COMPSCI 690RA: Randomized Algorithms and Probabilistic Data Analysis

Prof. Cameron Musco

University of Massachusetts Amherst. Spring 2022.

Lecture 2

Logistics

- Reminder that there is a weekly quiz, released after class on Wednesday and due the next Tuesday 8pm.
- Problem Set 1 was released Monday. Due next Friday 2/11. Download from the course website.
- See Piazza for a post to organize homework groups.
- Reminder that we encourage you to post your questions publicly on Piazza – you will receive extra credit for this. And help your classmates!

Talk This Week

Thursday at 4pm Talya Eden (BU, MIT) will be giving a Zoom talk on Sublinear-Time Graph Algorithms: Motif Counting and Uniform Sampling.

- This is a very cool line of work that heavily uses randomization.
- Link on CICS Events page.

`https://umass-amherst.zoom.us/j/94725490374?pwd=bGtsa0hjNGx5c1VyNnlGT21WbU5wQT09`

Summary

Last Time:

- Motivation behind randomized algorithms and some classic examples — polynomial identity testing, Freivald's algorithm.
- Complexity classes related to randomized algorithms – $P \subseteq ZPP \subseteq RP \subseteq BPP$.
- Probability review – linearity of expectation and variance.

Today:

- Concentation bounds – Markov's and Chebyshev's inequalities.
- The union bound.
- Exponential concentration bounds – Chernoff and Bernstein
- Applications of tools to Quicksort analysis, coupon collecting, statistical estimation, random hashing.

Application 1: Quicksort with Random Pivots

`Quicksort`(X): where $X = (x_1, \ldots, x_n)$ is a list of numbers.

1. If X is empty: return X.
2. Else: select pivot p uniformly at random from $\{1, \ldots, n\}$.
3. Let $X_{lo} = \{i \in X : x_i < x_p\}$ and $X_{hi} = \{i \in X : x_i \geq x_p\}$ (requires $n - 1$ comparisons with x_p to determine).
4. Return the concatenation of the lists [`Quicksort`(X_{lo}), (x_p), `Quicksort`(X_{hi})].

4	5	2	8	1	3	6	9	7	0	4	5	2	8	1	3	6

What is the worst case running time of this algorithm?

Randomized Quicksort Analysis

Theorem: Let T be the number of comparisions performed by `Quicksort`(X). Then $\mathbb{E}[\mathsf{T}] = O(n \log n)$.

- For any $i, j \in [n]$ with $i < j$, let $\mathbf{I}_{ij} = 1$ if x_i, x_j are compared at some point during the algorithm, and $\mathbf{I}_{ij} = 0$ if they are not. An indicator random variable.
- We can write $\mathsf{T} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathbf{I}_{ij}$. Thus, via linearity of expectation

$$\mathbb{E}[\mathsf{T}] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathbb{E}[\mathbf{I}_{ij}] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \Pr[x_i, x_j \text{ are compared}]$$

So we need to upper bound $\Pr[x_i, x_j \text{ are compared}]$.

Randomized Quicksort Analysis

Upper bounding $\Pr[x_i, x_j$ **are compared]:**

- Assume without loss of generality that $x_1 \le x_2 \le \ldots \le x_n$. This is just 'renaming' the elements of our list. Also recall that $i < j$.
- At exactly one step of the recursion, x_i, x_j will be 'split up' with one landing in X_{hi} and the other landing in X_{lo}, or one being chosen as the pivot. x_i, x_j are only ever compared in this later case – if one is chosen as the pivot when they are split up.
- The split occurs when some element between x_i and x_j is chosen as the pivot. The possible elements are $x_i, x_{i+1}, \ldots, x_j$.

