Online Class

- Attend synchronous class (if possible), participate, ask questions

On Zoom:
- Use the largest display you have. Your phone is not a good choice
- Join Zoom with video ON and audio Muted
- View options: Fit to Window
- To participate:
  - Use Raise Hand in Reactions
  - Unmute and talk
  - Chat to Everyone or Host.
Overview

- CSCI 1550/2540 is a mathematics course
- Modern mathematics at the interface of probability theory and computation
- Formulates, and explains many of the great successes of computing, such as machine learning
- Why probability theory?
- Why mathematics?
Why Probability?

Almost any advanced computing application today has some randomization/statistical/machine learning components:

- **Randomized algorithms** - random steps help!
  - Efficiency: Hashing, Quicksort, ...
  - Security and Privacy: Open key cryptography, one way function, ...
  - Monte Carlo methods: scientific computing, finance, weather forecast, ...

- **Probabilistic analysis of algorithms** - Theoretically “hard to solve” problems are often not that hard in practice.
  - Average case analysis
  - Almost always analysis

- **Modeling data**
  - Statistical machine learning
  - Data mining
  - Recommendation systems
Why mathematics / analytical methods?

If you don’t understand your program, and the library functions it uses,

- Obviously, you don’t know what your program is doing, and how good is its output.
- When randomization/probability/statistics is involved, wrong steps also lead to bias and fairness issues.

Why analytical methods?

- Probability is counterintuitive. - intuition doesn’t work.
- Only through rigorous mathematical analysis you can understand stochastic processes and identify errors, bias, etc.
Course Details - Main Topics

1. QUICK review of basic probability theory through analysis of randomized algorithms.
2. Large deviation bounds: Chernoff and Hoeffding bounds
3. Martingale (in discrete space)
4. Theory of statistical learning, PAC learning, VC-dimension
5. Monte Carlo methods
7. The probabilistic method
8. ...

This course emphasize rigorous mathematical approach, mathematical proofs, and analysis.
QUICK review of basic probability theory through analysis of randomized algorithms.

- Randomized algorithm for computing a min-cut in a graph
- Randomized algorithm for finding the $k$-smallest element in a set.
- Review of events, probability space, conditional probability, independence, expectation, ...
Course Details - Main Topics

• QUICK review of basic probability theory through analysis of randomized algorithms.
• Large deviation bounds: Chernoff and Hoeffding bounds
  • How many independent samples are needed for estimating a probability or an expectation?
  • Replacing limits with finite sample bounds
Course Details - Main Topics

• QUICK review of basic probability theory through analysis of randomized algorithms.
• Large deviation bounds: Chernoff and Hoeffding bounds
• Martingale (in discrete space)
  • Can we remove the independence assumption?
Course Details - Main Topics

1. QUICK review of basic probability theory through analysis of randomized algorithms.
2. Large deviation bounds: Chernoff and Hoeffding bounds
3. Martingale (in discrete space)
4. Theory of statistical learning, PAC learning, VC-dimension
   - What is learnable from random examples? What is not learnable?
   - How large training set do we need?
   - Can we use one sample to answer infinite many questions?
Course Details - Main Topics

1. QUICK review of basic probability theory through analysis of randomized algorithms.
2. Large deviation bounds: Chernoff and Hoeffding bounds
3. Martingale (in discrete space)
4. Theory of statistical learning, PAC learning, VC-dimension
5. Monte Carlo methods
   - Metropolis algorithm
   - What can be learned from simulations?
   - How many needles are in the haystack?
**Course Details - Main Topics**

1. QUICK review of basic probability theory through analysis of randomized algorithms.
2. Large deviation bounds: Chernoff and Hoeffding bounds
3. Martingale (in discrete space)
4. Theory of statistical learning, PAC learning, VC-dimension
5. Monte Carlo methods, Metropolis algorithm, ...
7. The probabilistic method
   - How to prove a deterministic statement using a probabilistic argument?
   - How is it useful for algorithm design?
Course Details - Main Topics

1. QUICK review of basic probability theory through analysis of randomized algorithms.
2. Large deviation bounds: Chernoff and Hoeffding bounds
3. Martingale (in discrete space)
4. Theory of statistical learning, PAC learning, VC-dimension
5. Monte Carlo methods, Metropolis algorithm, ...
7. The probabilistic method
8. ...

This course emphasize rigorous mathematical approach, mathematical proofs, and analysis.
Course Details

- **Pre-requisite**: CS145 or equivalent (first three chapters in the course textbook).
- **Course textbook**: 
  ![Book 1](image1.png) ![Book 2](image2.png)

- **You need to follow**: Piazza, Canvas, and the course website: https://cs.brown.edu/courses/csci1550/
Homeworks, Midterm and Final:

• Weekly assignments.
  • For learning, graded for effort and correctness.
  • Concise and correct proofs.
  • Can work together - but write in your own words.
  • Graded only if submitted on time.

• Midterm and final: take home exams, absolute no collaboration, cheaters get C.

• For 200 level - more challenging exams.

