Outline

Introduction

Upper Bounds on Testing Uniformity

Lower Bounds on Testing Uniformity

Extensions
Sublinear Algorithms: Algorithms that return reasonably good approximate answers without scanning or storing the entire input.
Sublinear Algorithms: Algorithms that return reasonably good approximate answers without scanning or storing the entire input

Usually these algorithms are randomised!
Sublinear Algorithms: Algorithms that return reasonably good approximate answers without scanning or storing the entire input.

Usually these algorithms are randomised!
**Sublinear Algorithms Overview**

**Sublinear Algorithms**: Algorithms that return reasonably good approximate answers without scanning or storing the entire input. Usually these algorithms are randomised!

- **Sublinear-(Time) Algorithms**: Algorithm may only inspect a small fraction of the whole input.
Sublinear Algorithms: Algorithms that return reasonably good approximate answers without scanning or storing the entire input.

Usually these algorithms are randomised!

Sublinear Algorithms

Sublinear-(Time) Algorithms
Algorithm may only inspect a small fraction of the whole input

Data Streaming Algorithms
Algorithm may only read input once and store a small fraction
Sublinear Algorithms Overview

**Sublinear Algorithms**: Algorithms that return reasonably good approximate answers without scanning or storing the entire input.

Usually these algorithms are randomised!

- **Sublinear-(Time) Algorithms**: Algorithm may only inspect a small fraction of the whole input.
- **Data Streaming Algorithms**: Algorithm may only read input once and store a small fraction.
- **Dimensionality Reduction**: Preprocess to reduce the size of the input.
Motivation

**Goal:** Estimate properties of **big** probability distributions

Transactions of 20-30 yr olds

Transactions of 30-40 yr olds

Testing closeness of two distributions: trend change?

Source: Slides by Ronitt Rubinfeld

Thanks to Krzysztof Onak (pointer) and Eric Price (graph)
Motivation

**Goal:** Estimate properties of **big** probability distributions

big means that the domain of the finite probability distribution is very large!
**Motivation**

**Goal:** Estimate properties of **big** probability distributions

big means that the domain of the finite probability distribution is very large!

- **Lottery** (are numbers equally likely?)
- **Birthday Distribution** (is the birthday distribution uniform over 365 days?)

![Chart showing distribution of first 100 drawings of Polish Multilotek](image)

Source: Slides by Ronitt Rubinfeld
Motivation

**Goal:** Estimate properties of **big** probability distributions

**big** means that the domain of the finite probability distribution is very large!

- **Lottery** (are numbers equally likely?)
- **Birthday Distribution** (is the birthday distribution uniform over 365 days?)
- **Shopping patterns** (are distributions the same or different?)

Source: Slides by Ronitt Rubinfeld
Motivation

Goal: Estimate properties of big probability distributions

big means that the domain of the finite probability distribution is very large!

- Lottery (are numbers equally likely?)
- Birthday Distribution (is the birthday distribution uniform over 365 days?)
- Shopping patterns (are distributions the same or different?)
- Physical Experiment (is the observed distribution close to the prediction?)

Source: Slides by Ronitt Rubinfeld
Motivation

**Goal:** Estimate properties of **big** probability distributions

Big means that the domain of the finite probability distribution is very large!

- **Lottery** (are numbers equally likely?)
- **Birthday Distribution** (is the birthday distribution uniform over 365 days?)
- **Shopping patterns** (are distributions the same or different?)
- **Physical Experiment** (is the observed distribution close to the prediction?)
- **Health** (are there correlations between zip code and health condition?)

Thanks to Krzysztof Onak (pointer) and Eric Price (graph)

Source: Slides by Ronitt Rubinfeld
Testing Probability Distribution (Formal Model)

Model

- Given one (or more) probability distribution \( p = (p_1, p_2, \ldots, p_n) \)
- distribution(s) are unknown, but can obtain independent samples
- also known: \( n \) (or a good estimate of it)
Testing Probability Distribution (Formal Model)

- Given one (or more) probability distribution \( p = (p_1, p_2, \ldots, p_n) \)
- distribution(s) are unknown, but can obtain independent samples
- also known: \( n \) (or a good estimate of it)

**Cost:** number of samples (queries)
Testing Probability Distribution (Formal Model)

- Given one (or more) probability distribution $p = (p_1, p_2, \ldots, p_n)$
- distribution(s) are unknown, but can obtain independent samples
- also known: $n$ (or a good estimate of it)

Cost: number of samples (queries)

Questions:
1. Is the distribution $p$ close to the uniform distribution $u$?
2. Is the distribution $p$ close to some other distribution $q$?
3. What is $\max_{1 \leq i \leq n} p_i$ (heavy hitter)?
4. Are the distributions $p$ and $q$ independent? \ldots
Testing Uniformity: Is the distribution $p$ close to the uniform distribution $u$?

Distance between Discrete Distributions

Examples:
1. $p = (1, 0, \ldots, 0)$ and $q = (0, 1, 0, \ldots, 0)$. Then $\|p - q\|_1 = 2$, $\|p - q\|_2 = \sqrt{2}$ and $\|p - q\|_\infty = 1$.
2. $p = (1, 0, \ldots, 0)$ and $q = (1/n, 1/n, \ldots, 1/n)$. Then $\|p - q\|_1 = 2 - 2n$, $\|p - q\|_2 = \sqrt{1 \cdot (1 - 1/n) + (n - 1) \cdot (1/n)^2} = \sqrt{1 - 1/n}$ and $\|p - q\|_\infty = 1 - 1/n$.
3. $p = (2/n, \ldots, 2/n)$ and $q = (0, \ldots, 0, 2/n)$. Then $\|p - q\|_1 = 2$, $\|p - q\|_2 = \sqrt{4/n \cdot (2/n)^2} = \sqrt{4/n}$ and $\|p - q\|_\infty = 2/n$.
Testing Uniformity: Is the distribution \( p \) close to the uniform distribution \( u \)?

Distance between Discrete Distributions

Let \( p \) and \( q \) be any two distributions over \( \{1, 2, \ldots, n\} \). Then:

1. \( L_1 \)-distance: \( \|p - q\|_1 = \sum_{i=1}^{n} |p_i - q_i| \in [0, 2] \),
2. \( L_2 \)-distance: \( \|p - q\|_2 = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \in [0, \sqrt{2}] \),
3. \( L_{\infty} \)-distance: \( \|p - q\|_{\infty} = \max_{i=1}^{n} |p_i - q_i| \in [0, 1] \).
**Testing Uniformity:** Is the distribution $p$ close to the uniform distribution $u$?

