

CS-GY 6763: Lecture 2

Chebyshev's Inequality, Union Bound, Exponential Tail Bounds

NYU Tandon School of Engineering, Prof. Christopher Musco

NOTE ON MATHEMATICAL PROOFS

It can be hard to know how formal to be. We will try to provide feedback on first problem set for anyone who is either too rigorous or too loose. It's a learning process.

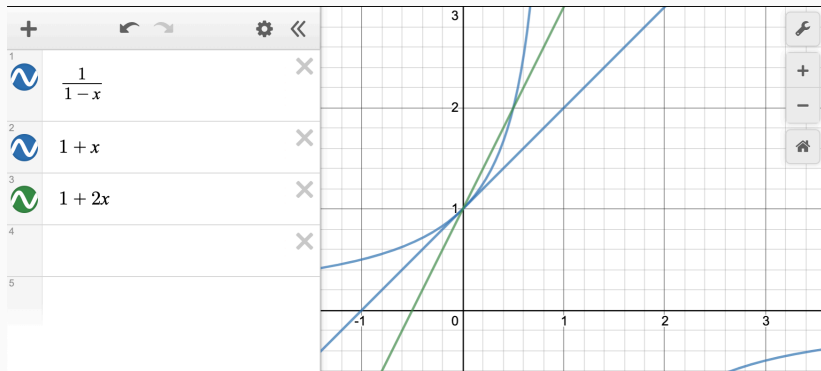
Things that are generally fine:

- Can assume input size n is $> C$ for some constant c . E.g. $n > 2, n > 10$.
- Similarly can assume $\epsilon < c$ for constant c . E.g. $\epsilon < .1, \epsilon < .01$.
- If I write $O(z)$, you are free to choose the constant. E.g., it's fine if your method only works for tables of size $1000 \cdot m^{1.5}$.
- Derivatives, integrals, etc. can be taken from e.g. WolframAlpha without working through steps.
- Basic inequalities can be used without proof, as long as you verify numerically. Don't need to include plot on problem set.

EXAMPLE INEQUALITY

$$1 + \epsilon \leq \frac{1}{1 - \epsilon} \leq 1 + 2\epsilon \text{ for } \epsilon \in [0, .5].$$

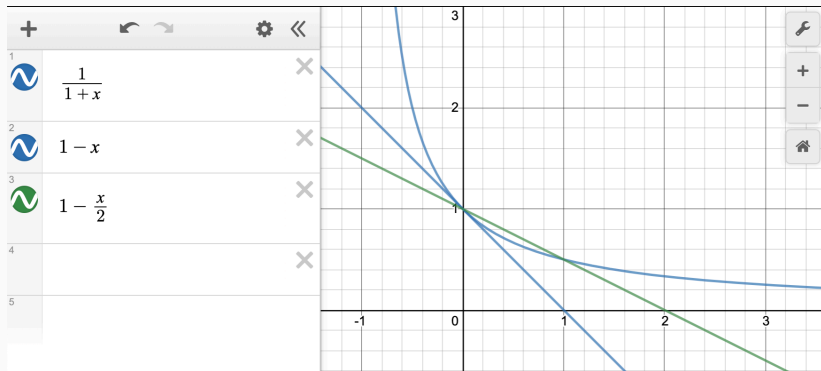
Proof by plotting:



EXAMPLE INEQUALITY

$$1 - \epsilon \leq \frac{1}{1 + \epsilon} \leq 1 + \epsilon \text{ for } \epsilon \in [0, 1].$$

Proof by plotting:



Tip: When confronted with a complex expression, try to simplify by using big-Oh notation, or just rounding things off. Then clean-up your proof after you get to a solution.

Examples:

- $(m - 1) \approx m$
- $\frac{1}{n} - \frac{1}{n^2} \approx \frac{1}{n}$
- $\left(\frac{m-1}{cm^{1.5}}\right)^2 \approx O\left(\frac{1}{m}\right).$
- $\log(n/2) \approx \log(n)$

LECTURE ROAD MAP

Last week we saw the power of Linearity of Expectation + Markov's. This week we will discuss four more tools:

- Linearity of Variance + Chebyshev's Inequality
- Union Bound + Exponential Tail Bounds



These six simple tools combined are surprising powerful and flexible. They form the cornerstone of randomized algorithm design.

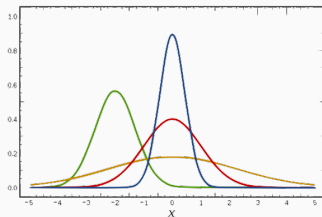
Chebyshev's Inequality

A new concentration inequality:

Lemma (Chebyshev's Inequality)

Let X be a random variable with expectation $\mathbb{E}[X]$ and variance $\sigma^2 = \text{Var}[X]$. Then for any $k > 0$,

$$\Pr[|X - \mathbb{E}[X]| \geq k \cdot \sigma] \leq \frac{1}{k^2}$$



$\sigma = \sqrt{\text{Var}[X]}$ is the standard deviation of X . Intuitively this bound makes sense: it is tighter when σ is smaller.

COMPARISON TO MARKOV'S INEQUALITY

Properties of Chebyshev's inequality:

- **Good:** No requirement of non-negativity. X can be anything.
- **Good:** Two-sided. Bounds the probability that $|X - \mathbb{E}X|$ is large, which means that X isn't too far above or below its expectation. Markov's only bounded probability that X exceeds $\mathbb{E}[X]$.
- **Bad/Good:** Requires a bound on the variance of X .

