**etherpad.wikimedia.org/p/312** for (anonymous) questions/comments!

# More applications!

## *More "bandits", Randomized Algorithms, etc.*

CSE 312 24Su

Lecture 23

# Logistics

- I lied to you…one more “concept check”, please do course evals!
  But it won’t be treated as a regular concept check, it’s one extra credit free point!
- Ed post about makeup opportunity for midterm this weekend
- If you requested a makeup exam, you should have received an email
  If there are other things that come up, please reach out as soon as possible

# The Problem

There are $K$ slot machines ("bandits" with "arms").

"Bandits" because they steal your money

You pull arms $T$ times, where the arm you pull at time $t$ *(arm pulled at time t is* $a_t \in \{1, \ldots, K\}$*)* will return a **random reward** to you



# The Problem

Your goal? Win as much money as possible! 🤑 That is, maximize your total expected reward after T pulls of an arm of the K slot machines.

*Question:* **How do you know which arm to pull (based on information you have – the past) to achieve your goal?**

*Assumptions:*

1. Rewards from each arm are independent.
2. The reward distribution of an arm does not change over time.

# Our Problem (summarized)

There are $k$ "arms". We get some reward when we pull the arm, but we don't know the distribution with which the rewards are given. Each time we pull an arm, we can observe the reward we get.

*Our goal:* Derive a strategy for picking which arm to pull to maximize the total reward based on observations from your previous pulls.

# Before we go on...why is this important?

Lot's of problems (reinforcement learning) can be phrased as a bandit problem!

**A/B Testing:** Experiment with releasing a new feature, or test ads to maximize click rate.
*Arms:* feature(s)/ad(s) to test *Maximize*: total ratings/click rate

**Networks:** Adaptive routings for picking best route for each pieces of data.
*Arms:* available routes *Maximize*: transmission speed

**Recommendations:** $K$ movies options. Recommend each person a movie and observe.
*Arms:* available movies *Maximize*: clicks/rating recommendations

**Clinical Trials:** $K$ treatment options. For each patient, give treatment and observe results.
*Arms:* different possible treatments *Maximize*: number of patients healed

***Very* real life:** For each meal, you choose food, and track your happiness after eating.
*Arms:* possible food options *Maximize*: total/average happiness

# Why is this a challenging problem?

There's a tradeoff between…

Exploitation **(act accordingly to what you know is going to work)**
Pulling arms that we know are "good" based on reward history.

Exploration **(explore other options that *might* increase reward)**
Pulling other arms in case they could be "good" or "better".

**Regret:** The difference between the optimal, best possible total (expected) reward and the actual reward from our choices of T pulls

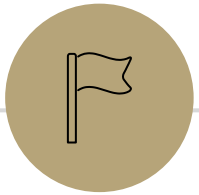
# Simplification for the problem

In this case, let's say that each "arm" rewards either $\$1$, $\$10$, or $\$100$.

We don't know what the probabilities are though…so the PMF for the reward of arm $i$ is:

$$p_{A_i}(k) = \begin{cases} 1 - \theta_1 - \theta_2 & k = 1 \\ \theta_1 & k = 10 \\ \theta_2 & \mathrm{k} = 100 \end{cases}$$

*Again, our goal is to find a strategy for picking an arm to pull that will maximize the total reward over the T pulls.*



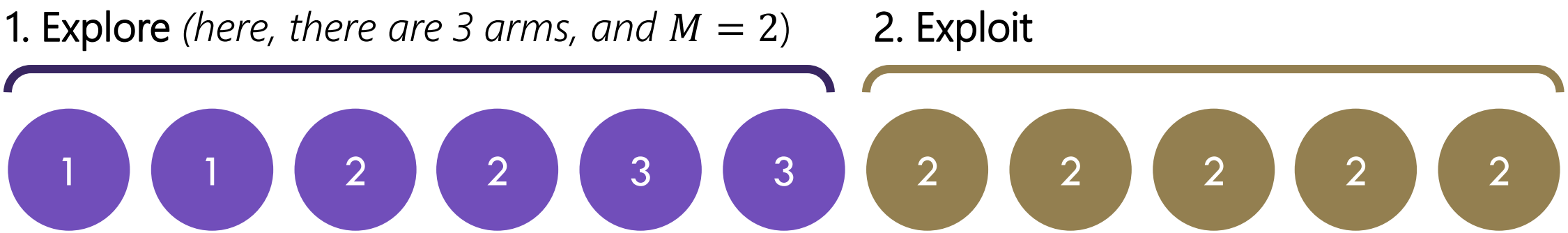
# Strategy 1: Naïve, Greedy Approach

# Strategy 1: Naïve, Greedy Approach

1. **Explore**. Pull each arm $M$ times and record the reward from each

*Based on data for each, estimate the parameters $\theta_1$ and $\theta_2$ for each arm*

2. **Exploit.** Pick arm with highest estimated expected value and *only* use that arm for the remaining pulls.

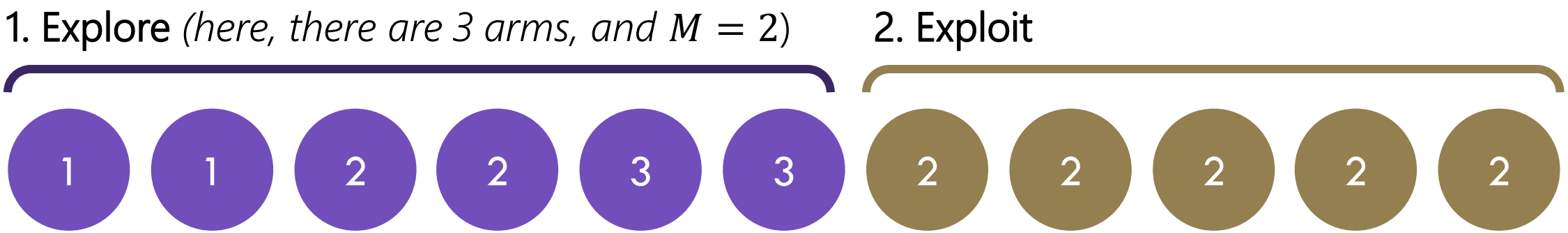


# Strategy 1: Naïve, Greedy Approach

1. **Explore**. Pull each arm $M$ times and record the reward from each

*Based on data for each, estimate the parameters $\theta_1$ and $\theta_2$ for each arm*
*How do we do this…..Maximum Likelihood Estimation!*

2. **Exploit.** Pick arm with highest estimated expected value and *only* use that arm for the remaining pulls.



# Strategy 1: Naïve, Greedy Approach

Pull arm $1$ $35$ times and record the reward from each.

We see rewards $x_1, x_2, \ldots, x_{35}$. Out of these, we get $\$1$ 5 times, $\$10$ 10 times and $\$100$ 20 times. What is the MLE for $\theta_1$ and $\theta_2$ for arm 1?

1. Likelihood Function

$$p_{A_1}(k) = \begin{cases} 1 - \theta_1 - \theta_2 & k = 1 \\ \theta_1 & k = 10 \\ \theta_2 & \mathrm{k} = 100 \end{cases}$$

2. Log-likelihood Function

3. Derivative(s) of Log-likelihood Function

....

4. Set the derivative(s) to 0 and solve for MLE(s)

5. Second derivative Test

> Next step would be to repeat this process and get estimates all arms
> Use the estimates to compute the expected reward from each
> Pick the arm with highest expected reward for the remaining pulls

# Strategy 1: Naïve, Greedy Approach

Pull arm $1$ $35$ times and record the reward from each.

We see rewards $x_1, x_2, \ldots, x_{35}$. Out of these, we get $\$1$ 5 times, $\$10$ 10 times and $\$100$ 20 times. What is the MLE for $\theta_1$ and $\theta_2$ for arm 1?

$$\widehat{\theta_1} = \frac{10}{35}, \widehat{\theta_2} = \frac{20}{35}$$

$$p_{A_i}(k) = \begin{cases} 1 - \theta_1 - \theta_2 & k = 1 \\ \theta_1 & k = 10 \\ \theta_2 & \mathrm{k} = 100 \end{cases}$$

How good is this estimator? Is it biased?

*1. Write a generalized form of the estimator*

Let $X_i \sim \mathrm{Ber}(\theta_1)$ and $Y_i \sim \mathrm{Ber}(\theta_2)$ --> $\widehat{\theta_1} = \frac{\sum_{i=1}^{35} X_i}{35}$, $\widehat{\theta_i} = \frac{\sum_{i=1}^{35} Y_i}{35}$

*2. Check if it's unbiased*

$$\mathbb{E}[\widehat{\theta_1}] = \mathbb{E}\left[\frac{\sum_{i=1}^{35} X_i}{35}\right] = \frac{\sum_{i=1}^{35} \mathbb{E}[X_i]}{35} = \theta_1 \text{ ✅} \quad \mathbb{E}[\widehat{\theta_2}] = \mathbb{E}\left[\frac{\sum_{i=1}^{35} Y_i}{35}\right] = \frac{\sum_{i=1}^{35} \mathbb{E}[Y_i]}{35} = \theta_2 \text{ ✅}$$

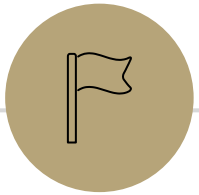
# Problems with this approach

We may not get an accurate idea from our first $M$ pulls of each arm

> If we choose the wrong best arm, we'd regret it for the rest of time!

> If we increase M, we are spending more time on sub-optimal arms

*Problem:* We did all exploration, and then all exploitation.
Why don't we blend the two a bit more?



# Strategy 2: Epsilon-Greedy

# Strategy 2: Epsilon Greedy

1. **Explore**. Pull each arm $M$ times and record the reward from each (same as before)

2. **Exploit (**with a mix of exploration!**).** With a small probability of $\epsilon$, try a random other arm. Otherwise calculate "best arm" *Based on data for each, estimate the parameters $\theta_1$ and $\theta_2$ for each arm and pick best arm with highest estimated expected value.*

***Better!***

> continuously updates estimated expected reward when it is pulled

> explores with some probability $\varepsilon$ which allows *you* to choose how to balance exploration and exploitation.

# Still some problems!

Is uniform exploration the optimal policy?
Is a higher estimate *always* a better choice?