---

**Distance between Discrete Distributions**

Let $p$ and $q$ be any two distributions over $\{1, 2, \ldots, n\}$. Then:

1. **$L_1$-distance:** $\|p - q\|_1 = \sum_{i=1}^{n} |p_i - q_i| \in [0, 2]$,
2. **$L_2$-distance:** $\|p - q\|_2 = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \in [0, \sqrt{2}]$,
3. **$L_\infty$-distance:** $\|p - q\|_\infty = \max_{i=1}^{n} |p_i - q_i| \in [0, 1]$.

---

**Examples:**

---
Testing Uniformity: Is the distribution $p$ close to the uniform distribution $u$?

Distance between Discrete Distributions

Let $p$ and $q$ be any two distributions over $\{1, 2, \ldots, n\}$. Then:

1. $L_1$-distance: $\|p - q\|_1 = \sum_{i=1}^{n} |p_i - q_i| \in [0, 2]$,  
2. $L_2$-distance: $\|p - q\|_2 = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \in [0, \sqrt{2}]$,  
3. $L_\infty$-distance: $\|p - q\|_\infty = \max_{i=1}^{n} |p_i - q_i| \in [0, 1]$.

Examples:

1. $p = (1, 0, \ldots, 0)$, $q = (0, 1, 0, \ldots, 0)$. Then $\|p - q\|_1 = 2$, $\|p - q\|_2 = \sqrt{2}$ and $\|p - q\|_\infty = 1$.  

Lecture 13-14: Sublinear-Time Algorithms
Testing Uniformity: Is the distribution $p$ close to the uniform distribution $u$?

**Distance between Discrete Distributions**

Let $p$ and $q$ be any two distributions over $\{1, 2, \ldots, n\}$. Then:

1. $L_1$-distance: $\|p - q\|_1 = \sum_{i=1}^{n} |p_i - q_i| \in [0, 2]$,
2. $L_2$-distance: $\|p - q\|_2 = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \in [0, \sqrt{2}]$,
3. $L_{\infty}$-distance: $\|p - q\|_{\infty} = \max_{i=1}^{n} |p_i - q_i| \in [0, 1]$.

**Examples:**

1. $p = (1, 0, \ldots, 0)$, $q = (0, 1, 0, \ldots, 0)$. Then $\|p - q\|_1 = 2$, $\|p - q\|_2 = \sqrt{2}$ and $\|p - q\|_{\infty} = 1$.
2. $p = (1, 0, \ldots, 0)$, $q = (1/n, 1/n, \ldots, 1/n)$. Then $\|p - q\|_1 = 2 - 2/n$,
   $\|p - q\|_2 = \sqrt{1 \cdot (1 - 1/n)^2 + (n - 1) \cdot (1/n)^2} = \sqrt{1 - 1/n}$ and $\|p - q\|_{\infty} = 1 - 1/n$. 

Lecture 13-14: Sublinear-Time Algorithms
Testing Uniformity: Is the distribution $p$ close to the uniform distribution $u$?

**Distance between Discrete Distributions**

Let $p$ and $q$ be any two distributions over $\{1, 2, \ldots, n\}$. Then:

1. $L_1$-distance: $\|p - q\|_1 = \sum_{i=1}^{n} |p_i - q_i| \in [0, 2]$,
2. $L_2$-distance: $\|p - q\|_2 = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \in [0, \sqrt{2}]$,
3. $L_{\infty}$-distance: $\|p - q\|_{\infty} = \max_{i=1}^{n} |p_i - q_i| \in [0, 1]$.

**Examples:**

1. $p = (1, 0, \ldots, 0)$, $q = (0, 1, 0, \ldots, 0)$. Then $\|p - q\|_1 = 2$, $\|p - q\|_2 = \sqrt{2}$ and $\|p - q\|_{\infty} = 1$.
2. $p = (1, 0, \ldots, 0)$, $q = (1/n, 1/n, \ldots, 1/n)$. Then $\|p - q\|_1 = 2 - 2/n$, $\|p - q\|_2 = \sqrt{1 \cdot (1 - 1/n)^2 + (n - 1) \cdot (1/n)^2} = \sqrt{1 - 1/n}$ and $\|p - q\|_{\infty} = 1 - 1/n$.
3. $p = (2/n, \ldots, 2/n, 0, \ldots, 0)$ and $q = (0, \ldots, 0, 2/n, \ldots, 2/n)$. Then $\|p - q\|_1 = 2$, $\|p - q\|_2 = \sqrt{2 \cdot (n/2) \cdot (2/n)^2} = \sqrt{4/n}$ and $\|p - q\|_{\infty} = 2/n$. 

Lecture 13-14: Sublinear-Time Algorithms
Testing Uniformity:

Is the distribution $p$ close to the uniform distribution $u$?

Distance between Discrete Distributions

Let $p$ and $q$ be any two distributions over $\{1, 2, \ldots, n\}$. Then:

1. $L_1$-distance: $\|p - q\|_1 = \sum_{i=1}^{n} |p_i - q_i| \in [0, 2]$,

2. $L_2$-distance: $\|p - q\|_2 = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \in [0, \sqrt{2}]$,

3. $L_\infty$-distance: $\|p - q\|_\infty = \max_{i=1}^{n} |p_i - q_i| \in [0, 1]$.

Examples:

1. $p = (1, 0, \ldots, 0), q = (0, 1, 0, \ldots, 0)$. Then $\|p - q\|_1 = 2, \|p - q\|_2 = \sqrt{2}$ and $\|p - q\|_\infty = 1$.

2. $p = (1, 0, \ldots, 0), q = (1/n, 1/n, \ldots, 1/n)$. Then $\|p - q\|_1 = 2 - 2/n, \|p - q\|_2 = \sqrt{1 \cdot (1 - 1/n)^2 + (n-1) \cdot (1/n)^2} = \sqrt{1 - 1/n}$ and $\|p - q\|_\infty = 1 - 1/n$.

3. $p = (2/n, \ldots, 2/n, 0, \ldots, 0)$ and $q = (0, \ldots, 0, 2/n, \ldots, 2/n)$. Then $\|p - q\|_1 = 2, \|p - q\|_2 = \sqrt{2 \cdot (n/2) \cdot (2/n)^2} = \sqrt{4/n}$ and $\|p - q\|_\infty = 2/n$.

