CSE 290A: Randomized Algorithms
Lecture 2: Concentration Inequalities

Lecturer: C. Seshadhri — University of California, Santa Cruz
Scribe: Sabyasachi Basu (sbasu3@ucsc.edu) — April 2, 2020

---

1 Types of Randomised Algorithms

The previous lecture focused on two popular randomised algorithms: randomised quciksort, and Karger's min-cut algorithm. However, even though they're both randomised algorithms, there's a fundamental difference between the two of them. The former, gave an algorithm that was always correct, but could possibly have bad running time; however, in expectation, this running time is satisfactory. In comparison, for the other, the runtime bound is concrete but it does not guarantee a correct outcome. However, it is correct with possibly high probability. The former is a *Las Vegas* algorithm, while the latter is called a *Monte Carlo* algorithm.

Our choice of the kind of algorithm depends on the scenario: in cases where even a single error is catastrophic, a Las Vegas algorithm is preferred. To convert a Monte Carlo algorithm into a Las Vegas one, a key step is to verify the correctness of solutions produced by the Monte Carlo algorithm; the time taken for this gives us a bound on the runtime of the Las Vegas algorithm; this can be hard. However, it is also worth noting that given a Las Vegas algorithm, we can always convert it to a Monte Carlo algorithm as well. This process involves using concentration inequalities. In today's lecture, we will be surveying some of the more popular concentration bounds.

2 Concentration Inequalities

What is a concentration inequality, and how do they help us here ?
A concentration inequality provides us a bound on the nature of a random variable, and how likely it is to deviate from the expectation. The deviation is often measured in terms of number of standard deviations. A good concentration bound is one where the probability drops drastically as the random variable moves away from the expectation.
Let us take the example of randomised quicksort (**RQS**). We have a nice bound on the expected runtime of the algorithm, $\mathbb{E}[T_{\mathbf{RQS}}] < cn \lg n$, where c is a constant and $T_{\mathbf{RQS}}$ is the running time of **RQS**. So, using concentration bounds, we can bound the probability of the runtime being larger than $cn \lg n$.

Theorem 2.1 (Markov's Inequality)**.** *For a non negative random variable X, for $k \geq 0$, $Pr[X \geq k\mathbb{E}[X]] \leq 1/k$.*

We use the aforementioned idea, in combination with Markov's inequality to get the following algorithm:

Algorithm 1 FixedRQSort(A)

for *d times* **do**
  Run **RQS(A)**
  if runtime exceeds $2cn \lg n$, stop and run the next iteration;
  else return output;
end
If all d runs fail, return FAIL;

As one can see, the algorithm has a runtime of $\Omega(n \lg n)$. Each of the d runs is independent, so we can say that

$$\Pr[\text{Failure of FixedRQSort}] = \Pr[T_{\mathbf{RQS}} \geq \mathbb{E}[T_{\mathbf{RQS}}]]^d = \frac{1}{2^d}. \tag{1}$$

This result is significant, because we increase the running time by just a linear factor, but we get an exponential bound on the error probability. However, the question remains if we can do better?

Can we analyse RQSort directly? Our goal in the following sections is to get a better understanding of when a random variable behaves like its expectation (i.e. takes values close to it) with high probability. To put it more generally, we would like to say that

$$\Pr[T_{\mathbf{RQS}} \leq k\mathbb{E}[T_{\mathbf{RQS}}]] \geq 1 - f(k), \tag{2}$$

where $f(k)$ is a decreasing function of k. Observe that Markov's inequality as shown in 2.1 does the same thing (flip the sign of comparisons to get the exact form in (2)). However, while this is apparently a decent bound, we can do better if we have some structure on the nature of the random variable.

Theorem 2.2 (Chebyshev's Inequality)**.** *Let X be an integrable random variable with finite mean and variance, $\mu = \mathbb{E}[X]$ and σ^2 respectively. Then $Pr[|X - \mu| \geq k\sigma] \leq 1/k^2$.*

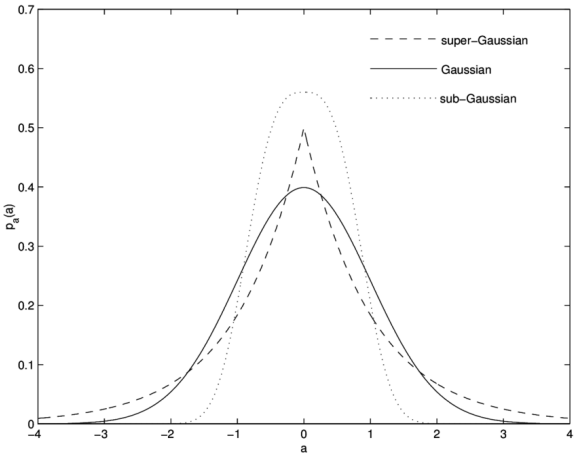
Observe that for this stronger bound, we needed a stronger condition on the random variable, that is its first and second moments must be finite. In a similar nature, we can get even sharper concentrations with more "structure" on the random variable. For instance, if we are to consider several i.i.d. random variables $X_1, \ldots, X_n$ and define the random variable $X = \sum_i X_i$, then the central limit theorem tells us that $X \xrightarrow{d} \mathcal{N}(0, 1)$. Then, upto certain constants,

