[CS3958: Lecture 1] Introduction, Morris's Algorithm, Concentration

Instructor: Chihao Zhang, Scribed by Yulin Wang

September 24, 2022

1 Introduction to the Course

See topics and references for references of the course. The course mainly consists of four parts.

1.1 Basic Probabilistic Tools

1.1.1 Concentration Inequalities

Concentration inequalities are ubiquitous in computer science. It is mainly used to prove that some random process behaves close to its expectation. Consider the following scenario: Given a (biased) coin which shows HEAD with probability p when tossed. We want to estimate the parameter p. Suppose we toss the coin T times. Let $X_i \sim \texttt{Ber}(p)$ be the indicator of whether the i-th toss gives HEAD, and $\hat{X}$ be the average of $X_1, X_2, \ldots, X_T$. We know that

$\texttt{Ber}(p)$ is a Bernoulli distribution which takes value 1 with probability p and value 0 with probability $1-p$. The expectation of a Bernoulli random variable $X \sim \texttt{Ber}(p)$ is $\mathbf{E}[X] = p$.

$$\mathbf{E}\left[\hat{X}\right] = \frac{1}{T}\sum_{t=1}^{T}\mathbf{E}\left[X_t\right] = p.$$

Concentration inequalities here provide bounds on how $\hat{X}$ deviates from its expectation p, i.e. inequalities of the form

$$\mathbf{Pr}\left[\left|\hat{X} - p\right| \geq \varepsilon\right] < \delta.$$

for parameters $\varepsilon, \delta > 0$.

We will establish several inequalities of this form in the course, and each of them has its own scope of applicability.

1.1.2 Martingale

You have met martingales in the probability course. In this course, we will first study concentration inequalities for martingales, which are important tools to analyze random process. Then we will introduce the optional stopping theorem (OST) for martingales. It is a powerful to argue about processes involving a random stopping condition.

Example 1 *Suppose there is a country in which people only want boys. In the following three scenarios, is the sex ratio of the country* $1:1$

1. *Each family continues to have children until they have a boy.*

2. *Each family continues to have children until there are more boys.*
3. *Each family continues to have children until there are more boys or there are 10 children.*

In fact, in case 1 and 3, the sex ratio is $1:1$*, which can be justified by OST.*

1.2 Optimization

1.2.1 Optimization

We shall study a few first-order optimization algorithms.

1.2.2 Online Optimization

We begin with a classic problem called Multi-Armed Bandit (MAB). It is a simple model for reinforcement learning.

Example 2 (Two-Armed Bandits) *Assume there is a two-armed bandit, and the reward of each arm follows a Bernoulli distribution. Our target is to maximize the expected reward of pulling the bandit for* T *rounds.*

The Explore-then-Commit(ETC) algorithm is a commonly used algorithm for Multi-Armed Bandit. This strategy try to identify the best arm as soon as possible, so it takes the most straightforward way that is trying each arm a few times and picking the one with best empirical reward. In fact, it is sub-optimal. We will introduce some more sophisticated algorithm for the problem.

Next we introduce the general framework of online learning. Assume there is a game that we play T rounds. In each round $t = 1, 2, \ldots, T$, we choose to play x_t and get reward $f_t(x_t)$. The target is

$$\max \sum_{t=1}^{T} f_t(x_t).$$

Here we take x_t from a set $V \in \mathbb{R}^d$, and $f_1, f_2, \ldots, f_T$ are functions from V to $\mathbb{R}$.

Algorithms for offline optimization can be adapted to work in the online setting.

1.3 Markov Chains and Sampling

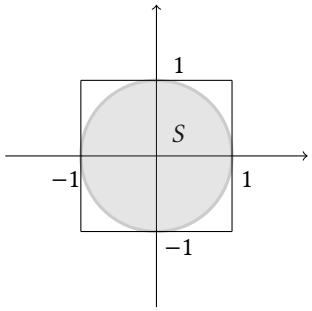
Recall that optimization problem is that

$$\min f(x) \quad \text{s.t.} \quad x \in D.$$

In the task of sample, instead of find a point minimizing $f(x)$, we sample with density proportional to $f(x)$. For example, given $f(x) = e^{-x^2/2}$, this is equivalent to sample from standard Gaussian.