Disjoint distributions, yet $L_2$ and $L_\infty$ distances are small!
Outline

Introduction

Upper Bounds on Testing Uniformity

Lower Bounds on Testing Uniformity

Extensions
Testing Uniformity in the $L_1$-distance

Objective

Find an efficient tester such that
- Given any probability distribution $p$ and $\epsilon \in (0, 1)$
Testing Uniformity in the $L_1$-distance

**Objective**

Find an **efficient** tester such that
- Given any probability distribution $p$ and $\epsilon \in (0, 1)$
  - If $p$ is the **uniform distribution**, then $P[\text{ACCEPT}] \geq 2/3$,
Testing Uniformity in the $L_1$-distance

Objective

Find an efficient tester such that
- Given any probability distribution $p$ and $\epsilon \in (0, 1)$
  - If $p$ is the uniform distribution, then $\Pr[\text{ACCEPT}] \geq 2/3$,
  - If $p$ is $\epsilon$-far from uniform ($\sum_{i=1}^{n} |p_i - 1/n| \geq \epsilon$), then $\Pr[\text{REJECT}] \geq 2/3$. 

\[ \begin{align*}
  &\text{\emph{Objective}} \\
  &\text{Find an efficient tester such that} \\
  &\quad \text{- Given any probability distribution } p \text{ and } \epsilon \in (0, 1) \\
  &\quad \quad \text{- If } p \text{ is the uniform distribution, then } \Pr[\text{ACCEPT}] \geq 2/3, \\
  &\quad \quad \text{- If } p \text{ is } \epsilon\text{-far from uniform (} \sum_{i=1}^{n} |p_i - 1/n| \geq \epsilon \text{), then } \Pr[\text{REJECT}] \geq 2/3.
\end{align*} \]
Testing Uniformity in the $L_1$-distance

**Objective**

Find an efficient tester such that
- Given any probability distribution $p$ and $\epsilon \in (0, 1)$
  - If $p$ is the uniform distribution, then $P[\text{ACCEPT}] \geq 2/3$,
  - If $p$ is $\epsilon$-far from uniform ($\sum_{i=1}^{n} |p_i - 1/n| \geq \epsilon$), then $P[\text{REJECT}] \geq 2/3$.

- tester efficient (sub-linear) $\leadsto$ different from standard statistical tests!
- tester is allowed to have two-sided error
- there is a “grey area” when $p$ is different from but close to uniform, where the tester may give any result

\[ p_i \]

\[ u_i \]

![Graphs showing $p_i$ and $u_i$ distributions for $i = 1, 2, 3, 4, 5$.]
High Level Idea

Recall: $L_1$-distance is

$$\sum_{i=1}^{n} \left| p_i - \frac{1}{n} \right|$$
Recall: $L_1$-distance is

$$\sum_{i=1}^{n} |p_i - \frac{1}{n}|$$

First Idea might be to approximate each $p_i - \frac{1}{n}$, but this takes at least $\Omega(n)$ queries.
High Level Idea

Recall: $L_1$-distance is

$$\sum_{i=1}^{n} |p_i - \frac{1}{n}|$$

First Idea might be to approximate each $p_i - \frac{1}{n}$, but this takes at least $\Omega(n)$ queries.

Birthday Paradox:
- If $p$ is (close to) uniform, expect to see collisions after $\approx \sqrt{n}$ samples.
- If $p$ is far from uniform, expect to see collisions with ??

$$P[\text{collision}] \approx 1 - \exp\left(-\frac{k(k-1)}{2 \cdot 365}\right)$$
**High Level Idea**

**Recall:** $L_1$-distance is

$$
\sum_{i=1}^{n} \left| p_i - \frac{1}{n} \right|
$$

First Idea might be to approximate each $p_i - \frac{1}{n}$, but this takes at least $\Omega(n)$ queries.

**Birthday Paradox:**

- If $p$ is (close to) uniform, expect to see collisions after $\approx \sqrt{n}$ samples
- If $p$ is far from uniform, expect to see collisions with even less samples

$$
P[\text{collision}] \approx 1 - \exp\left( -\frac{k(k-1)}{2 \cdot 365} \right)
$$
Collision Probability and $L_2$-distance

$$\|p - u\|_2^2$$
Collision Probability and $L_2$-distance

$$\|p - u\|_2^2 = \sum_{i=1}^{n} (p_i - 1/n)^2$$
Collision Probability and $L_2$-distance

\[ \|p - u\|_2^2 = \sum_{i=1}^{n} (p_i - 1/n)^2 = \sum_{i=1}^{n} p_i^2 - 2 \cdot \sum_{i=1}^{n} p_i \cdot \frac{1}{n} + \sum_{i=1}^{n} \left( \frac{1}{n} \right)^2 \]
Collision Probability and $L_2$-distance

\[
\|p - u\|_2^2 = \sum_{i=1}^{n} (p_i - 1/n)^2 = \sum_{i=1}^{n} p_i^2 - 2 \cdot \sum_{i=1}^{n} p_i \cdot \frac{1}{n} + \sum_{i=1}^{n} \left( \frac{1}{n} \right)^2 = \|p\|_2^2 - \frac{1}{n}
\]
Collision Probability and $L_2$-distance

\[ \| p - u \|_2^2 = \sum_{i=1}^{n} (p_i - 1/n)^2 = \sum_{i=1}^{n} p_i^2 - 2 \cdot \sum_{i=1}^{n} p_i \cdot \frac{1}{n} + \sum_{i=1}^{n} \left( \frac{1}{n} \right)^2 = \| p \|_2^2 - \frac{1}{n} \]

Hence $\| p \|_2^2 = \sum_{i=1}^{n} p_i^2$ captures the $L_2$-distance to the uniform distribution.
Collision Probability and $L_2$-distance

\[ \|p - u\|_2^2 = \sum_{i=1}^{n} (p_i - 1/n)^2 = \sum_{i=1}^{n} p_i^2 - 2 \cdot \sum_{i=1}^{n} p_i \cdot \frac{1}{n} + \sum_{i=1}^{n} \left( \frac{1}{n} \right)^2 = \|p\|_2^2 - \frac{1}{n} \]

Hence $\|p\|_2^2 = \sum_{i=1}^{n} p_i^2$ captures the $L_2$-distance to the uniform distribution

**APPROXIMATE $\|p\|_2^2$**

1. Sample $r$ elements from $p$, $x_1, x_2, \ldots, x_r \in \{1, \ldots, n\}$
2. For each $1 \leq i < j \leq r$,
   \[ \sigma_{i,j} := \begin{cases} 
   1 & \text{if } x_i = x_j, \\
   0 & \text{otherwise}. 
\end{cases} \]
3. Output $Y := \frac{1}{\binom{r}{2}} \cdot \sum_{1 \leq i < j \leq r} \sigma_{i,j}$. 

