COMPSCI 514: ALGORITHMS FOR DATA SCIENCE

Prof. Cameron Musco

University of Massachusetts Amherst. Fall 2021.

Lecture 1

FACE MASKS

Masks covering your nose and mouth are required in class (and generally indoors at UMass) regardless of vaccination status.

- There is no exception for eating/drinking, so if you need to take a drink, please step outside briefly to do so.

MOTIVATION FOR THIS CLASS

People are increasingly interested in analyzing and learning from massive datasets.

People are increasingly interested in analyzing and learning from massive datasets.

- Twitter receives 6,000 tweets per second, 500 million/day. Google receives 60,000 searches per second, 5.6 billion/day.
  - How do they process them to target advertisements? To predict trends? To improve their products?

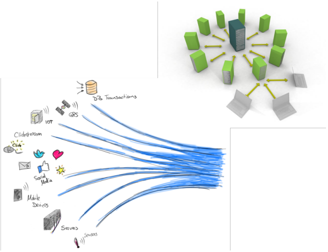
People are increasingly interested in analyzing and learning from massive datasets.

- Twitter receives 6,000 tweets per second, 500 million/day. Google receives 60,000 searches per second, 5.6 billion/day.
  - How do they process them to target advertisements? To predict trends? To improve their products?
- The Large Synoptic Survey Telescope will take high definition photographs of the sky, producing 15 terabytes of data/night.
  - How do they denoise and compress the images? How do they detect anomalies such as changing brightness or position of objects to alert researchers?

- Traditionally, algorithm design focuses on fast computation when data is stored in an efficiently accessible centralized manner (e.g., in RAM on a single machine).

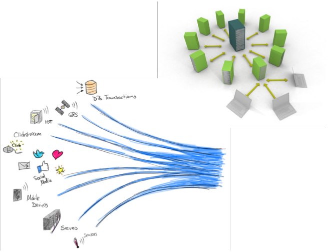
- Traditionally, algorithm design focuses on fast computation when data is stored in an efficiently accessible centralized manner (e.g., in RAM on a single machine).
- Massive data sets require storage in a distributed manner or processing in a continuous stream.

vs.



- Traditionally, algorithm design focuses on fast computation when data is stored in an efficiently accessible centralized manner (e.g., in RAM on a single machine).
- Massive data sets require storage in a distributed manner or processing in a continuous stream.

vs.



- Even ‘simple’ problems become very difficult in this setting.

For example:

For example:

- How can Twitter rapidly detect if an incoming Tweet is an exact duplicate of another Tweet made in the last year? Given that no machine can store all Tweets made in a year.

For example:

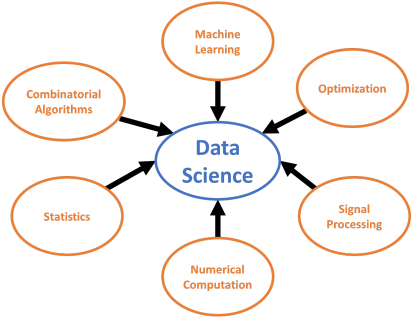
- How can Twitter rapidly detect if an incoming Tweet is an exact duplicate of another Tweet made in the last year? Given that no machine can store all Tweets made in a year.
- How can Google estimate the number of unique search queries that are made in a given week? Given that no machine can store the full list of queries.

For example:

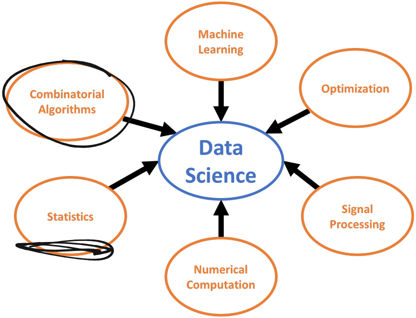
- How can Twitter rapidly detect if an incoming Tweet is an exact duplicate of another Tweet made in the last year? Given that no machine can store all Tweets made in a year.
- How can Google estimate the number of unique search queries that are made in a given week? Given that no machine can store the full list of queries.
- When you use Shazam to identify a song from a recording, how does it provide an answer in < 10 seconds, without scanning over all $\sim$ 8 million audio files in its database.

A Second Motivation: Data Science is highly interdisciplinary.

A Second Motivation: Data Science is highly interdisciplinary.



A Second Motivation: Data Science is highly interdisciplinary.



- Many techniques that aren't covered in the traditional CS algorithms curriculum.
- Emphasis on building comfort with mathematical tools that underly data science and machine learning.

WHAT WE'LL COVER

Section 1: Randomized Methods & Sketching

Section 1: Randomized Methods & Sketching

How can we efficiently compress large data sets in a way that lets us answer important algorithmic questions rapidly?

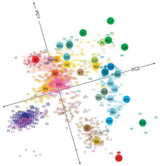
Section 1: Randomized Methods & Sketching

How can we efficiently compress large data sets in a way that lets us answer important algorithmic questions rapidly?

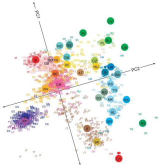
- Probability tools and concentration inequalities.
- Randomized hashing for efficient lookup, load balancing, and estimation. Bloom filters.
- Locality sensitive hashing and nearest neighbor search.
- Streaming algorithms: identifying frequent items in a data stream, counting distinct items, etc.
- Random compression of high-dimensional vectors: the Johnson-Lindenstrauss lemma, applications, and connections to the weirdness of high-dimensional geometry.

WHAT WE'LL COVER

Section 2: Spectral Methods

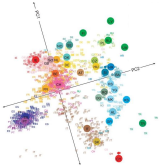


Section 2: Spectral Methods



How do we identify the most important features of a dataset using linear algebraic techniques?

