CS3230 – Design and Analysis of Algorithms (S1 AY2024/25)

Special lecture: Preview of CS5330 – Randomized Algorithms

Why randomness?

- There are **many scenarios** where randomness is **extremely useful**.

Why randomness?

- There are **many scenarios** where randomness is **extremely useful**.

Sublinear, parallel, and distributed computing

Overcoming known lower bounds for deterministic algorithms

Example 1: communication protocols

- **Equality testing:**
  - Alice holds a large n-bit string S_A.
  - Bob holds a large n-bit string S_B.
- **Goal:**
  - Alice and Bob want to decide whether $S_A = S_B$.
- **Example:**
  - After downloading a large file, you want to be sure that the file downloaded is correct.

Example 1: communication protocols

- **Equality testing:**
  - Alice holds a large n-bit string S_A.
  - Bob holds a large n-bit string S_B.
- **Goal:**
  - Alice and Bob want to decide whether $S_A = S_B$.
- **Example:**
  - After downloading a large file, you want to be sure that the file downloaded is correct.

Deterministic algorithm:

- Any deterministic algorithm requires $\Omega(n)$ bits of communication.

Randomized algorithm:

- With randomness, there is a communication protocol that only sends $O(\log n)$ bits, with success probability 0.99.

Example 2: sampling

- Given a long list of numbers, estimate its average value.
    - **Example:** How many friends does a Facebook user have on average?

Example 2: sampling

- Given a long list of numbers, estimate its average value.
  - **Example:** How many friends does a Facebook user have on average?

Deterministic algorithm:
- Requires seeing the entire input.

Randomized algorithm:
- Sampling a subset of input and calculating its average.
- The larger the sample size, the more accurate the estimate is.

Example 2: sampling

- Given a long list of numbers, estimate its average value.
  - **Example:** How many friends does a Facebook user have on average?

Deterministic algorithm:
- Requires seeing the entire input.

Randomized algorithm:
- Sampling a subset of input and calculating its average.
- The larger the sample size, the more accurate the estimate is.

Randomness is <u>extremely useful</u> in designing **sublinear-time algorithms** for approximately learning a property of a massive data set.

Two selected topics

- Concentration inequalities. Show that X is close to its expectation $\mathbb{E}[X]$ with high probability.
- Derandomization. Turning a randomized algorithm into a deterministic algorithm.

Two selected topics

- **Concentration inequalities.**
- Derandomization.

Programming assignment 2:
- There is a randomized guessing strategy using $2.25 \cdot n$ guesses in expectation.
- What is the probability that the number of guesses is at most $2.45 \cdot n$?

Two selected topics

- **Concentration inequalities.**
- Derandomization.

$\mathbb{E}[X] = 2.25 \cdot n$

Programming assignment 2:

- There is a randomized guessing strategy using $2.25 \cdot n$ guesses <u>in expectation</u>.
- What is the probability that <u>the number of guesses</u> is at most $2.45 \cdot n$?

X

Markov inequality: $\Pr\left[X \geq a \cdot \mathbb{E}[X]\right] \leq \frac{1}{a}$

Two selected topics

- **Concentration inequalities.**
- Derandomization.

$\mathbb{E}[X] = 2.25 \cdot n$

Programming assignment 2:

- There is a randomized guessing strategy using $2.25 \cdot n$ guesses <u>in expectation</u>.
- What is the probability that <u>the number of guesses</u> is at most $2.45 \cdot n$?

X

Markov inequality: $\Pr\left[X \geq a \cdot \mathbb{E}[X]\right] \leq \frac{1}{a}$

The success probability is **too small**.

$$\Pr[X \geq 2.45 \cdot n] = \Pr\left[X \geq \frac{2.45}{2.25} \cdot \mathbb{E}[X]\right] \leq \frac{2.25}{2.45} = 0.9183 \ldots \longrightarrow \Pr[X \leq 2.45 \cdot n] \geq 0.0816 \ldots$$

Two selected topics

- Concentration inequalities.
- **Derandomization.** Can we solve these problems deterministically?