Lecture 13-14: Sublinear-Time Algorithms
Collision Probability and $L_2$-distance

\[ \|p - u\|_2^2 = \sum_{i=1}^{n} (p_i - 1/n)^2 = \sum_{i=1}^{n} p_i^2 - 2 \cdot \sum_{i=1}^{n} p_i \cdot \frac{1}{n} + \sum_{i=1}^{n} \left( \frac{1}{n} \right)^2 = \|p\|_2^2 - \frac{1}{n} \]

Hence $\|p\|_2^2 = \sum_{i=1}^{n} p_i^2$ captures the $L_2$-distance to the uniform distribution.

**APPROXIMATE $\|p\|_2^2$**

1. Sample $r$ elements from $p$, $x_1, x_2, \ldots, x_r \in \{1, \ldots, n\}$
2. For each $1 \leq i < j \leq r$,
   \[ \sigma_{i,j} := \begin{cases} 
   1 & \text{if } x_i = x_j, \\
   0 & \text{otherwise}. 
   \end{cases} \]
3. Output $Y := \frac{1}{\binom{r}{2}} \cdot \sum_{1 \leq i < j \leq r} \sigma_{i,j}$. 

number of samples $r$ will be specified later!
Runtime Analysis

- Sampling/Query Complexity is obviously $r$
Runtime Analysis

- Sampling/Query Complexity is obviously $r$
- Time Complexity??
Runtime Analysis

- **Sampling/Query Complexity** is obviously \( r \)
- **Time Complexity??**
  - Evaluating \( \sum_{1 \leq i < j \leq r} \sigma_{i,j} \) directly takes time quadratic in \( r \)
Runtime Analysis

- **Sampling/Query Complexity** is obviously $r$
- **Time Complexity??**
  - Evaluating $\sum_{1 \leq i < j \leq r} \sigma_{i,j}$ directly takes time quadratic in $r$
  - **Linear-Time Solution:**
    1. Maintain array $A = (a_1, a_2, \ldots, a_n)$, where $a_i \in [0, r]$ counts the frequency of samples of item $i$
    2. Use formula
       
       $$\sum_{1 \leq i < j \leq r} \sigma_{i,j} \overset{(*)}{=} \sum_{k=1}^{n} \binom{a_k}{2}$$

3. Since at most $O(r)$ elements in $A$ will be non-zero, using hash-function allows computation in time $O(r)$
Runtime Analysis

- **Sampling/Query Complexity** is obviously $r$
- **Time Complexity??**
  - Evaluating $\sum_{1 \leq i < j \leq r} \sigma_{i,j}$ directly takes time quadratic in $r$
  - **Linear-Time Solution:**
    1. Maintain array $A = (a_1, a_2, \ldots, a_n)$, where $a_i \in [0, r]$ counts the frequency of samples of item $i$
    2. Use formula

$$\sum_{1 \leq i < j \leq r} \sigma_{i,j} \overset{\text{(\star)}}{=} \sum_{k=1}^{n} \binom{a_k}{2}$$

3. Since at most $O(r)$ elements in $A$ will be non-zero, using hash-function allows computation in time $O(r)$

**Proof of (\star):**

$$\sum_{1 \leq i < j \leq r} \sigma_{i,j}$$
Runtime Analysis

- **Sampling/Query Complexity** is obviously $r$
- **Time Complexity??**
  - Evaluating $\sum_{1 \leq i < j \leq r} \sigma_{i,j}$ directly takes time quadratic in $r$
  - **Linear-Time Solution:**
    1. Maintain array $A = (a_1, a_2, \ldots, a_n)$, where $a_i \in [0, r]$ counts the frequency of samples of item $i$
    2. Use formula
       \[
       \sum_{1 \leq i < j \leq r} \sigma_{i,j} = n \sum_{k=1}^{n} \binom{a_k}{2}
       \]
    3. Since at most $O(r)$ elements in $A$ will be non-zero, using hash-function allows computation in time $O(r)$

**Proof of ($\star$):**

\[
\sum_{1 \leq i < j \leq r} \sigma_{i,j} = \sum_{1 \leq i < j \leq r} 1_{x_i=x_j}
\]
Runtime Analysis

- **Sampling/Query Complexity** is obviously \( r \)
- **Time Complexity??**
  - Evaluating \( \sum_{1 \leq i < j \leq r} \sigma_{i,j} \) directly takes time quadratic in \( r \)
  - **Linear-Time Solution:**
    1. Maintain array \( A = (a_1, a_2, \ldots, a_n) \), where \( a_i \in [0, r] \) counts the frequency of samples of item \( i \)
    2. Use formula

\[
\sum_{1 \leq i < j \leq r} \sigma_{i,j} = \sum_{k=1}^{n} \binom{a_k}{2}
\]

3. Since at most \( O(r) \) elements in \( A \) will be non-zero, using hash-function allows computation in time \( O(r) \)

**Proof of \((\star)\):**

\[
\sum_{1 \leq i < j \leq r} \sigma_{i,j} = \sum_{1 \leq i < j \leq r} 1_{x_i = x_j}
\]

\[
= \sum_{1 \leq i < j \leq r} \sum_{k=1}^{n} 1_{x_i = x_j = k}
\]
Runtime Analysis

- **Sampling/Query Complexity** is obviously $r$
- **Time Complexity??**
  - Evaluating $\sum_{1 \leq i < j \leq r} \sigma_{i,j}$ directly takes time quadratic in $r$
  - **Linear-Time Solution:**
    1. Maintain array $A = (a_1, a_2, \ldots, a_n)$, where $a_i \in [0, r]$ counts the frequency of samples of item $i$
    2. Use formula

\[
\sum_{1 \leq i < j \leq r} \sigma_{i,j} \overset{(*)}{=} \sum_{k=1}^{n} \binom{a_k}{2}
\]

3. Since at most $O(r)$ elements in $A$ will be non-zero, using hash-function allows computation in time $O(r)$

**Proof of (\overset{(*)}{=})**:

\[
\sum_{1 \leq i < j \leq r} \sigma_{i,j} = \sum_{1 \leq i < j \leq r} 1_{x_i=x_j}
\]

\[
= \sum_{1 \leq i < j \leq r} \sum_{k=1}^{n} 1_{x_i=x_j=k} = \sum_{k=1}^{n} \sum_{1 \leq i < j \leq r} 1_{x_i=x_j=k}
\]
Runtime Analysis