No hard rule for which to apply! Both Markov's and Chebyshev's are useful in different settings.

PROOF OF CHEBYSHEV'S INEQUALITY

Idea: Apply Markov's inequality to the (non-negative) random variable $S = (X - \mathbb{E}[X])^2$.

Lemma (Chebyshev's Inequality)

Let X be a random variable with expectation $\mathbb{E}[X]$ and variance $\sigma^2 = \text{Var}[X]$. Then for any $k > 0$,

$$\Pr[|X - \mathbb{E}[X]| \geq k \cdot \sigma] \leq \frac{1}{k^2}$$

Markov's inequality: for positive r.v. S , $\Pr[S \geq t] \leq \mathbb{E}[S]/t$.

QUICK EXAMPLE

If I flip a fair coin 100 times, show that with 93% chance I get between 30 and 70 heads?

Let C_1, \dots, C_{100} be independent random variables that are 1 with probability $1/2$, 0 otherwise.

Let $H = \sum_{i=1}^{100} C_i$ be the number of heads that get flipped.

$$\mathbb{E}[H] =$$

$$\text{Var}[H] =$$

Fact: For pairwise independent random variables X_1, \dots, X_m ,

$$\text{Var}[X_1 + X_2 + \dots + X_m] = \text{Var}[X_1] + \text{Var}[X_2] + \dots + \text{Var}[X_m].$$

I.e., we require that for any i, j X_i and X_j are independent.

This is strictly weaker than mutual independence, which requires that for all possible values v_1, \dots, v_k ,

$$\Pr[X_1 = v_1, \dots, X_k = v_k] = \Pr[X_1 = v_1] \cdot \dots \cdot \Pr[X_k = v_k].$$

QUICK EXAMPLE

If I flip a fair coin 100 times, show that with 93% chance I get between 30 and 70 heads?

Let C_1, \dots, C_{100} be independent random variables that are 1 with probability $1/2$, 0 otherwise.

Let $H = \sum_{i=1}^{100} C_i$ be the number of heads that get flipped.

$$\text{Var}[H] = 25.$$

Chebyshev's:

Recall the set up from last lecture:

Draw items x_1, \dots, x_m uniformly at random from a set of size n and count the number of collisions:

$$D = \sum_{\substack{i,j \in \{1, \dots, m\} \\ i < j}} \mathbb{1}[x_i == x_j].$$

We showed that $\mathbb{E}[D] = \binom{m}{2} \frac{1}{n}$.

Exercise: Show that $\text{Var}[D] \leq \binom{m}{2} \frac{1}{n}$ and use to prove the claim on Slide 32 from Lecture 1.

Abstract architecture of a streaming algorithm:

Have massive dataset $D = d_1, \dots, d_n$ with n pieces of data that arrive in a sequential stream. There is far too much data to store or process it in a single location.

- Still want to analyze the data: i.e. fit a model or (approximately) compute some function $f(D)$.
- To do so, we must compress data “on-the-fly”, storing some smaller data structure which still contains interesting information.
- Often can only take a single-pass over the data.

Count-Min was our first example of a streaming algorithm for the (ϵ, k) -frequent items problem.

STREAMING ALGORITHMS IN PRACTICE

Sensor data: GPS or seismometer readings to detect geological anomalies, telescope images, satellite imagery, highway travel time sensors.

Web traffic and data: User data for website, including e.g. click data, web searches and API queries, posts and image uploads on social media.

Training machine learning models: Often done in a streaming setting when training dataset is huge, often with multiple passes.



Lots of software frameworks exist for easy development of streaming algorithms.

DISTINCT ELEMENTS PROBLEM

Input: $d_1, \dots, d_n \in \mathcal{U}$ where \mathcal{U} is a huge universe of items.

Output: Number of distinct inputs.

Example: $f(1, 10, 2, 4, 9, 2, 10, 4) \rightarrow 5$

Applications:

- Distinct users hitting a webpage.
- Distinct values in a database column (e.g. for estimating the size of group by queries)
- Number of distinct queries to a search engine.
- Distinct motifs in DNA sequence.

Implementations widely used at Google (Sawzall, Dremel, PowerDrill), Yahoo, Twitter, Facebook Presto, etc.

DISTINCT ELEMENTS PROBLEM

Input: $d_1, \dots, d_n \in \mathcal{U}$ where \mathcal{U} is a huge universe of items.

Output: Number of distinct inputs.

Example: $f(1, 10, 2, 4, 9, 2, 10, 4) \rightarrow 5$

Flajolet–Martin (simplified):

- Choose random hash function $h : \mathcal{U} \rightarrow [0, 1]$.
- $S = \infty$
- For $i = 1, \dots, n$
 - $S \leftarrow \min(S, h(d_i))$
- Return: $\frac{1}{S} - 1$

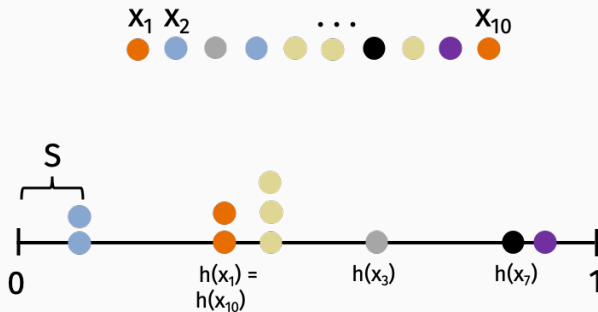
The hash function h maps from \mathcal{U} to a random point in $[0, 1]$?