Tutorial 5:
- Any graph $G = (V, E)$ admits a cut of size of at least $|E|/2$.
- Such a cut can be computed <u>in expectation</u>.

Two selected topics

- Concentration inequalities.
- **Derandomization.**

Can we solve these problems deterministically?

Tutorial 5:
- Any graph $G = (V, E)$ admits a cut of size of at least $|E|/2$.
- Such a cut can be computed in expectation.

Midterm exam:
- Let $G = (V, E)$ be any n-vertex bipartite graph where each vertex v is associated with a list $L(v)$ of $\lceil \log_2 n \rceil + 1$ colors.
- A proper coloring can be computed with probability $1/2$.

Recap

- **Markov inequality:**
  - If X is a non-negative random variable and $a > 0$, then
$$\Pr\left[X \geq a \cdot \mathbb{E}[X]\right] \leq \frac{1}{a}.$$

The tail bound obtained by Markov inequality is only **linear** in a^{-1}.

Can we improve this?

Chebyshev inequality

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

Chebyshev inequality

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

$\Pr\left[X \geq a \cdot \mathbb{E}[X]\right] \leq \frac{1}{a}$

  - As $(X - \mathbb{E}[X])^2 \geq 0$, we may apply Markov inequality to $(X - \mathbb{E}[X])^2$:

$$\Pr\left[(X - \mathbb{E}[X])^2 \geq a \cdot \mathbb{E}[(X - \mathbb{E}[X])^2]\right] \leq \frac{1}{a}.$$

Chebyshev inequality

- **Variance:**
  - $\mathrm{Var}[X] = \mathbb{E}[(X - \mathbb{E}[X])^2]$.
  - As $(X - \mathbb{E}[X])^2 \geq 0$, we may apply Markov inequality to $(X - \mathbb{E}[X])^2$:

$\Pr\left[X \geq a \cdot \mathbb{E}[X]\right] \leq \frac{1}{a}$

$$\Pr\left[(X - \mathbb{E}[X])^2 \geq a \cdot \mathbb{E}[(X - \mathbb{E}[X])^2]\right] \leq \frac{1}{a}.$$

Chebyshev inequality:

- $\Pr\left[|X - \mathbb{E}[X]| \geq b \cdot \sqrt{\mathrm{Var}[X]}\right] \leq \frac{1}{b^2}$.
- $\Pr[|X - \mathbb{E}[X]| \geq c] \leq \frac{\mathrm{Var}[X]}{c^2}$.

$b = \sqrt{a}$

$c = b \cdot \sqrt{\mathrm{Var}[X]}$

Chebyshev inequality

- **Variance:**
  - $\mathrm{Var}[X] = \mathbb{E}[(X - \mathbb{E}[X])^2]$.
  - As $(X - \mathbb{E}[X])^2 \geq 0$, we may apply Markov inequality to $(X - \mathbb{E}[X])^2$:

$\Pr\left[X \geq a \cdot \mathbb{E}[X]\right] \leq \frac{1}{a}$

$$\Pr\left[(X - \mathbb{E}[X])^2 \geq a \cdot \mathbb{E}[(X - \mathbb{E}[X])^2]\right] \leq \frac{1}{a}.$$

Chebyshev inequality:

- $\Pr\left[|X - \mathbb{E}[X]| \geq b \cdot \sqrt{\mathrm{Var}[X]}\right] \leq \frac{1}{b^2}$.
- $\Pr[|X - \mathbb{E}[X]| \geq c] \leq \frac{\mathrm{Var}[X]}{c^2}$.

$b = \sqrt{a}$

$c = b \cdot \sqrt{\mathrm{Var}[X]}$

An improvement over Markov inequality

The tail bound obtained by Chebyshev inequality is **quadratic**.

Application

$\mathbb{E}[X] = 2.25 \cdot n$

Programming assignment 2:

- There is a randomized guessing strategy using $2.25 \cdot n$ guesses <u>in expectation</u>.
- What is the probability that <u>the number of guesses</u> is at most $2.45 \cdot n$?