- Sampling/Query Complexity is obviously \( r \)
- Time Complexity??
  - Evaluating \( \sum_{1 \leq i < j \leq r} \sigma_{i,j} \) directly takes time quadratic in \( r \)
  - Linear-Time Solution:
    1. Maintain array \( A = (a_1, a_2, \ldots, a_n) \), where \( a_i \in [0, r] \) counts the frequency of samples of item \( i \)
    2. Use formula

\[
\sum_{1 \leq i < j \leq r} \sigma_{i,j} = \sum_{k=1}^{n} \binom{a_k}{2}
\]

3. Since at most \( O(r) \) elements in \( A \) will be non-zero, using hash-function allows computation in time \( O(r) \)

Proof of (\( \star \)):

\[
\sum_{1 \leq i < j \leq r} \sigma_{i,j} = \sum_{1 \leq i < j \leq r} 1_{x_i = x_j} = \sum_{k=1}^{n} \sum_{1 \leq i < j \leq r} 1_{x_i = x_j} = \sum_{k=1}^{n} \sum_{1 \leq i < j \leq r} 1_{x_i = x_j = k} = \sum_{k=1}^{n} \binom{a_k}{2}. \quad \Box
\]
Approximation Analysis

For any value $r \geq 30 \cdot \frac{\sqrt{n}}{e^2}$, the algorithm returns a value $Y$ such that

$$\mathbb{P}\left[|Y - \|p\|_2^2| \geq \epsilon \cdot \|p\|_2^2\right] \leq 1/3.$$
Approximation Analysis

For any value $r \geq 30 \cdot \frac{\sqrt{n}}{\epsilon^2}$, the algorithm returns a value $Y$ such that

$$P\left[\left|Y - \|p\|_2^2\right| \geq \epsilon \cdot \|p\|_2^2\right] \leq 1/3.$$

Proof (1/5):

- Let us start by computing $E[Y]$:

$$E[Y]$$
Approximation Analysis

For any value $r \geq 30 \cdot \frac{\sqrt{n}}{\epsilon^2}$, the algorithm returns a value $Y$ such that

$$\mathbb{P}\left[\left| Y - \|p\|_2^2 \right| \geq \epsilon \cdot \|p\|_2^2 \right] \leq \frac{1}{3}.$$

**Proof (1/5):**
- Let us start by computing $\mathbb{E}[Y]$:
  $$\mathbb{E}[Y] = \frac{1}{\binom{r}{2}} \cdot \sum_{1 \leq i < j \leq r} \mathbb{E}[\sigma_{i,j}]$$
Approximation Analysis

Analysis

For any value $r \geq 30 \cdot \frac{\sqrt{n}}{\epsilon^2}$, the algorithm returns a value $Y$ such that

$$
P\left[ \left| Y - \|p\|_2^2 \right| \geq \epsilon \cdot \|p\|_2^2 \right] \leq 1/3.
$$

Proof (1/5):
- Let us start by computing $E[ Y ]$:

$$
E[ Y ] = \frac{1}{\binom{r}{2}} \cdot \sum_{1 \leq i < j \leq r} \mathbb{E}[ \sigma_{i,j} ]
$$

$$
= \frac{1}{\binom{r}{2}} \cdot \sum_{1 \leq i < j \leq r} \sum_{k=1}^{n} \mathbb{P}[ x_i = k ] \cdot \mathbb{P}[ x_j = k ]
$$
Approximation Analysis

For any value \( r \geq 30 \cdot \frac{\sqrt{n}}{\epsilon^2} \), the algorithm returns a value \( Y \) such that

\[
P\left[ \left| Y - \|p\|_2^2 \right| \geq \epsilon \cdot \|p\|_2^2 \right] \leq \frac{1}{3}.
\]

Proof (1/5):

- Let us start by computing \( \mathbb{E}[Y] \):

\[
\mathbb{E}[Y] = \frac{1}{(r/2)} \cdot \sum_{1 \leq i < j \leq r} \mathbb{E}[\sigma_{i,j}]
\]

\[
= \frac{1}{(r/2)} \cdot \sum_{1 \leq i < j \leq r} \sum_{k=1}^{n} \mathbb{P}[x_i = k] \cdot \mathbb{P}[x_j = k]
\]

\[
= \frac{1}{(r/2)} \cdot \sum_{1 \leq i < j \leq r} \sum_{k=1}^{n} p_k^2 = \|p\|_2^2.
\]
Approximation Analysis

For any value \( r \geq 30 \cdot \frac{\sqrt{n}}{\epsilon^2} \), the algorithm returns a value \( Y \) such that

\[
P\left[ \left| Y - \|p\|_2^2 \right| \geq \epsilon \cdot \|p\|_2^2 \right] \leq \frac{1}{3}.
\]

Proof (1/5):
- Let us start by computing \( E[ Y ] \):

\[
E[ Y ] = \frac{1}{r \choose 2} \cdot \sum_{1 \leq i < j \leq r} E[ \sigma_{i,j} ]
\]

\[
= \frac{1}{r \choose 2} \cdot \sum_{1 \leq i < j \leq r} \sum_{k=1}^{n} P[ x_i = k ] \cdot P[ x_j = k ]
\]

\[
= \frac{1}{r \choose 2} \cdot \sum_{1 \leq i < j \leq r} \sum_{k=1}^{n} p_k^2 = \|p\|_2^2.
\]

- Analysis of the deviation more complex (see next slides):
Approximation Analysis

For any value \( r \geq 30 \cdot \frac{\sqrt{n}}{\epsilon^2} \), the algorithm returns a value \( Y \) such that

\[
P\left[ |Y - \|p\|_2^2| \geq \epsilon \cdot \|p\|_2^2 \right] \leq 1/3.
\]

**Proof (1/5):**

- Let us start by computing \( \mathbb{E}[Y] \):

\[
\mathbb{E}[Y] = \frac{1}{r} \cdot \sum_{1 \leq i < j \leq r} \mathbb{E}[\sigma_{i,j}]
\]

\[
= \frac{1}{r} \cdot \sum_{1 \leq i < j \leq r} \sum_{k=1}^{n} \mathbb{P}[x_i = k] \cdot \mathbb{P}[x_j = k]
\]

\[
= \frac{1}{r} \cdot \sum_{1 \leq i < j \leq r} \sum_{k=1}^{n} p_{k}^2 = \|p\|_2^2.
\]