• This year: we encourage group work, we’ll set up random pairing each week, and group working hours with TA help.
We treat you as adults...

- You don’t need to attend class - but you cannot ask the instructor/TA’s to repeat information given in class.
- You don’t need to submit homework - but homework grades can improve your course grade.
- \( \text{CourseGrade} = 0.4 \times \text{Final} + 0.3 \times \max(\text{Midterm, Final}) + 0.3 \times \max(\text{Hw, Final}) \)
  \( \text{Hw} = \text{Average of the best 6 homework grades.} \)
- HW-0, not graded, out today.
- DON’T take this course if you don’t want to have HW-0 type exercises every week.
Questions?
Min-Cut

A graph $G = (V, E)$, $V$-set of vertices, $E$ set of edges.

A Min-Cut set - A minimum set of edges that disconnects the graph.

Fundamental computation problem in transportation, network reliability, arbitrage, ...
Min-Cut Algorithm

Input: An \( n \)-node graph \( G \).
Output: A minimal set of edges that disconnects the graph.

1. **Repeat** \( n - 2 \) **times:**
   1. Pick an edge uniformly at random.
   2. Contract the two vertices connected by that edge, eliminate all edges connecting the two vertices.

2. Output the set of edges connecting the two remaining vertices.

How good is this algorithm?
Min-Cut Algorithm

**Input:** An $n$-node graph $G$.

**Output:** A minimal set of edges that disconnects the graph.

1. **Repeat** $n - 2$ times:
   1. Pick an edge uniformly at random.
   2. Contract the two vertices connected by that edge, eliminate all edges connecting the two vertices.

2. Output the set of edges connecting the two remaining vertices.

**Theorem**

1. The algorithm outputs a min-cut edge-set with probability
   \[ \geq \frac{2}{n(n-1)}. \]
2. The smallest output in $O(n^2 \log n)$ iterations of the algorithm gives a correct answer with probability $1 - 1/n^2$. 
A probability space has three components:

1. A sample space $\Omega$, which is the set of all possible outcomes of the random process modeled by the probability space;
2. A family of sets $\mathcal{F}$ representing the allowable events, where each set in $\mathcal{F}$ is a subset of the sample space $\Omega$;
3. A probability function $\Pr : \mathcal{F} \rightarrow [0, 1]$ defining a measure.

In a discrete probability, an element of $\Omega$ is a simple event, and $\mathcal{F} = 2^\Omega$. 
A **probability function** is any function $\Pr : \mathcal{F} \rightarrow \mathbb{R}$ that satisfies the following conditions:

1. For any event $E$, $0 \leq \Pr(E) \leq 1$;
2. $\Pr(\Omega) = 1$;
3. For any finite or countably infinite sequence of pairwise mutually disjoint events $E_1, E_2, E_3, \ldots$

$$\Pr \left( \bigcup_{i \geq 1} E_i \right) = \sum_{i \geq 1} \Pr(E_i).$$

The probability of an event is the sum of the probabilities of its simple events.
Min-Cut Algorithm

**Input:** An $n$-node graph $G$.

**Output:** A minimal set of edges that disconnects the graph.

1. **Repeat** $n - 2$ **times:**
   1. Pick an edge uniformly at random.
   2. Contract the two vertices connected by that edge, eliminate all edges connecting the two vertices.

2. Output the set of edges connecting the two remaining vertices.

**Theorem**

The algorithm outputs a min-cut edge-set with probability

$$\geq \frac{2}{n(n-1)}.$$

What’s the probability space? The space changes each step.
Conditional Probabilities

**Definition**

The **conditional probability** that event $E_1$ occurs given that event $E_2$ occurs is

$$\Pr(E_1 \mid E_2) = \frac{\Pr(E_1 \cap E_2)}{\Pr(E_2)}.$$  

The conditional probability is only well-defined if $\Pr(E_2) > 0$.

By conditioning on $E_2$ we restrict the sample space to the set $E_2$. Thus we are interested in $\Pr(E_1 \cap E_2)$ “normalized” by $\Pr(E_2)$. A condition $E_2$ defines a new sample space, with a new probability function $P(\cdot \mid E_2)$.
Analysis of the Algorithm

Assume that the graph has a min-cut set of \( k \) edges. We compute the probability of finding one such set \( C \).

**Lemma**

If no edge of \( C \) was contracted, no edge of \( C \) was eliminated.

**Proof.**

Let \( X \) and \( Y \) be the two set of vertices cut by \( C \). If the contracting edge connects two vertices in \( X \) (res. \( Y \)), then all its parallel edges also connect vertices in \( X \) (res. \( Y \)).

**Corollary**

If the algorithm terminates before contracting any edge of \( C \), the algorithm gives the correct answer.
Let $E_i = "the edge contracted in iteration \ i \ is \ not \ in \ C."$
Let $F_i = \cap_{j=1}^{i} E_j = "no edge of \ C \ was \ contracted \ in \ the \ first \ i \ iterations".$
We need to compute $Pr(F_{n-2})$
Since the minimum cut-set has \( k \) edges, all vertices have degree \( \geq k \), and the graph has \( \geq nk/2 \) edges.