X

Application

X_i = number of guesses in iteration i.

- $\Pr[X_i = 1] = \frac{1}{4}$
- $\Pr[X_i = 2] = \frac{1}{4}$
- $\Pr[X_i = 3] = \frac{1}{2}$
- $\mathbb{E}[X_i] = 2.25$

$\mathbb{E}[X] = 2.25 \cdot n$

$$X = \sum_{i=1}^{n} X_i$$

Programming assignment 2:
- There is a randomized guessing strategy using $2.25 \cdot n$ guesses in expectation.
- What is the probability that the number of guesses is at most $2.45 \cdot n$?

X

Application

X_i = number of guesses in iteration i.

- $\Pr[X_i = 1] = \frac{1}{4}$
- $\Pr[X_i = 2] = \frac{1}{4}$
- $\Pr[X_i = 3] = \frac{1}{2}$
- $\mathbb{E}[X_i] = 2.25$
- $\mathrm{Var}[X_i] = \frac{1}{4} \cdot (1 - 2.25)^2 + \frac{1}{4} \cdot (2 - 2.25)^2 + \frac{1}{2} \cdot (3 - 2.25)^2 = 0.6875$

$\mathbb{E}[X] = 2.25 \cdot n$

$$X = \sum_{i=1}^{n} X_i$$

Programming assignment 2:

- There is a randomized guessing strategy using $2.25 \cdot n$ guesses in expectation.
- What is the probability that the number of guesses is at most $2.45 \cdot n$?

X

Application

X_i = number of guesses in iteration i.

- $\Pr[X_i = 1] = \frac{1}{4}$
- $\Pr[X_i = 2] = \frac{1}{4}$
- $\Pr[X_i = 3] = \frac{1}{2}$
- $\mathbb{E}[X_i] = 2.25$
- $\mathrm{Var}[X_i] = \frac{1}{4} \cdot (1 - 2.25)^2 + \frac{1}{4} \cdot (2 - 2.25)^2 + \frac{1}{2} \cdot (3 - 2.25)^2 = 0.6875$

$X_1, X_2, \ldots X_n$ are independent.

$$\mathrm{Var}[X] = \sum_{i=1}^{n} \mathrm{Var}[X_i] = 0.6875 \cdot n$$

$$\mathbb{E}[X] = 2.25 \cdot n$$

$$X = \sum_{i=1}^{n} X_i$$

Programming assignment 2:

- There is a randomized guessing strategy using $2.25 \cdot n$ guesses <u>in expectation</u>.
- What is the probability that <u>the number of guesses</u> is at most $2.45 \cdot n$?

X

Application

$$\mathrm{Var}[X] = \sum_{i=1}^{n} \mathrm{Var}[X_i] = 0.6875 \cdot n$$

Chebyshev inequality:

- $\Pr[|X - \mathbb{E}[X]| \geq c] \leq \frac{\mathrm{Var}[X]}{c^2}$.

$$\Pr[X \geq 2.45 \cdot n] \leq \Pr[|X - \mathbb{E}[X]| \geq 0.2 \cdot n] \leq \frac{\mathrm{Var}[X]}{(0.2 \cdot n)^2} = \frac{17.1875}{n}$$

$$\mathbb{E}[X] = 2.25 \cdot n$$

$$X = \sum_{i=1}^{n} X_i$$

Programming assignment 2:

- There is a randomized guessing strategy using $2.25 \cdot n$ guesses <u>in expectation</u>.
- What is the probability that <u>the number of guesses</u> is at most $2.45 \cdot n$?