- Analysis of the deviation more complex (see next slides):
  - requires a careful analysis of the variance
  (note that the \( \sigma_{i,j} \)'s are not even pairwise independent! - **Exercise**)

Lecture 13-14: Sublinear-Time Algorithms
Approximation Analysis

For any value $r \geq 30 \cdot \frac{\sqrt{n}}{\epsilon^2}$, the algorithm returns a value $Y$ such that

$$P \left[ \left| Y - \|p\|_2^2 \right| \geq \epsilon \cdot \|p\|_2^2 \right] \leq \frac{1}{3}.$$

Proof (1/5):

- Let us start by computing $E[Y]$: 

  $$E[Y] = \frac{1}{\binom{r}{2}} \cdot \sum_{1 \leq i < j \leq r} E[\sigma_{i,j}]$$

  $$= \frac{1}{\binom{r}{2}} \cdot \sum_{1 \leq i < j \leq r} \sum_{k=1}^{n} P[x_i = k] \cdot P[x_j = k]$$

  $$= \frac{1}{\binom{r}{2}} \cdot \sum_{1 \leq i < j \leq r} \sum_{k=1}^{n} p_k^2 = \|p\|_2^2.$$

Analysis of the deviation more complex (see next slides):

- requires a careful analysis of the variance
  (note that the $\sigma_{i,j}$’s are not even pairwise independent! - Exercise)
- final step is an application of Chebysheff’s inequality
Approximation Analysis

For any value $r \geq 30 \cdot \frac{\sqrt{n}}{\epsilon^2}$, the algorithm returns a value $Y$ such that

$$P\left[|Y - \|p\|_2^2| \geq \epsilon \cdot \|p\|_2^2\right] \leq \frac{1}{3}.$$ 

Proof (2/5):
For any value \( r \geq 30 \cdot \frac{\sqrt{n}}{e^2} \), the algorithm returns a value \( Y \) such that

\[
P\left[ \left| Y - \|p\|_2^2 \right| \geq \epsilon \cdot \|p\|_2^2 \right] \leq \frac{1}{3}.
\]

Proof (2/5):

- Define \( \hat{\sigma}_{i,j} := \sigma_{i,j} - E[\sigma_{i,j}] \). Note \( E[\hat{\sigma}_{i,j}] = 0 \), \( \hat{\sigma}_{i,j} \leq \sigma_{i,j} \) and
Approximation Analysis

For any value \( r \geq 30 \cdot \frac{\sqrt{n}}{\epsilon^2} \), the algorithm returns a value \( Y \) such that

\[
P\left[ \left| Y - \|p\|_2^2 \right| \geq \epsilon \cdot \|p\|_2^2 \right] \leq \frac{1}{3}.
\]

Proof (2/5):

- Define \( \widehat{\sigma}_{i,j} := \sigma_{i,j} - \mathbb{E}[\sigma_{i,j}] \). Note \( \mathbb{E}[\widehat{\sigma}_{i,j}] = 0 \), \( \widehat{\sigma}_{i,j} \leq \sigma_{i,j} \) and

\[
\text{Var} \left[ \sum_{1 \leq i < j \leq r} \sigma_{i,j} \right] = \mathbb{E} \left[ \left( \sum_{1 \leq i < j \leq r} \sigma_{i,j} - \sum_{1 \leq i < j \leq r} \mathbb{E}[\sigma_{i,j}] \right)^2 \right] = \mathbb{E} \left[ \left( \sum_{1 \leq i < j \leq r} \widehat{\sigma}_{i,j} \right)^2 \right].
\]
Approximation Analysis

For any value $r \geq 30 \cdot \sqrt{\frac{n}{\epsilon^2}}$, the algorithm returns a value $Y$ such that

$$P \left[ |Y - \|p\|_2^2| \geq \epsilon \cdot \|p\|_2^2 \right] \leq \frac{1}{3}.$$

Proof (2/5):

- Define $\hat{\sigma}_{i,j} := \sigma_{i,j} - E[\sigma_{i,j}]$. Note $E[\hat{\sigma}_{i,j}] = 0$, $\hat{\sigma}_{i,j} \leq \sigma_{i,j}$ and

$$\text{Var} \left[ \sum_{1 \leq i < j \leq r} \sigma_{i,j} \right] = E \left[ \left( \sum_{1 \leq i < j \leq r} \sigma_{i,j} - \sum_{1 \leq i < j \leq r} E[\sigma_{i,j}] \right)^2 \right] = E \left[ \left( \sum_{1 \leq i < j \leq r} \hat{\sigma}_{i,j} \right)^2 \right].$$

- Expanding yields:

$$\sum_{1 \leq i < j \leq r} E \left[ \hat{\sigma}_{i,j}^2 \right] + \sum_{i,j,k,\ell \text{ diff.}} E[\hat{\sigma}_{i,j} \cdot \hat{\sigma}_{k,\ell}] + 4 \cdot \sum_{1 \leq i < j < k \leq r} E[\hat{\sigma}_{i,j} \cdot \hat{\sigma}_{j,k}].$$
Approximation Analysis

For any value $r \geq 30 \cdot \frac{\sqrt{n}}{\epsilon^2}$, the algorithm returns a value $Y$ such that

$$P \left[ \left| Y - \|p\|_2^2 \right| \geq \epsilon \cdot \|p\|_2^2 \right] \leq 1/3.$$

Proof (3/5):

$$A = \sum_{1 \leq i < j \leq r} E \left[ \sigma_{i,j}^2 \right].$$
Approximation Analysis

For any value \( r \geq 30 \cdot \frac{\sqrt{n}}{\epsilon^2} \), the algorithm returns a value \( Y \) such that

\[
P\left[ \left| Y - \|p\|_2^2 \right| \geq \epsilon \cdot \|p\|_2^2 \right] \leq \frac{1}{3}.
\]

Proof (3/5):

\[
A = \sum_{1 \leq i < j \leq r} E\left[ \hat{\sigma}_{i,j}^2 \right] \leq \sum_{1 \leq i < j \leq r} E\left[ \sigma_{i,j}^2 \right]
\]
Approximation Analysis

For any value \( r \geq 30 \cdot \frac{\sqrt{n}}{\epsilon^2} \), the algorithm returns a value \( Y \) such that

\[
P \left[ \left| Y - \|p\|_2^2 \right| \geq \epsilon \cdot \|p\|_2^2 \right] \leq \frac{1}{3}.
\]