$$\Pr[|X - \mu| > k\sigma] \leq c'.e^{-k^2}. \tag{3}$$

A very simple calculation for $k = 10$ shows us that Chebyshev gives us an error bound of the order of 10^{-2}, whereas the latter gives us something of the order of 10^{-44}.
Thus, it is instructive to, as a rule of thumb, consider this bound to understand random variables that can be expressed as a sum of (typically independent) random variables.

A random variable that exhibits this property is said to be *sub-Gaussian*. In this case, most of the "mass" of the distribution exists within a few standard deviations from the mean. The upper tail (the tail that extends to $+\infty$) and the lower tail (likewise for $-\infty$) are both below the corresponding tails for a Gaussian asymptotically; hence the name. Likewise, we can define a *super Gaussian*. Figure 1 has a helpful illustrative diagram showing the three kinds. A very preliminary concentration inequality that builds on this idea is the Höffing bound, which is a special case of another such tool called the Azuma-Höffding inequality.

Figure 1: Examples of Gaussian, Subgaussian, and Supergaussian distributions, taken from `https://arxiv.org/abs/1603.03089`.



Theorem 2.3 (Höffding Bound)**.** *Let* $X_i|_{i=1}^n \in [0,1]$ *be independent random variables,* $X = \sum_{i=1}^n X_i$*. Then, for some* $t > 0$*,* $Pr[|X - \mathbb{E}[X]| \geq t] \leq 2\exp(-t^2/n)$.

Because the X_i are all independent, $\text{Var}(X) = \sum_{i=1}^n \text{Var}(X_i) \leq \sum_{i=1}^n \mathbb{E}[X_i^2] \leq n$. This implies that $\sigma(X) \leq \sqrt{n}$. Then, we can express t as $t \geq (t/\sqrt{n}).\sigma$. Thus, 2.3 can be translated into a bound for how likely the random variable is to deviate from the mean by $t/\sqrt{n}$ standard deviations. Using the intuition in the previous paragraph, we can then say that

Pr[Deviating from the mean by a distance of greater than $(t/\sqrt{n})\sigma] \leq c'.\exp(-t^2/n)$.

3 Population estimates

Suppose that in a group of people of size n, there is a smaller community of size b. We wish to estimate b by random sampling. Let a sample of size s be created by sampling s uniformly at random people independently, with replacement, of whom b_s belong to the aforementioned subcommunity. A natural estimate for the size of the subcommunity is then $b_s/s \times n$. The following theorem talks about how accurate this bound is:

Theorem 3.1. *For any* $\alpha > 0$ *and* $\delta > 0$*, we can estimate* b *upto additive error* αn *with probability greater than* $1 - \delta$ *using* $s = \ln(2/\delta)/\alpha^2$*. In other words,*

$$Pr\left[\left|\frac{b_s}{s} \times n - b\right| > \alpha n\right] < \delta. \tag{4}$$

Here, αn is the accuracy and δ is the confidence. It is interesting to note that the size of the sample is in no way related to the size of the entire population n. It does however depend on both α and δ. It is also worth noting here that to increase the accuracy, one would want a smaller α, which is expensive as the sample size is inversely proportional to α^2. However, it is much cheaper to increase confidence, which can be done by simply decreasing δ, which is not as bad. We now prove this theorem.

Proof. Let X_i be the indicator random variable for the i-th sample to be in the subcommunity. Since the total number of these is b_s, it follows that $X = \sum_{i=1}^{s} X_i = b_s$. We now observe that for each i, $\mathbb{E}[X_i] = b/n$. Then, by linearity of expectation we have that $\mathbb{E}[X] = sb/n$. Then, applying 2.3, we have that

$$\Pr[|X - \mathbb{E}[X]| > \alpha s] \leq 2 \exp\left(-\frac{\alpha^2 s^2}{s}\right). \tag{5}$$

Plugging in $s = \ln(2/\delta)/\alpha^2$ into (5), we see that the probability is actually at most δ. However, at first glance, it might appear that this is not exactly what we wanted, because this tells us about b_s, the deviation of which we don't apparently care about. However, upon closer inspection, we can see that if we multiply the terms by n/s and plug in the values for X and $\mathbb{E}[X]$, we arrive at

$$\delta > \Pr\left[\frac{n}{s} \times |X - \mathbb{E}[X]| > \frac{n}{s} \times \alpha s\right] = \Pr\left[\left|\frac{b_s}{s} \times n - b\right| > \alpha n\right], \tag{6}$$

which is the statement of the theorem. Thus our choice of s was good enough to obtain this bound. □

Despite the fact that we allow replacement, the fact that the trials are independent allows the process to go through without hiccups. It is useful to note that:

1. We can accurately express the quantity of interest as a random variable that is the sum of some other random variables, which allows us to use the Höffding bound
2. We allow additive error bounds on αn,
3. X that has expectation proportional to s/n (this is facilitated by the uniform at random sampling).