We have our estimates $\widehat{\theta_1}$ and $\widehat{\theta_2}$ for each arm that we use to find the expected reward for each arm. As we see more data, we can update these estimates.

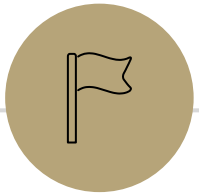
But, for example…

> After 300 samples from arm 1, $\widehat{\theta_2} = \mathbf{0.3}$

> After 3 samples from arm 2, $\widehat{\theta_2} = \mathbf{0.2}$

are very different! In this case, arm 2 still has a *potential* to have a much higher true probability of getting $100, but with arm 1, it's less likely

***We don't want to explore these equally!***



# Strategy 3: Upper Confidence Bound

Explore arms that have a higher *potential* of a better expectation

# Confidence Interval for Our Estimates

Instead of picking the arm with the highest expected value, pick the arm with the **highest *potential* for expected value**?

How to calculate "potential"? Use a confidence interval!

For each of the estimated parameters, what range can we be 95% sure the *true parameter* lies in?

*E.g. for arm 1,* $\widehat{\theta_1} = 0.2$ --> $\theta_1 \in [0.2 - 0.15., 0.2 + 0.15] = [0.05,0.35]$

$\widehat{\theta_2} = 0.3$ --> $\theta_2 \in [0.3 - 0.1, 0.3 + 0.1] = [0.2,0.4]$

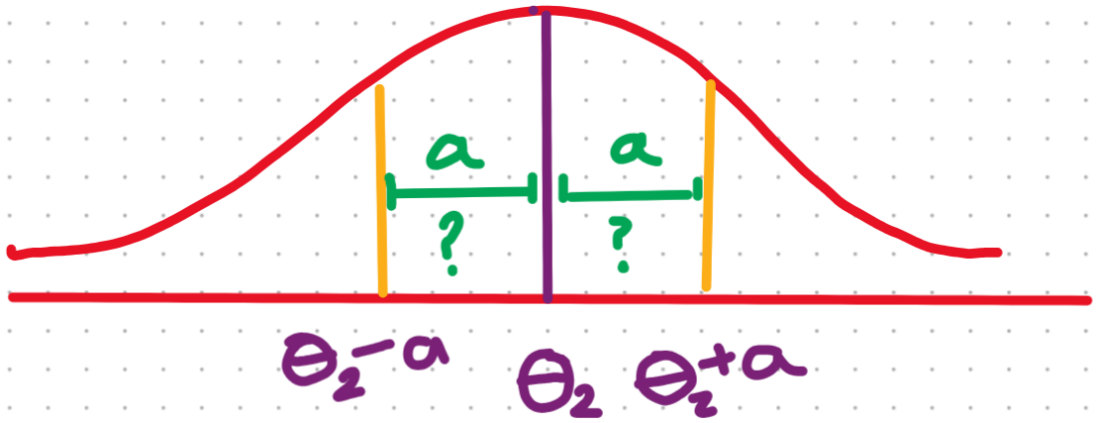
On the next slide, we're going to just look at the estimate for one of the parameters, $\theta_2$, but ideally, we would also look at the other parameter, and use it to estimate the expected value $\left(1 - \widehat{\theta_1} - \widehat{\theta_2}\right) + \left(10 \cdot \widehat{\theta_1}\right) + (100 \cdot \widehat{\theta_2})$

# Confidence Interval for Our Estimates

For an estimate $\widehat{\theta_2}$ on arm $i$ after seeing $35$ samples, what is the smallest value of $a$ such that the distance between the true $\theta_2$ and the estimate $\widehat{\theta_2}$ is at most $a$ with at least $95\%$ confidence?

*Translating to math notation….*

$$\mathbb{P}\left(\theta_2 - a \leq \widehat{\theta_2} \leq \theta_2 + a\right) \geq 0.95$$




*And what exactly is $\widehat{\theta_2}$ again?*

$\widehat{\theta_2} = \frac{\sum_{i=1}^{35} Y_i}{35} = \sum_{i=1}^{35} \frac{Y_i}{35}$, where $Y_i \sim \text{Ber}(\theta_2)$

^ is a sum of i.i.d RVs! So, what can we use to solve for $a$?

# Outline of CLT steps

1. Setup the problem (e.g., $X = \sum_{i=1}^{n} X_i$, $X_i$ are i.i.d., and we want $\mathbb{P}(X \leq k)$)
   Write event you are interested in, in terms of sum of random variables.

   ⭐ Apply *continuity correction here* if RVs are discrete.

2. Apply CLT (e.g., approx $X$ as $Y \sim N(n\mu, n\sigma^2)$ -> $\mathbb{P}(X \leq k) \approx \mathbb{P}(Y \leq k)$
   Approximate sum of RVs as normal with appropriate mean and variance

*from here, we're working with a normal distribution, which we've worked with before!*

3. Compute probability approximation using Phi table

   > *Standardize* $(Z = \frac{N-\mu}{\sigma})$ -> $\mathbb{P}(Y \leq k) = \mathbb{P}\left(\frac{Y-\mu}{\sigma} \leq \frac{k-\mu}{\sigma}\right) = \mathbb{P}\left(Z \leq \frac{k-\mu}{\sigma}\right)$

   > *Write in terms of* $\Phi(z) = \mathbb{P}(Z \leq z)$

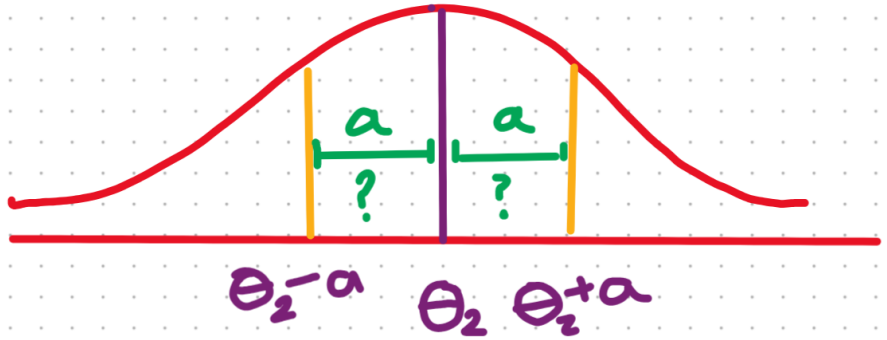
   > *Look up in table*

# Confidence Interval for Our Estimates

We have this MLE estimate: $\widehat{\theta_2} = \frac{\sum_{i=1}^{35} Y_i}{35} = \sum_{i=1}^{35} \frac{Y_i}{35}$, where $Y_i \sim \text{Ber}(\theta_2)$

What is the value of a such that: $\mathbb{P}\left(\theta_2 - a \leq \widehat{\theta_2} \leq \theta_2 + a\right) \geq 0.95$

1. Setup the problem




2. Apply CLT

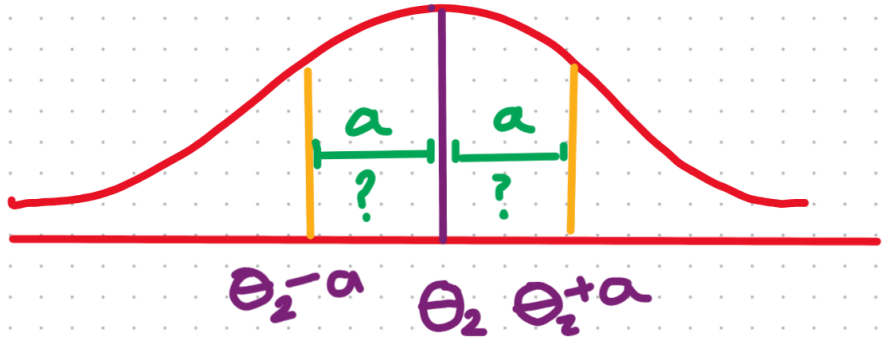
# 3. Compute probability approximation using Phi table

# Confidence Interval for Our Estimates

We have this MLE estimate: $\widehat{\theta_2} = \frac{\sum_{i=1}^{35} Y_i}{35} = \sum_{i=1}^{35} \frac{Y_i}{35}$, where $Y_i \sim \text{Ber}(\theta_2)$

What is the value of a such that: $\mathbb{P}\left(\theta_2 - a \leq \widehat{\theta_2} \leq \theta_2 + a\right) \geq 0.95$

1. Setup the problem




2. Apply CLT

# Handling $\sqrt{\theta_2(1-\theta_2)}$

**Justification 1:** If we make a mistake, we want it to be making $n$ bigger. (since we're trying to say "take $n$ at least this big, and you'll be safe").

The bigger the standard deviation, the bigger $n$ will need to be to control it. So assume the biggest possible standard deviation.

**Justification 2:**

As $\sqrt{\theta_2(1-\theta_2)}$ gets bigger, the interval gets smaller (it's in the denominator), so assuming the biggest value of $\sqrt{\theta_2(1-\theta_2)}$ gives us the most restricted interval. So no matter what the true interval is we have a subset of it. And if our probability is at least $.95$ then the true probability is at least $.95$.

What's the maximum of $\sqrt{\theta_2(1-\theta_2)}$?

# Worst value of $p$

Calculus time!