Proof (3/5):

\[
A = \sum_{1 \leq i < j \leq r} E \left[ \hat{\sigma}_{i,j}^2 \right] \leq \sum_{1 \leq i < j \leq r} E \left[ \sigma_{i,j}^2 \right] = \sum_{1 \leq i < j \leq r} E \left[ \sigma_{i,j} \right]
\]
Approximation Analysis

For any value $r \geq 30 \cdot \frac{\sqrt{n}}{\epsilon^2}$, the algorithm returns a value $Y$ such that

$$P\left[Y - \|p\|_2^2 \geq \epsilon \cdot \|p\|_2^2\right] \leq 1/3.$$

Proof (3/5):

$$A = \sum_{1 \leq i < j \leq r} E\left[\hat{\sigma}_{i,j}^2\right] \leq \sum_{1 \leq i < j \leq r} E\left[\sigma_{i,j}^2\right] = \sum_{1 \leq i < j \leq r} E[\sigma_{i,j}] = \binom{r}{2} \cdot \|p\|_2^2.$$
Approximation Analysis

For any value $r \geq 30 \cdot \frac{\sqrt{n}}{\epsilon^2}$, the algorithm returns a value $Y$ such that

$$P \left[ |Y - \|p\|_2^2| \geq \epsilon \cdot \|p\|_2^2 \right] \leq 1/3.$$

Proof (3/5):

$$A = \sum_{1 \leq i < j \leq r} E \left[ \hat{\sigma}_{i,j}^2 \right] \leq \sum_{1 \leq i < j \leq r} E \left[ \sigma_{i,j}^2 \right] = \sum_{1 \leq i < j \leq r} E [\sigma_{i,j}] = \binom{r}{2} \cdot \|p\|_2^2.$$

$$B = \sum_{i, j, k, \ell \text{ diff.}} E [\hat{\sigma}_{i,j} \cdot \hat{\sigma}_{k,\ell}]$$
Approximation Analysis

Analysis

For any value \( r \geq 30 \cdot \frac{\sqrt{n}}{\epsilon^2} \), the algorithm returns a value \( Y \) such that

\[
P\left[ |Y - \|p\|_2^2| \geq \epsilon \cdot \|p\|_2^2 \right] \leq 1/3.
\]

Proof (3/5):

\[
A = \sum_{1 \leq i < j \leq r} E\left[ \hat{\sigma}_{i,j}^2 \right] \leq \sum_{1 \leq i < j \leq r} E\left[ \sigma_{i,j}^2 \right] = \sum_{1 \leq i < j \leq r} E[\sigma_{i,j}] = \binom{r}{2} \cdot \|p\|_2^2.
\]

\[
B = \sum_{i, j, k, \ell \text{ diff.}} E[\hat{\sigma}_{i,j} \cdot \hat{\sigma}_{k,\ell}] = \sum_{i, j, k, \ell \text{ diff.}} E[\hat{\sigma}_{i,j}] \cdot E[\hat{\sigma}_{k,\ell}]
\]

Covariance Formula:

\[
E\left[ (X - E[X])(Y - E[Y]) \right] = E[XY] - E[X]E[Y]
\]
Approximation Analysis

For any value \( r \geq 30 \cdot \frac{\sqrt{n}}{\epsilon^2} \), the algorithm returns a value \( Y \) such that

\[
P\left[ \left| Y - \|p\|_2^2 \right| \geq \epsilon \cdot \|p\|_2^2 \right] \leq \frac{1}{3}.
\]

Proof (3/5):

\[
A = \sum_{1 \leq i < j \leq r} \mathbb{E}\left[ \hat{\sigma}_{i,j}^2 \right] \leq \sum_{1 \leq i < j \leq r} \mathbb{E}\left[ \sigma_{i,j}^2 \right] = \sum_{1 \leq i < j \leq r} \mathbb{E}\left[ \sigma_{i,j} \right] = \binom{r}{2} \cdot \|p\|_2^2.
\]

\[
B = \sum_{i, j, k, \ell \text{ diff.}} \mathbb{E}\left[ \hat{\sigma}_{i,j} \cdot \hat{\sigma}_{k,\ell} \right] = \sum_{i, j, k, \ell \text{ diff.}} \mathbb{E}\left[ \hat{\sigma}_{i,j} \right] \cdot \mathbb{E}\left[ \hat{\sigma}_{k,\ell} \right] = 0.
\]
Approximation Analysis

For any value \( r \geq 30 \cdot \frac{\sqrt{n}}{\epsilon^2} \), the algorithm returns a value \( Y \) such that

\[
P \left[ \left| Y - \|p\|_2^2 \right| \geq \epsilon \cdot \|p\|_2^2 \right] \leq \frac{1}{3}.
\]

Proof (3/5):

\[
A = \sum_{1 \leq i < j \leq r} E \left[ \hat{\sigma}_{i,j}^2 \right] \leq \sum_{1 \leq i < j \leq r} E \left[ \sigma_{i,j}^2 \right] = \sum_{1 \leq i < j \leq r} E \left[ \sigma_{i,j} \right] = \binom{r}{2} \cdot \|p\|_2^2.
\]

\[
B = \sum_{i, j, k, \ell \text{ diff.}} E \left[ \hat{\sigma}_{i,j} \cdot \hat{\sigma}_{k,\ell} \right] = \sum_{i, j, k, \ell \text{ diff.}} E \left[ \hat{\sigma}_{i,j} \right] \cdot E \left[ \hat{\sigma}_{k,\ell} \right] = 0.
\]

\[
C = \sum_{1 \leq i < j < k \leq r} E \left[ \hat{\sigma}_{i,j} \hat{\sigma}_{i,k} \right]
\]
Approximation Analysis

For any value \( r \geq 30 \cdot \sqrt[3]{n} \), the algorithm returns a value \( Y \) such that

\[
P \left[ \left| Y - \| p \|_2^2 \right| \geq \epsilon \cdot \| p \|_2^2 \right] \leq 1/3.
\]

Proof (3/5):

\[
A = \sum_{1 \leq i < j \leq r} E \left[ \hat{\sigma}_{i,j}^2 \right] \leq \sum_{1 \leq i < j \leq r} E \left[ \sigma_{i,j}^2 \right] = \sum_{1 \leq i < j \leq r} E \left[ \sigma_{i,j} \right] = \binom{r}{2} \cdot \| p \|_2^2.
\]

\[
B = \sum_{i, j, k, \ell \text{ diff.}} E \left[ \hat{\sigma}_{i,j} \cdot \hat{\sigma}_{k,\ell} \right] = \sum_{i, j, k, \ell \text{ diff.}} E \left[ \hat{\sigma}_{i,j} \right] \cdot E \left[ \hat{\sigma}_{k,\ell} \right] = 0.
\]

\[
C = \sum_{1 \leq i < j < k \leq r} E \left[ \hat{\sigma}_{i,j} \hat{\sigma}_{i,k} \right] \leq \sum_{1 \leq i < j < k \leq r} E \left[ \sigma_{i,j} \sigma_{i,k} \right]
\]

