CS-GY 6763: Lecture 3
Finish Chebyshev's, Exponential Concentration Inequalities

NYU Tandon School of Engineering, Prof. Christopher Musco

DISTINCT ELEMENTS PROBLEM

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

Output: Number of distinct inputs, D.

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

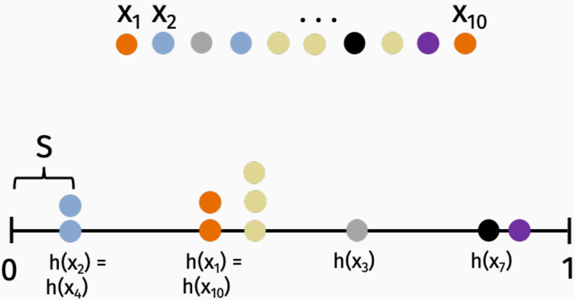
Flajolet–Martin (simplified):

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

VISUALIZATION

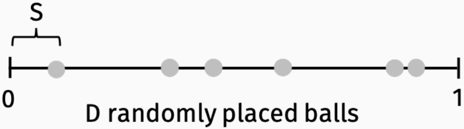
Flajolet–Martin (simplified):

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



proof "from the book"

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



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

Final Estimate: $\tilde{D} = \frac{1}{S} - 1$.

PROVING CONCENTRATION

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

$$\text{If } (1-\epsilon)\mathbb{E}S \leq S \leq (1+\epsilon)\mathbb{E}S\text{, then:}$$

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

So, it suffices to show that S concentrates around its mean. I.e. that $|S - \mathbb{E}S| \leq \epsilon \cdot \mathbb{E}S$. We will use Chebyshev's inequality as our concentration bound.

ϵ MANIPULATION TRICKS

Recall:

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

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

CALCULUS PROOF

Lemma

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

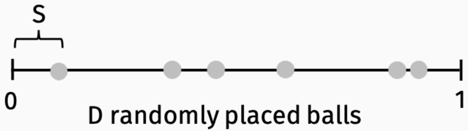
Proof:

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

`www.wolframalpha.com/input?i=antiderivative+of+%281-sqrt%28x%29%29%5ED`

PROOF "FROM THE BOOK"

$\mathbb{E}[S^2] = ??.$



FM ANALYSIS

Recall we want to show that, with high probability, $(1-\epsilon)\mathbb{E}[S] \leq S \leq (1-\epsilon)\mathbb{E}[S]$.

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

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

Vacuous bound. Our variance is way too high!

VARIANCE REDUCTION

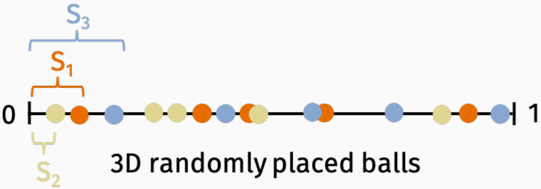
Trick of the trade: Repeat many independent trials and take the mean to get a better estimator.

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

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

FM ANALYSIS

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



Flajolet–Martin:

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

FM ANALYSIS

1 estimator:

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

k estimators:

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

Setting $c = 1/\sqrt{\delta}$ and $k = \frac{1}{\epsilon^2\delta}$ gives:

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

FM ANALYSIS

- Recall that to ensure $(1-\bar{\epsilon})D \leq \frac{1}{S} - 1 \leq (1+\bar{\epsilon})D$, we needed $|S - \mu| \leq \frac{\bar{\epsilon}}{4}\mu$.
- So apply the result from the previous slide with $\epsilon = \bar{\epsilon}/4$.
- Need to store $k = \frac{1}{\epsilon^2\delta} = \frac{1}{(\bar{\epsilon}/4)^2\delta} = \frac{16}{\epsilon^2\delta}$ counters.

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

NOTE ON FAILURE PROBABILITY

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

- $1/\epsilon^2$ dependence cannot be improved.
- No linear dependence on number of distinct elements D.[1]
- But... $1/\delta$ dependence is not ideal. For 95% success rate, pay a $\frac{1}{5\%} = 20$ factor overhead in space.

