. For example, we would like to know the number of times a certain IP address makes a request.

The amount of information which passes through the router greatly exceeds its available storage. Therefore, we cannot simply store copies of data passing through the router, and then compute based on the stored data.

Counting problem Cont'd

Solution

Many of these tasks are impossible. In order to solve this problems, we relax the guarantees.

Counting problem Cont'd

Solution

Many of these tasks are impossible. In order to solve this problems, we relax the guarantees.Instead of exact computations, we approximate. Instead of requiring these approximations to work all the time, we require they work with large probability.

Counting problem Cont'd

Solution

Many of these tasks are impossible. In order to solve this problems, we relax the guarantees.Instead of exact computations, we approximate. Instead of requiring these approximations to work all the time, we require they work with large probability.
One way to approximate is to say:

$$\text{true answer} \leq \text{output} \leq \alpha \cdot \text{ true answer.}$$

We often think of $\alpha = 1 + \epsilon$, where ϵ is very small. As ϵ becomes smaller, the output becomes closer to the true answer.

Counting problem Cont'd

Solution

Many of these tasks are impossible. In order to solve this problems, we relax the guarantees.Instead of exact computations, we approximate. Instead of requiring these approximations to work all the time, we require they work with large probability.
One way to approximate is to say:

$$\text{true answer} \leq \text{output} \leq \alpha \cdot \text{ true answer.}$$

We often think of $\alpha = 1 + \epsilon$, where ϵ is very small. As ϵ becomes smaller, the output becomes closer to the true answer.
We will require the equation to hold with probability $1 - \delta$. We think of δ as being a very small number, so the probability is close to 1. In other words, the approximation holds often.

Counting problem

Problem definition

The algorithm must monitor a sequence of events, then at any given time output of the number of events thus far. More formally, this is a data structure maintaining a single integer n and supporting the following two operations:

- **update():** increments n by 1;
- **query():** must output (an estimate of) n.

Morris algorithm (Morris 1978)

1: initialize $X \leftarrow 0$;
2: for each update, increment X with probability $\frac{1}{2^X}$;
3: for a query, output $\widehat{n} = 2^X - 1$.

Running example of Morris

input	True	Sampling probability	X	Estimator
	0	1	0	0
1	1	$\frac{1}{2}$	1	1
1	2	$\frac{1}{2}$	1	1
1	3	$\frac{1}{4}$	2	3
1	4	$\frac{1}{4}$	2	3
1	5	$\frac{1}{4}$	2	3
1	6	$\frac{1}{4}$	2	3
1	7	$\frac{1}{8}$	3	7
1	8	$\frac{1}{8}$	3	7

Table: Running example of Morris

Running example of Morris

input	True	Sampling probability	X	Estimator
	0	1	0	0
1	1	$\frac{1}{2}$	1	1
1	2	$\frac{1}{2}$	1	1
1	3	$\frac{1}{4}$	2	3
1	4	$\frac{1}{4}$	2	3
1	5	$\frac{1}{4}$	2	3
1	6	$\frac{1}{4}$	2	3
1	7	$\frac{1}{8}$	3	7
1	8	$\frac{1}{8}$	3	7

Table: Running example of Morris

Regarding this example, do you find any drawbacks of Morris?

Analysis of Morris' algorithm

Analysis

Let X_N denote X in Morris' algorithm after N updates. Then, we have:

$$E2^{X_N} = N + 1$$

Proof

$$\begin{aligned}
E2^{X_{N+1}} &= \sum_{i=0}^{\infty} P(X_N = j) E(2^{X_{N+1}} | X_N = j) \\
&= \sum_{i=0}^{\infty} P(X_N = j)(2^j(1 - \frac{1}{2^j}) + \frac{1}{2^j} 2^{j+1}) \\
&= \sum_{i=0}^{\infty} P(X_N = j) 2^j + \sum_{i=0}^{\infty} P(X_N = j) = E2^{X_N} + 1
\end{aligned}$$

So by induction, we have $E(2^{X_N}) = N + 1$.

