COMPSCI 514: Algorithms for Data Science

Prof. Cameron Musco

University of Massachusetts Amherst. Fall 2023.

Lecture 1

Motivation For this Class

The ability to analyze and learn from massive datasets is critical across many industries, the sciences, and beyond.

Motivation For this Class

The ability to analyze and learn from massive datasets is critical across many industries, the sciences, and beyond.

- Google receives 99,000 searches per second, 8.5 billion/day.
  - How do they process them to target advertisements? To predict trends? To improve their products?

Motivation For this Class

The ability to analyze and learn from massive datasets is critical across many industries, the sciences, and beyond.

- Google receives 99,000 searches per second, 8.5 billion/day.
  - How do they process them to target advertisements? To predict trends? To improve their products?
- The Vera C. Rubin Observatory in Chile is planned to produce 30 terabytes of data/night.
  - How do they denoise and compress the images? How do they detect anomalies such as changing object brightness? Will generate roughly 10 million alerts per night, within 60 seconds of the triggering event.

Motivation For this Class

The ability to analyze and learn from massive datasets is critical across many industries, the sciences, and beyond.

- Google receives 99,000 searches per second, 8.5 billion/day.
  - How do they process them to target advertisements? To predict trends? To improve their products?
- The Vera C. Rubin Observatory in Chile is planned to produce 30 terabytes of data/night.
  - How do they denoise and compress the images? How do they detect anomalies such as changing object brightness? Will generate roughly 10 million alerts per night, within 60 seconds of the triggering event.
- Meta's LLaMA-2 large language model was trained on 2 trillion tokens of data.
  - How is this data collected and cleaned? How is it used to train a language model with up to 65 billion parameters?.

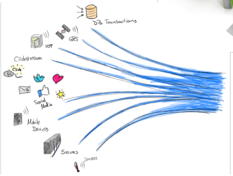
A New Paradigm for Algorithm Design

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

A New Paradigm for Algorithm Design

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

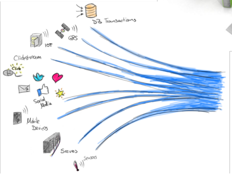
vs.

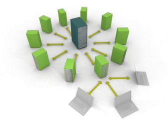


A New Paradigm for Algorithm Design

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

vs.





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

A New Paradigm for Algorithm Design

For example:

A New Paradigm for Algorithm Design

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.

A New Paradigm for Algorithm Design

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.

A New Paradigm for Algorithm Design

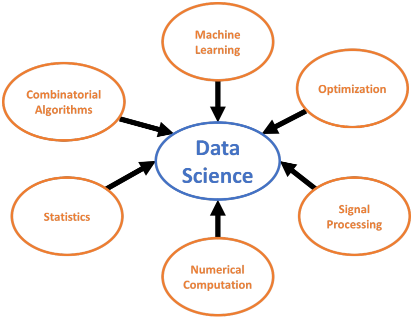
For example:

- 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 or perform a Google reverse image search, how does it provide an answer in < 10 seconds, without scanning over all of the millions or billions of possible images/audio files.

A Second Motivation: Data Science is highly interdisciplinary.

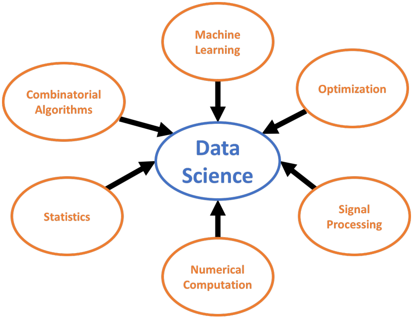
Motivation for This Class

A Second Motivation: Data Science is highly interdisciplinary.



Motivation for This Class

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

What We'll Cover

Section 1: Randomized Methods & Sketching

What We'll Cover

Section 1: Randomized Methods & Sketching

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

What We'll Cover

Section 1: Randomized Methods & Sketching

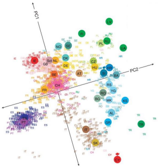
How can we efficiently compress large data sets in a way that lets us answer important algorithmic questions rapidly?

- 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

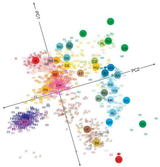
What We'll Cover

Section 2: Spectral Methods



What We'll Cover

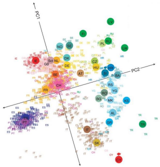
Section 2: Spectral Methods



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

What We'll Cover

Section 2: Spectral Methods

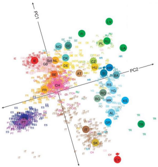


How do we identify the most important features of a dataset using linear algebraic techniques?

- 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 matrices 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?

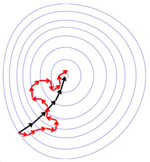
If you open up the codes that are underneath [most data science applications] this is all linear algebra on arrays.