X

Application

If n is large, then with a very high probability, the number of guesses is at most $2.45 \cdot n$.

$$\mathrm{Var}[X] = \sum_{i=1}^{n} \mathrm{Var}[X_i] = 0.6875 \cdot n$$

Chebyshev inequality:

- $\Pr[|X - \mathbb{E}[X]| \geq c] \leq \frac{\mathrm{Var}[X]}{c^2}$.

$$\Pr[X \geq 2.45 \cdot n] \leq \Pr[|X - \mathbb{E}[X]| \geq 0.2 \cdot n] \leq \frac{\mathrm{Var}[X]}{(0.2 \cdot n)^2} = \frac{17.1875}{n} \in O\left(\frac{1}{n}\right)$$

$$\mathbb{E}[X] = 2.25 \cdot n$$

$$X = \sum_{i=1}^{n} X_i$$

Programming assignment 2:

- There is a randomized guessing strategy using $2.25 \cdot n$ guesses in expectation.
- What is the probability that the number of guesses is at most $2.45 \cdot n$?

X

Application

If n is large, then with a very high probability,
the number of guesses is at most $2.45 \cdot n$.

Chebyshev inequality:

- $\Pr[|X - \mathbb{E}[X]| \geq c] \leq \frac{\mathrm{Var}[X]}{c^2}$.

$$\Pr[X \geq 2.45 \cdot n] \leq \Pr[|X - \mathbb{E}[X]| \geq 0.2 \cdot n] \leq \frac{\mathrm{Var}[X]}{(0.2 \cdot n)^2} = \frac{17.1875}{n} \in O\left(\frac{1}{n}\right)$$

Is it possible to get an even better bound?

Higher moments

- It is possible to extend Chebyshev inequality to **higher moments**.

$$\Pr[|X - \mathbb{E}[X]| \geq a] = \Pr[|X - \mathbb{E}[X]|^k \geq a^k] \leq \frac{\mathbb{E}[|X - \mathbb{E}[X]|^k]}{a^k}.$$

If X is non−negative, then $\Pr\left[X \geq a \cdot \left(\mathbb{E}[X^k]\right)^{1/k}\right] = \Pr\left[X^k \geq a^k \cdot \mathbb{E}[X^k]\right] \leq \frac{1}{a^k}.$

Higher moments

- It is possible to extend Chebyshev inequality to **higher moments**.

$$\boxed{\Pr[|X - \mathbb{E}[X]| \geq a] = \Pr[|X - \mathbb{E}[X]|^k \geq a^k] \leq \frac{\mathbb{E}[|X - \mathbb{E}[X]|^k]}{a^k}.}$$

$$\boxed{\text{If } X \text{ is non-negative, then } \Pr\left[X \geq a \cdot \left(\mathbb{E}[X^k]\right)^{1/k}\right] = \Pr\left[X^k \geq a^k \cdot \mathbb{E}[X^k]\right] \leq \frac{1}{a^k}.}$$

With these concentration inequalities, we should be able to get an **improved bound**:

$$\Pr[X \geq 2.45 \cdot n] \in O\left(\frac{1}{n^{k-1}}\right)$$

Disclaimer: I am confident this will work, though I have not personally done the calculations.

Further improvements

- What is the limit of this approach?

Further improvements

- What is the limit of this approach?

Hoeffding inequality:

- $X = \sum_{i=1}^{n} X_i$, where $X_1, X_2, \dots, X_n$ are independent random variables taking values in $[a_i, b_i]$.

- $\Pr[X \leq \mathbb{E}[X] - t] \leq e^{-\frac{2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}}$.
- $\Pr[X \geq \mathbb{E}[X] + t] \leq e^{-\frac{2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}}$.

Further improvements

The probability that the number of guesses exceeds $2.45 \cdot n$ is exponentially small.

$$\Pr[X \geq 2.45 \cdot n] \leq \Pr[X \geq \mathbb{E}[X] + 0.2 \cdot n] \leq e^{-\frac{2(0.2 \cdot n)^2}{\sum_{i=1}^{n} 2^2}} = e^{-\frac{n}{50}}$$

Hoeffding inequality:

- $X = \sum_{i=1}^{n} X_i$, where $X_1, X_2, \dots, X_n$ are independent random variables taking values in $[a_i, b_i]$.
- $\Pr[X \leq \mathbb{E}[X] - t] \leq e^{-\frac{2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}}$.
- $\Pr[X \geq \mathbb{E}[X] + t] \leq e^{-\frac{2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}}$.