1.3.1 An application of Markov Chain Monte Carlo (MCMC)

An important application of MCMC is to estmate the volume of a set when a membership oricle is given. For example, we want to measure the volume of a figure S in $[-1,1]^d$ space. We can use Monte-Carlo algorithm directly: sample points from $[-1,1]^d$ uniformly and output the ratio of points located in S. However, if S is a sphere at the origin with radius 1, the size of



space $[-1,1]^d$ is exponential in d, while the volume of S is upper bounded by a constant. The probability that a point locating in S is exponentially small and therefore the cost for the Monte-Carlo algorithm to obtain an reasonable estimate is huge.

Sometimes one can use Markov Chain Monte Carlo to construct a more efficient estimator. Informally, instead of sampling from $[-1,1]^d$ directly, we construct a sequence of sets $S_1 \subset S_2 \subset \cdots \subset S_n = S$, and compute the ratio of volume for each pair of adjacent sets, i.e. $\frac{|S_i|}{|S_{i+1}|}$ for $i = 1, 2, \ldots, n-1$. This can be done by constructing a random walk inside each S_i. In this course, we will develop tools to analyze such random walks.

1.3.2 Perspectives for studying MCMC

We shall develop tools from many different perspetives to study MCMC.

Stochastic Process: We consider Markov chains as a stochastic processes, and use probabilistic methods such as martingales and the coupling method to study its behavior.

Spectrum: We will study the spectrum of the transition matrices of Markov chains using linear algebra tools.

Operator: We can more generally view Markov chains as linear operators and analyze them using tools from functional analysis.

2 Concentration

Now we start to study basic concentration inequalities. As a motivating example, let's first look at the streaming model.

2.1 Streaming Model

Example 3 *Suppose we have a router with limited memory, but need to solve some computational tasks with large input data such as monitoring the id of devices visiting it. We can ask the following three natural questions in this scenario,*

- *How many numbers in a given data streaming?*
- *How many distinct numbers?*
- *What is the most frequent number?*

In order to study these problems systematically, we now formally define the streaming model.

In the streaming model, the input is a sequence $\sigma = \langle a_1, a_2, \cdots, a_m \rangle$ where each $a_i \in [n]$. We should notice that the data arrive one by one as suggested by the word "streaming" in the name. We focus the following basic problem:

- How many numbers in the stream (What is m)?

Clearly we can maintain a counter k, and whenever a number a_i arrives, increase k by one. It is not hard to see that we need $\lceil \log_2 m \rceil$ bits of memory.

Can we design a more clever algorithm with only $o(\log m)$ memory? It turns out that computing the exact answer is impossible even with $\lceil \log_2 m \rceil - 1$ memory. The reason is as follows: suppose we have an algorithm $\mathcal{M}$ using only $\lceil \log_2 m \rceil - 1$ memory. Denote $\mathcal{M}(i)$ as the output of the algorithm with a input σ of length i. Then there exists $i, j \in \{m\}$ such that $i \neq j$ while $\mathcal{M}(i) = \mathcal{M}(j)$.

Even though we can not get a better algorithm for the exact answer, it is possible to save lots of memories if approximation is allowed. That is, for every $\varepsilon > 0$, the algorithm computes a number $\widehat{m}$ such that

$$1 - \varepsilon \leq \frac{\widehat{m}}{m} \leq 1 + \varepsilon$$

with high probability.

The Morris' algorithm is presented as Algorithms 1. It is a randomized algorithm. Therefore we look at the expectation of its output.

Theorem 1 *The output of Morris' algorithm $\widehat{m}$ satisfies $\mathbf{E}[\widehat{m}] = m$.*

Proof. We prove it by induction on m. Since $X = 1$ when $m = 1$, we have $\mathbf{E}[\widehat{m}] = 1$. Assume it is true for smaller m, let X_i denote the value of X after

Input: *An instance* $\sigma = \langle a_1, a_2, \cdots, a_m \rangle$ *where each* $a_i \in [n]$.
Output: *The length* m *of the sequence* σ.