– Michael Stonebraker

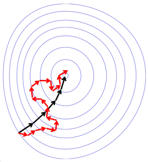
What We’ll Cover

What We'll Cover

Section 3: Optimization



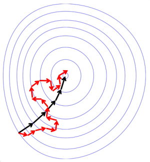
Section 3: Optimization



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

What We'll Cover

Section 3: Optimization

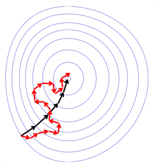


Fundamental continuous optimization approaches that drive methods in machine learning and statistics.

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

What We'll Cover

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 emphasize algorithm design, correctness proofs, and asymptotic analysis (relatively little 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 emphasize algorithm design, correctness proofs, and asymptotic analysis (relatively little required coding).
- A strong algorithms and mathematical background (particularly in linear algebra and probability) **are required**.
- Prereqs: COMPSCI 240 and COMPSCI 311. If you are an MS student and unsure about your background, email me or come chat.

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 emphasize algorithm design, correctness proofs, and asymptotic analysis (relatively little required coding).
- A strong algorithms and mathematical background (particularly in linear algebra and probability) **are required**.
- Prereqs: COMPSCI 240 and COMPSCI 311. If you are an MS student and unsure about your background, email me or come chat.

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 emphasize algorithm design, correctness proofs, and asymptotic analysis (relatively little required coding).
- A strong algorithms and mathematical background (particularly in linear algebra and probability) **are required**.
- Prereqs: COMPSCI 240 and COMPSCI 311. If you are an MS student and unsure about your background, email me or come chat.

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.

Course Logistics

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

http://people.cs.umass.edu/~cmusco/CS514F23/

See Moodle page for this link if you lose it, or search my name and follow the link from my homepage.

Moodle will be used for weekly quizzes and posting of exam grades but the course page for mostly everything else.

Personnel

Professor: Cameron Musco

- Email: cmusco@cs.umass.edu
- Office Hours: ~~Over Zoom~~, Tuesdays, 2:30pm-3:30pm (directly after class) in CS 234.
- 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:

- Weronika Nguyen
- Ed Almusalamy
- Mohit Yadav

See website for office hours (~~some TBD~~) and contact info.

Piazza and Participation

We will use Piazza for class discussion and questions.

- See website for link to sign up.

Piazza and Participation

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.
- Answering other students' or instructor questions on Piazza.
- Posting helpful links on Piazza, e.g., resources that cover class material, research articles related to the class, etc.
- It is completely fine to post private questions on Piazza, but these don't count towards participation credit.
- You can post anonymously on Piazza. Instructors will see the author behind all posts, so we can assign participation credit.

Textbooks and Materials

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 on Echo360.
- Sometimes it takes a lecture or two to get the Echo360 set up working properly.

Problem Sets

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

Problem Sets

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 easier and 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.
- You can change groups as you like over the course of the semester.

Problem Sets

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 easier and 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.
- You can change groups as you like over the course of the semester.

Problem set submissions will be via Gradescope.

- See website for a link to join and an entry code

Problem Sets

The problem sets will have two components:

- **Core Competency Problems:** Must be completed. Graded numerically. Similar in difficulty to exam problems and designed to prepare you for the exams.
- **Challenge Problems:** Designed to strengthen your ability to think creatively about algorithmic problems and push beyond what is taught in class, to design solutions of your own.
    - Will take significantly longer to tackle than the core competency problems.
    - Graded on a X, ✓-, ✓, ✓+ scale.
    - Can choose which ones your solve and attempt as many as you'd like.

Challenge Problem Grading

- A ✓is worth 1 point, a ✓+ is worth 2 points.
- Full credit is obtained by scoring 10 points over the course of the semester.
- E.g., if you complete 6 challenge problems with a ✓and 2 with a ✓+, you'll receive 100% on this component of the course.
- Roughly three challenge problems will be given per problem set (so roughly 15 total).
- Your team for the challenge problems does not need to match your team for the core competency problems.

Weekly Quizzes

We will release an online quiz in Moodle each Thursday after lecture, due the next Monday at 8pm.

Weekly Quizzes

We will release an online 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.

Grading

Grade Breakdown:

- Problem Sets (5 total): 40% total. 20% core competency problems split equally across problem sets, 20% challenge problems.
- Weekly Quizzes: 10%, weighted equally.
- Midterm (October 24th, in class): 25%.
- Final (December 14th, 10:30am - 12:30pm): 25%.
- Extra Credit: Up to 5% for participation. Potentially more available on problem sets and exams.

Grading

Grade Breakdown:

- Problem Sets (5 total): 40% total. 20% core competency problems split equally across problem sets, 20% challenge problems.
- Weekly Quizzes: 10%, weighted equally.
- Midterm (October 24th, in class): 25%.
- Final (December 14th, 10:30am - 12:30pm): 25%.
- Extra Credit: Up to 5% for participation. Potentially more available on problem sets and exams.