Analysis of Morris' algorithm Cont'd

Let X_N denote X in Morris' algorithm after N updates. Then, we have:

$$E2^{2X_N} = O(N^2).$$

Proof

$Var(2^{X_N}) = E((2^{X_N})^2) - (E(2^{X_N}))^2 = E(2^{2X_N}) - (N+1)^2,$

$$E(2^{2X_N}) = \sum_{i\geq 1} 2^{2i} P(X_N = i) = \sum_{i\geq 1} 2^{2i} \Big(\frac{1}{2^{i-1}} P(X_{N-1} = i-1) + (1 - \frac{1}{2^i}) P(X_{N-1} = i) \Big)$$

$$= \sum_{i\geq 1} 2^{i+1} P(X_{N-1} = i-1) + \sum_{i\geq 1} 2^{2i} P(X_{N-1} = i) - \sum_{i\geq 1} 2^{i} P(X_{N-1} = i)$$

$$= 4 \sum_{i\geq 1} 2^{i-1} P(X_{N-1} = i-1) + E(2^{2X_{N-1}}) - \sum_{i\geq 1} 2^{i} P(X_{N-1} = i)$$

$$= E(2^{2X_{N-1}}) + 3 \sum_{i\geq 1} 2^{i} P(X_{N-1} = i) = E(2^{2X_{N-1}}) + 3E(2^{X_{N-1}})$$

Analysis of Morris' algorithm Cont'd

Let X_N denote X in Morris' algorithm after N updates. Then, we have:

$$E2^{2X_N} = O(N^2).$$

Proof

$$Var(2^{X_N}) = E((2^{X_N})^2) - (E(2^{X_N}))^2 = E(2^{2X_N}) - (N+1)^2,$$

$$E(2^{2X_N}) = \sum_{i\geq 1} 2^{2i} P(X_N = i) = \sum_{i\geq 1} 2^{2i}\Big(\frac{1}{2^{i-1}} P(X_{N-1} = i-1) + (1 - \frac{1}{2^i})P(X_{N-1} = i)\Big)$$

$$= \sum_{i\geq 1} 2^{i+1} P(X_{N-1} = i-1) + \sum_{i\geq 1} 2^{2i} P(X_{N-1} = i) - \sum_{i\geq 1} 2^{i} P(X_{N-1} = i)$$

$$= 4\sum_{i\geq 1} 2^{i-1} P(X_{N-1} = i-1) + E(2^{2X_{N-1}}) - \sum_{i\geq 1} 2^{i} P(X_{N-1} = i)$$

$$= E(2^{2X_{N-1}}) + 3\sum_{i\geq 1} 2^{i} P(X_{N-1} = i) = E(2^{2X_{N-1}}) + 3E(2^{X_{N-1}})$$

So by induction, noting that $E(2^{2X_0}) = 1$, it follows that $E(2^{2X_N}) = 3\sum_{i=0}^{i-1} E(2^{X_i}) + 1$.

Analysis of Morris' algorithm Cont'd

Let X_N denote X in Morris' algorithm after N updates. Then, we have:

$$E2^{2X_N} = O(N^2).$$

Proof

$Var(2^{X_N}) = E((2^{X_N})^2) - (E(2^{X_N}))^2 = E(2^{2X_N}) - (N+1)^2,$

$$E(2^{2X_N}) = \sum_{i\geq 1} 2^{2i} P(X_N = i) = \sum_{i\geq 1} 2^{2i}\Big(\frac{1}{2^{i-1}} P(X_{N-1} = i-1) + (1-\frac{1}{2^i})P(X_{N-1} = i)\Big)$$

$$= \sum_{i\geq 1} 2^{i+1} P(X_{N-1} = i-1) + \sum_{i\geq 1} 2^{2i} P(X_{N-1} = i) - \sum_{i\geq 1} 2^{i} P(X_{N-1} = i)$$

$$= 4\sum_{i\geq 1} 2^{i-1} P(X_{N-1} = i-1) + E(2^{2X_{N-1}}) - \sum_{i\geq 1} 2^{i} P(X_{N-1} = i)$$

$$= E(2^{2X_{N-1}}) + 3\sum_{i\geq 1} 2^{i} P(X_{N-1} = i) = E(2^{2X_{N-1}}) + 3E(2^{X_{N-1}})$$

So by induction, noting that $E(2^{2X_0}) = 1$, it follows that $E(2^{2X_N}) = 3\sum_{i=0}^{i-1} E(2^{X_i}) + 1$. Thus, we have $Var(2^{X_N}) = O(N^2)$.