Set $\frac{d}{dp}\sqrt{p-p^2}=0$

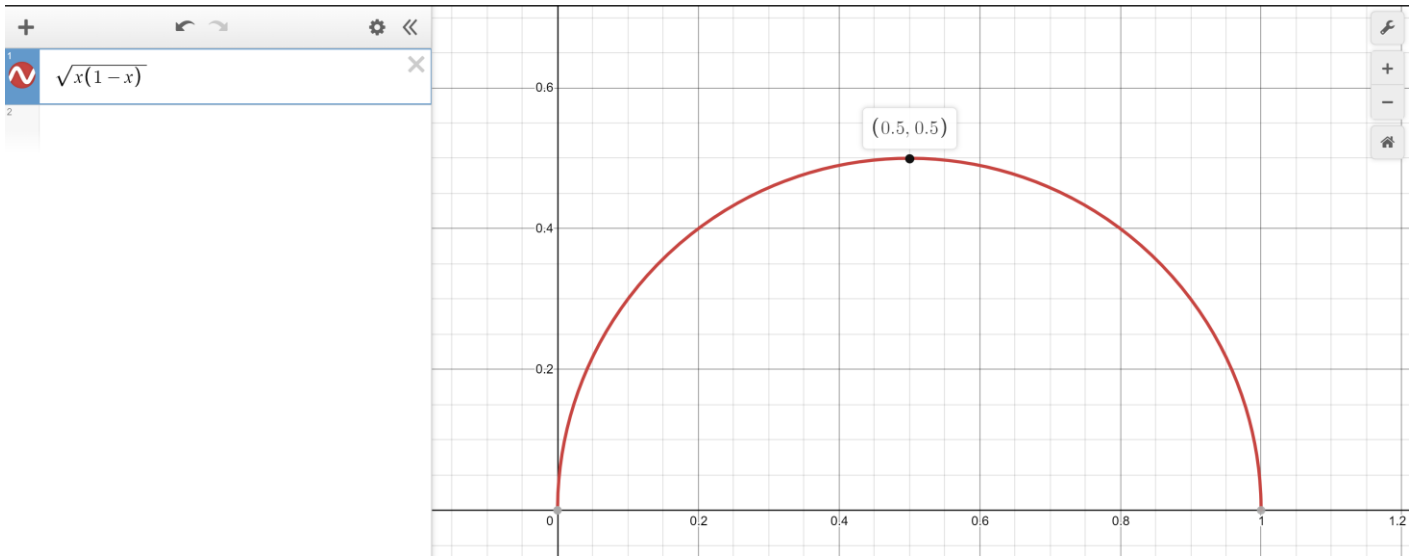
$\frac{1}{\sqrt{p-p^2}}(1-2p)=0$

$1-2p=0 \rightarrow p=1/2$

Second derivative test will confirm $p=\frac{1}{2}$ is a maximizer

Or just plot it.

$\sqrt{\frac{1}{2}\left(1-\frac{1}{2}\right)}=\sqrt{1/4}.$



**Side note:** A similar process can be used when we're trying to figure out things like how many people do we need to poll to be confident in our result when we know *NOTHING* about the true mean or variance of the population's votes!

$\Phi$ Table: $\mathbb{P}(Z \leq z)$ when $Z \sim \mathcal{N}(0,1)$

| $z$ | 0.00 | 0.01 | 0.02 | 0.03 | 0.04 | 0.05 | 0.06 | 0.07 | 0.08 | 0.09 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0.0 | 0.5 | 0.50399 | 0.50798 | 0.51197 | 0.51595 | 0.51994 | 0.52392 | 0.5279 | 0.53188 | 0.53586 |
| 0.1 | 0.53983 | 0.5438 | 0.54776 | 0.55172 | 0.55567 | 0.55962 | 0.56356 | 0.56749 | 0.57142 | 0.57535 |
| 0.2 | 0.57926 | 0.58317 | 0.58706 | 0.59095 | 0.59483 | 0.59871 | 0.60257 | 0.60642 | 0.61026 | 0.61409 |
| 0.3 | 0.61791 | 0.62172 | 0.62552 | 0.6293 | 0.63307 | 0.63683 | 0.64058 | 0.64431 | 0.64803 | 0.65173 |
| 0.4 | 0.65542 | 0.6591 | 0.66276 | 0.6664 | 0.67003 | 0.67364 | 0.67724 | 0.68082 | 0.68439 | 0.68793 |
| 0.5 | 0.69146 | 0.69497 | 0.69847 | 0.70194 | 0.7054 | 0.70884 | 0.71226 | 0.71566 | 0.71904 | 0.7224 |
| 0.6 | 0.72575 | 0.72907 | 0.73237 | 0.73565 | 0.73891 | 0.74215 | 0.74537 | 0.74857 | 0.75175 | 0.7549 |
| 0.7 | 0.75804 | 0.76115 | 0.76424 | 0.7673 | 0.77035 | 0.77337 | 0.77637 | 0.77935 | 0.7823 | 0.78524 |
| 0.8 | 0.78814 | 0.79103 | 0.79389 | 0.79673 | 0.79955 | 0.80234 | 0.80511 | 0.80785 | 0.81057 | 0.81327 |
| 0.9 | 0.81594 | 0.81859 | 0.82121 | 0.82381 | 0.82639 | 0.82894 | 0.83147 | 0.83398 | 0.83646 | 0.83891 |
| 1.0 | 0.84134 | 0.84375 | 0.84614 | 0.84849 | 0.85083 | 0.85314 | 0.85543 | 0.85769 | 0.85993 | 0.86214 |
| 1.1 | 0.86433 | 0.8665 | 0.86864 | 0.87076 | 0.87286 | 0.87493 | 0.87698 | 0.879 | 0.881 | 0.88298 |
| 1.2 | 0.88493 | 0.88686 | 0.88877 | 0.89065 | 0.89251 | 0.89435 | 0.89617 | 0.89796 | 0.89973 | 0.90147 |
| 1.3 | 0.9032 | 0.9049 | 0.90658 | 0.90824 | 0.90988 | 0.91149 | 0.91309 | 0.91466 | 0.91621 | 0.91774 |
| 1.4 | 0.91924 | 0.92073 | 0.9222 | 0.92364 | 0.92507 | 0.92647 | 0.92785 | 0.92922 | 0.93056 | 0.93189 |
| 1.5 | 0.93319 | 0.93448 | 0.93574 | 0.93699 | 0.93822 | 0.93943 | 0.94062 | 0.94179 | 0.94295 | 0.94408 |
| 1.6 | 0.9452 | 0.9463 | 0.94738 | 0.94845 | 0.9495 | 0.95053 | 0.95154 | 0.95254 | 0.95352 | 0.95449 |
| 1.7 | 0.95543 | 0.95637 | 0.95728 | 0.95818 | 0.95907 | 0.95994 | 0.9608 | 0.96164 | 0.96246 | 0.96327 |
| 1.8 | 0.96407 | 0.96485 | 0.96562 | 0.96638 | 0.96712 | 0.96784 | 0.96856 | 0.96926 | 0.96995 | 0.97062 |
| 1.9 | 0.97128 | 0.97193 | 0.97257 | 0.9732 | 0.97381 | 0.97441 | 0.975 | 0.97558 | 0.97615 | 0.9767 |
| 2.0 | 0.97725 | 0.97778 | 0.97831 | 0.97882 | 0.97932 | 0.97982 | 0.9803 | 0.98077 | 0.98124 | 0.98169 |
| 2.1 | 0.98214 | 0.98257 | 0.983 | 0.98341 | 0.98382 | 0.98422 | 0.98461 | 0.985 | 0.98537 | 0.98574 |
| 2.2 | 0.9861 | 0.98645 | 0.98679 | 0.98713 | 0.98745 | 0.98778 | 0.98809 | 0.9884 | 0.9887 | 0.98899 |
| 2.3 | 0.98928 | 0.98956 | 0.98983 | 0.9901 | 0.99036 | 0.99061 | 0.99086 | 0.99111 | 0.99134 | 0.99158 |
| 2.4 | 0.9918 | 0.99202 | 0.99224 | 0.99245 | 0.99266 | 0.99286 | 0.99305 | 0.99324 | 0.99343 | 0.99361 |
| 2.5 | 0.99379 | 0.99396 | 0.99413 | 0.9943 | 0.99446 | 0.99461 | 0.99477 | 0.99492 | 0.99506 | 0.9952 |
| 2.6 | 0.99534 | 0.99547 | 0.9956 | 0.99573 | 0.99585 | 0.99598 | 0.99609 | 0.99621 | 0.99632 | 0.99643 |
| 2.7 | 0.99653 | 0.99664 | 0.99674 | 0.99683 | 0.99693 | 0.99702 | 0.99711 | 0.9972 | 0.99728 | 0.99736 |
| 2.8 | 0.99744 | 0.99752 | 0.9976 | 0.99767 | 0.99774 | 0.99781 | 0.99788 | 0.99795 | 0.99801 | 0.99807 |
| 2.9 | 0.99813 | 0.99819 | 0.99825 | 0.99831 | 0.99836 | 0.99841 | 0.99846 | 0.99851 | 0.99856 | 0.99861 |
| 3.0 | 0.99865 | 0.99869 | 0.99874 | 0.99878 | 0.99882 | 0.99886 | 0.99889 | 0.99893 | 0.99896 | 0.999 |

# Upper Confidence Bound

Our strategy now is:

> First sample K of each arm to get some initial data

> Now, for every choice, we will use the previous data to figure out confidence intervals for each arm, use this to get ranges for the potential expectations of each arm, and pick the arm with the *highest* upper bound