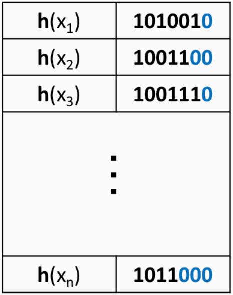
We can get a better bound depending on $O(\log(1/\delta))$ using exponential tail bounds. We will see next.

[1] Technically, if we account for the bit complexity of storing $S_1, \ldots, S_k$ and the hash functions $h_1, \ldots, h_k$, the space complexity is $O\left(\frac{\log D}{\epsilon^2\delta}\right)$.

DISTINCT ELEMENTS IN PRACTICE

In practice, we cannot hash to real numbers on $[0, 1]$. Could use a finite grid, but more popular choice is to hash to integers (bit vectors).

Real Flajolet-Martin / HyperLogLog:



$h(x_1)$	1010010
$h(x_2)$	1001100
$h(x_3)$	1001110
⋮	
$h(x_n)$	1011000

- 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.

LOGLOG SPACE

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.

LOGLOG SPACE

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.

LOGLOG SPACE

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

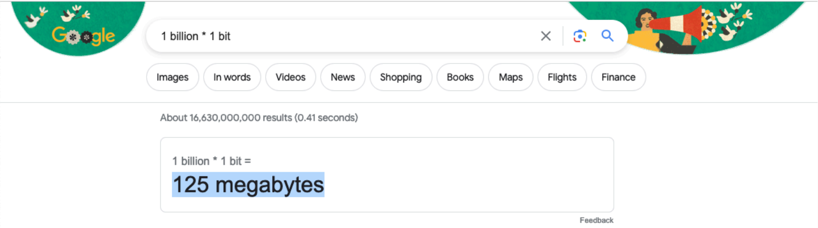
"Using an auxiliary memory smaller than the size of this abstract, the LogLog algorithm makes it possible to estimate in a single pass and within a few percents the number of different words in the whole of Shakespeare's works." – Flajolet, Durand.

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

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

HYPERLOGLOG IN PRACTICE

Although, to be fair, storing a dictionary with 1 billion bits only takes 125 megabytes. Not tiny, but not unreasonable.



About 16,630,000,000 results (0.41 seconds)

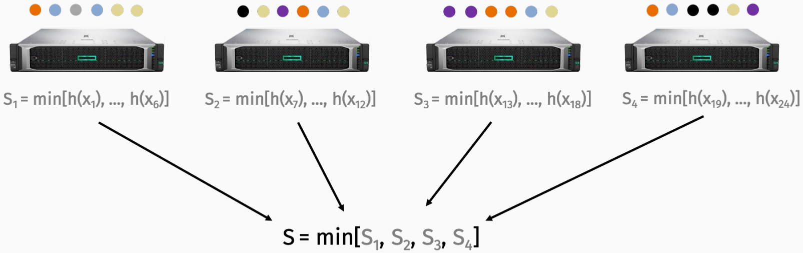
1 billion * 1 bit =

125 megabytes

These estimators become more important when you want to count many different things (e.g., a software company tracking clicks on 100s of UI elements).

DISTRIBUTED DISTINCT ELEMENTS

Also very important in distributed settings.



$S_1 = \min[h(x_1), \ldots, h(x_6)]$ $S_2 = \min[h(x_7), \ldots, h(x_{12})]$ $S_3 = \min[h(x_{13}), \ldots, h(x_{18})]$ $S_4 = \min[h(x_{19}), \ldots, h(x_{24})]$

$S = \min[S_1, S_2, S_3, S_4]$

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.

HYPERLOGLOG IN PRACTICE

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.

HYPERLOGLOG IN PRACTICE

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.

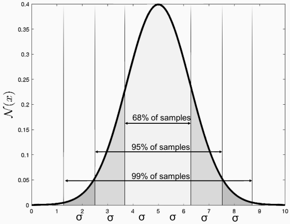
Answering a query requires a (distributed) linear scan over the database: 2 seconds in Google's distributed implementation.

Google Paper: "Processing a Trillion Cells per Mouse Click"

BEYOND CHEBYSHEV

Motivating question: Is Chebyshev's Inequality tight?

It is the worst case, but often not in reality.