Analysis of Morris' algorithm cont'd

Bound

It is now clear why we output our estimate of n as $\widehat{n} = 2^X - 1$ since it is an unbiased estimator of n.

- Note that $E(\widehat{n} - n)^2 < \frac{1}{2}n^2$.

Analysis of Morris' algorithm cont'd

Bound

It is now clear why we output our estimate of n as $\widehat{n} = 2^X - 1$ since it is an unbiased estimator of n.

- Note that $E(\widehat{n} - n)^2 < \frac{1}{2}n^2$.
- If r.v. X has mean and variance $\mu = E(X)$ and $\sigma^2 = E[(X-\mu)^2]$, then $P(|X - \mu| > a) \leq \frac{\sigma^2}{a^2}$ or $P(|X - \mu| > aE(X)) \leq \frac{\sigma^2}{a^2 E(X)^2}$.

Analysis of Morris' algorithm cont'd

Bound

It is now clear why we output our estimate of n as $\widehat{n} = 2^X - 1$ since it is an unbiased estimator of n.

- Note that $E(\widehat{n} - n)^2 < \frac{1}{2}n^2$.
- If r.v. X has mean and variance $\mu = E(X)$ and $\sigma^2 = E[(X-\mu)^2]$, then $P(|X-\mu| > a) \leq \frac{\sigma^2}{a^2}$ or $P(|X-\mu| > aE(X)) \leq \frac{\sigma^2}{a^2 E(X)^2}$.
- $P(|\widehat{n} - n| > \epsilon n) < \frac{n^2}{2\epsilon^2 n^2} = \frac{1}{2\epsilon^2}$.

Analysis of Morris' algorithm cont'd

Bound

It is now clear why we output our estimate of n as $\widehat{n} = 2^X - 1$ since it is an unbiased estimator of n.

- Note that $E(\widehat{n} - n)^2 < \frac{1}{2}n^2$.
- If r.v. X has mean and variance $\mu = E(X)$ and $\sigma^2 = E[(X-\mu)^2]$, then $P(|X-\mu| > a) \leq \frac{\sigma^2}{a^2}$ or $P(|X-\mu| > aE(X)) \leq \frac{\sigma^2}{a^2 E(X)^2}$.
- $P(|\widehat{n} - n| > \epsilon n) < \frac{n^2}{2\epsilon^2 n^2} = \frac{1}{2\epsilon^2}$.
- It is not particularly meaningful since the right hand side is only better than $\frac{1}{2}$ failure probability when $\epsilon \geq 1$.

Analysis of Morris' algorithm cont'd

Bound

It is now clear why we output our estimate of n as $\widehat{n} = 2^X - 1$ since it is an unbiased estimator of n.

- Note that $E(\widehat{n} - n)^2 < \frac{1}{2}n^2$.
- If r.v. X has mean and variance $\mu = E(X)$ and $\sigma^2 = E[(X-\mu)^2]$, then $P(|X-\mu| > a) \leq \frac{\sigma^2}{a^2}$ or $P(|X-\mu| > aE(X)) \leq \frac{\sigma^2}{a^2 E(X)^2}$.
- $P(|\widehat{n} - n| > \epsilon n) < \frac{n^2}{2\epsilon^2 n^2} = \frac{1}{2\epsilon^2}$.
- It is not particularly meaningful since the right hand side is only better than $\frac{1}{2}$ failure probability when $\epsilon \geq 1$.
- Now we apply the Chebyshev bound. Although we don't have the variance computed exactly, an upper bound on the variance will still give us a Chebyshev bound. Our estimator, $2^X - 1$ is close to its expectation $\pm\sqrt{n^2}$ with high constant probability.

Morris +

Boosting success probability

To decrease the failure probability of Morris' basic algorithm, we need to reduce the variance of $\widehat{n}$. How to do that?

Morris +

Boosting success probability

To decrease the failure probability of Morris' basic algorithm, we need to reduce the variance of $\widehat{n}$. How to do that? We propose the Morris + algorithm

- Run k independent copies of Morris algorithm. Keeping $(X_1, \cdots, X_k)$;
- At the end, output $\frac{1}{k}\sum_{i=1}^{k}(2^{X_i} - 1)$.