There is no online option for the exams. If you must miss an exam due to sickness or another emergency, we'll schedule an in-person make-up.

Academic Honesty and Exceptions

- No late homework submissions, unless there are extenuating circumstances, approved by the instructor before the deadline.
- No exceptions are given for missed quizzes. Instead, we just drop the lowest quiz grade for all students, so it is ok if you skip a week.

Academic Honesty and Exceptions

- No late homework submissions, unless there are extenuating circumstances, approved by the instructor before the deadline.
- No exceptions are given for missed quizzes. Instead, we just drop the lowest quiz grade for all students, so it is ok if you skip a week.

Academic Honesty:

- A first violation cheating on a homework, quiz, or other assignment will result in a 0 on that assignment. For problem sets, this will include all components of the assignment – core competency and challenge problems.
- 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 9/14** so that we can make arrangements.

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 9/14** 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

Some Probability Review

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

$\frac{1}{6}$ $\frac{1}{6}$ . . . $\frac{1}{6}$

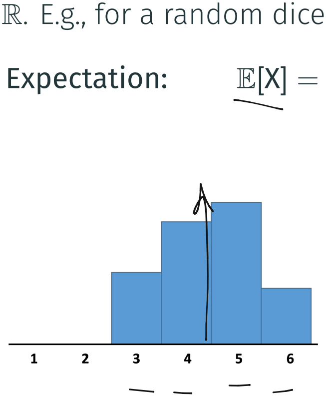
Some Probability Review

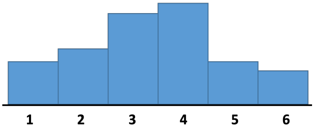
Consider a random variable X taking values in some finite set $S \subset \mathbb{R}$. E.g., for a random dice roll, $S = \{1, 2, 3, 4, 5, 6\}$.

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

$\sum_{i=\{1,2..6\}} \frac{1}{6} \cdot i$

$= 3.5$

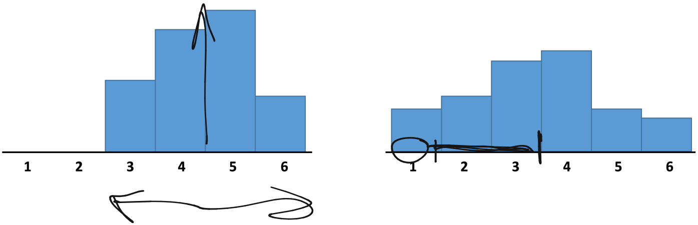




Some Probability Review

Consider a random variable X 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:** $\mathbb{E}[X] = \sum_{s \in S} \Pr(X = s) \cdot s$.
- **Variance:** $\mathrm{Var}[X] = \mathbb{E}[(X - \mathbb{E}[X])^2]$.

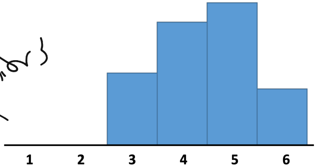


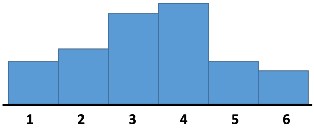
Some Probability Review

Consider a random variable X 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:** $\mathbb{E}[X] = \sum_{s \in S} \Pr(X = s) \cdot s$.
- **Variance:** $\mathrm{Var}[X] = \mathbb{E}[(X - \mathbb{E}[X])^2]$.

Distributions
X





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

$\alpha = 2$

Independence

Consider two random events A and B.

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

Independence

Consider two random events A and B.

- Conditional Probability:

$$\Pr(A|B) = \frac{\Pr(A \cap B)}{\Pr(B)}. \quad \frac{1/5}{1/2} = \frac{2}{5}$$

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

Independence

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.

Independence

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.

Independence

Example 1: What is the probability that for two independent dice rolls the first is a 6 and the second is odd?

$$P(A \cap B) = P(A) \cdot P(B)$$

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

Independence

Example 1: What is the probability that for two independent dice rolls the first is a 6 and the second is odd?

Example 2: What is the probability that a random dice roll is a prime number, conditioned on it being even.

$\{2, 4, 6\}$

$\frac{1}{3}$

prime, even

$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{1/6}{1/2} = \frac{1}{3}$

$P(B) = 1/2 \quad P(A) = \frac{1}{2}$

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:

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

Linearity of Expectation and Variance

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

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

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

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

and

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

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

X, Y: any two random variables.

Linearity of Expectation

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

Linearity of Expectation

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

Linearity of Expectation

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

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

Linearity of Expectation

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

Proof:

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

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

Proof:

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

Linearity of Expectation

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

Proof:

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

(law of total probability)

Linearity of Expectation

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

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