Derandomization

Tutorial 5:

- Any graph $G = (V, E)$ admits a cut of size of at least $|E|/2$.
- Such a cut can be computed <u>in expectation</u>.

Can we obtain such a cut deterministically?

Derandomization

Tutorial 5:

- Any graph $G = (V, E)$ admits a cut of size of at least $|E|/2$.
- Such a cut can be computed <u>in expectation</u>.

Can we obtain such a cut deterministically?

Recall: A randomized algorithm and its analysis.

Random partition:

- $V = \{v_1, v_2, \dots, v_n\}$.
- Compute a partition $V = V_1 \cup V_2$ randomly:
  - $x_i \in \{1, 2\}$ is the outcome of a fair coin flip.
    - $v_i \in V_1$ if $x_i = 1$.
    - $v_i \in V_2$ if $x_i = 2$.

Analysis:

- $X_e =$ the indicator random variable for the event that e crosses V_1 and V_2.
- $X = \sum_{e \in E} X_e$ is the size of the cut.
- $\mathbb{E}[X] = \mathbb{E}[\sum_{e \in E} X_e] = \sum_{e \in E} \mathbb{E}[X_e] = \sum_{e \in E} \frac{1}{2} = \frac{|E|}{2}$

Derandomization

Tutorial 5:
- Any graph $G = (V, E)$ admits a cut of size of at least $|E|/2$.
- Such a cut can be computed in expectation.

Can we obtain such a cut deterministically?

Random partition:
- $V = \{v_1, v_2, \ldots, v_n\}$.
- Compute a partition $V = V_1 \cup V_2$ randomly:
  - $x_i \in \{1, 2\}$ is the outcome of a fair coin flip.
    - $v_i \in V_1$ if $x_i = 1$.
    - $v_i \in V_2$ if $x_i = 2$.

Analysis:
- $X_e =$ the indicator random variable for the event that e crosses V_1 and V_2.
- $X = \sum_{e \in E} X_e$ is the size of the cut.
- $\mathbb{E}[X] = \mathbb{E}[\sum_{e \in E} X_e] = \sum_{e \in E} \mathbb{E}[X_e] = \sum_{e \in E} \frac{1}{2} = \frac{|E|}{2}$

Derandomization:
Set the random variables $x_1, x_2, \ldots, x_n$ one by one deterministically to maximize the conditional expectation.

$$\mathbb{E}[X] = \Pr[x_1 = 1] \cdot \mathbb{E}[X | x_1 = 1] + \Pr[x_1 = 2] \cdot \mathbb{E}[X | x_1 = 2]$$

The method of conditional expectations

$$\mathbb{E}[X] = \Pr[x_1 = 1] \cdot \mathbb{E}[X|x_1 = 1] + \Pr[x_1 = 2] \cdot \mathbb{E}[X|x_1 = 2]$$

At least one of the following holds:

- $\mathbb{E}[X] \leq \mathbb{E}[X|x_1 = 1]$
- $\mathbb{E}[X] \leq \mathbb{E}[X|x_1 = 2]$

Choose $a_1 \in \{1, 2\}$ to maximize $\mathbb{E}[X|x_1 = a_1]$ $\longrightarrow$ $\mathbb{E}[X] \leq \mathbb{E}[X|x_1 = a_1]$

Fix $x_1 = a_1$, and then repeat the process to fix the rest of the variables

The method of conditional expectations

$$\mathbb{E}[X] = \Pr[x_1 = 1] \cdot \mathbb{E}[X|x_1 = 1] + \Pr[x_1 = 2] \cdot \mathbb{E}[X|x_1 = 2]$$

At least one of the following holds:

- $\mathbb{E}[X] \le \mathbb{E}[X|x_1 = 1]$
- $\mathbb{E}[X] \le \mathbb{E}[X|x_1 = 2]$

Can be computed in polynomial time.