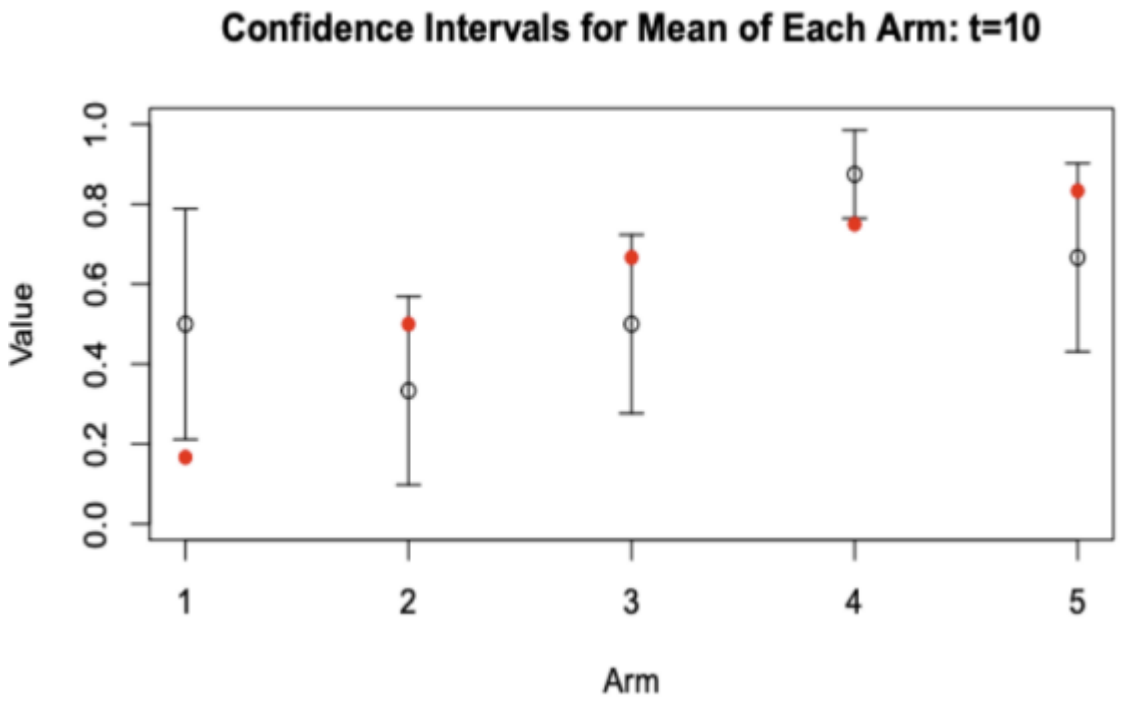
# Upper Confidence Bound

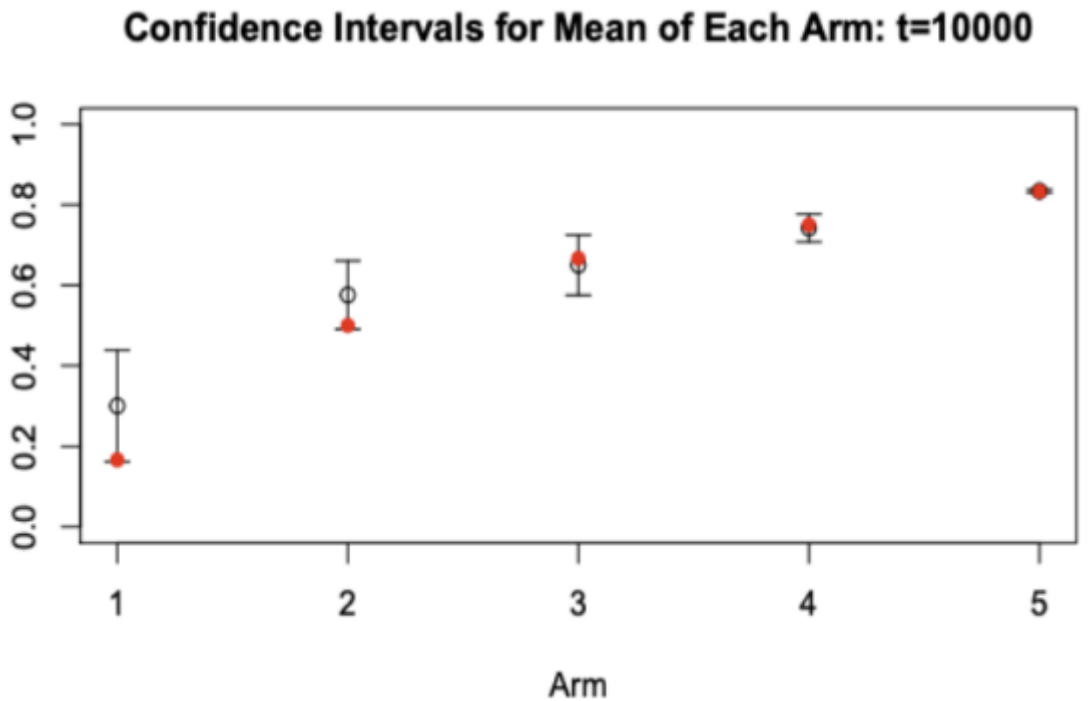
*More frequent pulls from an arm* --> smaller upper confidence bound

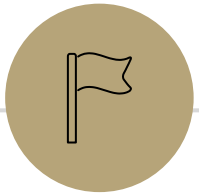
After seeing more observations, the variance of distribution decreases

*Less frequent pulls from an arm* --> larger upper confidence bound

After not seeing many observations, more variation is still possible





# Strategy 4: Thompson Sampling

See Alex Tsun's textbook for details about this ☺

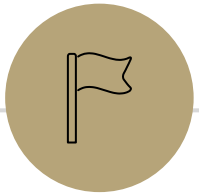
# Muti-Armed Bandits aren't *perfect*

- They don't personalize!
  they only find overall "winners", but sometimes, different customers need different things. There are other external factors that affect what the best choice is
- When there are too many arms, MABs don't work as fast and as well
  e.g., if the arms are the time of day to send an email, if you find that if you send an email at 3pm, the customer buys the product, that tells you that 2pm and 4pm are also more likely to be better times

*Contextual Bandits* are the next step! If you're curious, here are some great resources for learning about this.

article, another article, another *another* article

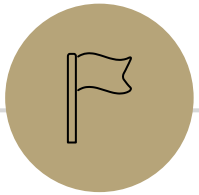
If you like these topics, this is also the basic for *reinforcement learning*



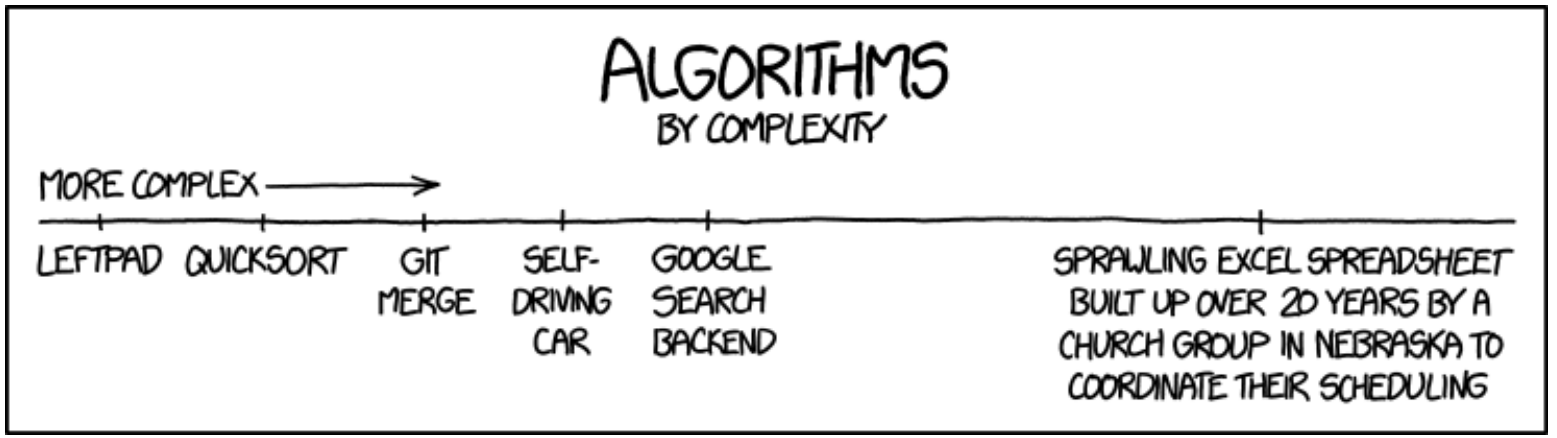
# Tail Bounds *In The Wild*

# Tail Bounds – Summary

- **Markov's inequality** - $\mathbb{P}(X \geq t) \leq \frac{\mathbb{E}[X]}{t}$
  - Use if $X$ is non-negative and we know the expectation
  - Useful when we don't know much about $X$
- **Chebyshev's inequality** - $\mathbb{P}(|X - \mathbb{E}[X]| \geq t) \leq \frac{\mathrm{Var(t)}}{t^2}$
  - Use if we know the expectation **and** variance of $X$
  - Gives better bounds with small variances
- **Chernoff Bound**
  $\mathbb{P}(X \leq (1-\delta)\mu) \leq e^{\left(-\frac{\delta^2\mu}{2}\right)}$ *and* $\mathbb{P}(X \geq (1+\delta)\mu) \leq e^{\left(-\frac{\delta^2\mu}{3}\right)}$
  - Use if $X$ is a sum of independent Bernoulli random variables
  - Gives a very good bound usually, and is especially helpful when $X$ is binomial and we can't easily computationally compute some summations/probability
- **Union Bound** - $\mathbb{P}(A \cup B) \leq \mathbb{P}(A) + \mathbb{P}(B)$ (*technically* not a *tail* bound…)
  - Use if we don't have enough information to find the union (e.g,. ways for at least of __ to occur, for A, or B, or C, or … to occur)



# Algorithm Analysis




https://xkcd.com/1667

# Randomized Algorithms

**Randomized algorithm use *randomness* in the computation**

Many algorithms incorporate some level of randomness

We can use the probabilistic techniques we've learned about in this class to analyze these algorithms!

Today…using tail bounds for analysis in randomized algorithms

# Two Common Types

**Las Vegas Algorithms:** We will keep running the algorithm (randomly looking for the solution) till we get a good solution.

*What is a bound on the running time for this?*

**Monte Carlo Algorithms:** We will stop at some fixed number of attempts regardless of whether a good solution was found.

*What is the probability a correct solution was found?*

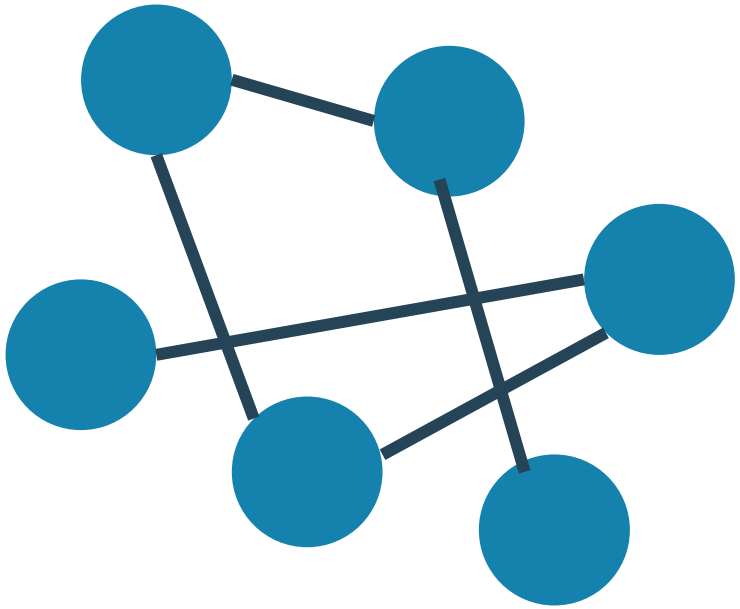
# Graphs

A pair of

> Set of **vertices/nodes**

> Set of **edges** between the vertices

- ***Weighted graph***s have weighted edges
- ***Directed graphs*** have edges that either go from A to B, or B to A