4	5	2	1	3	0	6	8	9	7

- $\Pr[x_i, x_j$ are compared$]$ is equal to the probability that either x_i or x_j are chosen as the splitting pivot from this list. Thus, $\Pr[x_i, x_j$ are compared$] =$

So Far: Expected number of comparisons is given as:

$$\mathbb{E}[T] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \Pr[x_i, x_j \text{ are compared}].$$

And we computed $Pr[x_i, x_j \text{ are compared}] = \frac{2}{j-i+1}$. Plugging in:

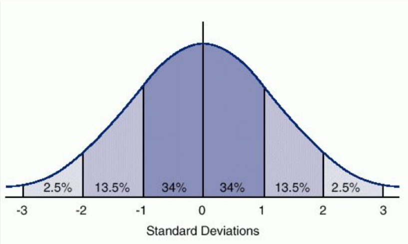
$$\mathbb{E}[T] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j-i+1} = \sum_{i=1}^{n-1} \sum_{k=2}^{n-i+1} \frac{2}{k}$$

$$\leq \sum_{i=1}^{n-1} \sum_{k=1}^{n} \frac{2}{k} \leq 2 \cdot (n-1) \cdot \sum_{k=1}^{n} \frac{1}{k} = 2n \cdot H_n = O(n \log n).$$

Concentration Inequalities

Concentration Inequalities

Concentration inequalities are bounds showing that a random variable lies close to it's expectation with good probability. Key tools in the analysis of randomized algorithms.



Markov's Inequality

The most fundamental concentration bound: **Markov's inequality.**

For any non-negative random variable X and any $t > 0$:

$$\Pr[X \geq t] \leq \frac{\mathbb{E}[X]}{t}.$$

Proof:

$$\begin{aligned}\mathbb{E}[X] = \sum_{s} \Pr(X = u) \cdot u &\geq \sum_{u \geq t} \Pr(X = u) \cdot u \\ &\geq \sum_{u \geq t} \Pr(X = u) \cdot t \\ &= t \cdot \Pr(X \geq t).\end{aligned}$$

Plugging in $t = \mathbb{E}[X] \cdot s$, $\Pr[X \geq s \cdot \mathbb{E}[X]] \leq 1/s$. The larger the deviation s, the smaller the probability.

Think-Pair-Share: You have a Las Vegas algorithm that solves some decision problem in expected running time T. Show how to turn this into a Monte-Carlo algorithm with worst case running time $3T$ and success probability $2/3$.

Chebyshev's inequality

With a very simple twist, Markov's Inequality can be made much more powerful in many settings.

For any random variable X and any value $t > 0$:

$$\Pr(|X| \geq t) = \Pr(X^2 \geq t^2).$$

X^2 is a nonnegative random variable. So can apply Markov's:

$$\Pr(|X| \geq t) = \Pr(X^2 \geq t^2) \leq \frac{\mathbb{E}[X^2]}{t^2}.$$

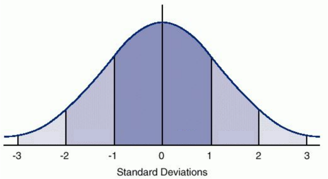
Plugging in the random variable $X - \mathbb{E}[X]$, gives the standard form of **Chebyshev's inequality:**

$$\Pr(|X - \mathbb{E}[X]| \geq t) \leq \frac{\mathbb{E}[(X - \mathbb{E}[X])^2}{t^2} = \frac{\mathrm{Var}(X)}{t^2}.$$

Chebyshev's inequality

$$\Pr(|X - \mathbb{E}[X]| \geq t) \leq \frac{\mathrm{Var}[X]}{t^2}$$

What is the probability that X falls s standard deviations from it's mean?



$$\Pr(|X - \mathbb{E}[X]| \geq s \cdot \sqrt{\mathrm{Var}[X]}) \leq \frac{\mathrm{Var}[X]}{s^2 \cdot \mathrm{Var}[X]} = \frac{1}{s^2}.$$

Application 2: Statistical Estimation + Law of Large Numbers

Concentration of Sample Mean

Theorem: Let $X_1, \ldots, X_n$ be pairwise independent random variables with $\mathbb{E}[X_i] = \mu$ and $\text{Var}[X_i] = \sigma^2$. Let $\bar{X} = \frac{1}{n}\sum_{i=1}^n X_n$ be their sample average.

For any $\epsilon > 0$, $\Pr[|\bar{X} - \mu| \geq \epsilon\sigma] \leq \frac{1}{n\epsilon^2}$.