Morris +

Boosting success probability

To decrease the failure probability of Morris' basic algorithm, we need to reduce the variance of $\widehat{n}$. How to do that? We propose the Morris + algorithm

- Run k independent copies of Morris algorithm. Keeping $(X_1, \cdots, X_k)$;
- At the end, output $\frac{1}{k}\sum_{i=1}^{k}(2^{X_i} - 1)$.

That is, we obtain independent estimators $\widehat{n}_1, \cdots, \widehat{n}_k$ from independent instantiations of Morris' algorithm, and our output to a query is

$$\widehat{n} = \frac{1}{k}\sum_{i=1}^{k}\widehat{n}_i.$$

Morris +

Boosting success probability

To decrease the failure probability of Morris' basic algorithm, we need to reduce the variance of $\widehat{n}$. How to do that? We propose the Morris + algorithm

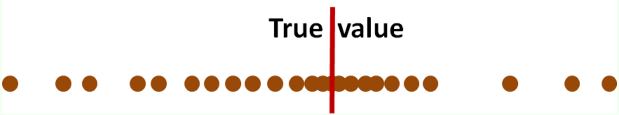
- Run k independent copies of Morris algorithm. Keeping $(X_1, \cdots, X_k)$;
- At the end, output $\frac{1}{k}\sum_{i=1}^{k}(2^{X_i} - 1)$.

That is, we obtain independent estimators $\widehat{n}_1, \cdots, \widehat{n}_k$ from independent instantiations of Morris' algorithm, and our output to a query is

$$\widehat{n} = \frac{1}{k}\sum_{i=1}^{k}\widehat{n}_i.$$

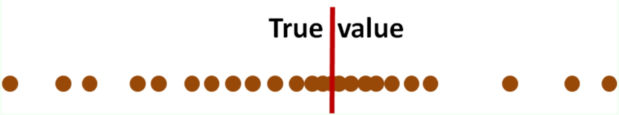
According to the Chebyshev bound, $P(|\widehat{n} - n| > \epsilon n) < \frac{1}{2k\epsilon^2} < \delta$ for $k > \frac{1}{2\epsilon^2\delta}$.

Further boosting the success probability

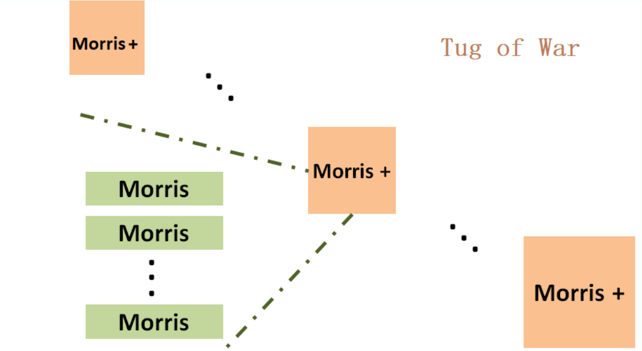


Observing this figure, do you have any idea?

Further boosting the success probability



Observing this figure, do you have any idea?



Morris++

Tug of War

It is a simple technique to reduce the dependence on the failure probability δ from $\frac{1}{\delta}$ to $\log 1/\delta$.

Morris++

Tug of War

It is a simple technique to reduce the dependence on the failure probability δ from $\frac{1}{\delta}$ to $\log 1/\delta$.

Morris++

Tug of War

It is a simple technique to reduce the dependence on the failure probability δ from $\frac{1}{\delta}$ to $\log 1/\delta$.

- We run t instantiations of Morris+, each with failure probability $\frac{1}{3}$,

$$P(|\widehat{n} - n| > \epsilon n) < \frac{1}{2k\epsilon^2} = \frac{1}{3},$$

that is $k = O(1/\epsilon^2)$.

Morris++

Tug of War

It is a simple technique to reduce the dependence on the failure probability δ from $\frac{1}{\delta}$ to $\log 1/\delta$.

- We run t instantiations of Morris+, each with failure probability $\frac{1}{3}$,

$$P(|\widehat{n} - n| > \epsilon n) < \frac{1}{2k\epsilon^2} = \frac{1}{3},$$

  that is $k = O(1/\epsilon^2)$.