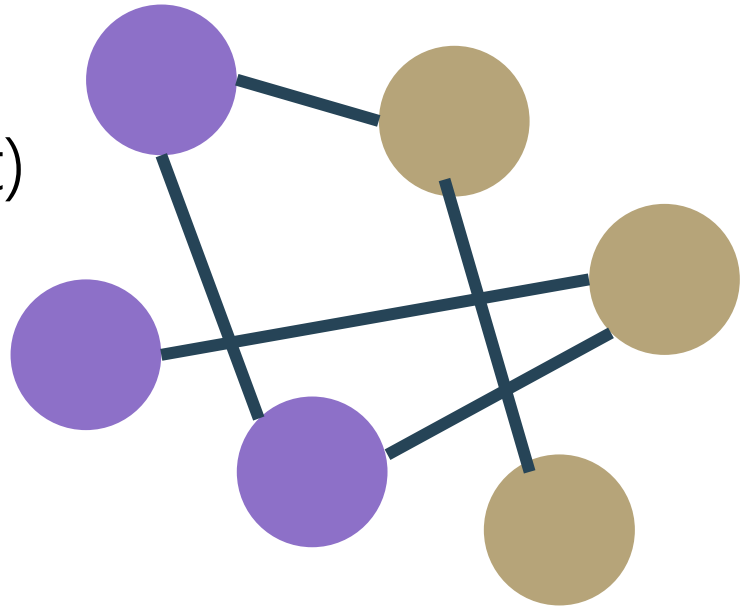


# Maximum Cut Problem

**The problem:** partition the nodes of a graph into two sets A and B such that the number of edges between the sets is maximized

*real world examples: binary classification*

The *cut* is the set of edges between
the nodes in the two sets
(goal: maximize the number of edges in the cut)

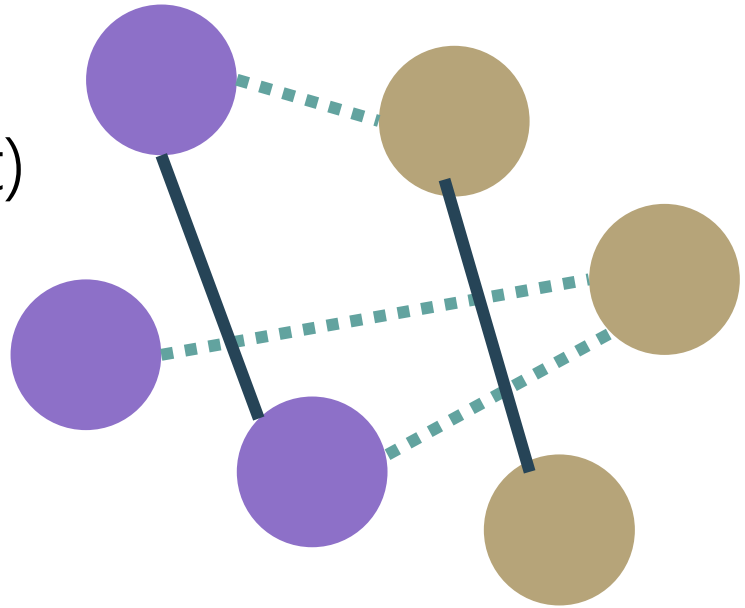


# Maximum Cut Problem

**The problem:** partition the nodes of a graph into two sets A and B such that the number of edges between the sets is maximized

*real world examples: binary classification*

The *cut* is the set of edges between
the nodes in the two sets
(goal: maximize the number of edges in the cut)

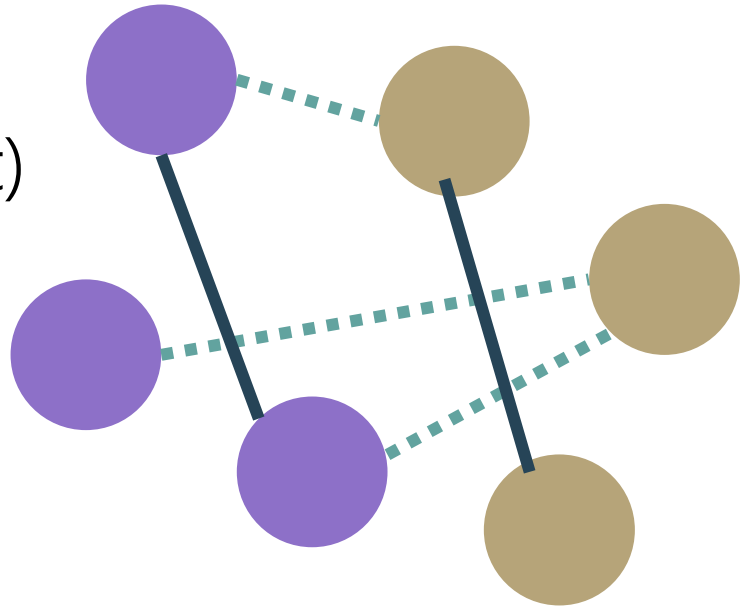


# Maximum Cut Problem

**The problem:** partition the nodes of a graph into two sets A and B such that the number of edges between the sets is maximized

*real world examples: binary classification*

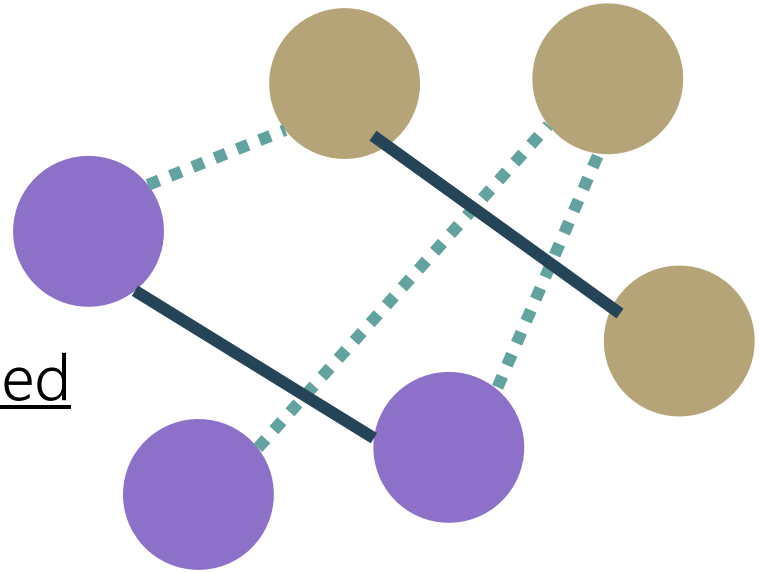
The *cut* is the set of edges between the nodes in the two sets
(goal: maximize the number of edges in the cut)



*Simple randomized algorithm:*

*Each node goes to A or B with probability 1/2*

# Maximum Cut Problem



**The problem:** partition the vertices of a graph into two sets such that the number of edges between the sets is maximized

*Simple algorithm:*
*Each node goes to A or B with probability 1/2*

***What's the probability of a "small" cut?***

$n$ is number of edges, X is number of edges in cut

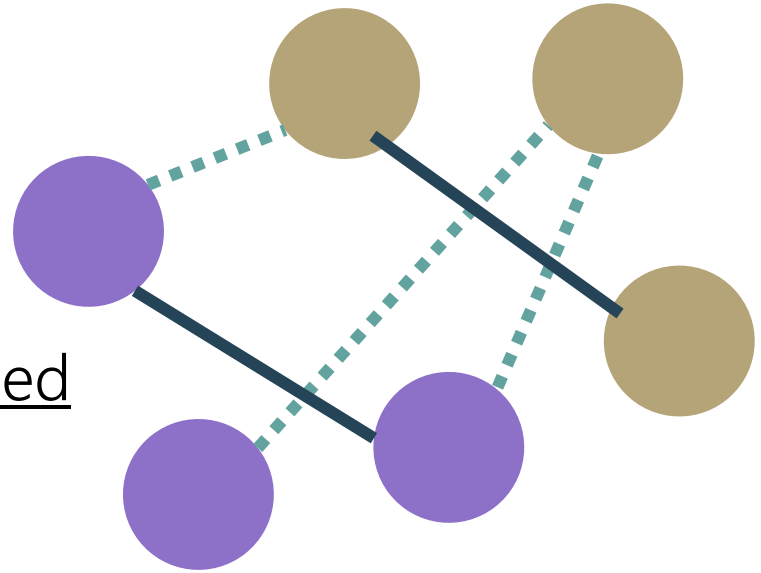
Use Markov's inequality to bound $\mathbb{P}(X \leq n/3)$

$$\mathbb{P}(X \geq k) \leq \frac{\mathbb{E}[X]}{k}$$

1. Find $\mathbb{E}[X]$

2. Apply Markov's Ineqality.

# Maximum Cut Problem



**The problem:** partition the vertices of a graph into two sets such that the number of edges between the sets is maximized

*Simple algorithm:*
*Each node goes to A or B with probability 1/2*

***What's the probability of a "small" cut?***

$n$ is number of edges, X is number of edges in cut

$$\mathbb{P}(X \geq k) \leq \frac{\mathbb{E}[X]}{k}$$

Use Markov's inequality to bound $\mathbb{P}(X \leq n/3)$

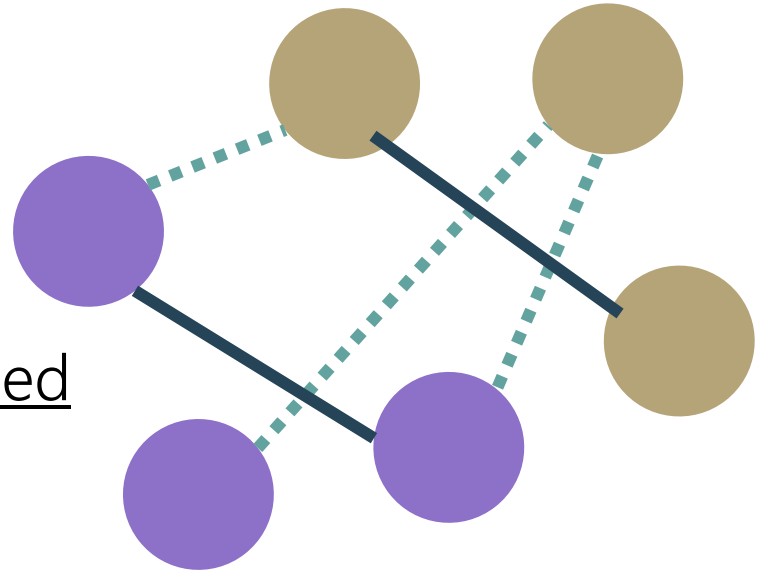
1. Find $\mathbb{E}[X]$. $X_i = 1$ of $i$'th edge is in the cut. $\mathbb{P}(X_i = 1) = \frac{1}{2}$