- By linearity of expectation, $\mathbb{E}[\bar{X}] = \frac{1}{n}\sum_{i=1}^n \mathbb{E}[X_i] = \mu$.
- By linearity of variance, $\mathbb{E}[\bar{X}] = \frac{1}{n^2}\sum_{i=1}^n \text{Var}[X_i] = \frac{\sigma^2}{n}$.
- Plugging into Chebyshev's inequality:

$$\Pr[|\bar{X} - \mu| \geq \epsilon\sigma] \leq \frac{\text{Var}[\bar{X}]}{\epsilon^2\sigma^2} = \frac{1}{n\epsilon^2}.$$

This is the weak law of large numbers.

Application to statistical estimation: There is a large population of individuals. A p fraction of them have a certain property (e.g., 55% of people support decreased taxation, 10% of people are greater than 6' tall, etc.). Want to estimate p from a small sample of individuals.

- Sample n individuals uniformly at random, with replacement.
- Let $X_i = 1$ if the i^{th} individual has the property, and 0 otherwise. $X_1, \ldots, X_n$ are i.i.d. draws from $Bern(p)$ – each is 1 with probability p and 0 with probability $1 - p$.

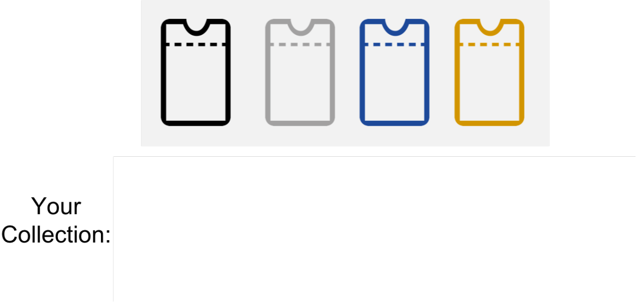
Application to Success Boosting

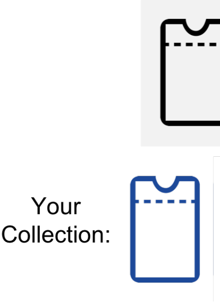
Think-Pair-Share: You have a Monte-Carlo algorithm with worst case running time T and success probability $2/3$. Show how to obtain, for any $\delta \in (0, 1)$, a Monte-Carlo algorithm with worse case running time $O(T/\delta)$ and success probability $1 - \delta$.

Application 3: Coupon Collecting

Coupon Collector Problem

There is a set of n unique coupons. At each step you draw a random coupon from this set. How many steps does it take you to collect all the coupons?





Think-Pair-Share: Say you have collected i coupons so far. Let T_{i+1} denote the number of draws needed to collect the $(i+1)^{st}$ coupon. What is $\mathbb{E}[T_i]$?

Coupon Collector Analysis

Think-Pair-Share: Say you have collected i coupons so far. Let T_{i+1} denote the number of draws needed to collect the $(i+1)^{st}$ coupon. What is $\mathbb{E}[\mathrm{T}_i]$?

- T_i is a geometric random variable with success probability $p_i = \frac{n-i}{n}$. I.e., $Pr[\mathrm{T}_i = j] = p_i(1-p_i)^{j-1}$.
- **Exercise:** verify that $\mathbb{E}[\mathrm{T}_i] = 1/p_i = \frac{n}{n-i}$.
- By linearity of expectation, the expected number of draws to collect all the coupons is:

$$\mathbb{E}[\mathrm{T}] = \sum_{i=0}^{n-1} \mathbb{E}[\mathrm{T}_i] = \frac{n}{n} + \frac{n}{n-1} + \ldots \frac{n}{2} + \ldots \frac{n}{1}$$

$$= n \cdot H_n.$$

- By Markov's inequality, $\Pr[\mathrm{T} \geq cn \cdot H_n] \leq$

Quiz Question

Consider rolling a fair 6-sided dice, which takes a value in $\{1, 2, 3, 4, 5, 6\}$ each with probability $1/6$. What is the expected number of rolls needed to see each odd number (i.e., see each of $\{1, 3, 5\}$) at least once?

Coupon Collector Analysis

Can get a tighter tail bound using Chebyshev's inequality in place of Markov's.