1 $X \longleftarrow 0$;
2 *On each input:* $X \longleftarrow X + 1$ with probability 2^{-X};
3 **return** $2^X - 1$

Algorithm 1: Morris' Algorithms for Counting Elements

processing i-th input. We have the following fact,

$$\begin{aligned}
\mathbf{E}\left[\widehat{m}\right] &= \mathbf{E}\left[2^{X_m}\right] - 1 \\
&= \sum_{i=0}^{m} \mathbf{Pr}\left[X_m = i\right] \cdot 2^i - 1 \\
&= \sum_{i=0}^{m} \left(\mathbf{Pr}\left[X_{m-1} = i\right] \cdot \left(1 - 2^{-i}\right) + \mathbf{Pr}\left[X_{m-1} = i-1\right] \cdot 2^{1-i}\right) \cdot 2^i - 1 \\
&= \sum_{i=0}^{m-1} \mathbf{Pr}\left[X_{m-1} = i\right] \cdot \left(2^i + 1\right) - 1 \\
&= \mathbf{E}\left[2^{X_{m-1}}\right] \\
&= m
\end{aligned}$$

where the last equation holds due to the induction hypothesis. □

It it now clear that Morris' algorithm is an unbiased estimator for m and uses approximately $O(\log \log m)$ bits of memory. However, for a practical randomized algorithm, we further require its output to concentrate on the expectation. That is, we want to establish concentration inequality of the form

$$\mathbf{Pr}\left[|\widehat{m} - m| > \varepsilon\right] \leq \delta$$

for $\varepsilon, \delta > 0$. It is natural to see that for fixed ε, the smaller δ is, the better the algorithm is.

How to analyze a Morris' algorithm:

- Analyze its concentration
- Improve the concentration

We will learn more about concentration later to study on this example.

2.2 Concentration Inequalities

We start with Markov inequality.

Theorem 2 *(Markov Inequality) For any* non-negative *random variable* X *and* $a > 0$,

$$\mathbf{Pr}\left[X \geq a\right] \leq \frac{\mathbf{E}\left[X\right]}{a}.$$

Proof. Since X is non-negative, we have

$$\mathbf{E}\left[X\right] \geq a \cdot \mathbf{Pr}\left[X \geq a\right] + 0 \cdot \mathbf{Pr}\left[X < a\right].$$

This is equivalent to

$$\mathbf{Pr}\left[X \geq a\right] \leq \frac{\mathbf{E}\left[X\right]}{a}.$$

□

Example 4 (Coupon Collector) *There are n types of coupons. Each time one draws a coupon whose type is uniformly at random. How many times one needs to draw to collect all n types of coupons in expectation?*

Let X be the number of draws. For each $i = 0, 1, \ldots, n-1$, let X_i be the number of draws to get a new type of coupon while i different types of coupons are already in hand. All these numbers are random variable and clearly $X = \sum_{i=0}^{n-1} X_i$. Then by the linearity of the expectation

$$\mathbf{E}\left[X\right] = \mathbf{E}\left[\sum_{i=0}^{n-1} X_i\right] = \sum_{i=0}^{n-1} \mathbf{E}\left[X_i\right].$$

Note that $X_i \sim \texttt{Geom}\left(\frac{n-i}{n}\right)$ and therefore $\mathbf{E}\left[X_i\right] = \frac{n}{n-i}$. We have

$$\mathbf{E}\left[X\right] = \sum_{i=0}^{n-1} \frac{n}{n-i} = n\left(1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}\right) = nH_n,$$

where $H_n = \sum_{i=1}^{n} \frac{1}{i} \to \log n + \gamma$ when $n \to \infty$ with Euler constant $\gamma = 0.577\ldots$.

Therefore, by Markov inequality,

$$\mathbf{Pr}\left[X \geq c \cdot nH_n\right] \leq \frac{\mathbf{E}\left[X\right]}{c \cdot nH_n} = \frac{1}{c}$$

for any $c > 0$.

The geometric distribution is the probability distribution of the number X of Bernoulli trials needed to get one success, supported on the set {1,2,3,…}. If the Bernoulli trial successes with probability p, then

$$\mathbf{Pr}\left[X = k\right] = (1-p)^{k-1}p$$

One way to obtain sharper concentration inequality is to apply the Markov inequality to $f(|X - \mathbf{E}\left[x\right]|)$ for some f increasing on $\mathbb{R}_{\geq 0}$. This gives

$$\mathbf{Pr}\left[f(|X - \mathbf{E}\left[X\right]|) \geq f(a)\right] \leq \frac{\mathbf{E}\left[f(|X - \mathbf{E}\left[x\right]|)\right]}{f(a)}.$$

If we take $f(x) = x^2$, we obtain Chebyshev's Inequality.