Section 2: Spectral Methods

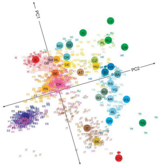


How do we identify the most important features of a dataset using linear algebraic techniques?

- Principal component analysis, low-rank approximation, dimensionality reduction.
- The singular value decomposition (SVD) and its applications to PCA, low-rank approximation, LSI, MDS, …
- Spectral graph theory. Spectral clustering, community detection, network visualization.
- Computing the SVD on large datasets via iterative methods.

WHAT WE’LL COVER

Section 2: Spectral Methods



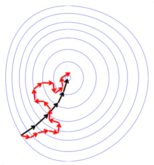
How do we identify the most important features of a dataset using linear algebraic techniques?

If you open up the codes that are underneath [most data science applications] this is all linear algebra on arrays.

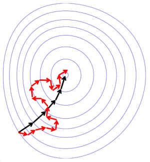
– Michael Stonebraker

WHAT WE'LL COVER

Section 3: Optimization

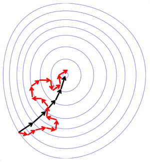


Section 3: Optimization



Fundamental continuous optimization approaches that drive methods in machine learning and statistics.

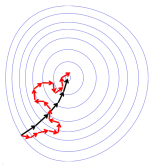
Section 3: Optimization



Fundamental continuous optimization approaches that drive methods in machine learning and statistics.

- Gradient descent. Analysis for convex functions.
- Stochastic and online gradient descent.
- Focus on convergence analysis.

Section 3: Optimization



Fundamental continuous optimization approaches that drive methods in machine learning and statistics.

- Gradient descent. Analysis for convex functions.
- Stochastic and online gradient descent.
- Focus on convergence analysis.

A small taste of what you can find in COMPSCI 590OP or 690OP.

IMPORTANT TOPICS WE WON'T COVER

IMPORTANT TOPICS WE WON'T COVER

- Systems/Software Tools.

IMPORTANT TOPICS WE WON'T COVER

- Systems/Software Tools.

  - COMPSCI 532: Systems for Data Science

IMPORTANT TOPICS WE WON'T COVER

- **Systems/Software Tools.**
  - COMPSCI 532: Systems for Data Science
- **Machine Learning/Data Analysis Methods and Models.**
  - E.g., regression methods, kernel methods, random forests, SVM, deep neural networks.

IMPORTANT TOPICS WE WON'T COVER

- **Systems/Software Tools.**
  - COMPSCI 532: Systems for Data Science
- **Machine Learning/Data Analysis Methods and Models.**
  - E.g., regression methods, kernel methods, random forests, SVM, deep neural networks.
  - COMPSCI 589/689: Machine Learning

STYLE OF THE COURSE

This is a theory course.

STYLE OF THE COURSE

This is a theory course.

- Build general mathematical tools and algorithmic strategies that can be applied to a wide range of problems.

STYLE OF THE COURSE

This is a theory course.

- Build general mathematical tools and algorithmic strategies that can be applied to a wide range of problems.
- Assignments will emphasize algorithm design, correctness proofs, and asymptotic analysis (minimal required coding).

STYLE OF THE COURSE

This is a theory course.

- Build general mathematical tools and algorithmic strategies that can be applied to a wide range of problems.
- Assignments will emphasize algorithm design, correctness proofs, and asymptotic analysis (minimal required coding).
- The homework is designed to make you think beyond what is taught in class. You will get stuck, and not see the solutions right away. This is a great (the only?) way to build mathematical and algorithm design skills.

STYLE OF THE COURSE

This is a theory course.

- Build general mathematical tools and algorithmic strategies that can be applied to a wide range of problems.
- Assignments will emphasize algorithm design, correctness proofs, and asymptotic analysis (minimal required coding).
- The homework is designed to make you think beyond what is taught in class. You will get stuck, and not see the solutions right away. This is a great (the only?) way to build mathematical and algorithm design skills.
- A strong algorithms and mathematical background (particularly in linear algebra and probability) **are required**.
- UMass prereqs: COMPSCI 240 and COMPSCI 311.

This is a theory course.

- Build general mathematical tools and algorithmic strategies that can be applied to a wide range of problems.
- Assignments will emphasize algorithm design, correctness proofs, and asymptotic analysis (minimal required coding).
- The homework is designed to make you think beyond what is taught in class. You will get stuck, and not see the solutions right away. This is a great (the only?) way to build mathematical and algorithm design skills.
- A strong algorithms and mathematical background (particularly in linear algebra and probability) **are required**.
- UMass prereqs: COMPSCI 240 and COMPSCI 311.

For example: Baye's rule in conditional probability. What it means for a vector x to be an eigenvector of a matrix A, orthogonal projection, greedy algorithms, divide-and-conquer algorithms.

See course webpage for logistics, policies, lecture notes, assignments, etc.:

http://people.cs.umass.edu/~cmusco/CS514F21/

See Moodle page for this link if you lose it.

Professor: Cameron Musco

- Email: cmusco@cs.umass.edu
- Office Hours: Over Zoom, Tuesdays, 2:30pm-3:30pm (directly after class). See website for Zoom link. ?
- I encourage you to come as regularly as possible to ask questions/work together on practice problems.
- If you need to chat individually, please email meet to set up a time.

TAs:

- Pratheba Selvaraju
- Shiv Shankar
- Weronika Nguyen

See website for office hours and contact info.

There is also an online version of 514 taught this semester by Andrew McGregor.

- The sections will closely parallel each other, and share the same TAs.
- You may attend Prof. McGregor's lectures and office hours (both over Zoom) if it is helpful.
- See course webpage for schedule and Moodle for Zoom links.