The last two allow the sample size (s) to be free of n.

3.1 How Tight is this?

The first obvious question is: how tight is this bound obtained? Do we actually need the quadratic dependence on α, or can we say something stronger? Can it depend on, some $\alpha^{2-\epsilon}$? **Yes, the quadratic dependence on α is necessary because of some bad regimes**, although it is not required in some subregimes.

To understand this, let us restate the problem. Imagine that we were instead asked to estimate the probability of getting heads in some coin, and our 'sample' here is the number of flips we must do to make an accurate estimate. Consider s independent coin flips, and the indicator random variables X_i that indicate when the i-th flip is a head. Assume that the 'true' probability is $p = b/n$; relating back to the previous problem, this n and b again correspond to analogues for the sizes of the population and the specific community of interest. Then this is a binomial distribution $\text{Bin}(s, p)$. The number of heads you get here in s flips corresponds to the b_s, i.e. the size of the subcommunity in the previous theorem.

Suppose the true probability were $p = 0.5$. Then what is the probability of seeing $s/2+\alpha s$ heads? This is

$$f(\alpha) = \frac{1}{2^s}\binom{s}{s/2+\alpha s}. \tag{7}$$

Let us introduce a change of notation: $s = 2N$, $\alpha s = k$. Define the new function

$$g(k) = \left(\frac{1}{2}\right)^{2N} \frac{2N!}{(N+k)!(N-k)!}. \tag{8}$$

If we use Stirling's approximation of the form $n! = \sqrt{2\pi n}e^{n\ln n - n}$, then, with some arduous calculation and approximations, key among which are the fact that N is large and k is small, we arrive at the expression

$$g(k) = \frac{1}{\sqrt{\pi N}}e^{-k^2/N} \Rightarrow f(\alpha) = \sqrt{\frac{2}{\pi s}}e^{-\phi(\alpha)s}, \tag{9}$$

where $\phi(\alpha) = \Theta(\alpha^2)$. The exact constants do not matter, and the analysis follows through irrespective of it. In the extreme case where $\alpha = 0$, $f(\alpha) = \Theta(1/\sqrt{s})$. On the other hand, if you have $\alpha = o(1/\sqrt{s})$, then $e^{-\phi(\alpha)s} = \Theta(1)$, which means that even in this case, $f(\alpha) = \Theta(1/\sqrt{s})$. This translates into the fact that for such values of α, you are likely to see more or less the same number of heads. To write this in the language of probabilities, we should say that

$$\Pr\left(\frac{s}{2} + \Delta(s) \text{ heads}\right) = \Theta\left(\Pr\left(\frac{s}{2} \text{ heads}\right)\right) \tag{10}$$

for all $\Delta(s) = o(\sqrt{s})$. For such Δ, if you get $s/2 + \Delta(s)$ heads, you estimate the bias to be $p = 0.5 + \Delta'(s)$, where $\Delta'(s) = o(1/\sqrt{s})$. But our true p was equal to 0.5, so we have an error $\alpha = o(1/\sqrt{s})$, and flipping this around this means that if we want our error to be at most alpha, we need $\Omega(1/\alpha^2)$ samples. This analysis holds true for general population estimation problems as well, with possible changes in the values of the parameters; but the quadratic dependence remains.

The next question that comes up then is: if we do have a smaller sample size, how much information can we actually infer from it?

4 Motivation: Multiplicative Chernoff Bound

We repurpose the population estimate problem to a problem of estimating parameters for the binomial distribution. If we fix a p and $s \to \infty$, then the binomial approaches the normal distribution. Typically, the estimates are hardest to do when $p = 0.5$; at the extreme cases, if you either get all or no heads and your estimate is perfect. The following theorem gives a bound for a smallest sample size with linear dependence instead of quadratic.