Theorem 3 (Chebyshev's Inequality) *For any random variable with bounded $\mathbf{E}\left[X\right]$ and $a \geq 0$, it holds that*

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

Proof. Let $Y = |X - \mathbf{E}\left[X\right]|$, then clearly $Y \geq 0$. Therefore

$$\begin{aligned}\mathbf{Pr}\left[|X - \mathbf{E}\left[X\right]| \geq a\right] = \mathbf{Pr}\left[Y \geq a\right] = \mathbf{Pr}\left[Y^2 \geq a^2\right] &\leq \frac{\mathbf{E}\left[Y^2\right]}{a^2} \\ &= \frac{\mathbf{E}\left[(X - \mathbf{E}\left[X\right])^2\right]}{a^2} = \frac{\mathbf{Var}\left[X\right]}{a^2}.\end{aligned}$$

□

With these concentration inequalities, let us return to Morris's algorithm. First, we have to compute the variance of $\hat{m}$.

Lemma 4

$$\mathbf{E}\left[\left(2^{X_m}\right)^2\right] = \frac{3}{2}m^2 + \frac{3}{2}m + 1$$

Proof. We can prove the claim using an induction argument similar to our proof for the expectation. When $m = 1, \mathrm{E}\left[\left(2^{X_m}\right)^2\right] = 4$. We assume it is true for smaller m and use the same notation X_i. We have that

$$\begin{aligned}
\mathrm{E}[\widehat{m}] &= \mathrm{E}\left[2^{X_m}\right] - 1 \\
&= \sum_{i=0}^{m} \Pr\left[X_m = i\right] \cdot 2^{2i} \\
&= \sum_{i=0}^{m} \left(\Pr\left[X_{m-1} = i\right]\left(-2^{-i}\right) + \Pr\left[X_{m-1} = i-1\right] \cdot 2^{1-i}\right) \cdot 2^{2i} \\
&= \sum_{i=0}^{m} \left(\Pr\left[X_{m-1} = i\right]\left(2^{2i} - 2^{i}\right) + \Pr\left[X_{m-1} = i-1\right] \cdot 2^{i+1}\right) \\
&= \sum_{i=0}^{m-1} \Pr\left[X_{m-1} = i\right]\left(2^{2i} + 3 \cdot 2^{i}\right) \\
&= \mathrm{E}\left[\left(2^{X_{m-1}}\right)^2\right] + 3\mathrm{E}\left[2^{X_{m-1}}\right] \\
&= \frac{3}{2}m^2 + \frac{3}{2}m + 1
\end{aligned}$$

□

With Lemma 4, we can compute the variance as follows,

$$\mathbf{Var}\left[\widehat{m}\right] = \mathbf{E}\left[\widehat{m}^2\right] - \mathbf{E}\left[\widehat{m}^2\right] = \mathbf{E}\left[\left(2^{X_m} - 1\right)^2\right] - m^2 \leq \frac{m^2}{2}.$$

Applying Chebyshev's inequality, we obtain that for every $\varepsilon > 0$,

$$\mathbf{Pr}\left[|\widehat{m} - m| \geq \varepsilon m\right] \leq \frac{1}{2\varepsilon^2}.$$

However, we observe that as ε becomes smaller, the above bound is not useful. Thus, it is necessary to improve the concentration of the algorithm. We now introduce to common tricks to achieve this.

2.3 The Averaging Trick

The Chebyshev's inequality tells us that we can improve the concentration by reducing the variance. Let's first review some properties of variances. Let X be a random variable, we have

$$\mathbf{Var}\left[a \cdot X\right] = a^2 \cdot \mathbf{Var}\left[X\right],$$

for any constant a. For any two independent random variables X and Y, we have

$$\mathbf{Var}\left[X + Y\right] = \mathbf{Var}\left[X\right] + \mathbf{Var}\left[Y\right].$$

We can design a new algorithm by independently running Morris's algorithm t time in parallel. Denote the corresponding outputs be $\widehat{m}_1, \cdots, \widehat{m}_t$. The final output is

$$\widehat{m}^* := \frac{\sum_{i=1}^{t} \widehat{m}_i}{t}.$$

By the above two properties, we have $\mathbf{Var}\left[\widehat{m}^*\right] = \frac{\mathbf{Var}[\widehat{m}_1]}{t}$.