   $X = \sum_{i=1}^{n} X_i$ --> $\mathbb{E}[X] = \sum_{i=1}^{n} \mathbb{E}[X_i] = \frac{n}{2}$

2. Apply Markov's Ineqality. $\mathbb{P}(X \geq n/3) \leq \frac{n/2}{n/3}$ --> *taking complement..*

$\mathbb{P}(X \leq n/3) = 1 - \mathbb{P}\left(X \geq \frac{n}{3}\right) \geq 1 - 3/2 = -0.5$ *a trivial bound*

# Maximum Cut Problem



**The problem:** partition the vertices of a graph into two sets such that the number of edges between the sets is maximized

*Simple algorithm:*
*Each node goes to A or B with probability 1/2*

***What's the probability of a "small" cut?***

$n$ is number of edges, X is number of edges in cut

*You don't need to know this trick in this class*

Use Markov's inequality to bound $\mathbb{P}(X \leq n/3)$

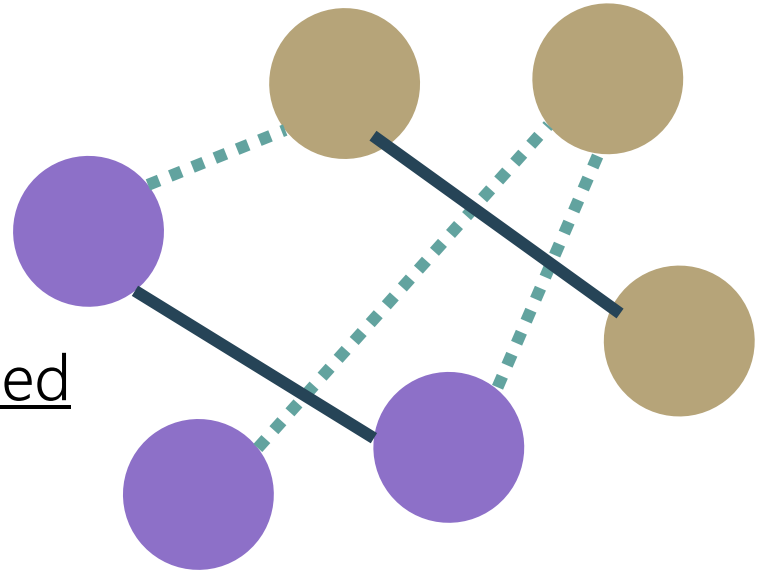
1. Find $\mathbb{E}[X]$. $X_i = 1$ of $i$'th edge is in the cut. $\mathbb{P}(X_i = 1) = \frac{1}{2}$

$X = \sum_{i=1}^{n} X_i$ --> $\mathbb{E}[X] = \sum_{i=1}^{n} \mathbb{E}[X_i] = \frac{n}{2}$

2. Apply Markov's Ineqality.

$\mathbb{P}(X \leq n/3) = \mathbb{P}(n - X \geq n - n/3) \leq \frac{\mathbb{E}[n-X]}{n-\frac{n}{3}} = \frac{n-n/2}{n-n/3} = \frac{n/2}{2n/3} = 3/4$ --> *a trick!*

# Maximum Cut Problem



**The problem:** partition the vertices of a graph into two sets such that the number of edges between the sets is maximized

*Simple algorithm:*
*Each node goes to A or B with probability 1/2*

***What's the probability of a "small" cut?***

$n$ is number of edges, X is number of edges in cut

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

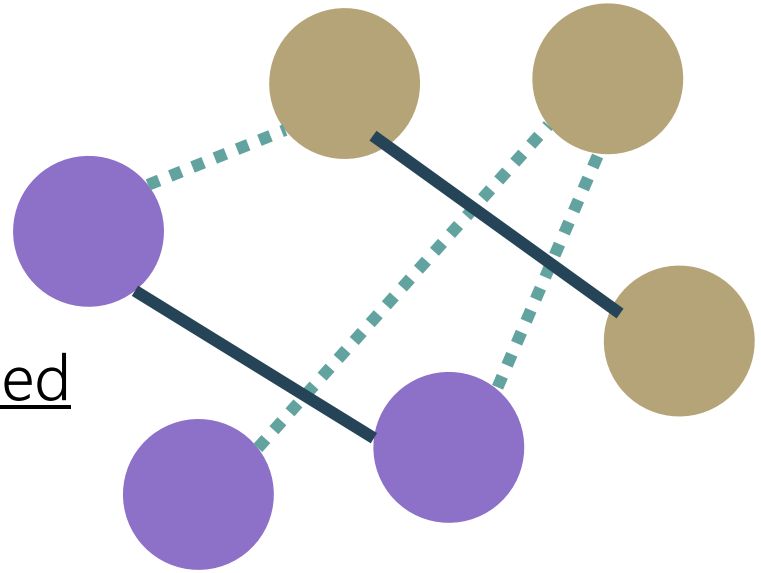
Use Chebyshev's inequality to bound $\mathbb{P}(X \leq n/3)$

1. Find $\mathbb{E}[X]$. $\mathbb{E}[X] = \frac{n}{2}$ 2. Find $\mathrm{Var}(X)$. $\mathrm{Var}(X) = \frac{n}{4}$ (see 3.2 here for explanation)

2. Apply Chebyshev's Ineqality.

$\mathbb{P}(X \leq n/3) =$

# Maximum Cut Problem



**The problem:** partition the vertices of a graph into two sets such that the number of edges between the sets is maximized

*Simple algorithm:*
*Each node goes to A or B with probability 1/2*

***What's the probability of a "small" cut?***

$n$ is number of edges, X is number of edges in cut

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

Use Chebyshev's inequality to bound $\mathbb{P}(X \leq n/3)$

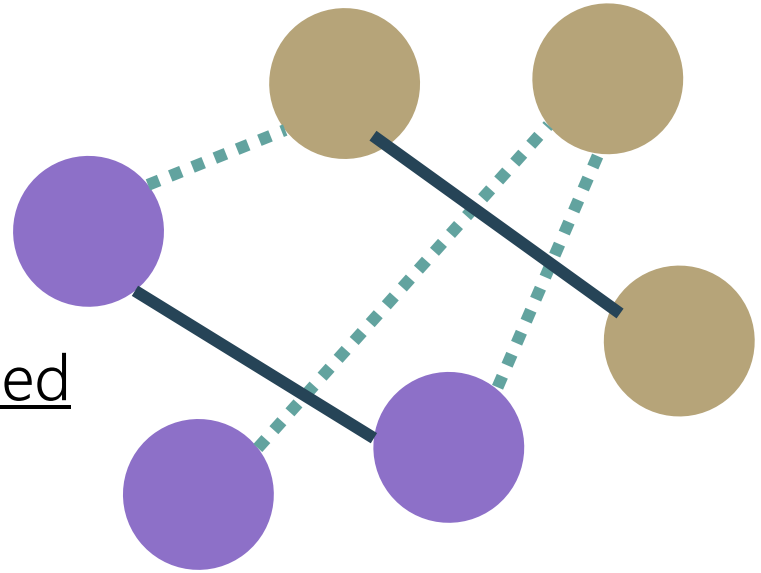
1. Find $\mathbb{E}[X]$. $\mathbb{E}[X] = \frac{n}{2}$ 2. Find $\mathrm{Var}(X)$. $\mathrm{Var}(X) = \frac{n}{4}$

2. Apply Chebyshev's Ineqality.

$$\mathbb{P}(X \leq n/3) = \mathbb{P}\left(X - \frac{n}{2} \leq \frac{n}{3} - \frac{n}{2}\right) \leq \mathbb{P}\left(X - \frac{n}{2} \leq -\frac{n}{6}\right) + \mathbb{P}\left(X - \frac{n}{2} \geq \frac{n}{6}\right)$$

$$\leq \mathbb{P}\left(\left|X - \frac{n}{2}\right| \geq \frac{n}{6}\right) \leq \frac{n/4}{(n/6)^2} = \frac{9}{n}$$

# Maximum Cut Problem



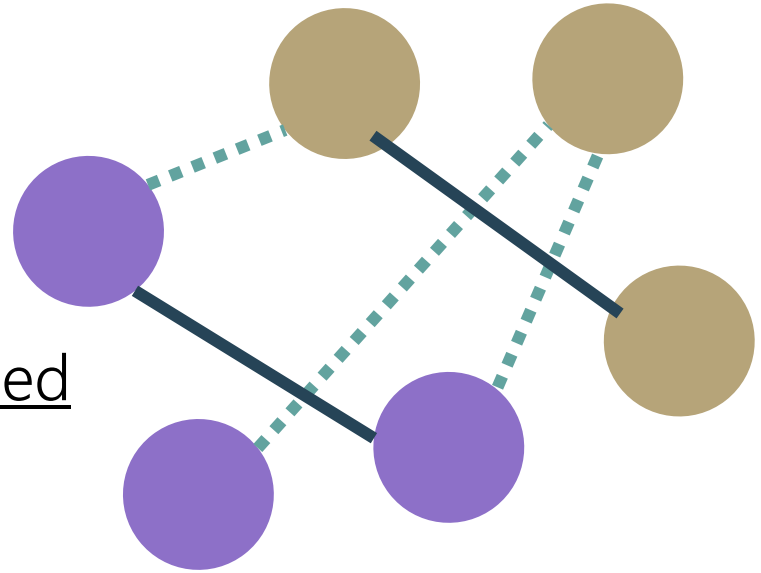
**The problem:** partition the vertices of a graph into two sets such that the number of edges between the sets is maximized

*Simple algorithm:*
*Each node goes to A or B with probability ½*

*Better, Las Vegas algorithm:*
*Keep doing this till there is a large cut found (i.e., $X \geq n/3$)*