- We then output the median estimate from all the t Morris instantiations.

Morris++

Tug of War

It is a simple technique to reduce the dependence on the failure probability δ from $\frac{1}{\delta}$ to $\log 1/\delta$.

- We run t instantiations of Morris+, each with failure probability $\frac{1}{3}$,

$$P(|\widehat{n} - n| > \epsilon n) < \frac{1}{2k\epsilon^2} = \frac{1}{3},$$

  that is $k = O(1/\epsilon^2)$.

- We then output the median estimate from all the t Morris instantiations.

Note that the expected number of Morris+ instantiations that failure is at most $\frac{t}{3}$.

Morris++

Tug of War

It is a simple technique to reduce the dependence on the failure probability δ from $\frac{1}{\delta}$ to $\log 1/\delta$.

- We run t instantiations of Morris+, each with failure probability $\frac{1}{3}$,

$$P(|\widehat{n} - n| > \epsilon n) < \frac{1}{2k\epsilon^2} = \frac{1}{3},$$

  that is $k = O(1/\epsilon^2)$.

- We then output the median estimate from all the t Morris instantiations.

Note that the expected number of Morris+ instantiations that failure is at most $\frac{t}{3}$. If the median to is a bad estimator, at least half the Morris + instantiations can failure, implying the number of failure instantiations deviated from its expectation by at least $\frac{t}{6}$.

Analysis of Morris++

Analysis

Define

$$Y_i = \begin{cases} 1, & \text{if } |\frac{1}{k}\sum_{j=1}^{k}\widehat{n_{ij}} - n| > \epsilon n; \\ 0, & \text{otherwise.} \end{cases}$$

Analysis of Morris++

Analysis

Define

$$Y_i = \begin{cases} 1, & \text{if } |\frac{1}{k}\sum_{j=1}^{k}\widehat{n_{ij}} - n| > \epsilon n; \\ 0, & \text{otherwise.} \end{cases}$$

- For $k = O(1/\epsilon^2)$, we have $P(Y_i = 1) < \frac{1}{3}$.

Analysis of Morris++

Analysis

Define

$$Y_i = \begin{cases} 1, & \text{if } |\frac{1}{k}\sum_{j=1}^{k}\widehat{n_{ij}} - n| > \epsilon n; \\ 0, & \text{otherwise.} \end{cases}$$

- For $k = O(1/\epsilon^2)$, we have $P(Y_i = 1) < \frac{1}{3}$.
- Note that $\mu = E(\sum_i Y_i) \leq \frac{t}{3}$. Then by the Chernoff bound,

$$P(\sum_i Y_i > \frac{t}{2}) \leq P(\sum_i Y_i > (1 + \frac{1}{2})\mu) \leq \exp(-\mu(1/2)^2/4),$$

$$\exp(-t/48) \leq \exp(-\mu(1/2)^2/4) < \delta$$

thus, we have $t = O(\log 1/\delta)$.

Analysis of Morris++

Analysis

Define

$$Y_i = \begin{cases} 1, & \text{if } |\frac{1}{k}\sum_{j=1}^{k}\widehat{n_{ij}} - n| > \epsilon n; \\ 0, & \text{otherwise.} \end{cases}$$

- For $k = O(1/\epsilon^2)$, we have $P(Y_i = 1) < \frac{1}{3}$.
- Note that $\mu = E(\sum_i Y_i) \leq \frac{t}{3}$. Then by the Chernoff bound,

$$P(\sum_i Y_i > \frac{t}{2}) \leq P(\sum_i Y_i > (1+\frac{1}{2})\mu) \leq \exp(-\mu(1/2)^2/4),$$

$$\exp(-t/48) \leq \exp(-\mu(1/2)^2/4) < \delta$$

  thus, we have $t = O(\log 1/\delta)$.
- Finally, we can get an $(\epsilon, \delta)-$approximation in complexity $O(\frac{\log 1/\delta}{\epsilon^2})$.

Take-aways

- Probability inequality
  - Markov inequality
  - Chebyshev inequality
  - Chernoff bound
- Applications
  - Morris
  - Morris +
  - Morris ++