We will use Piazza for class discussion and questions.

- See website for link to sign up.

We will use Piazza for class discussion and questions.

- See website for link to sign up.

You may earn up to 5% extra credit for participation.

- Asking good clarifying questions and answering questions during the lecture or on Piazza.
- Actively participating in office hours.
- Answering other students' or instructor questions on Piazza.
- Posting helpful/interesting links on Piazza, e.g., resources that cover class material, research articles related to the topics covered in class, etc.

We will use material from two textbooks (links to free online versions on the course webpage): *Foundations of Data Science* and *Mining of Massive Datasets*, but will follow neither closely.

- I will post optional readings a few days prior to each class.
- Lecture notes will be posted before each class, and annotated notes posted after class.
- Recordings of the live lectures will also be posted.

We will have 5 problem sets, which you may complete in **groups of up to 3 students**.

HOMEWORK

We will have 5 problem sets, which you may complete in **groups of up to 3 students**.

- We strongly encourage working in groups, as it will make completing the problem sets much easier/more educational.
- Collaboration with students outside your group is limited to discussion at a high level. You may not work through problems in detail or write up solutions together.
- See Piazza for a thread to help you organize groups.

We will have 5 problem sets, which you may complete in **groups of up to 3 students**.

- We strongly encourage working in groups, as it will make completing the problem sets much easier/more educational.
- Collaboration with students outside your group is limited to discussion at a high level. You may not work through problems in detail or write up solutions together.
- See Piazza for a thread to help you organize groups.

Problem set submissions will be via Gradescope.

- See website for a link to join. **Entry Code: P5NKXN**

I will release a multiple choice quiz in Moodle each Thursday after lecture, due the next Monday at 8pm.

I will release a multiple choice quiz in Moodle each Thursday after lecture, due the next Monday at 8pm.

- Designed as a check-in that you are following the material, and to help me make adjustments as needed.
- Will take around 15-30 minutes per week, open notes.
- Will also include free response check-in questions to get your feedback on how the course is going, what material from the past week you find most confusing, interesting, etc.

Grade Breakdown:

- Problem Sets (5 total): 40%, weighted equally.
- Weekly Quizzes: 10%, weighted equally.
- Midterm (October 19th, in class): 25%.
- Final (December 16th, 10:30am - 12:30pm): 25%.
- Extra Credit: Up to 5% for participation, and lots more available on problem sets, for questions asked in class, etc.

Grade Breakdown:

- Problem Sets (5 total): 40%, weighted equally.
- Weekly Quizzes: 10%, weighted equally.
- Midterm (October 19th, in class): 25%.
- Final (December 16th, 10:30am - 12:30pm): 25%.
- Extra Credit: Up to 5% for participation, and lots more available on problem sets, for questions asked in class, etc.

Academic Honesty:

- A first violation cheating on a homework, quiz, or other assignment will result in a 0 on that assignment.
- A second violation, or cheating on an exam will result in failing the class.
- For fairness, I adhere very strictly to these policies.

DISABILITY SERVICES AND ACCOMODATIONS

UMass Amherst is committed to making reasonable, effective, and appropriate accommodations to meet the needs to students with disabilities.

- If you have a documented disability **on file with Disability Services**, you may be eligible for reasonable accommodations in this course.
- If your disability requires an accommodation, please email me by **next Thursday 9/9** so that we can make arrangements.

UMass Amherst is committed to making reasonable, effective, and appropriate accommodations to meet the needs to students with disabilities.

- If you have a documented disability **on file with Disability Services**, you may be eligible for reasonable accommodations in this course.
- If your disability requires an accommodation, please email me by **next Thursday 9/9** so that we can make arrangements.

I understand that people have different learning needs, home situations, etc. If something isn't working for you in the class, please reach out and let's try to work it out.

Questions?

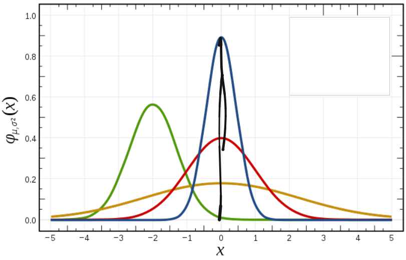
Section 1: Randomized Methods & Sketching

Consider a random X variable taking values in some finite set $S \subset \mathbb{R}$. E.g., for a random dice roll, $S = \{1, 2, 3, 4, 5, 6\}$.

Consider a random **X** variable taking values in some finite set $S \subset \mathbb{R}$. E.g., for a random dice roll, $S = \{1, 2, 3, 4, 5, 6\}$.

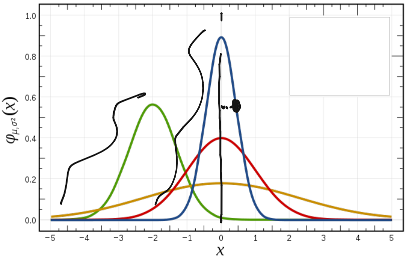
- **Expectation:** $\mathbb{E}[\mathbf{X}] = \sum_{s \in S} \Pr(\mathbf{X} = s) \cdot s$.

$E[X] = \frac{1}{6} \cdot 1 + \frac{1}{6} \cdot 2 + \ldots \frac{1}{6} \cdot 6 = 3.5$



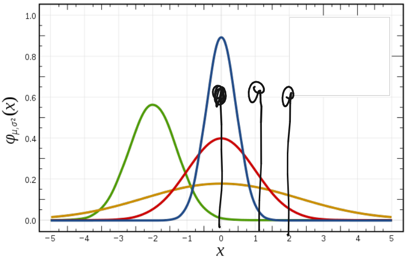
Consider a random **X** variable taking values in some finite set $S \subset \mathbb{R}$. E.g., for a random dice roll, $S = \{1, 2, 3, 4, 5, 6\}$.