Hashing to real numbers:

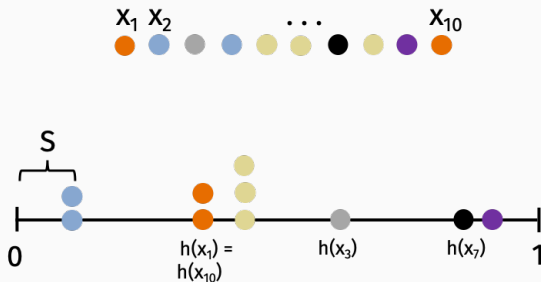
- Impossible to implement $h(x)$ in reality, but you can replace it with $\frac{g(x)}{k}$, where g is a hash function that maps to $\{0, 1, \dots, k\}$ for sufficiently large k .
- All results hold if this “discrete” hash is used instead, but the analysis is simpler if we assume access to h .
- Just like when we assumed uniform random hash functions, this is a useful abstraction which makes understanding and analyzing the underlying algorithms easier.

Flajolet–Martin (simplified):

- Choose random hash function $h : \mathcal{U} \rightarrow [0, 1]$.
- $S = \infty$
- For $i = 1, \dots, n$
 - $S \leftarrow \min(S, h(d_i))$
- Return: $\tilde{D} = \frac{1}{S} - 1$



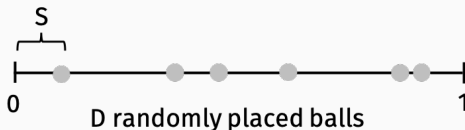
Let D equal the number of distinct elements in our stream.



D unique locations after hashing

Intuition: When D is larger, S will be smaller. Makes sense to return the estimate $\tilde{D} = \frac{1}{S} - 1$.

What is $\mathbb{E}S$?



Let D equal the number of distinct elements in our stream.

Lemma

$$\mathbb{E}S = \frac{1}{D+1}.$$

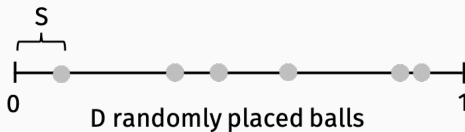
Proof:

$$\begin{aligned}\mathbb{E}[S] &= \int_0^1 \Pr[S \geq \lambda] d\lambda \\ &= \int_0^1 (1 - \lambda)^D d\lambda \\ &= \frac{-(1 - \lambda)^{D+1}}{D + 1} \Big|_{\lambda=0}^1 \\ &= \frac{1}{D + 1}\end{aligned}$$

Exercise: Why?

PROOF “FROM THE BOOK”

$\mathbb{E}[S] = \Pr[(D + 1)^{\text{st}} \text{ item has the smallest hash value}]$.



By symmetry, this equals $\frac{1}{D+1}$ (since every ball is equally likely to be first).

$\mathbb{E}S = \frac{1}{D+1}$. **Estimate:** $\tilde{D} = \frac{1}{S} - 1$. We have for $\epsilon < \frac{1}{2}$:

If $|S - \mathbb{E}S| \leq \epsilon \cdot \mathbb{E}S$, then:

$$(1 - 4\epsilon)D \leq \tilde{D} \leq (1 + 4\epsilon)D.$$

So, it suffices to show that S concentrates around its mean. We will use Chebyshev's inequality as our concentration bound.

Lemma

$$\text{Var}[S] = \mathbb{E}[S^2] - \mathbb{E}[S]^2 = \frac{2}{(D+1)(D+2)} - \frac{1}{(D+1)^2} \leq \frac{1}{(D+1)^2}.$$

Proof:

$$\begin{aligned} \mathbb{E}[S^2] &= \int_0^1 \Pr[S^2 \geq \lambda] d\lambda \\ &= \int_0^1 \Pr[S \geq \sqrt{\lambda}] d\lambda \\ &= \int_0^1 (1 - \sqrt{\lambda})^D d\lambda \\ &= \frac{2}{(D+1)(D+2)} \end{aligned}$$

www.wolframalpha.com/input/?i=integral+from+0+to+1+of+%281-sqrt%28x%29%29%5ED

PROOF “FROM THE BOOK”

$$\mathbb{E}[S] = ??.$$



- $\mathbb{E}[S] = \frac{1}{D+1} = \mu.$
- $\text{Var}[S] \leq \mu^2$. Standard deviation: $\sigma \leq \mu.$
- Want to bound $\Pr[|S - \mu| \leq \epsilon\mu] \leq \delta.$

Chebyshev's: $\Pr[|S - \mu| \leq \epsilon\mu] = \Pr[|S - \mu| \leq \epsilon\sigma] \leq \frac{1}{\epsilon^2}.$

Vacuous bound. Our variance is way too high!

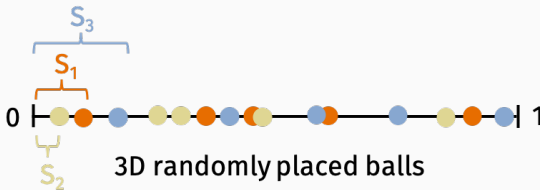
Trick of the trade: Repeat many independent trials and take the mean to get a better estimator.

