CS155: Probability and Computing: Randomized Algorithms and Probabilistic Analysis

Eli Upfal
Eli_Upfal@brown.edu
Office: 319

TA’s: Lorenzo De Stefani and Sorin Vatasoiu

cs155tas@cs.brown.edu
“It is remarkable that this science, which originated in the consideration of games and chances, should have become the most important object of human knowledge... The most important questions of life are, for the most part, really only problems of probability”

Why Probability in Computing?

The ideal "naive" computation paradigm:

- Deterministic algorithm computes efficiently a correct output for any input.

In "reality":

- Algorithms are often randomized (no deterministic algorithm, or randomized algorithm is simpler and more efficient)
- Input is assumed from a distribution - probabilistic analysis. (No efficient worst case solution or iterative computation on unknown future input).
- Correct Output is not known, need to learn from examples.
Almost any advance computing application today has some randomization/statistical/machine learning components:

- Network security
- Cryptography
- Web search and Web advertising
- Spam filtering
- Social network tools
- Recommendation systems: Amazon, Netflix,..
- Communication protocols
- Computational finance
- System biology
- DNA sequencing and analysis
- Data mining
Probability and Computing

- **Randomized algorithms** - random steps help! - cryptography and security, fast algorithms, simulations
- **Probabilistic analysis of algorithms** - Why “hard to solve” problems in theory are often not that hard in practice.
- **Statistical inference** - Machine learning, data mining...

All are based on the same (mostly discrete) probability theory principles and techniques
Course Details - Main Topics

- Review basic probability theory through analysis of randomized algorithms.
- Large deviation: Chernoff and Hoeffding bounds
- Martingale (in discrete space)
- The theory of learning, PAC learning and VC-dimension
- Monte Carlo methods, Metropolis algorithm, ...
- Convergence of Monte Carlo Markov Chains methods.
- The probabilistic method
- The Poisson process and some queueing theory.
- ...

Course Details

- Pre-requisite: CS 145, CS 45 or equivalent (first three chapters in the textbook).
- Textbook:
Homeworks, Midterm and Final:

- Weekly assignments.
- Typeset in Latex (or readable like typed) - template on the website
- Concise and correct proofs.
- Can work together - but write in your own words.
- Work **must** be submitted on time.
- Midterm and final: take home exams, absolute no collaboration, cheaters get C.
Verifying Matrix Multiplication

Given three $n \times n$ matrices $A$, $B$, and $C$ in a Boolean field, we want to verify

$$AB = C.$$ 

**Standard method:** Matrix multiplication - takes $\Theta(n^3)$ ($\Theta(n^{2.37})$) operations.
**Randomized algorithm:**

1. Chooses a random vector $\bar{r} = (r_1, r_2, \ldots, r_n) \in \{0, 1\}^n$.
2. Compute $B\bar{r}$;
3. Compute $A(B\bar{r})$;
4. Computes $C\bar{r}$;
5. If $A(B\bar{r}) \neq C\bar{r}$ return $AB \neq C$, else return $AB = C$.

The algorithm takes $\Theta(n^2)$ time.

---

**Theorem**

If $AB \neq C$, and $\bar{r}$ is chosen uniformly at random from $\{0, 1\}^n$, then

$$\Pr(AB\bar{r} = C\bar{r}) \leq \frac{1}{2}.$$
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 \( \Pr : \mathcal{F} \to \mathbb{R} \), satisfying the definition below.

An element of \( \Omega \) is a simple event. In a discrete probability space we use \( \mathcal{F} = 2^\Omega \).
A **probability function** is any function $\Pr : \mathcal{F} \to \mathbb{R}$ 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.
**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 $I \subseteq \{1, k\}$,

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

Randomized algorithm:
1. Chooses a random vector \( \bar{r} = (r_1, r_2, \ldots, r_n) \in \{0, 1\}^n \).
2. Compute \( B\bar{r} \);
3. Compute \( A(B\bar{r}) \);
4. Computes \( C\bar{r} \);
5. If \( A(B\bar{r}) \neq C\bar{r} \) return \( AB \neq C \), else return \( AB = C \).