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

Chebyshev's Inequality:

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

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

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

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

Truth:

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

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

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

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

GAUSSIAN CONCENTRATION

$X \sim \mathcal{N}(\mu, \sigma^2)$ has probability density function (PDF) p with:

$$p(\mu \pm x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-x^2/2\sigma^2}$$

Lemma (Gaussian Tail Bound)

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

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

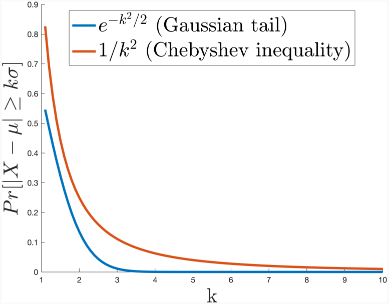
Compare this to:

Lemma (Chebyshev's Inequality)

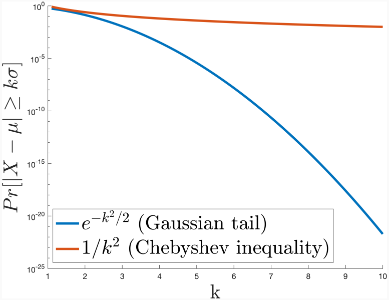
For $X \sim \mathcal{N}(\mu, \sigma^2)$:

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

GAUSSIAN CONCENTRATION



Standard y-scale.



Logarithmic y-scale.

Takeaway: Gaussian random variables concentrate much tighter around their expectation than variance alone predicts (i.e., than Chebyshevs's inequality predicts).

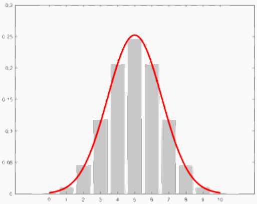
Why does this matter for algorithm design?

CENTRAL LIMIT THEOREM

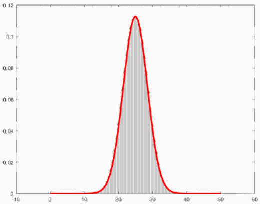
Theorem (CLT – Informal)

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

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



(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.

INDEPENDENCE

Recall:

Definition (Mutual Independence)

Random variables $X_1, \ldots, X_n$ are mutually independent if, for all possible values $v_1, \ldots, v_n$,

$$\Pr[X_1 = v_1, \ldots, X_n = v_n] = \Pr[X_1 = v_1] \cdot \ldots \cdot \Pr[X_n = v_n]$$

Strictly stronger than pairwise independence.

EXERCISE

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

Let's approximate the probability by assuming the limit of the CLT holds exactly – i.e., that this sum looks exactly like a Gaussian random variable.

Lemma (Gaussian Tail Bound)

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

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

$2e^{-8} = .06\%$. Chebyshev's inequality gave a bound of 6.25%.

QUANTITATIVE VERSIONS OF THE CLT

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

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

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

QUANTITATIVE VERSIONS OF THE CLT

Theorem (Chernoff Bound)

Let $X_1, X_2, \ldots, X_n$ *be independent* $\{0,1\}$*-valued random variables and let* $p_i = \mathbb{E}[X_i]$*, where* $0 < p_i < 1$*. Then the sum* $S = \sum_{i=1}^{n} X_i$*, which has mean* $\mu = \sum_{i=1}^{n} p_i$*, 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 (Chernoff Bound Corollary)

Let $X_1, X_2, \ldots, X_n$ be independent $\{0,1\}$-valued random variables and let $p_i = \mathbb{E}[X_i]$, where $0 < p_i < 1$. Let $S = \sum_{i=1}^n X_i$ and $\mathbb{E}[S] = \mu$. For $\epsilon \in (0,1)$,

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

Why does this look like the Gaussian tail bound of $\Pr[|S - \mu| \geq k \cdot \sigma] \lesssim 2e^{-k^2/2}$? What is $\sigma(S)$?