Given i.i.d. (independent, identically distributed) random variables X_1, \dots, X_k with mean μ and variance σ^2 , what is:

- $\mathbb{E} \left[\frac{1}{k} \sum_{i=1}^k X_i \right] =$

- $\text{Var} \left[\frac{1}{k} \sum_{i=1}^k X_i \right] =$

Using independent hash functions, maintain k independent sketches S_1, \dots, S_k .



Flajolet-Martin:

- Choose k random hash function $h_1, \dots, h_k : \mathcal{U} \rightarrow [0, 1]$.
- $S_1 = \infty, \dots, S_k = \infty$
- For $i = 1, \dots, n$
 - $S_j \leftarrow \min(S_j, h_j(d_i))$ for all $j \in 1, \dots, k$.
- $S = (S_1 + \dots + S_k)/k$
- Return: $\frac{1}{S} - 1$

1 estimator:

- $\mathbb{E}[S] = \frac{1}{D+1} = \mu.$
- $\text{Var}[S] = \mu^2$

k estimators:

- $\mathbb{E}[S] = \frac{1}{D+1} = \mu.$
- $\text{Var}[S] = \mu^2/k$
- By Chebyshev, $\Pr[|S - \mathbb{E}S| \geq c\mu/\sqrt{k}] \leq \frac{1}{c^2}.$

Setting $c = 1/\sqrt{\delta}$ and $k = O\left(\frac{1}{\epsilon^2\delta}\right)$ gives:

$$\Pr[|S - \mu| \geq \epsilon\mu] \leq \delta.$$

Total space complexity: $O\left(\frac{1}{\epsilon^2\delta}\right)$ to estimate distinct elements up to error ϵ with success probability $1 - \delta$.

$O\left(\frac{1}{\epsilon^2\delta}\right)$ space is an impressive bound:

- Achieves any accuracy desired. $1/\epsilon^2$ dependence cannot be improved.
- No dependence on number of distinct elements D . Naive algorithm takes $O(D)$ space.
- But... $1/\delta$ dependence is not ideal. For 95% success rate, pay a $\frac{1}{5\%} = 20$ factor overhead in space.

We can get a better bound depending on $O(\log(1/\delta))$ using exponential tail bounds.

DISTINCT ELEMENTS IN PRACTICE

In practice, we cannot hash to real numbers on $[0, 1]$. Instead, map to bit vectors.

Real Flajolet-Martin / HyperLogLog:

$h(x_1)$	101001 0
$h(x_2)$	10011 00
$h(x_3)$	100111 0
⋮	
$h(x_n)$	1011 000

- Estimate # distinct elements based on maximum number of trailing zeros m .
- The more distinct hashes we see, the higher we expect this maximum to be.

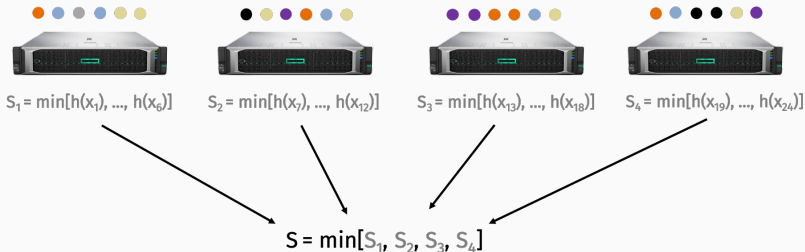
Total Space: $O\left(\frac{\log \log D}{\epsilon^2} + \log D\right)$ for an ϵ approximate count.

“Using an auxiliary memory smaller than the size of this abstract, the LogLog algorithm makes it possible to estimate in a single pass and within a few percents the number of different words in the whole of Shakespeare’s works.” – Flajolet, Durand.

Using HyperLogLog to count 1 billion distinct items with 2% accuracy:

$$\begin{aligned}\text{space used} &= O\left(\frac{\log \log D}{\epsilon^2} + \log D\right) \\ &= \frac{1.04 \cdot \lceil \log_2 \log_2 D \rceil}{\epsilon^2} + \lceil \log_2 D \rceil \text{ bits} \\ &= \frac{1.04 \cdot 5}{.02^2} + 30 = 13030 \text{ bits} \approx 1.6 \text{ kB!}\end{aligned}$$

DISTRIBUTED DISTINCT ELEMENTS



Distinct elements summaries are “mergeable”. No need to share lists of distinct elements if those elements are stored on different machines. Just share minimum hash value.

Implementations: Google PowerDrill, Facebook Presto, Twitter Algebird, Amazon Redshift.

Use Case: Exploratory SQL-like queries on tables with 100's of billions of rows.

- **Count** number of **distinct** users in Germany that made at least one search containing the word 'auto' in the last month.
- **Count** number of **distinct** subject lines in emails sent by users that have registered in the last week, in comparison to number of emails sent overall (to estimate rates of spam accounts).

Answering a query requires a (distributed) linear scan over the database: 2 seconds in Google's distributed implementation.