- We wrote $\mathsf{T} = \sum_{i=0}^{n-1} \mathsf{T}_i$, which let us compute $\mathbb{E}[\mathsf{T}] = n \cdot H_n$.
- Also have $\mathrm{Var}[\mathsf{T}] = \sum_{i=0}^{n-1} \mathrm{Var}[\mathsf{T}_i]$. Why?
- **Exercise:** show that $\mathrm{Var}[\mathsf{T}_i] = \frac{1-p_i}{p_i^2}$, and recall that $p_i = \frac{n-i}{n}$.
- Putting these together:

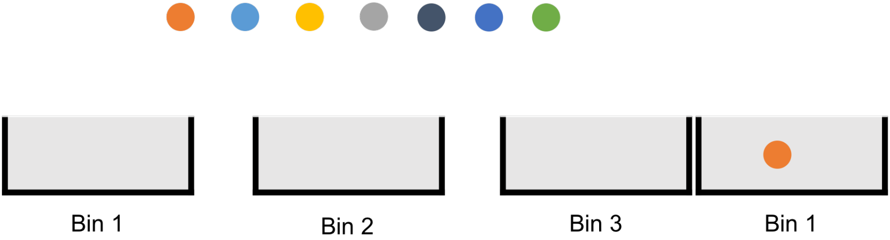
$$\mathrm{Var}[\mathsf{T}] = \sum_{i=0}^{n} \frac{1-p_i}{p_i^2} = \sum_{i=0}^{n} \frac{1}{p_i^2} - \sum_{i=0}^{n} \frac{1}{p_i}$$
$$\leq n^2 \cdot \frac{\pi^2}{6} - n \cdot H_n \leq n^2 \cdot \frac{\pi^2}{6}.$$

- Via Chebyshev's inequality, $\Pr[|\mathsf{T} - n \cdot H_n| \geq cn] \leq$

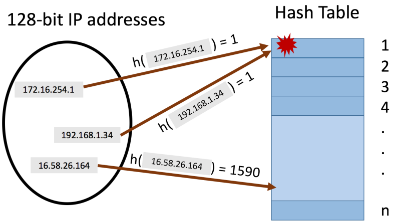
Application 4: Randomized Load Balancing and Hashing, and ‘Ball Into Bins’

Balls Into Bins

I throw m balls independently and uniformly at random into n bins. What is the maximum number of balls any bin?



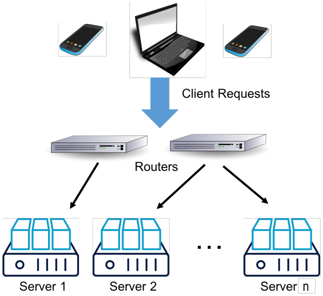
Application: Hash Tables



- **hash function** $h : U \to [n]$ maps elements to indices of an array.
- Repeated elements in the same bucket are stored as a linked list – 'chaining'.
- Worse-case look up time is proportional to the maximum list length – i.e., the maximum number of 'balls' in a 'bin'.

Note: A 'fully random hash function' maps items independently and uniformly at random to buckets. This is a theoretical idealization of practical hash functions.

Application: Randomized Load Balancing



- m requests are distributed randomly to n servers. Want to bound the maximum number of requests that a single server must handle.
- Assignment is often is done via a random hash function so that repeated requests or related requests can be mapped to the same server, to take advantages of caching and other optimizations.

Balls Into Bins Analysis

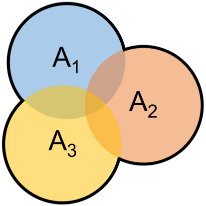
Let $\mathbf{b}_i$ be the number of balls landing in bin i. For n balls into m bins what is $\mathbb{E}[\mathbf{b}_i]$?

$$\Pr\left[\max_{i=1,\ldots,n} \mathbf{b}_i \geq k\right] = \Pr\left[\bigcup_{i=1}^{n} A_i\right],$$

where A_i is the event that $\mathbf{b}_i \geq k$.

Union Bound: For any random events $A_1, A_2, ..., A_n$,

$$\Pr\left(A_1 \cup A_2 \cup \ldots \cup A_n\right) \leq \Pr(A_1) + \Pr(A_2) + \ldots + Pr(A_n).$$



Exercise: Show that the union bound is a special case of Markov's inequality with indicator random variables.