Choose $a_1 \in \{1, 2\}$ to maximize $\mathbb{E}[X|x_1 = a_1]$ $\longrightarrow$ $\mathbb{E}[X] \le \mathbb{E}[X|x_1 = a_1]$

Fix $x_1 = a_1$, and then repeat the process to fix the rest of the variables

The method of conditional expectations

$$\frac{|E|}{2} \leq \mathbb{E}[X]$$
$$\leq \mathbb{E}[X | x_1 = a_1]$$
$$\leq \mathbb{E}[X | x_1 = a_1, x_2 = a_2]$$
$$\leq \mathbb{E}[X | x_1 = a_1, x_2 = a_2, x_3 = a_3]$$
$$\cdots$$
$$\leq \mathbb{E}[X | \underbrace{x_1 = a_1, \dots, x_n = a_n}]$$

A cut with size $\geq \frac{|E|}{2}$ is computed deterministically.

Fix $x_1 = a_1$, and then repeat the process to fix the rest of the variables

$$\mathbb{E}[X] \leq \mathbb{E}[X | x_1 = a_1]$$

Graph coloring

Can we find this coloring deterministically?

Midterm exam:

- Let $G = (V, E)$ be any n-vertex bipartite graph where each vertex v is associated with a list $L(v)$ of $\lceil \log_2 n \rceil + 1$ colors.
- A proper coloring can be computed <u>with probability $1/2$</u>.

Graph coloring

Can we find this coloring deterministically?

Midterm exam:

- Let $G = (V, E)$ be any n-vertex bipartite graph where each vertex v is associated with a list $L(v)$ of $\lceil \log_2 n \rceil + 1$ colors.
- A proper coloring can be computed <u>with probability $1/2$</u>.

Randomized algorithm:

- Assign each color to one of the two parts randomly.
- The algorithm is successful is every vertex v has a color in its list $L(v)$ assigned to its part.

Graph coloring

Can we find this coloring deterministically?

Midterm exam:
- Let $G = (V, E)$ be any n-vertex bipartite graph where each vertex v is associated with a list $L(v)$ of $\lceil \log_2 n \rceil + 1$ colors.
- A proper coloring can be computed <u>with probability $1/2$</u>.

Randomized algorithm:
- Assign each color to one of the two parts randomly.
- The algorithm is successful is every vertex v has a color in its list $L(v)$ assigned to its part.

- $X_v =$ the indicator random variable for the **bad event** that all colors in $L(v)$ are assigned to the opposite side.
- The algorithm is **successful** if $X = \sum_{v \in V} X_v < 1$.
- $\mathbb{E}[X] = \frac{1}{2}$.

Graph coloring

Can we find this coloring deterministically?

Midterm exam:

- Let $G = (V, E)$ be any n-vertex bipartite graph where each vertex v is associated with a list $L(v)$ of $\lceil \log_2 n \rceil + 1$ colors.
- A proper coloring can be computed <u>with probability $1/2$</u>.

Randomized algorithm:

- Assign each color to one of the two parts randomly.
- The algorithm is successful is every vertex v has a color in its list $L(v)$ assigned to its part.

- $X_v =$ the indicator random variable for the **bad event** that all colors in $L(v)$ are assigned to the opposite side.
- The algorithm is **successful** if $X = \sum_{v \in V} X_v < 1$.
- $\mathbb{E}[X] = \frac{1}{2}$.

Use **the method of conditional expectations** to find an allocation of the colors such that $X \leq \mathbb{E}[X] = \frac{1}{2} < 1$.

Summary

- Many randomized algorithms can be derandomized:

Deterministic **greedy** algorithm that sets the variables sequentially to optimize the conditional expectation.

In many cases, it is difficult to obtain such a greedy algorithm from scratch without first designing a randomized algorithm.

Summary

- Many randomized algorithms can be derandomized:

For some problems, we still do not know how to derandomize existing randomized algorithms.
https://en.wikipedia.org/wiki/Polynomial_identity_testing