Google Paper: “Processing a Trillion Cells per Mouse Click”

“The system has been in production since end of 2008 and was made available for internal users across all of Google mid 2009. Each month it is used by more than 800 users sending out about 4 million SQL queries. **After a hard day’s work, one of our top users has spent over 6 hours in the UI, triggering up to 12 thousand queries.** When using our column-store as a backend, this may amount to scanning as much as 525 trillion cells in (hypothetical) full scans.”

BREAK

Load balancing problem:

Suppose Google answers map search queries using servers A_1, \dots, A_q . Given a query like “new york to rhode island”, common practice is to choose a random hash function $h \rightarrow \{1, \dots, q\}$ and to route this query to server:

$$A_h(\text{“new york to rhode island”})$$

Why use a hash function instead of just distributing requests randomly?

Goal: Ensure that requests are distributed evenly, so no one server gets loaded with too many requests. We want to avoid downtime and slow responses to clients.

Suppose we have n servers and m requests, x_1, \dots, x_m . Let s_i be the number of requests sent to server $i \in \{1, \dots, n\}$:

$$s_i = \sum_{j=1}^m \mathbb{1}[h(x_j) = i].$$

Formally, our goal is to understand the value of maximum load on any server, which can be written as the random variable:

$$S = \max_{i \in \{1, \dots, n\}} s_i.$$

A good first step in any analysis of random variables is to first think about expectations. If we have n servers and m requests, for any $i \in \{1, \dots, n\}$:

$$\mathbb{E}[S_i] = \sum_{j=1}^m \mathbb{E} [\mathbb{1}[h(x_j) = i]] = \frac{m}{n}.$$

But it's very unclear what the expectation of $S = \max_{i \in \{1, \dots, n\}} S_i$ is... in particular, $\mathbb{E}[S] \neq \max_{i \in \{1, \dots, n\}} \mathbb{E}[S_i]$.

Exercise: Convince yourself that for two random variables A and B , $\mathbb{E}[\max(A, B)] \neq \max(\mathbb{E}[A], \mathbb{E}[B])$ even if those random variable are independent.

SIMPLIFYING ASSUMPTIONS

Number of servers: To reduce notation and keep the math simple, let's assume that $m = n$. I.e., we have exactly the same number of servers and requests.

Hash function: Continue to assume a fully (uniformly) random hash function h .



Often called the “balls-into-bins” model.

$\mathbb{E}[s_i]$ = expected number of balls per bin = $\frac{m}{n} = 1$. We would like to prove a bound of the form:

$$\Pr[\max_i s_i \geq C] \leq \frac{1}{10}.$$

for as tight a value of C . I.e., something much better than $C = n$.

BOUNDING A UNION OF EVENTS

Goal: Prove that for some C ,

$$\Pr[\max_i s_i \geq C] \leq \frac{1}{10}.$$

\cup means “or”. \cap means “and”.

Equivalent statement: Prove that for some C ,

$$\Pr[(s_1 \geq C) \cup (s_2 \geq C) \cup \dots \cup (s_n \geq C)] \leq \frac{1}{10}.$$

Need to bound the probability of a union of different events.

These events are not independent!!

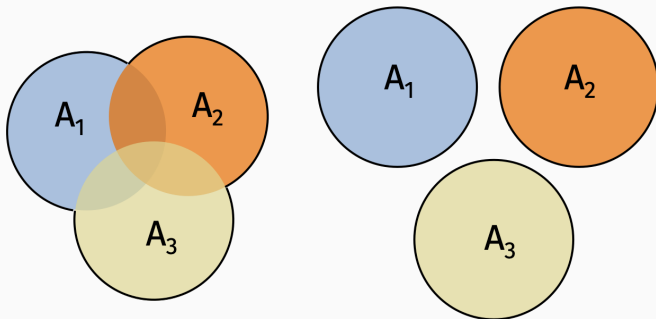
n = number of balls and number of bins. s_i is number of balls in bin i . C = upper bound on maximum number of balls in any bin.

USE A UNION BOUND

Lemma (Union Bound)

For any random events A_1, \dots, A_k :

$$\Pr[A_1 \cup A_2 \cup \dots \cup A_k] \leq \Pr[A_1] + \Pr[A_2] + \dots + \Pr[A_k].$$



Proof by picture.

APPLICATION OF UNION BOUND

We want to prove that:

$$\Pr[\max_i s_i \geq C] = \Pr[(s_1 \geq C) \cup (s_2 \geq C) \cup \dots \cup (s_n \geq C)] \leq \frac{1}{10}.$$

To do so, it suffices to prove that for all i :

$$\Pr[s_i \geq C] \leq \frac{1}{10n}.$$

Why? Because then by the union bound,

$$\begin{aligned} \Pr[\max_i s_i \geq C] &\leq \sum_{i=1}^n \Pr[s_i \geq C] \quad (\text{Union bound}) \\ &\leq \sum_{i=1}^n \frac{1}{10n} = \frac{1}{10}. \quad \square \end{aligned}$$

NEW GOAL

Prove that for some C ,

$$\Pr[s_i \geq C] \leq \frac{1}{10n}.$$

This should look hard! We need to prove that $s_i < C$ (i.e. the i^{th} bin has a small number of balls) with very high probability (specifically $1 - \frac{1}{10n}$).

Markov's inequality is too weak of a bound for this.

n = number of balls and number of bins. s_i is number of balls in bin i . C = upper bound on maximum number of balls in any bin.