- **Expectation:** $\mathbb{E}[\mathbf{X}] = \sum_{s \in S} \Pr(\mathbf{X} = s) \cdot s$.
- **Variance:** $\text{Var}[\mathbf{X}] = \mathbb{E}[(\mathbf{X} - \mathbb{E}[\mathbf{X}])^2]$.



Consider a random **X** variable taking values in some finite set $S \subset \mathbb{R}$. E.g., for a random dice roll, $S = \{1, 2, 3, 4, 5, 6\}$.

- **Expectation:** $\mathbb{E}[\mathbf{X}] = \sum_{s \in S} \Pr(\mathbf{X} = s) \cdot s$.
- **Variance:** $\text{Var}[\mathbf{X}] = \mathbb{E}[(\mathbf{X} - \mathbb{E}[\mathbf{X}])^2]$.



Exercise: Show that for any scalar α, $\mathbb{E}[\alpha \cdot \mathbf{X}] = \alpha \cdot \mathbb{E}[\mathbf{X}]$ and $\text{Var}[\alpha \cdot \mathbf{X}] = \alpha^2 \cdot \text{Var}[\mathbf{X}]$.

Consider two random events A and B.

$A \cap B$: event that both events *A and B* happen.

Consider two random events A and B. P(A)

- **Conditional Probability:**

$$\Pr(A|B) = \frac{\Pr(A \cap B)}{\Pr(B)}.$$

$A \cap B$: event that both events *A and B* happen.

Consider two random events A and B.

- **Conditional Probability:**

$$\Pr(A|B) = \frac{\Pr(A \cap B)}{\Pr(B)}.$$

- **Independence:** A and B are independent if:

$$\Pr(A|B) = \Pr(A).$$

$A \cap B$: event that both events A *and* B happen.

Consider two random events A and B.

- **Conditional Probability:**

$$\Pr(A|B) = \frac{\Pr(A \cap B)}{\Pr(B)}.$$

- **Independence:** A and B are independent if:

$$\Pr(A|B) = \Pr(A).$$

Using the definition of conditional probability, independence means:

$$\frac{\Pr(A \cap B)}{\Pr(B)} = \Pr(A) \implies \Pr(A \cap B) = \Pr(A) \cdot \Pr(B).$$

$A \cap B$: event that both events A *and* B happen.

For Example: What is the probability that for two independent dice rolls the first is a 6 and the second is odd?

$$P(D_1 = 6 \cap D_2 \in \{1,3,5\})$$

$$P(D_1 = 6) \cdot P(D_2 \in \{1,3,5\})$$

$$\frac{1}{6} \cdot \frac{1}{2} = \frac{1}{12}$$

Independent Random Variables: Two random variables X, Y are independent if for all s, t, $X = s$ and $Y = t$ are independent events. In other words:

independent: $\Pr(X = s \cap Y = t) = \Pr(X = s) \cdot \Pr(Y = t)$.

uncorrelated: $\mathbb{E}[XY] = \mathbb{E}[X] \cdot \mathbb{E}[Y]$

Think-Pair-Share: When are the expectation and variance linear?

I.e., under what conditions on X and Y do we have:

$$\mathbb{E}[X+Y] = \mathbb{E}[X] + \mathbb{E}[Y]$$

and

$$\text{Var}[X+Y] = \text{Var}[X] + \text{Var}[Y].$$

X, Y: any two random variables.

- ~~when independent~~

$$\mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y]$$

$\mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y]$ for any random variables X and Y.

$\mathbb{E}[X+Y] = \mathbb{E}[X] + \mathbb{E}[Y]$ for any random variables X and Y.

Proof:

$\mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y]$ for any random variables X and Y.

Proof:

$$\mathbb{E}[X + Y] = \sum_{s \in S} \sum_{t \in T} \Pr(X = s \cap Y = t) \cdot (s + t)$$

$\mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y]$ for any random variables X and Y.

Proof:

$$\mathbb{E}[X+Y] = \sum_{s \in S} \sum_{t \in T} \Pr(X = s \cap Y = t) \cdot (s + t)$$
$$= \sum_{s \in S} \sum_{t \in T} \Pr(X = s \cap Y = t) \cdot s + \sum_{t \in T} \sum_{s \in S} \Pr(X = s \cap Y = t) \cdot t$$

$\mathbb{E}[X+Y] = \mathbb{E}[X] + \mathbb{E}[Y]$ for any random variables X and Y.

Proof:

$$\mathbb{E}[X+Y] = \sum_{s\in S}\sum_{t\in T} \Pr(X = s \cap Y = t)\cdot(s+t)$$

$$= \sum_{s\in S}\underbrace{\sum_{t\in T} \Pr(X = s \cap Y = t)}_{\Pr(X=s)}\cdot s + \sum_{t\in T}\underbrace{\sum_{s\in S} \Pr(X = s \cap Y = t)}_{\Pr(y=t)}\cdot t$$

$\mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y]$ for any random variables X and Y.

Proof:

$$\mathbb{E}[X+Y] = \sum_{s \in S} \sum_{t \in T} \Pr(X = s \cap Y = t) \cdot (s + t)$$

$$= \sum_{s \in S} \sum_{t \in T} \Pr(X = s \cap Y = t) \cdot s + \sum_{t \in T} \sum_{s \in S} \Pr(X = s \cap Y = t) \cdot t$$

$$= \sum_{s \in S} \Pr(X = s) \cdot s + \sum_{t \in T} \Pr(Y = t) \cdot t$$

$\mathbb{E}[X] + \mathbb{E}[Y]$ (law of total probability)

$\mathbb{E}[X+Y] = \frac{1}{2}\cdot 0 + \frac{1}{2}\cdot 2 = 1$

$\mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y]$ for any random variables X and Y.