Theorem 4.1. *For any $\alpha, \delta > 0$, if $p \geq \alpha$, then one can estimate p upto a constant multiplicative factor w.p. greater than $1 - \delta$ using a sample size $O(\log(1/\delta)/\alpha)$. When $p = \Theta(\alpha)$, the estimate is also an additive estimate.*

When $\alpha = 0.5$, this theorem says the following:

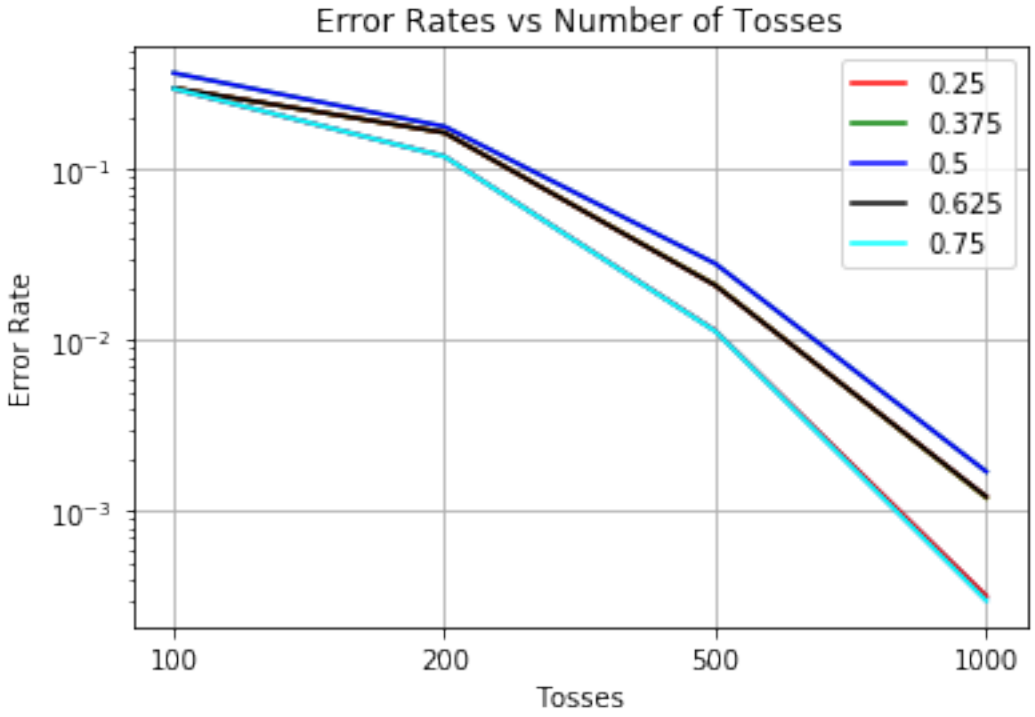
$$\Pr[\text{ Estimate} \in [p/2, 2p]] > 1 - \delta. \tag{11}$$

The case with this α is important because it is at $p = 0.5$ that you have an unbiased coin, making estimation the hardest, and for the same p, this is an additive bound as the theorem states. This can be proved by what is known as the *multiplicative Chernoff bound*, which, like the Höffding inequality holds true not just for binomials but for all random variables that take values between 0 and 1.

5 Illustrative Simulations

Some simulations were carried out in python to show the change in accuracy with an increase in the sample size. Here, we look at the number of flips of a coin with different biases, and an error rate (α) of 0.05, and observe how often the estimate deviates the aforementioned bounds. This probability of deviation beyond the 'accepted' margin is referred to as 'Error Rate' in the log-plot in Figure 2

Figure 2: Error Rate plot



Python Code

```
import numpy as np
import matplotlib.pyplot as plt
n=[100,200,500,1000]
p=[0.25,0.375,0.5,0.625,0.75]
alpha=0.05
s=0
times=4
counts=np.zeros(times)
runs=1000000
col=[‘Red’,‘Green’,‘Blue’,‘Black’,‘Cyan’]
for bias in range(5):
    for k in range(times):
        count=0
        for j in range(runs):
            s = sum(np.random.binomial(n[k], p[bias], 1))
            if abs(s-n[k]*p[bias])>=n[k]*alpha:
                 count+=1
        counts[k]=count/runs
    x=np.arange(times)
    my_xticks = [‘100’,‘200’,‘500’,‘1000’]
    plt.xticks(x, n)
    plt.semilogy(counts)
    plt.plot(x,counts, label=p[bias], color=col[bias])
    plt.grid(True)
    plt.legend()
plt.title(‘Error Rates vs Number of Tosses’)
plt.xlabel(‘Tosses’)
plt.ylabel(‘Error Rate’)
plt.show()
```

Note that in the code, the ’ has been replaced in places with a ‘ for visual purposes. The code will not run if copy-pasted.

What is worth noting in the graph is that the error rate is higher when bias is 0.5, and roughly symmetric about it otherwise (which makes sense as it just flips the chances of heads and tails). Also, as expected, the error rate dips sharply as the number of tosses increases. Upon actual computation, it can be seen that the Höffding bound is actually fairly loose in most of these cases.