APPLICATION TO BALLS INTO BINS

Goal: Prove that $\Pr[s_i \geq C] \leq \frac{1}{10n}$.

- **Step 1.** To apply Chebyshev's inequality, we need to understand $\sigma^2 = \text{Var}[s_i]$.

Use linearity of variance. Let $s_{i,j}$ be a $\{0, 1\}$ indicator random variable for the event that ball j falls in bin i . Clearly:

$$s_i = \sum_{j=1}^n s_{i,j}.$$

And $s_{i,1}, \dots, s_{i,n}$ are (mutually) independent so:

$$\text{Var}[s_i] = \sum_{j=1}^n \text{Var}[s_{i,j}].$$

n = number of balls and number of bins. s_i is number of balls in bin i . $\mathbb{E}[s_i] = 1$. C = upper bound on max number of balls in bin.

$$s_{i,j} = \begin{cases} 1 & \text{with probability } \frac{1}{n} \\ 0 & \text{otherwise.} \end{cases}$$

$$\mathbb{E}[s_{i,j}] =$$

$$\mathbb{E}[s_{i,j}^2] =$$

So:

$$\text{Var}[s_{i,j}] = \mathbb{E}[s_{i,j}^2] - \mathbb{E}[s_{i,j}]^2 =$$

n = number of balls and number of bins. $s_{i,j}$ is event ball j lands in bin i .

APPLYING CHEBYSHEV'S

Goal: Prove that $\Pr[s_i \geq C] \leq \frac{1}{10n}$.

Step 1. To apply Chebyshev's inequality, we need to understand $\sigma^2 = \text{Var}[s_i]$.

$$\text{Var}[s_i] = \sum_{j=1}^n \text{Var}[s_{i,j}] = \sum_{j=1}^n \frac{1}{n} - \frac{1}{n^2} = 1 - \frac{1}{n}.$$

Step 2. Apply Chebyshev's inequality:

$$\Pr \left[|s_i - \mathbb{E}[s_i]| \geq k \cdot \sqrt{1 - 1/n} \right] \leq \frac{1}{k^2}$$

$$\text{which implies } \Pr [|s_i - 1| \geq k \cdot 1] \leq \frac{1}{k^2}.$$

n = number of balls and number of bins. s_i = number of balls in bin i . $s_{i,j}$ is event ball j lands in bin i . $\mathbb{E}[s_i] = 1$.

APPLYING CHEBYSHEV'S

Goal: Prove that $\Pr[s_i \geq C] \leq \frac{1}{10n}$.

We just proved: $\Pr[|s_i - 1| \geq k] \leq \frac{1}{k^2}$.

Setting $k = \sqrt{10n}$ gives:

$$\Pr[|s_i - 1| \geq \sqrt{10n}] \leq \frac{1}{10n}.$$

So, we have that:

$$\Pr[s_i \geq \sqrt{10n} + 1] \leq \frac{1}{10n}.$$

By the union bound argument from earlier, it thus holds that:

$$\Pr[\max_{i \in \{1, \dots, n\}} s_i \geq \sqrt{10n} + 1] \leq \frac{1}{10}.$$

n = number of balls and number of bins. s_i is number of balls in bin i . C = upper bound on maximum number of balls in any bin.

When hashing n balls into n bins, the maximum bin contains $o(\sqrt{n})$ balls with probability $\frac{9}{10}$.



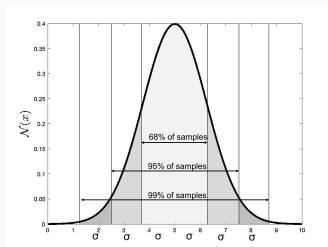
Much better than the trivial bound of n !

Techniques used that will appear again:

- Union bound to control the maximum of many random variables.
- Chebyshev's inequality to bound a variable whose variance we can compute.
- To compute the variance, break down random variable into smaller pieces and apply linearity of variance.

But... For this problem we can actually use even stronger tools to prove a better bound of $O(\log n)$ for the most loaded bin.

Motivating question: Is Chebyshev's Inequality tight?



68-95-99 rule for Gaussian bell-curve. $X \sim N(0, \sigma^2)$

Chebyshev's Inequality:

$$\Pr(|X - \mathbb{E}[X]| \geq 1\sigma) \leq 100\%$$

$$\Pr(|X - \mathbb{E}[X]| \geq 2\sigma) \leq 25\%$$

$$\Pr(|X - \mathbb{E}[X]| \geq 3\sigma) \leq 11\%$$

$$\Pr(|X - \mathbb{E}[X]| \geq 4\sigma) \leq 6\%.$$

Truth:

$$\Pr(|X - \mathbb{E}[X]| \geq 1\sigma) \approx 32\%$$

$$\Pr(|X - \mathbb{E}[X]| \geq 2\sigma) \approx 5\%$$

$$\Pr(|X - \mathbb{E}[X]| \geq 3\sigma) \approx 1\%$$

$$\Pr(|X - \mathbb{E}[X]| \geq 4\sigma) \approx .01\%$$

GAUSSIAN CONCENTRATION

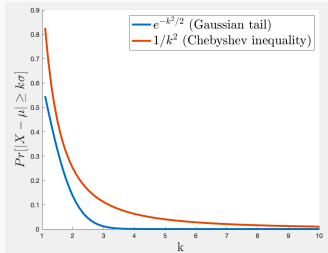
For $X \sim \mathcal{N}(\mu, \sigma^2)$:

$$\Pr[X = \mu \pm x] = \frac{1}{\sigma\sqrt{2\pi}} e^{-x^2/2\sigma^2}$$

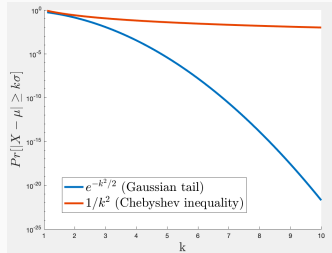
Lemma (Gaussian Tail Bound)

For $X \sim \mathcal{N}(\mu, \sigma^2)$:

$$\Pr[|X - \mathbb{E}X| \geq k \cdot \sigma] \leq 2e^{-k^2/2}.$$



Standard y-scale.