Proof:

$$\begin{aligned}
\mathbb{E}[X+Y] &= \sum_{s\in S}\sum_{t\in T} \Pr(X = s \cap Y = t)\cdot(s+t) \\
&= \sum_{s\in S}\sum_{t\in T} \Pr(X = s \cap Y = t)\cdot s + \sum_{t\in T}\sum_{s\in S} \Pr(X = s \cap Y = t)\cdot t \\
&= \sum_{s\in S} \Pr(X = s)\cdot s + \sum_{t\in T} \Pr(Y = t)\cdot t
\end{aligned}$$

(law of total probability)

$$= \mathbb{E}[X] + \mathbb{E}[Y].$$

$X = 0$ w.p $\frac{1}{2}$, 1 w.p $\frac{1}{2}$

$Y = 0$ w.p $\frac{1}{2}$, 1 w.p. $\frac{1}{2}$

$\mathbb{E}[X+Y] = \frac{1}{4}\cdot 0 + \frac{1}{4}\cdot 1 + \frac{1}{4}\cdot 1 + \frac{1}{4}\cdot 2$

$= 1$

$$\mathrm{Var}[X + Y] = \mathrm{Var}[X] + \mathrm{Var}[Y]$$ → when X and Y are independent

$Y = X$

$\mathrm{Var}(X+Y) = \mathrm{Var}(2X) = 4 \cdot \mathrm{Var}(X)$

$\text{Var}[X + Y] = \text{Var}[X] + \text{Var}[Y]$ when X and Y are independent.

$\text{Var}[X + Y] = \text{Var}[X] + \text{Var}[Y]$ when X and Y are independent.

Claim 1: (exercise) $\text{Var}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$ $= \mathbb{E}[(X - \mathbb{E}X)^2]$

$\text{Var}[X+Y] = \text{Var}[X] + \text{Var}[Y]$ when X and Y are independent.

Claim 1: (exercise) $\text{Var}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$ (via linearity of expectation)

$\text{Var}[X+Y] = \text{Var}[X] + \text{Var}[Y]$ when X and Y are independent.

Claim 1: (exercise) $\text{Var}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$ (via linearity of expectation)

Claim 2: (exercise) $\mathbb{E}[XY] = \mathbb{E}[X] \cdot \mathbb{E}[Y]$ (i.e., X and Y are uncorrelated) when X, Y are independent.

$\mathrm{Var}[X + Y] = \mathrm{Var}[X] + \mathrm{Var}[Y]$ when X and Y are independent.

Claim 1: (exercise) $\mathrm{Var}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$ (via linearity of expectation)

Claim 2: (exercise) $\mathbb{E}[XY] = \mathbb{E}[X] \cdot \mathbb{E}[Y]$ (i.e., X and Y are uncorrelated) when X, Y are independent.

Together give:

$\text{Var}[X+Y] = \text{Var}[X] + \text{Var}[Y]$ when X and Y are independent.

Claim 1: (exercise) $\text{Var}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$ (via linearity of expectation)

Claim 2: (exercise) $\mathbb{E}[XY] = \mathbb{E}[X] \cdot \mathbb{E}[Y]$ (i.e., X and Y are uncorrelated) when X, Y are independent.

Together give:

$$\text{Var}[X+Y] = \mathbb{E}[(X+Y)^2] - \mathbb{E}[X+Y]^2$$

$\text{Var}[X + Y] = \text{Var}[X] + \text{Var}[Y]$ when X and Y are independent.

Claim 1: (exercise) $\text{Var}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$ (via linearity of expectation)

Claim 2: (exercise) $\mathbb{E}[XY] = \mathbb{E}[X] \cdot \mathbb{E}[Y]$ (i.e., X and Y are uncorrelated) when X, Y are independent.

Together give: $\mathbb{E}[(X^2 + 2XY + Y^2)]$

$$\text{Var}[X + Y] = \mathbb{E}[(X + Y)^2] - \mathbb{E}[X + Y]^2$$

$$= \mathbb{E}[X^2] + 2\mathbb{E}[XY] + \mathbb{E}[Y^2] - (\mathbb{E}[X] + \mathbb{E}[Y])^2$$

(linearity of expectation)

$\text{Var}[X+Y] = \text{Var}[X] + \text{Var}[Y]$ when X and Y are independent.

Claim 1: (exercise) $\text{Var}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$ (via linearity of expectation)

Claim 2: (exercise) $\mathbb{E}[XY] = \mathbb{E}[X] \cdot \mathbb{E}[Y]$ (i.e., X and Y are uncorrelated) when X, Y are independent.

Together give:

$$\text{Var}[X+Y] = \mathbb{E}[(X+Y)^2] - \mathbb{E}[X+Y]^2$$

$$= \mathbb{E}[X^2] + 2\mathbb{E}[XY] + \mathbb{E}[Y^2] - (\mathbb{E}[X] + \mathbb{E}[Y])^2$$

(linearity of expectation)

$$= \mathbb{E}[X^2] + 2\mathbb{E}[XY] + \mathbb{E}[Y^2] - \mathbb{E}[X]^2 - 2\mathbb{E}[X] \cdot \mathbb{E}[Y] - \mathbb{E}[Y]^2$$

$\text{Var}(X) + \text{Var}(Y) + 2E[XY] - 2E[X] \cdot E[Y]$

$\text{Var}[X + Y] = \text{Var}[X] + \text{Var}[Y]$ when X and Y are independent.