*What is the probability that in the first 20 trials, we will have succeeded?*

# Maximum Cut Problem



**The problem:** partition the vertices of a graph into two sets such that the number of edges between the sets is maximized

*Simple algorithm:*
*Each node goes to A or B with probability ½*

*Better, Las Vegas algorithm:*
*Keep doing this till there is a large cut found (i.e., $X \geq n/3$)*

***What is the probability that in the first 20 trials, we will have succeeded?***

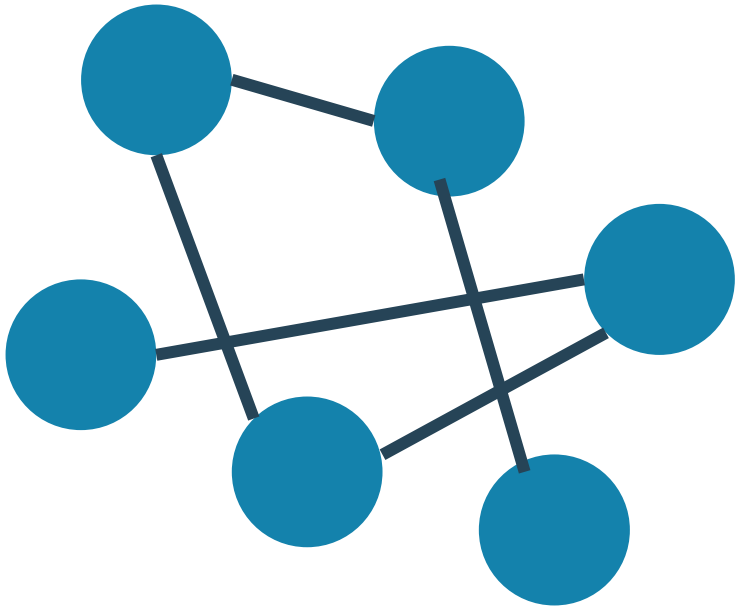
Let $X$ be the number of trials is takes. $X \sim \text{Geo}\left(p \leq \frac{9}{n}\right)$

So, $\mathbb{P}(X \leq 20) \leq 1 - \left(1 - \frac{9}{n}\right)^{20}$

# Graph Coloring Problem

**The problem:** color each node red, blue, or green, BUT minimize nodes with the same color sharing an edge (i.e., max. edges between distinct)

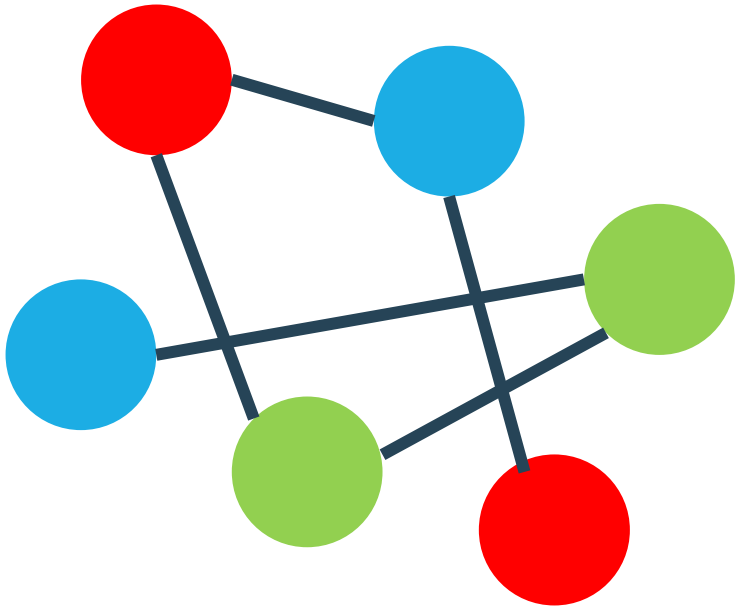
*real world examples: scheduling, coloring a map, sudoku solver, CPU allocation*



# Graph Coloring Problem

**The problem:** color each node red, blue, or green, BUT minimize nodes with the same color sharing an edge (i.e., max. edges between distinct)

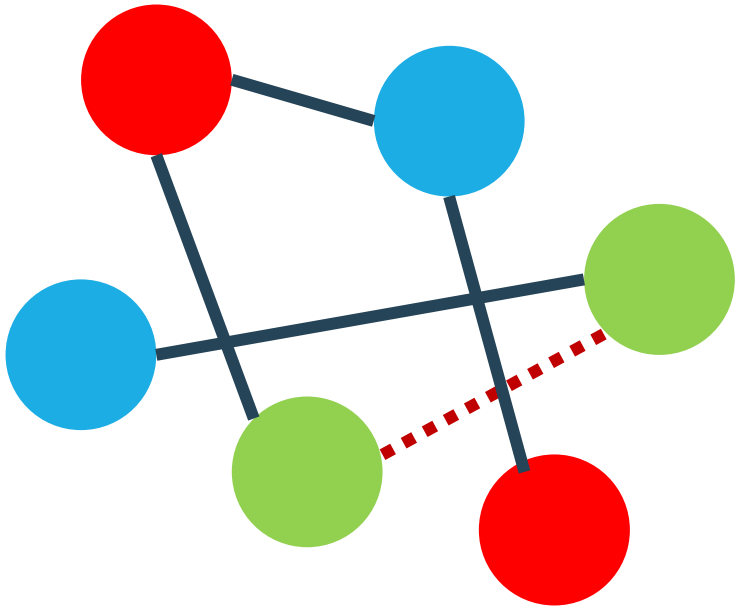
*real world examples: scheduling, coloring a map, sudoku solver, CPU allocation*



# Graph Coloring Problem

**The problem:** color each node red, blue, or green, BUT minimize nodes with the same color sharing an edge (i.e., max. edges between distinct)

*real world examples: scheduling, coloring a map, sudoku solver, CPU allocation*



# Graph Coloring Problem

**The problem:** color each node red, blue, or green, BUT minimize nodes with the same color sharing an edge (i.e., max. edges between distinct)

*real world examples: scheduling, coloring a map, sudoku solver, CPU allocation*

**Simple algorithm:** Randomly pick a color for each node

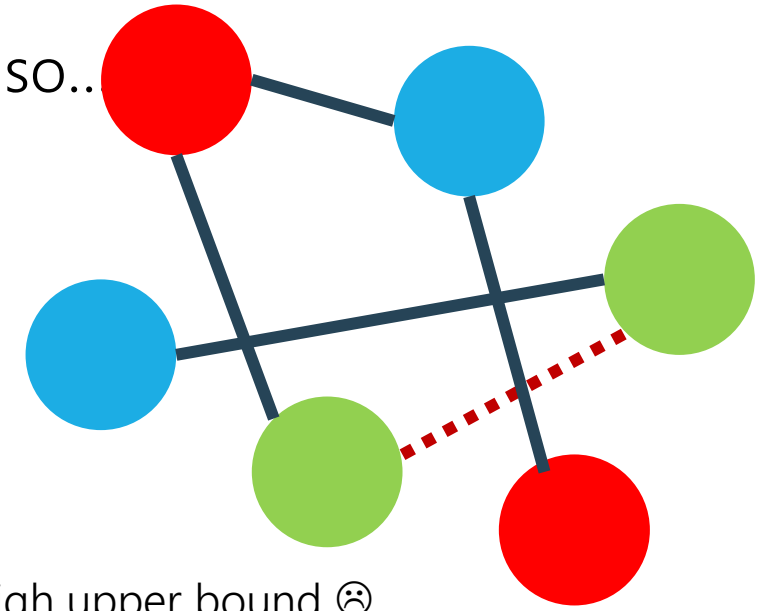
Probability of edge $e$ sharing a color (miscoloring) is $\frac{1}{3}$, so..

$\mathbb{E}[S_e] = \frac{1}{3}$ where $S_e$ is whether edge is miscolored

$S$ (num. of miscolored edges): $S = \sum_i^n S_e$ -->$\mathbb{E}[S] = \frac{n}{3}$

So, by Markov's inequality,

$$\mathbb{P}\left(S \geq 1.1 \cdot \frac{n}{3}\right) \leq \frac{n/3}{1.1 \cdot n/3} = \frac{1}{1.1} \approx 0.91$$



The probability of the algorithm miscoloring more than a third edges has a high upper bound ☹

# Graph Coloring Problem

**The problem:** color each node red, blue, or green, BUT minimize nodes with the same color sharing an edge (i.e., max. edges between distinct)
*real world examples: scheduling, coloring a map, sudoku solver, CPU allocation*

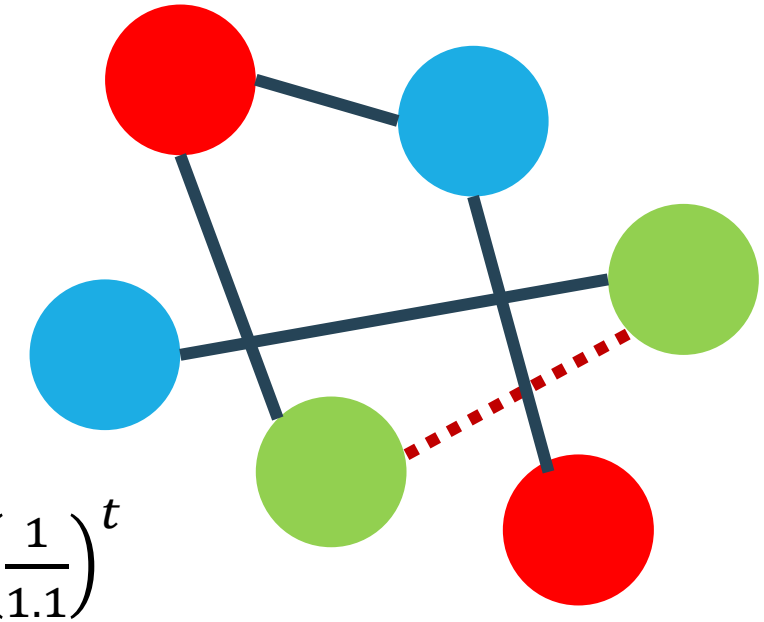
**Simple algorithm:** Randomly pick a color for each node

$S$ (num. of miscolored edges): $S = \sum_i^n S_e$ --> $\mathbb{E}[S] = \frac{n}{3}$

So, by Markov's inequality,

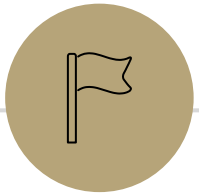
$$\mathbb{P}\left(S \geq 1.1 \cdot \frac{n}{3}\right) \leq \frac{n/3}{1.1 \cdot n/3} = \frac{1}{1.1} \approx 0.91$$

We can use a Monte Carlo algorithm!
Keep repeating the algorithm $t$ times.
Probability you **fail to find a good coloring** is at most $\left(\frac{1}{1.1}\right)^t$
probability is *very* low with high values of $t$



# If you like this kind of stuff…

- CSE 421 covers algorithms (like min cut, graph color, and more!)
- CSE 431 covers the theory behind this algorithms (which includes analysis of randomize algorithms!)