We can apply Chebyshev's inequality to $\widehat{m}^*$ and obtain that

$$\mathbf{Pr}\left[|\widehat{m}^* - m| \geq \varepsilon m\right] \leq \frac{1}{t \cdot 2\varepsilon^2}.$$

For $t \geq \frac{1}{2\varepsilon^2\delta}$, we have

$$\mathbf{Pr}\left[|\widehat{m}^* - m| \geq \varepsilon m\right] \leq \delta.$$

The new algorithm uses $O\left(\frac{\log\log n}{\varepsilon^2\delta}\right)$ bits of memory. It shows a trade-off between the precision of the randomized algorithm and the consumption of the memory space. We will further improve the bound using the Chernoff bound below.

2.4 *Chernoff Bound*

Like Chebyshev's inequality, if we choose $f(x) = e^{\alpha x}$ for $\alpha > 0$ and apply Markov inequality on $f(X)$, the bound amounts to bound $\mathbf{E}\left[e^{\alpha X}\right]$ which is the *moment generating function* of X. In case $\mathbf{E}\left[e^{\alpha X}\right]$ can be well bounded, we obtain sharp concentration bounds.

Theorem 5 (Chernoff Bound) *Let $X_1, \ldots, X_n$ be independent random variables such that $X_i \sim \mathrm{Ber}(p_i)$ for each $i = 1, 2, \ldots, n$. Let $X = \sum_{i=1}^{n} X_i$ and denote $\mu \triangleq \mathbf{E}[X] = \sum_{i=1}^{n} p_i$, we have*

$$\mathbf{Pr}\left[X \geq (1+\delta)\mu\right] \leq \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu}.$$

If $0 < \delta < 1$, then we have

$$\mathbf{Pr}\left[X \leq (1-\delta)\mu\right] \leq \left(\frac{e^{-\delta}}{(1-\delta)^{1-\delta}}\right)^{\mu}.$$

Proof. We only prove the upper tail bound and the proof of lower tail bound is similar. For every $\alpha > 0$, we have

$$\mathbf{Pr}\left[X \geq (1+\delta)\mu\right] = \mathbf{Pr}\left[e^{\alpha X} \geq e^{\alpha(1+\delta)\mu}\right] \leq \frac{\mathbf{E}\left[e^{\alpha X}\right]}{e^{\alpha(1+\delta)\mu}}.$$

Therefore, we need to estimate the moment generating function $\mathbf{E}\left[e^{\alpha X}\right]$. Since $X = \sum_{i=1}^{n} X_i$ is the sum of independent Bernoulli variables, we have

$$\mathbf{E}\left[e^{\alpha X}\right] = \mathbf{E}\left[e^{\alpha \sum_{i=1}^{n} X_i}\right] = \mathbf{E}\left[\prod_{i=1}^{n} e^{\alpha X_i}\right] = \prod_{i=1}^{n} \mathbf{E}\left[e^{\alpha X_i}\right].$$

Since $X_i \sim \text{Ber}(p_i)$, we can compute $\mathbf{E}\left[e^{\alpha X_i}\right]$ directly:

$$\mathbf{E}\left[e^{\alpha X_i}\right] = p_i e^{\alpha} + (1-p_i) = 1 + (e^{\alpha}-1)p_i \le \exp\left((e^{\alpha}-1)p_i\right).$$

Therefore,

$$\mathbf{E}\left[e^{\alpha X}\right] \le \prod_{i=1}^{n} \exp\left((e^{\alpha}-1)p_i\right) = \exp\left\{\left((e^{\alpha}-1)\sum_{i=1}^{n} p_i\right)\right\} = \exp\{((e^{\alpha}-1)\mu)\}.$$

Therefore,

$$\mathbf{Pr}\left[X \le (1+\delta)\mu\right] \le \frac{\mathbf{E}\left[e^{\alpha x}\right]}{e^{\alpha(1+\delta)\mu}} \le \left(\frac{\exp\{(e^{\alpha}-1)\}}{\exp\{(\alpha(1+\delta))\}}\right)^{\mu}.$$