Balls Into Bins Direct Analysis

Let $\mathbf{b}_i$ be the number of balls landing in bin i. If we can prove that for any i, $\Pr[A_i] = \Pr[\mathbf{b}_i \geq k] \leq p$, then by the union bound:

$$\Pr\left[\max_{i=1,\ldots,n} \mathbf{b}_i \geq k\right] = \Pr\left[\bigcup_{i=1}^{n} A_i\right] \leq n \cdot p.$$

Claim 1: Assume $m = n$. For $k \geq \frac{c \ln n}{\ln \ln n}$, $\Pr[\mathbf{b}_i \geq k] \leq \frac{1}{n^{c-o(1)}}$.

- $\mathbf{b}_i$ is a binomial random variable with n draws and success probability $1/n$.

$$\Pr[\mathbf{b}_i = j] = \binom{n}{j} \cdot \frac{1}{n^j} \cdot \left(1 - \frac{1}{n}\right)^{n-j}.$$

- We have $\binom{n}{j} \leq \left(\frac{en}{j}\right)^j$, giving $\Pr[\mathbf{b}_i = j] \leq \left(\frac{e}{j}\right)^j \cdot (1 - \frac{1}{n})^{n-j} \leq \left(\frac{e}{j}\right)^j$.
- Summing over $j \geq k$ we have:

$$\Pr[\mathbf{b}_i \geq k] \leq \sum_{j \geq k} \left(\frac{e}{j}\right)^j = \left(\frac{e}{k}\right)^k \cdot \frac{1}{1 - e/k}.$$

Balls Into Bins Direct Analysis

We just showed: When $n = m$ (i.e., n balls into n bins)

$$\Pr[\mathbf{b}_i \geq k] \leq \left(\frac{e}{k}\right)^k \cdot \frac{1}{1 - e/k}$$

For $k = \frac{c \ln n}{\ln \ln n}$ we have:

$$\Pr[\mathbf{b}_i \geq k] \leq \left(\frac{\ln \ln n}{\ln n}\right)^{\frac{c \ln n}{\ln \ln n}} \cdot \frac{1}{1 - (e \ln \ln n)/(c \ln n)} = \frac{1}{n^{c - o(1)}}.$$

Upshot: By the union bound, For $k = c\frac{\ln n}{\ln \ln n}$ for sufficiently large c,

$$\Pr\left[\max_{i=1,\ldots,n} \mathbf{b}_i \geq k\right] \leq n \cdot \frac{1}{n^{c-o(1)}} = \frac{1}{n^{c-1-o(1)}}.$$

When throwing n balls in to n bins, with very high probability the maximum number of balls in a bin will be $O\left(\frac{\ln n}{\ln \ln n}\right)$.

Balls Into Bins Via Chebyshev's Inequality

In our balls into bins analysis we directly bound $\Pr[\mathbf{b}_i \geq k] \leq \left(\frac{e}{k}\right)^k \cdot \frac{1}{1-e/k}$.

Think Pair Share: Give an upper bound on this probability using Chebyshev's inequality. Hint: write $\mathbf{b}_i$ as a sum of n indicator random variables and compute $\text{Var}[\mathbf{b}_i]$.

Balls Into Bins Via Chebyshev's Inequality

By Chebyshev's Inequality: $\Pr[\mathbf{b}_i \geq k] \leq \frac{2}{k^2}$.

Setting $k = c\sqrt{n}$, $\Pr\left[\mathbf{b}_i \geq c\sqrt{n}\right] \leq \frac{2}{c^2 n}$. So via a union bound:

$$\Pr\left[\max_{i=1,\ldots,n} \mathbf{b}_i \geq c\sqrt{n}\right] \leq n \cdot \frac{2}{c^2 n} \leq \frac{2}{c^2}.$$

Upshot: Chebyshev's inequality bounds the maximum load by $O(\sqrt{n})$ with good probability, as compared to $O\left(\frac{\log n}{\log \log n}\right)$ for the direct proof. It is quite loose here.

Chebyshev's and Markov's inequalities are extremely valuable because they are very general – require few assumptions on the underlying random variable. But by using assumptions, we can often get tighter analysis.