Logarithmic y-scale.

Takeaway: Gaussian random variables concentrate much tighter around their expectation than variance alone predicts.

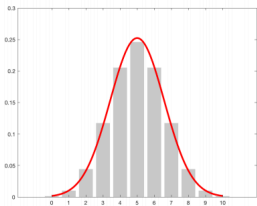
Why does this matter for algorithm design?

CENTRAL LIMIT THEOREM

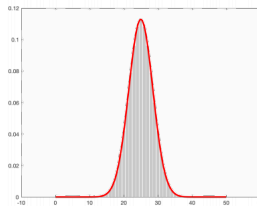
Theorem (CLT – Informal)

Any sum of *mutually independent, (identically distributed)* r.v.'s X_1, \dots, X_k with mean μ and finite variance σ^2 converges to a Gaussian r.v. with mean $k \cdot \mu$ and variance $k \cdot \sigma^2$, as $k \rightarrow \infty$.

$$S = \sum_{i=1}^n X_i \implies \mathcal{N}(k \cdot \mu, k \cdot \sigma^2).$$



(a) Distribution of # of heads after 10 coin flips, compared to a Gaussian.



(b) Distribution of # of heads after 50 coin flips, compared to a Gaussian.

Recall:

Definition (Mutual Independence)

Random variables X_1, \dots, X_k are mutually independent if, for all possible values v_1, \dots, v_k ,

$$\Pr[X_1 = v_1, \dots, X_k = v_k] = \Pr[X_1 = v_1] \cdot \dots \cdot \Pr[X_k = v_k]$$

Strictly stronger than pairwise independence.

If I flip a fair coin 100 times, lower bound the chance I get between 30 and 70 heads?

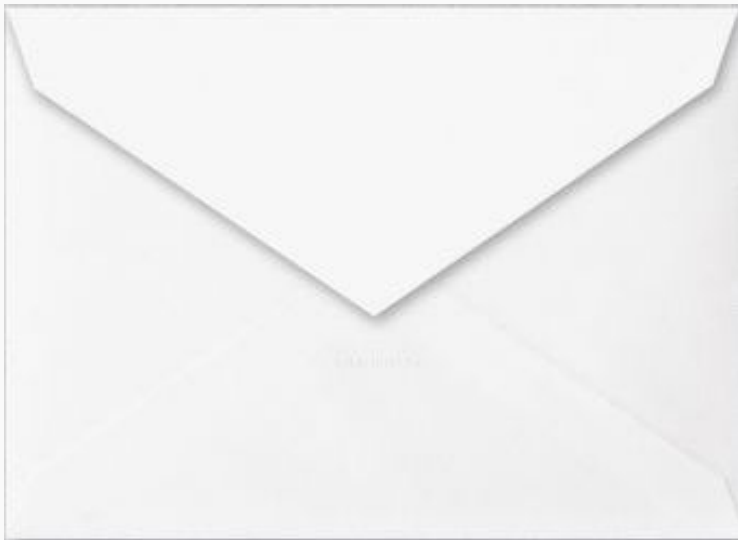
For this problem, we will assume the CLT holds exactly for a sum of independent random variables – i.e., that this sum looks exactly like a Gaussian random variable.

Lemma (Gaussian Tail Bound)

For $X \sim \mathcal{N}(\mu, \sigma^2)$:

$$\Pr[|X - \mathbb{E}X| \geq k \cdot \sigma] \leq 2e^{-k^2/2}.$$

BACK-OF-THE-ENVELOP CALCULATION



These back-of-the-envelope calculations can be made rigorous! Lots of different “versions” of bound which do so.

- Chernoff bound
- Bernstein bound
- Hoeffding bound
- ...

Different assumptions on random variables (e.g. binary vs. bounded), different forms (additive vs. multiplicative error), etc. **Wikipedia is your friend.**

Theorem (Chernoff Bound)

Let X_1, X_2, \dots, X_k be independent $\{0, 1\}$ -valued random variables and let $p_i = \mathbb{E}[X_i]$, where $0 < p_i < 1$. Then the sum $S = \sum_{i=1}^k X_i$, which has mean $\mu = \sum_{i=1}^k p_i$, satisfies

$$\Pr[S \geq (1 + \epsilon)\mu] \leq e^{\frac{-\epsilon^2 \mu}{2 + \epsilon}}.$$

and for $0 < \epsilon < 1$

$$\Pr[S \leq (1 - \epsilon)\mu] \leq e^{\frac{-\epsilon^2 \mu}{2}}.$$

Theorem (Bernstein Inequality)