Claim 1: (exercise) $\text{Var}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$ (via linearity of expectation)

Claim 2: (exercise) $\mathbb{E}[XY] = \mathbb{E}[X] \cdot \mathbb{E}[Y]$ (i.e., X and Y are uncorrelated) when X, Y are independent.

Together give:

$$\begin{aligned}\text{Var}[X + Y] &= \mathbb{E}[(X + Y)^2] - \mathbb{E}[X + Y]^2 \\ &= \mathbb{E}[X^2] + 2\mathbb{E}[XY] + \mathbb{E}[Y^2] - (\mathbb{E}[X] + \mathbb{E}[Y])^2 \\ &\qquad\qquad\qquad \text{(linearity of expectation)} \\ &= \mathbb{E}[X^2] + 2\mathbb{E}[XY] + \mathbb{E}[Y^2] - \mathbb{E}[X]^2 - 2\mathbb{E}[X] \cdot \mathbb{E}[Y] - \mathbb{E}[Y]^2\end{aligned}$$

$\text{Var}[X+Y] = \text{Var}[X] + \text{Var}[Y]$ when X and Y are independent.

Claim 1: (exercise) $\text{Var}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$ (via linearity of expectation)

Claim 2: (exercise) $\mathbb{E}[XY] = \mathbb{E}[X] \cdot \mathbb{E}[Y]$ (i.e., X and Y are uncorrelated) when X, Y are independent.

Together give:

$$\begin{aligned}
\text{Var}[X+Y] &= \mathbb{E}[(X+Y)^2] - \mathbb{E}[X+Y]^2 \\
&= \mathbb{E}[X^2] + 2\mathbb{E}[XY] + \mathbb{E}[Y^2] - (\mathbb{E}[X] + \mathbb{E}[Y])^2 \\
&\qquad \text{(linearity of expectation)} \\
&= \mathbb{E}[X^2] + 2\mathbb{E}[XY] + \mathbb{E}[Y^2] - \mathbb{E}[X]^2 - 2\mathbb{E}[X] \cdot \mathbb{E}[Y] - \mathbb{E}[Y]^2 \\
&= \mathbb{E}[X^2] + \mathbb{E}[Y^2] - \mathbb{E}[X]^2 - \mathbb{E}[Y]^2
\end{aligned}$$

$\mathrm{Var}[X+Y] = \mathrm{Var}[X] + \mathrm{Var}[Y]$ when X and Y are independent.

Claim 1: (exercise) $\mathrm{Var}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$ (via linearity of expectation)

Claim 2: (exercise) $\mathbb{E}[XY] = \mathbb{E}[X] \cdot \mathbb{E}[Y]$ (i.e., X and Y are uncorrelated) when X, Y are independent.

Together give:

$$\begin{aligned}
\mathrm{Var}[X+Y] &= \mathbb{E}[(X+Y)^2] - \mathbb{E}[X+Y]^2 \\
&= \mathbb{E}[X^2] + 2\mathbb{E}[XY] + \mathbb{E}[Y^2] - (\mathbb{E}[X] + \mathbb{E}[Y])^2 \\
&\qquad \text{(linearity of expectation)} \\
&= \mathbb{E}[X^2] + 2\mathbb{E}[XY] + \mathbb{E}[Y^2] - \mathbb{E}[X]^2 - 2\mathbb{E}[X] \cdot \mathbb{E}[Y] - \mathbb{E}[Y]^2 \\
&= \mathbb{E}[X^2] + \mathbb{E}[Y^2] - \mathbb{E}[X]^2 - \mathbb{E}[Y]^2 \\
&= \mathrm{Var}[X] + \mathrm{Var}[Y].
\end{aligned}$$

Questions?

You have contracted with a new company to provide CAPTCHAS for your website.

You have contracted with a new company to provide CAPTCHAS for your website.

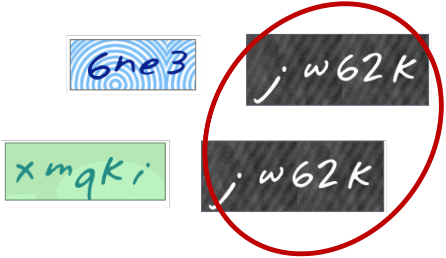
- They claim that they have a database of $1,000,000$ unique CAPTCHAS. A random one is chosen for each security check.
- You want to independently verify this claimed database size.

You have contracted with a new company to provide CAPTCHAS for your website.

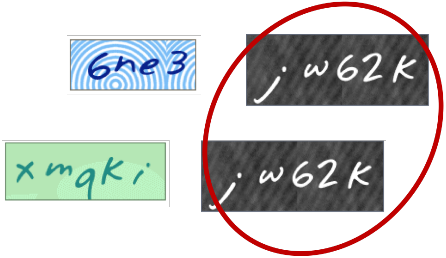
- They claim that they have a database of $1,000,000$ unique CAPTCHAS. A random one is chosen for each security check.
- You want to independently verify this claimed database size.
- You could make test checks until you see $1,000,000$ unique CAPTCHAS: would take $\geq 1,000,000$ checks!

AN ALGORITHMIC APPLICATION

An Idea: You run some test security checks and see if any duplicate CAPTCHAS show up. If you're seeing duplicates after not too many checks, the database size is probably not too big.