The algorithm takes \( \Theta(n^2) \) time.

Theorem

If \( AB \neq C \), and \( \bar{r} \) is chosen uniformly at random from \( \{0, 1\}^n \), then

\[
\Pr(AB\bar{r} = C\bar{r}) \leq \frac{1}{2}.
\]
Lemma

Choosing $\bar{r} = (r_1, r_2, \ldots, r_n) \in \{0, 1\}^n$ uniformly at random is equivalent to choosing each $r_i$ independently and uniformly from $\{0, 1\}$.

Proof.

If each $r_i$ is chosen independently and uniformly at random, each of the $2^n$ possible vectors $\bar{r}$ is chosen with probability $2^{-n}$, giving the lemma.
Proof:

Let $D = AB - C \neq 0$.

$AB\bar{r} = C\bar{r}$ implies that $D\bar{r} = 0$.

Since $D \neq 0$ it has some non-zero entry; assume $d_{11}$.

For $D\bar{r} = 0$, it must be the case that

$$\sum_{j=1}^{n} d_{1j} r_j = 0,$$

or equivalently

$$r_1 = -\frac{\sum_{j=2}^{n} d_{1j} r_j}{d_{11}}. \quad (1)$$

Here we use $d_{11} \neq 0$. 
Assume that we fixed $r_2, \ldots, r_n$. The RHS is already determined, the only variable is $r_1$.

$$r_1 = -\frac{\sum_{j=2}^{n} d_{1j} r_j}{d_{11}}. \tag{2}$$

Probability that $r_1 = \text{RHS}$ is no more than $1/2$. 
More formally, summing over all collections of values 
\((x_2, x_3, x_4, \ldots , x_n) \in \{0, 1\}^{n-1}\), we have

\[
\Pr(AB\bar{r} = C\bar{r}) \leq \sum_{(x_2,\ldots,x_n)\in\{0,1\}^{n-1}} \Pr((r_2,\ldots,r_n) = (x_2,\ldots,x_n)) \cdot \Pr(AB\bar{r} = C\bar{r} | (r_2,\ldots,r_n) = (x_2,\ldots,x_n))
\]

\[
= \sum_{(x_2,\ldots,x_n)\in\{0,1\}^{n-1}} \Pr((AB\bar{r} = C\bar{r}) \cap ((r_2,\ldots,r_n) = (x_2,\ldots,x_n)))
\]

\[
\leq \sum_{(x_2,\ldots,x_n)\in\{0,1\}^{n-1}} \Pr\left(\left(r_1 = -\frac{\sum_{j=2}^{n} d_{1j} r_j}{d_{11}}\right) \cap ((r_2,\ldots,r_n) = (x_2,\ldots,x_n))\right)
\]

\[
= \sum_{(x_2,\ldots,x_n)\in\{0,1\}^{n-1}} \Pr\left(r_1 = -\frac{\sum_{j=2}^{n} d_{1j} r_j}{d_{11}}\right) \cdot \Pr((r_2,\ldots,r_n) = (x_2,\ldots,x_n))
\]

\[
\leq \sum_{(x_2,\ldots,x_n)\in\{0,1\}^{n-1}} \frac{1}{2} \Pr((r_2,\ldots,r_n) = (x_2,\ldots,x_n))
\]

\[
= \frac{1}{2}.
\]
Computing 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 $\Pr(E_2)$.
Theorem (Law of Total Probability)

Let $E_1, E_2, \ldots, E_n$ be mutually disjoint events in the sample space $\Omega$, and $\bigcup_{i=1}^n E_i = \Omega$, then

$$
Pr(B) = \sum_{i=1}^n Pr(B \cap E_i) = \sum_{i=1}^n Pr(B \mid E_i) Pr(E_i).
$$

Proof.

Since the events $E_i, i = 1, \ldots, n$ are disjoint and cover the entire sample space $\Omega$,

$$
Pr(B) = \sum_{i=1}^n Pr(B \cap E_i) = \sum_{i=1}^n Pr(B \mid E_i) Pr(E_i).
$$
Smaller Error Probability

The test has a one side error, repeated tests are independent.