Let X_1, X_2, \dots, X_k be independent random variables with each $X_i \in [-1, 1]$. Let $\mu_i = \mathbb{E}[X_i]$ and $\sigma_i^2 = \text{Var}[X_i]$. Let $\mu = \sum_i \mu_i$ and $\sigma^2 = \sum_i \sigma_i^2$. Then, for $k \leq \frac{1}{2}\sigma$, $S = \sum_i X_i$ satisfies

$$\Pr[|S - \mu| > k \cdot \sigma] \leq 2 \exp\left(-\frac{k^2}{4}\right).$$

Theorem (Hoeffding Inequality)

Let X_1, X_2, \dots, X_k be independent random variables with each $X_i \in [a_i, b_i]$. Let $\mu_i = \mathbb{E}[X_i]$ and $\mu = \sum_i \mu_i$. Then, for any $\alpha > 0$, $S = \sum_i X_i$ satisfies:

$$\Pr[|S - \mu| > \alpha] \leq 2 \exp\left(-\frac{\alpha^2}{\sum_{i=1}^k (b_i - a_i)^2}\right).$$

HOW ARE THESE BOUNDS PROVEN?

Variance is a natural measure of central tendency, but there are others.

q^{th} central moment: $\mathbb{E}[(X - \mathbb{E}X)^q]$

$k = 2$ gives the variance. Proof of Chebyshev's applies Markov's inequality to the random variable $(X - \mathbb{E}X)^2$.

Idea in brief: Apply Markov's inequality to $\mathbb{E}[(X - \mathbb{E}X)^q]$ for larger q , or more generally to $f(X - \mathbb{E}X)$ for some other non-negative function f . E.g., to $\exp(X - \mathbb{E}X)$.

We will explore this approach in the problem set.

CHERNOFF BOUND APPLICATION

Sample Application: Flip biased coin k times: i.e. the coin is heads with probability b . As long as $k \geq O\left(\frac{\log(1/\delta)}{\epsilon^2}\right)$,

$$\Pr[|\# \text{ heads} - b \cdot k| \geq \epsilon k] \leq \delta$$

Setup: Let $X_i = \mathbb{1}[i^{\text{th}} \text{ flip is heads}]$. Want bound probability that $\sum_{i=1}^k X_i$ deviates from it's expectation.

Corollary of Chernoff bound: Let $S = \sum_{i=1}^k X_i$ and $\mu = \mathbb{E}[S]$. For $0 < \Delta < 1$,

$$\Pr[|S - \mu| \geq \Delta\mu] \leq 2e^{-\Delta^2\mu/3}$$

Sample Application: Flip biased coin k times: i.e. the coin is heads with probability b . As long as $k \geq O\left(\frac{\log(1/\delta)}{\epsilon^2}\right)$,

$$\Pr[|\# \text{ heads} - b \cdot k| \geq \epsilon k] \leq \delta$$

Pay very little for higher probability – if you increase the number of coin flips by $2x$, δ goes from

$$1/10 \rightarrow 1/100 \rightarrow 1/10000$$

LOAD BALANCING

As in the first video lecture, we want to use concentration bounds to study the randomized load balancing problem. n jobs are distributed randomly to n servers using a hash function. Let S_i be the number of jobs sent to server i . What's the smallest **B** for which we can prove:

$$\Pr[\max_i S_i \geq \mathbf{B}] \leq 1/10$$



Recall: Suffices to prove that, for any i , $\Pr[S_i \geq \mathbf{B}] \leq 1/10n$:

$$\begin{aligned} \Pr[\max_i S_i \geq \mathbf{B}] &= \Pr[S_1 \geq \mathbf{B} \text{ or } \dots \text{ or } S_n \geq \mathbf{B}] \\ &\leq \Pr[S_1 \geq \mathbf{B}] + \dots + \Pr[S_n \geq \mathbf{B}] \quad (\text{union bound}). \end{aligned}$$

Theorem (Chernoff Bound)

Let X_1, X_2, \dots, X_n be independent $\{0, 1\}$ -valued random variables and let $p_i = \mathbb{E}[X_i]$, where $0 < p_i < 1$. Then the sum $S = \sum_{j=1}^n X_j$, which has mean $\mu = \sum_{j=1}^n p_j$, satisfies

$$\Pr[X \geq (1 + \epsilon)\mu] \leq e^{\frac{-\epsilon^2 \mu}{3+3\epsilon}}.$$

Consider a single bin. Let $X_j = \mathbb{1}[\text{ball } j \text{ lands in that bin}]$.

$\mathbb{E}[X_j] = \frac{1}{n}$, so $\mu = 1$.

$$\Pr[S \geq (1 + c \log n)\mu] \leq e^{\frac{-c^2 \log^2 n}{c + c \log n}} \leq e^{\frac{-c \log^2 n}{2 \log n}} \leq e^{-.5c \log n} \leq \frac{1}{10n},$$

for sufficiently large c

So max load for randomized load balancing is $O(\log n)$! Best we could prove with Chebyshev's was $O(\sqrt{n})$.

Power of 2 Choices: Instead of assigning job to random server, choose 2 random servers and assign to the least loaded. With probability 1/10 the maximum load is bounded by:

- (a) $O(\log n)$
- (b) $O(\sqrt{\log n})$
- (c) $O(\log \log n)$
- (d) $O(1)$