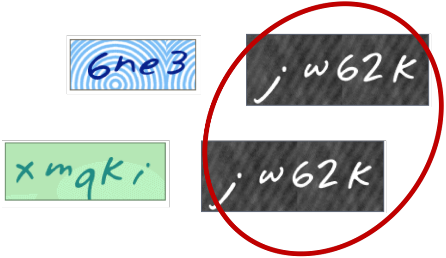


An Idea: You run some test security checks and see if any duplicate CAPTCHAS show up. If you're seeing duplicates after not too many checks, the database size is probably not too big.



'Mark and recapture' method in ecology.

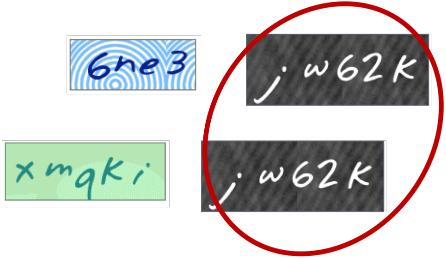
An Idea: You run some test security checks and see if any duplicate CAPTCHAS show up. If you're seeing duplicates after not too many checks, the database size is probably not too big.



'Mark and recapture' method in ecology.

If you run m security checks, and there are n unique CAPTCHAS, how many pairwise duplicates do you see in expectation?

An Idea: You run some test security checks and see if any duplicate CAPTCHAS show up. If you're seeing duplicates after not too many checks, the database size is probably not too big.



'Mark and recapture' method in ecology.

If you run m security checks, and there are n unique CAPTCHAS, how many pairwise duplicates do you see in expectation?

If e.g. the same CAPTCHA shows up three times, on your i^{th}, j^{th}, and k^{th} test, this is three duplicates: (i, j), (i, k) and (j, k).

Let $\mathbf{D}_{i,j} = 1$ if tests i and j give the same CAPTCHA, and 0 otherwise. An indicator random variable.

n: number of CAPTCHAS in database, m: number of random CAPTCHAS drawn to check database size, $\mathbf{D}$: number of pairwise duplicates in m random CAPTCHAS

Let $\mathsf{D}_{i,j} = 1$ if tests i and j give the same CAPTCHA, and 0 otherwise. An indicator random variable.

The number of pairwise duplicates (a random variable) is:

$$\mathsf{D} = \sum_{i,j \in [m]} \mathsf{D}_{i,j}.$$

n: number of CAPTCHAS in database, m: number of random CAPTCHAS drawn to check database size, D: number of pairwise duplicates in m random CAPTCHAS

Let $\mathsf{D}_{i,j} = 1$ if tests i and j give the same CAPTCHA, and 0 otherwise. An indicator random variable.

The number of pairwise duplicates (a random variable) is:

$$\mathbb{E}[\mathsf{D}] = \sum_{i,j \in [m]} \mathbb{E}[\mathsf{D}_{i,j}].$$

n: number of CAPTCHAS in database, m: number of random CAPTCHAS drawn to check database size, D: number of pairwise duplicates in m random CAPTCHAS

Let $\mathsf{D}_{i,j} = 1$ if tests i and j give the same CAPTCHA, and 0 otherwise. An indicator random variable.

The number of pairwise duplicates (a random variable) is:

$$\mathbb{E}[\mathsf{D}] = \sum_{i,j \in [m]} \mathbb{E}[\mathsf{D}_{i,j}].$$

For any pair $i, j \in [m]$: $\mathbb{E}[\mathsf{D}_{i,j}] = \Pr[\mathsf{D}_{i,j} = 1] = \frac{1}{n}$.

n: number of CAPTCHAS in database, m: number of random CAPTCHAS drawn to check database size, D: number of pairwise duplicates in m random CAPTCHAS

Let $\mathsf{D}_{i,j} = 1$ if tests i and j give the same CAPTCHA, and 0 otherwise. An indicator random variable.

The number of pairwise duplicates (a random variable) is:

$$\mathbb{E}[\mathsf{D}] = \sum_{i,j \in [m]} \mathbb{E}[\mathsf{D}_{i,j}].$$

For any pair $i, j \in [m]$: $\mathbb{E}[\mathsf{D}_{i,j}] = \Pr[\mathsf{D}_{i,j} = 1] = \frac{1}{n}$.

$$\mathbb{E}[\mathsf{D}] = \sum_{i,j \in [m]} \frac{1}{n} = \frac{\binom{m}{2}}{n} = \frac{m(m-1)}{2n}.$$

n: number of CAPTCHAS in database, m: number of random CAPTCHAS drawn to check database size, D: number of pairwise duplicates in m random CAPTCHAS

Let $\mathbf{D}_{i,j} = 1$ if tests i and j give the same CAPTCHA, and 0 otherwise. An indicator random variable.

The number of pairwise duplicates (a random variable) is:

$$\mathbb{E}[\mathbf{D}] = \sum_{i,j \in [m]} \mathbb{E}[\mathbf{D}_{i,j}].$$

For any pair $i, j \in [m]$: $\mathbb{E}[\mathbf{D}_{i,j}] = \Pr[\mathbf{D}_{i,j} = 1] = \frac{1}{n}$.

$$\mathbb{E}[\mathbf{D}] = \sum_{i,j \in [m]} \frac{1}{n} = \frac{\binom{m}{2}}{n} = \frac{m(m-1)}{2n}.$$

Note that the $\mathbf{D}_{i,j}$ random variables are not independent!

n: number of CAPTCHAS in database, m: number of random CAPTCHAS drawn to check database size, $\mathbf{D}$: number of pairwise duplicates in m random CAPTCHAS

You take $m = 1000$ samples. If the database size is as claimed ($n = 1,000,000$) then expected number of duplicates is:

$$\mathbb{E}[\mathbf{D}] = \frac{m(m-1)}{2n} = .4995$$

n: number of CAPTCHAS in database, m: number of random CAPTCHAS drawn to check database size, $\mathbf{D}$: number of pairwise duplicates in m random CAPTCHAS.