QUANTITATIVE VERSIONS OF THE CLT

Theorem (Bernstein Inequality)

Let $X_1, X_2, \ldots, X_n$ 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 $\mu = \sum_{i=1}^n \mu_i$ and $\sigma^2 = \sum_{i=1}^n \sigma_i^2$. Then, for $k \leq \frac{1}{2}\sigma$, $S = \sum_{i=1}^n X_i$ satisfies

$$\Pr[|S - \mu| > k \cdot \sigma] \leq 2e^{-k^2/4}.$$

QUANTITATIVE VERSIONS OF THE CLT

Theorem (Hoeffding Inequality)

Let $X_1, X_2, \ldots, X_n$ be independent random variables with each $X_i \in [a_i, b_i]$. Let $\mu_i = \mathbb{E}[X_i]$ and $\mu = \sum_{i=1}^n \mu_i$. Then, for any $k > 0$, $S = \sum_{i=1}^n X_i$ satisfies:

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

how are these bounds proven?

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

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

$q = 2$ 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 $\exp(X - \mathbb{E}X)$. Doing so requires higher-order independence.

EXERCISE

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

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

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

Here $X_i = \mathbb{1}[i^{\text{th}}$ flip is heads].

1.4%.

CHERNOFF BOUND APPLICATION

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

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

Pay very little for higher probability – if you increase the number of coin flips by 4x, δ goes from $1/10 \to 1/100 \to 1/10000$

Load balancing problem:

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

$$A_{h(\text{“new york to rhode island'})}$$

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.

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

LOAD BALANCING

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

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

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

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

LOAD BALANCING

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

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

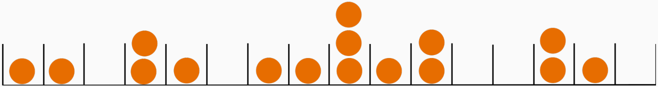
But it's unclear what the expectation of $S = \max_{i \in \{1,\ldots,n\}} s_i$ is... in particular, $\mathbb{E}[S] \neq \max_{i \in \{1,\ldots,n\}} \mathbb{E}[s_i]$.

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,

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

Equivalent statement: Prove that for some C,

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

These events are not independent, but we can apply union bound!

$n =$ 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 OF UNION BOUND

We want to prove that:

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

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,

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

$n =$ number of balls and number of bins. s_i is number of balls in bin i.

NEW GOAL

Prove that for some C,

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

Let's try doing this with Markov's, Chebyshev, and exponential concentration.

attempt with markov's inequality

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

- **Step 1.** Verify we can apply Markov's: s_i takes on non-negative values only. Good to go!
- **Step 2.** Apply Markov's: $\Pr[s_i \geq C] \leq \frac{\mathbb{E}[s_i]}{C} = \frac{1}{C}$.

To prove our target statement, need to see $C = 10n$.

Meaningless! There are only n balls, so of course there can't be more than $10n$ in the most overloaded bin.

$n =$ 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 maximum number of balls in any bin. **Markov's inequality**: for positive r.v. X, $\Pr[X \geq t] \leq \mathbb{E}[X]/t$.

ATTEMPT WITH CHEBYSHEV'S INEQUALITY

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. We have:

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

$n =$ 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.

VARIANCE ANALYSIS

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

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

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

So:

$$\mathrm{Var}[s_i] = \mathrm{Var}\left[\sum_{j=1}^{n} s_{i,j}\right] =$$

$n =$ 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 $\sigma^2 = \text{Var}[s_i]$.

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

Step 2. Apply Chebyshev's inequality:

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

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

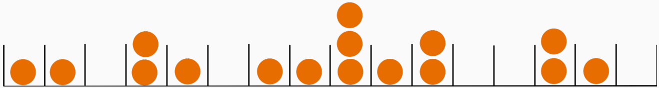
Goal: Prove that $\Pr[s_i \geq C] \leq \frac{1}{10n}$.

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

$n =$ 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.

FINAL RESULT FOR CHEBYSHEV'S

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!

attempt with exponential concentration

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

Recall: $s_i = \sum_{j=1}^{n} s_{i,j}$, where $s_{i,j} = \mathbb{1}[\text{ball } j \text{ lands in bin } i]$.

What bound might we use?

ATTEMPT WITH EXPONENTIAL CONCENTRATION

Theorem (Chernoff Bound)

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

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

Apply with $S = s_i$, $X_j = s_{i,j}$.

$\Pr[S \geq (1 + c\log n)\mu] \leq$

LOAD BALANCING

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

POWER OF TWO CHOICES

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) $O(1)$