# Differential Privacy

# Privacy Preservation

*A real-world example (adapted from The Ethical Algorithm by Kearns and Roth; based on protocol by Warner [1965]).*

*And gives a sense of how randomness is actually used to protect privacy.*

# Privacy Preservation with Randomness

**You're working with a social scientist. They want to get accurate data on the rate at which people cheat on their romantic partners.**

We know about polling accuracy!

> Use CLT or a tail-bound to estimate the needed number $n$ get a guaranteed good estimate, right?

> Do a poll, call up a random sample of adults and ask them "have you ever cheated on your romantic partner?"

You do that, and somehow, no one says they cheated. I wonder why…

# What's the problem?

People lie.

Or they might be concerned about you keeping this data.

Databases can be leaked (or infiltrated. Or subpoenaed).

*You don't want to hold this data, and the people you're calling don't want you to hold this data.*

# Doing It Better!

You don't need to know **who** was cheating. Just how many people were.

Here's a protocol:

*Please flip a coin.*
*If the coin is heads, or you have ever cheated, please tell me 'heads'*
*If the coin is tails and you have not ever cheated, please tell me 'tails'*

We have two concerns with this:

> Will it now be private?

> Will we be able to make accurate estimates using this data?

# Will it be private?

*Please flip a coin.*

*If the coin is heads, or you have ever cheated, please tell me 'heads'*

*If the coin is tails and you have not ever cheated, please tell me 'tails'*

If you are someone who has cheated, and you report heads can that be used against you? Not substantially – just say "no the coin came up heads!"

You discover your partner said heads, what's the probability that they cheated?

# Will it be private?

If you are someone who has cheated on your spouse, and you report heads can that be used against you? Not substantially – just say "no the coin came up heads!"

$$\mathbb{P}(C|H) = \frac{\mathbb{P}(H|C)\cdot\mathbb{P}(C)}{\mathbb{P}(H)} = \frac{1\cdot\mathbb{P}(C)}{\frac{1}{2}\mathbb{P}(\overline{C}) + 1\cdot\mathbb{P}(C)}$$

Is this a substantial change?

No. For real world values (~15%) of $\mathbb{P}(C)$, the probability estimate would increase (to ~26%). But that isn't too damaging.

# But will it be accurate?

But we've lost our data haven't we? People answered a different question. Can we still estimate how many people cheated?

Suppose you poll $n$ people, and let $X$ be the number of people who said "heads" We'll find an estimate $Y$ of the number of people who cheated in the sample, and let $p$ be the true probability of cheating in the population. What should $Y$ be? Can we draw a margin of error around $Y$?

$$\mathbb{P}(X_i = 1) = \frac{1}{2} + \frac{1}{2} \cdot p$$

$$\mathbb{E}[X] = \frac{n}{2} + \frac{1}{2}\mathbb{E}[Y]$$

We'll define $Y$ to be: $Y = 2\left(X - \frac{n}{2}\right)$.

This is a definition, based on how the $\mathbb{E}[Y]$ should relate to the $\mathbb{E}[X]$.

# But will it be accurate?

$\mathbb{E}[X] = \frac{n}{2} + \frac{1}{2}\mathbb{E}[Y]$

$Y = 2\left(X - \frac{n}{2}\right)$

$\mathrm{Var}(X) = \mathrm{Var}(\sum X_i) = \sum \mathrm{Var}(X_i)$

$\mathrm{Var}(X_i)$? It's an indicator with parameter $p + (1-p) \cdot \frac{1}{2} = \frac{1}{2} + \frac{p}{2}$

So $\mathrm{Var}(X_i) = \left(\frac{1}{2} + \frac{p}{2}\right)\left(\frac{1}{2} - \frac{p}{2}\right)$

$\mathrm{Var}(Y) = 4\mathrm{Var}(X) = 4n\mathrm{Var}(X_i) = 4n\left(\frac{1}{2} + \frac{p}{2}\right)\left(\frac{1}{2} - \frac{p}{2}\right) \leq \frac{4n}{4} = n$

The variance is 4 times as much as it would have been for a non-anonymous poll.

# Can we use Chernoff?

## (Multiplicative) Chernoff Bound

Let $X_1, X_2, \ldots, X_n$ be *independent* Bernoulli random variables.
*Let* $X = \sum X_i$, and $\mu = \mathbb{E}[X]$. For any $0 \leq \delta \leq 1$

$$\mathbb{P}(X \geq (1+\delta)\mu) \leq \exp\left(-\frac{\delta^2\mu}{3}\right) \text{ and } \mathbb{P}(X \leq (1-\delta)\mu) \leq \exp\left(-\frac{\delta^2\mu}{2}\right)$$

What happens with $\mathbf{n = 1000}$ people?

What range will we be within at least $95\%$ of the time?

# ☹ Can't bound $\delta$ without bounding $p$

The right tail is the looser bound, so ensuring the right tail is less than 2.5% gives us the needed guarantee.

$$\mathbb{P}(X \geq (1+\delta)\mu) \leq \exp\left(-\frac{\delta^2\mu}{3}\right) = \exp\left(-\frac{\delta^2 1000p}{3}\right) \leq .025$$

$$-\frac{\delta^2 1000p}{3} \leq \ln(.025)$$

$$-\delta^2 \leq \frac{3\cdot\ln(.025)}{1000p}$$

$$\delta \geq \sqrt{\frac{-3\ln(.025)}{1000p}}$$

As $p \to 0$, $\delta \to \infty$ – we're not actually making a claim anymore.

# A different inequality

If we try to use Chernoff, we'll hit a frustrating block.

Since $\mu$ depends on $p$, $p$ appears in the formula for $\delta$. And we wouldn't get an absolute guarantee unless we could plug in a $p$.

And it'll turn out that as $p \to 0$ that $\delta \to \infty$ so we don't say anything then.

Luckily, there's always another bound…

# Hoeffding's Inequality

## Hoeffding's Inequality

Let $X_1, X_2, \ldots, X_n$ be *independent* RVs, each with range $[0,1]$.
*Let* $\bar{X} = \sum X_i/n$, and $\mu = \mathbb{E}[\bar{X}]$. For any $t \geq 0$

$$\mathbb{P}(|\bar{X} - \mathbb{E}[\bar{X}]| \geq t) \leq 2\exp(-2nt^2)$$

$|X - \mathbb{E}[X]| \geq t$ if and only if $|Y - \mathbb{E}[Y]| \geq 2t$. Why?

$$Y = 2\left(X - \frac{n}{2}\right) \text{ or } X = \frac{Y+n}{2}$$

$$|X - \mathbb{E}[X]|$$
$$= \left|\frac{Y+n}{2} - \mathbb{E}\left[\frac{Y+n}{2}\right]\right|$$
$$= \left|\frac{Y+n}{2} - \mathbb{E}\left[\frac{Y}{2}\right] - \frac{n}{2}\right|$$
$$= \left|\frac{Y}{2} - \mathbb{E}\left[\frac{Y}{2}\right]\right|$$
$$= \frac{1}{2}|Y - \mathbb{E}[Y]|$$

So $|X - \mathbb{E}[X]| \geq t$ if and only if $\frac{1}{2}|Y - \mathbb{E}[Y]| \geq t$ iff $|Y - \mathbb{E}[Y]| \geq 2t$.

# Hoeffding's Inequality

## Hoeffding's Inequality

Let $X_1, X_2, \ldots, X_n$ be *independent* RVs, each with range $[0,1]$.
*Let* $\bar{X} = \sum X_i/n$, and $\mu = \mathbb{E}[\bar{X}]$. For any $t \geq 0$

$$\mathbb{P}(|\bar{X} - \mathbb{E}[\bar{X}]| \geq t) \leq 2\exp(-2nt^2)$$

How close will we be with *n=1000 with probability at least* .95*?*

$|X - \mathbb{E}[X]| \geq t$ if and only if $|Y - \mathbb{E}[Y]| \geq 2t$.

# Margin of Error

$\mathbb{P}(|Y - \mathbb{E}[Y]| \geq t) = \mathbb{P}(|X - \mathbb{E}[X]| \geq t/2) \leq 2\exp(-2nt^2) \leq .05$

For $n = 1000$, we get:

$2\exp\left(-2n\left(\frac{t}{2}\right)^2\right) \leq .05 \Rightarrow -\frac{2000t^2}{4} \leq \ln(.025) \Rightarrow t \leq .086.$

$\mathbb{P}(|Y - \mathbb{E}[Y]| \geq .086) \leq .05$

So our margin of error is about 8.6%.

To get a margin-of-error of $5\%$ need $2\exp\left(-2n\left(\frac{.05}{2}\right)^2\right) \leq .05$

$n \geq 2952$

# How much do we lose?

We lose a factor of two in the length of the margin (equivalently, we'd need to talk to $4$ times as many people to have the same confidence.

You can also control this tradeoff.

Want more accuracy? Make it roll a die: report 1 if cheated (truth o/w)

Want more security? Make it Bernoulli with probability $p \gg \frac{1}{2}$ or cheated have the same report (e.g. report "die roll 1 [and didn't cheat]" or "die roll 2-6 [or did cheat]"

# In The Real World

Injecting randomness to preserve privacy is a real thing.

Instead of having everyone flip a coin, "random noise" can be inserted after all the data has been collected.

**Differential privacy** is being used to protect the 2020 Census data.

The overall count of people in each state is exact (well, exactly the data they collected). But the data per block or per city will be randomized to protect against .

This video nicely explains what's involved. Notice that the accuracy guarantees come in the same "inside-margin-of-error-with-probability" guarantees we've been giving for our randomness (just much stronger).