You take $m = 1000$ samples. If the database size is as claimed ($n = 1,000,000$) then expected number of duplicates is:

$$\mathbb{E}[\mathbf{D}] = \frac{m(m-1)}{2n} = .4995$$

You see 10 pairwise duplicates and suspect that something is up. But how confident can you be in your test?

n: number of CAPTCHAS in database, m: number of random CAPTCHAS drawn to check database size, $\mathbf{D}$: number of pairwise duplicates in m random CAPTCHAS.

You take $m = 1000$ samples. If the database size is as claimed ($n = 1,000,000$) then expected number of duplicates is:

$$\mathbb{E}[\mathbf{D}] = \frac{m(m-1)}{2n} = .4995$$

You see 10 pairwise duplicates and suspect that something is up. But how confident can you be in your test?

Concentration Inequalities: Bounds on the probability that a random variable deviates a certain distance from its mean.

n: number of CAPTCHAS in database, m: number of random CAPTCHAS drawn to check database size, $\mathbf{D}$: number of pairwise duplicates in m random CAPTCHAS.

You take $m = 1000$ samples. If the database size is as claimed ($n = 1,000,000$) then expected number of duplicates is:

$$\mathbb{E}[\mathbf{D}] = \frac{m(m-1)}{2n} = .4995$$

You see 10 pairwise duplicates and suspect that something is up. But how confident can you be in your test?

Concentration Inequalities: Bounds on the probability that a random variable deviates a certain distance from its mean.

- Useful in understanding how statistical tests perform, the behavior of randomized algorithms, the behavior of data drawn from different distributions, etc.

n: number of CAPTCHAS in database, m: number of random CAPTCHAS drawn to check database size, $\mathbf{D}$: number of pairwise duplicates in m random CAPTCHAS.

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

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

For any non-negative random variable X:

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

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

For any non-negative random variable X:

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

Proof:

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

For any non-negative random variable X:

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

Proof:

$$\mathbb{E}[X] = \sum_{s} \Pr(X = s) \cdot s$$

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

For any non-negative random variable X:

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

Proof:

$$\mathbb{E}[X] = \sum_{s} \Pr(X = s) \cdot s \geq \sum_{s \geq t} \Pr(X = s) \cdot s$$

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

For any non-negative random variable X:

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

Proof:

$$\mathbb{E}[X] = \sum_{s} \Pr(X = s) \cdot s \geq \sum_{s \geq t} \Pr(X = s) \cdot s$$

$$\geq \sum_{s \geq t} \Pr(X = s) \cdot t$$

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

For any non-negative random variable X:

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

Proof:

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

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

For any non-negative random variable X:

$$\Pr[X \geq t \cdot \mathbb{E}[X]] \leq \frac{1}{t}.$$

Proof:

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

Expected number of duplicate CAPTCHAS:

$\mathbb{E}[D] = \frac{m(m-1)}{2n} = .4995.$

You see $D = 10$ duplicates.

n: number of CAPTCHAS in database ($n = 1,000,000$ claimed) , m: number of random CAPTCHAS drawn to check database size ($m = 1000$ in this example), D: number of pairwise duplicates in m random CAPTCHAS.

Expected number of duplicate CAPTCHAS:

$\mathbb{E}[\mathrm{D}] = \frac{m(m-1)}{2n} = .4995.$

You see $\mathrm{D} = 10$ duplicates.

Applying Markov's inequality, if the real database size is $n = 1,000,000$ the probability of this happening is:

$$\Pr[\mathrm{D} \geq 10] \leq \frac{\mathbb{E}[\mathrm{D}]}{10} = \frac{.4995}{10} \approx .05$$

n: number of CAPTCHAS in database ($n = 1,000,000$ claimed) , m: number of random CAPTCHAS drawn to check database size ($m = 1000$ in this example), D: number of pairwise duplicates in m random CAPTCHAS.

Expected number of duplicate CAPTCHAS:

$\mathbb{E}[\mathrm{D}] = \frac{m(m-1)}{2n} = .4995.$

You see $\mathrm{D} = 10$ duplicates.

Applying Markov's inequality, if the real database size is $n = 1,000,000$ the probability of this happening is:

$$\Pr[\mathrm{D} \geq 10] \leq \frac{\mathbb{E}[\mathrm{D}]}{10} = \frac{.4995}{10} \approx .05$$

This is pretty small – you feel pretty sure the number of unique CAPTCHAS is much less than $1,000,000$. But how can you boost your confidence?

n: number of CAPTCHAS in database ($n = 1,000,000$ claimed) , m: number of random CAPTCHAS drawn to check database size ($m = 1000$ in this example), D: number of pairwise duplicates in m random CAPTCHAS.

Expected number of duplicate CAPTCHAS:

$\mathbb{E}[\mathbf{D}] = \frac{m(m-1)}{2n} = .4995.$

You see $\mathbf{D} = 10$ duplicates.

Applying Markov's inequality, if the real database size is $n = 1,000,000$ the probability of this happening is:

$$\Pr[\mathbf{D} \geq 10] \leq \frac{\mathbb{E}[\mathbf{D}]}{10} = \frac{.4995}{10} \approx .05$$

This is pretty small – you feel pretty sure the number of unique CAPTCHAS is much less than $1,000,000$. But how can you boost your confidence? We'll discuss next class.

n: number of CAPTCHAS in database ($n = 1,000,000$ claimed) , m: number of random CAPTCHAS drawn to check database size ($m = 1000$ in this example), $\mathbf{D}$: number of pairwise duplicates in m random CAPTCHAS.

Questions?