Note that above holds for any $\alpha > 0$. Therefore, we can choose α so as to minimize $\frac{\exp\{(e^{\alpha}-1)\}}{\exp\{(\alpha(1+\delta))\}}$. To this end, we let

$$\left(\frac{\exp\{(e^{\alpha}-1)\}}{\exp\{(\alpha(1+\delta))\}}\right)' = \exp\{(e^{\alpha}-1-\alpha-\alpha\delta)\cdot(e^{\alpha}-1-\delta)\} = 0.$$

This gives $\alpha = \log(1+\delta)$. Therefore

$$\mathbf{Pr}\left[X \le (1+\delta)\mu\right] \le \left(\frac{\exp\{(e^{\alpha}-1)\}}{\exp\{(\alpha(1+\delta))\}}\right)^{\mu} = \left(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right)^{\mu}.$$

□

The following form of Chernoff bound is more convenient to use (but weaker):

Corollary 6 *For any* $0 < \delta < 1$,

$$\mathbf{Pr}\left[X \ge (1+\delta)\mu\right] \le \exp\left\{\left(-\frac{\delta^2}{3}\mu\right)\right\};$$
$$\mathbf{Pr}\left[X \le (1-\delta)\mu\right] \le \exp\left\{\left(-\frac{\delta^2}{2}\mu\right)\right\}.$$

Proof. We only prove the upper tail. It suffices to verify that for $0 < \delta < 1$, we have

$$\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}} \le \exp\left\{\left(-\frac{\delta^2}{3}\right)\right\}.$$

Taking logarithm of both sides, this is equivalent to

$$\delta - (1+\delta)\ln(1+\delta) \le -\frac{\delta^2}{3}.$$

Let $f(\delta) = \delta - (1+\delta)\ln(1+\delta) + \frac{\delta^2}{3}$ and note that

$$f'(\delta) = -\ln(1+\delta) + \frac{2}{3}\delta, \quad f''(\delta) = -\frac{1}{1+\delta} + \frac{2}{3}.$$

Then for $0 < \delta < 1/2$, $f''(\delta) < 0$, and for $1/2 < \delta < 1$, $f''(\delta) > 0$. Therefore, $f'(\delta)$ first decreases and then increases in $[0,1]$. Also note that $f'(0) = 0$, $f'(1) < 0$ and $f'(\delta) \le 0$ when $0 \le \delta \le 1$. Therefore $f(\delta) \le f(0) = 0$. □

2.5 The Median Trick

We can further boost the performance of Morris's algorithm using the the median trick. We choose $t = \frac{3}{2\varepsilon^2}$ in the algorithm introduced in the averaging trick and independently run it s time in parallel. Denote the outputs as $\widehat{m}_1^*, \widehat{m}_2^*, \cdots, \widehat{m}_s^*$ respectively. It holds that for every $i = 1, \cdots, s$,

$$\mathbf{Pr}\left[\left|\widehat{m}_i^* - m\right| \geq \varepsilon m\right]\right] \leq \frac{1}{3}.$$

At last, we output the median $\widehat{m}^{**}$ of $\widehat{m}_1^*, \widehat{m}_2^*, \cdots, \widehat{m}_s^*$.

Then we can apply the Chernoff bound to analyze the result obtained by the median trick. For every $i = 1, \cdots, s$, we let Y_i be the indicator of the (good) event that

$$\left|\widehat{m}_i^* - m\right| < \varepsilon \cdot m.$$

Then $Y \triangleq \sum_{i=1}^{s} Y_i$ satisfies $\mathbf{E}[Y] \geq \frac{2}{3}s$. If the median $\widehat{m}^{**}$ is bad (namely $|\widehat{m}^{**} - m| \geq \varepsilon \cdot m$), then at least half of $\widehat{m}^*$'s are bad. Equivalently, $Y \leq \frac{1}{2}s$. By Chernoff bound,

$$\mathbf{Pr}\left[|Y - \mathbf{E}[Y]| \geq \frac{1}{6}s\right] \leq 2\exp\left(-\frac{s}{72}\right).$$

Therefore, for $t = O\left(\frac{1}{\varepsilon^2}\right)$ and $s = O\left(\log \frac{1}{\delta}\right)$, we have

$$\mathbf{Pr}\left[|\widehat{m}^{**} - m| \geq \varepsilon m\right] \leq \delta.$$

This new algorithm uses $O(\frac{1}{\varepsilon^2} \cdot \log \frac{1}{\delta} \cdot \log\log n)$ bits of memory.