- Run the test $k$ times.
- Accept $AB = C$ if it passed all $k$ tests.

**Theorem**

The probability of making a mistake is $\leq (1/2)^k$. 
Bayes’ Law

Theorem (Bayes’ Law)

Assume that $E_1, E_2, \ldots, E_n$ are mutually disjoint sets such that $\bigcup_{i=1}^{n} E_i = \Omega$, then

$$
\Pr(E_j \mid B) = \frac{\Pr(E_j \cap B)}{\Pr(B)} = \frac{\Pr(B \mid E_j) \Pr(E_j)}{\sum_{i=1}^{n} \Pr(B \mid E_i) \Pr(E_i)}.
$$
Application: Finding a Biased Coin

- We are given three coins, two of the coins are fair and the third coin is biased, landing heads with probability $2/3$. We need to identify the biased coin.
- We flip each of the coins. The first and second coins come up heads, and the third comes up tails.
- What is the probability that the first coin is the biased one?
Let $E_i$ be the event that the $i$-th coin flipped is the biased one, and let $B$ be the event that the three coin flips came up heads, heads, and tails. Before we flip the coins we have $\Pr(E_i) = 1/3$ for $i = 1, \ldots, 3$, thus

$$\Pr(B \mid E_1) = \Pr(B \mid E_2) = \frac{2}{3} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{6},$$

and

$$\Pr(B \mid E_3) = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{3} = \frac{1}{12}. $$

Applying Bayes’ law we have

$$\Pr(E'_1 \mid B) = \frac{\Pr(B \mid E_1) \Pr(E_1)}{\sum_{i=1}^3 \Pr(B \mid E_i) \Pr(E_i)} = \frac{2}{5}.$$ 

The outcome of the three coin flips increases the probability that the first coin is the biased one from $1/3$ to $2/5$. 
Bayesian approach

• Start with an *prior* model, giving some initial value to the model parameters.
• This model is then modified, by incorporating new observations, to obtain a *posterior* model that captures the new information.
Min-Cut

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 set of edges with probability* \[ \geq \frac{2}{n(n-1)}. \]

Lemma

*Vertex contraction does not reduce the size of the min-cut set. (Contraction can only increase the size of the min-cut set.)*

Proof.

Every cut set in the new graph is a cut set in the original graph.
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. $Y$).
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 $C$.

$$Pr(E_1) = Pr(F_1) \geq 1 - \frac{2k}{nk} = 1 - \frac{2}{n}.$$
Assume that the first contraction did not eliminate an edge of \( C \) (conditioning on the event \( E_1 = F_1 \)). 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.

\[
Pr(E_2 \mid F_1) \geq 1 - \frac{k}{k(n-1)/2} \geq 1 - \frac{2}{n-1}.
\]

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

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

We use

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

\[ Pr(F_{n-2}) = \]

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

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

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

\[ Pr(F_{n-2}) = Pr(\cap_{j=1}^{n-2} E_j) = Pr(F_1) \prod_{j=2}^{n-2} Pr(E_j \mid F_{j-1}) \]

\[ \geq \prod_{i=1}^{n-2} \left(1 - \frac{2}{n-i+1}\right) = \prod_{i=1}^{n-2} \left(\frac{n-i-1}{n-i+1}\right) \]

\[ = \left(\frac{n-2}{n}\right) \left(\frac{n-3}{n-1}\right) \left(\frac{n-4}{n-2}\right) \cdots \]

\[ \frac{2}{n(n-1)}. \]
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

\[
\left(1 - \frac{2}{n(n-1)}\right)^{n(n-1) \log n} \leq e^{-2 \log n} = \frac{1}{n^2}.
\]

Proof.

The algorithm has a one side error: the output is never smaller than the min-cut value.
The Taylor series expansion of $e^{-x}$ gives

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

Thus, for $x < 1$,

$$1 - x \leq e^{-x}.$$