There are at least \( nk/2 \) edges in the graph, \( k \) edges are in \( C \). Thus, \( Pr(E_1) = Pr(F_1) \geq 1 - \frac{2k}{nk} = 1 - \frac{2}{n} \).

Conditioning on \( E_1 \), after the first vertex contraction we are left with an \( n-1 \) node graph, with minimum cut set, and minimum degree \( \geq k \). The new graph has at least \( k(n-1)/2 \) edges, thus \( Pr(E_2 \mid F_1) \geq 1 - \frac{k}{k(n-1)/2} \geq 1 - \frac{2}{n-1} \).

Similarly, \( Pr(E_i \mid F_{i-1}) \geq 1 - \frac{k}{k(n-i+1)/2} = 1 - \frac{2}{n-i+1} \).

We need to compute \( Pr(F_{n-2}) = Pr(\cap_{j=1}^{n-2} E_j) \)
Useful identities:

\[ Pr(A \mid B) = \frac{Pr(A \cap B)}{Pr(B)} \]

\[ Pr(A \cap B) = Pr(A \mid B)Pr(B) \]

\[ Pr(A \cap B \cap C) = Pr(A \mid B \cap C)Pr(B \cap C) \]

\[ = Pr(A \mid B \cap C)Pr(B \mid C)Pr(C) \]

Let \( A_1, \ldots, A_n \) be a sequence of events. Let \( E_i = \bigcap_{j=1}^{i} A_i \)

\[ Pr(E_n) = Pr(A_n \mid E_{n-1})Pr(E_{n-1}) = \]

\[ Pr(A_n \mid E_{n-1})Pr(A_{n-1} \mid E_{n-2}) \ldots P(A_2 \mid E_1)Pr(A_1) \]
We need to compute

\[ Pr(F_{n-2}) = Pr(\bigcap_{j=1}^{n-2} E_j) \]

We have

\[ Pr(E_1) = Pr(F_1) \geq 1 - \frac{2k}{nk} = 1 - \frac{2}{n} \]

and

\[ Pr(E_i \mid F_{i-1}) \geq 1 - \frac{k}{k(n - i + 1)/2} = 1 - \frac{2}{n - i + 1}. \]

\[ Pr(F_{n-2}) = Pr(E_{n-2} \cap F_{n-3}) = Pr(E_{n-2} \mid F_{n-3}) Pr(F_{n-3}) = \]

\[ Pr(E_{n-2} \mid F_{n-3}) Pr(E_{n-3} \mid F_{n-4}) \ldots Pr(E_2 \mid F_1) Pr(F_1) = \]

\[ Pr(F_1) \prod_{j=2}^{n-2} Pr(E_j \mid F_{j-1}) \]
The probability that the algorithm computes the minimum cut-set is

\[
Pr(F_{n-2}) = Pr(\cap_{j=1}^{n-2} E_j) = Pr(F_1) \prod_{j=2}^{n-2} Pr(E_j \mid F_{j-1}) \\
\geq \prod_{i=1}^{n-2} \left(1 - \frac{2}{n - i + 1}\right) = \prod_{i=1}^{n-2} \left(\frac{n - i - 1}{n - i + 1}\right) \\
= \left(\frac{n - 2}{n}\right) \left(\frac{n - 3}{n - 1}\right) \left(\frac{n - 4}{n - 2}\right) \cdots \\
\quad \frac{2}{n(n - 1)}.
\]
Theorem

Assume that we run the randomized min-cut algorithm \( n(n - 1) \log n \) times and output the minimum size cut-set found in all the iterations. The probability that the output is not a min-cut set is bounded by \( \frac{1}{n^2} \).

Lemma

Vertex contraction does not reduce the size of the min-cut set. Every cut set in the new graph is a cut set in the original graph.

Proof.

The algorithm has a one side error: the output is never smaller than the min-cut value.
\[
\left(1 - \frac{2}{n(n-1)}\right)^{n(n-1)\log n} \leq e^{-2\log n} = \frac{1}{n^2}.
\]

The Taylor series expansion of $e^{-x}$ gives

\[
e^{-x} = 1 - x + \frac{x^2}{2!} - \ldots.
\]

Thus, for $x < 1$,

\[
1 - x \leq e^{-x}.
\]
Theorem

1. The algorithm outputs a min-cut edge set with probability \( \geq \frac{2}{n(n-1)} \).

2. The smallest output in \( O(n^2 \log n) \) iterations of the algorithm gives a correct answer with probability \( 1 - \frac{1}{n^2} \).
Independent Events

Definition

Two events $E$ and $F$ are independent if and only if

$$Pr(E \cap F) = Pr(E) \cdot Pr(F).$$

More generally, events $E_1, E_2, \ldots, E_k$ are mutually independent if and only if for any subset $I \subseteq [1, k]$,

$$Pr \left( \bigcap_{i \in I} E_i \right) = \prod_{i \in I} Pr(E_i).$$