Covariance Formula: \( E \left[ (X - E[X])(Y - E[Y]) \right] = E[XY] - E[X]E[Y] \)
Approximation Analysis

For any value \( r \geq 30 \cdot \frac{\sqrt{n}}{\epsilon^2} \), the algorithm returns a value \( Y \) such that

\[
P\left[ \left| Y - \|p\|_2^2 \right| \geq \epsilon \cdot \|p\|_2^2 \right] \leq \frac{1}{3}.
\]

Proof (3/5):

\[
A = \sum_{1 \leq i < j \leq r} E\left[ \hat{\sigma}_{i,j}^2 \right] \leq \sum_{1 \leq i < j \leq r} E\left[ \sigma_{i,j}^2 \right] = \sum_{1 \leq i < j \leq r} E[\sigma_{i,j}] = \left( \begin{array}{c} r \\ 2 \end{array} \right) \cdot \|p\|_2^2.
\]

\[
B = \sum_{i, j, k, \ell \text{ diff.}} E[\hat{\sigma}_{i,j} \cdot \hat{\sigma}_{k,\ell}] = \sum_{i, j, k, \ell \text{ diff.}} E[\hat{\sigma}_{i,j}] \cdot E[\hat{\sigma}_{k,\ell}] = 0.
\]

Covariance Formula:

\[
E\left[ (X - E[X])(Y - E[Y]) \right] = E[XY] - E[X]E[Y]
\]

\[
C = \sum_{1 \leq i < j < k \leq r} E[\hat{\sigma}_{i,j} \hat{\sigma}_{i,k}] \leq \sum_{1 \leq i < j < k \leq r} E[\sigma_{i,j} \sigma_{i,k}]
\]

\[
= \sum_{1 \leq i < j < k \leq r} \sum_{\ell \in [n]} P[X_i = X_j = X_k = \ell]
\]
Approximation Analysis

For any value \( r \geq 30 \cdot \frac{\sqrt{n}}{\epsilon^2} \), the algorithm returns a value \( Y \) such that

\[
P \left[ \left| Y - \|p\|_2^2 \right| \geq \epsilon \cdot \|p\|_2^2 \right] \leq 1/3.
\]

Proof (3/5):

\[
A = \sum_{1 \leq i < j \leq r} E \left[ \hat{\sigma}_{i,j}^2 \right] \leq \sum_{1 \leq i < j \leq r} E \left[ \sigma_{i,j}^2 \right] = \sum_{1 \leq i < j \leq r} E \left[ \sigma_{i,j} \right] = \binom{r}{2} \cdot \|p\|_2^2.
\]

\[
B = \sum_{i, j, k, \ell \text{ diff.}} E \left[ \hat{\sigma}_{i,j} \cdot \hat{\sigma}_{k,\ell} \right] = \sum_{i, j, k, \ell \text{ diff.}} E \left[ \hat{\sigma}_{i,j} \right] \cdot E \left[ \hat{\sigma}_{k,\ell} \right] = 0.
\]

Covariance Formula: \( E \left[ (X - E[X])(Y - E[Y]) \right] = E[XY] - E[X] E[Y] \)

\[
C = \sum_{1 \leq i < j < k \leq r} E \left[ \hat{\sigma}_{i,j} \hat{\sigma}_{i,k} \right] \leq \sum_{1 \leq i < j < k \leq r} E \left[ \sigma_{i,j} \sigma_{i,k} \right]
\]

\[
= \sum_{1 \leq i < j < k \leq r} \sum_{\ell \in [n]} P[ X_i = X_j = X_k = \ell ] = \binom{r}{3} \cdot \sum_{\ell \in [n]} p_{\ell}^3
\]
Approximation Analysis

For any value $r \geq 30 \cdot \sqrt{\frac{n}{\epsilon^2}}$, the algorithm returns a value $Y$ such that

$$
P\left[ |Y - \|p\|_2^2| \geq \epsilon \cdot \|p\|_2^2 \right] \leq 1/3.
$$

Proof (3/5):

$$
A = \sum_{1 \leq i < j \leq r} \mathbb{E}\left[ \sigma^2_{i,j} \right] \leq \sum_{1 \leq i < j \leq r} \mathbb{E}\left[ \sigma^2_{i,j} \right] = \sum_{1 \leq i < j \leq r} \mathbb{E}[\sigma_{i,j}] = \left( \frac{r}{2} \right) \cdot \|p\|_2^2.
$$

$$
B = \sum_{i, j, k, \ell \text{ diff.}} \mathbb{E}[\hat{\sigma}_{i,j} \cdot \hat{\sigma}_{k,\ell}] = \sum_{i, j, k, \ell \text{ diff.}} \mathbb{E}[\hat{\sigma}_{i,j}] \cdot \mathbb{E}[\hat{\sigma}_{k,\ell}] = 0.
$$

$$
C = \sum_{1 \leq i < j < k \leq r} \mathbb{E}[\hat{\sigma}_{i,j} \hat{\sigma}_{i,k}] \leq \sum_{1 \leq i < j < k \leq r} \mathbb{E}[\sigma_{i,j} \sigma_{i,k}]
$$

$$
= \sum_{1 \leq i < j < k \leq r} \sum_{\ell \in [n]} \mathbb{P}[X_i = X_j = X_k = \ell] = \binom{r}{3} \cdot \sum_{\ell \in [n]} p^3_\ell \leq \frac{\sqrt{3}}{2} \left( \binom{r}{2} \|p\|_2^2 \right)^{3/2}
$$
For any value \( r \geq 30 \cdot \frac{\sqrt{n}}{\epsilon^2} \), the algorithm returns a value \( Y \) such that

\[
P\left[ \left| Y - \|p\|_2^2 \right| \geq \epsilon \cdot \|p\|_2^2 \right] \leq \frac{1}{3}.
\]

Proof (4/5):
- We have just shown that:

\[
\text{Var} \left[ \sum_{1 \leq i < j \leq r} \sigma_{i,j} \right] = A + B + 4C
\]
Approximation Analysis

For any value $r \geq 30 \cdot \frac{\sqrt{n}}{\epsilon^2}$, the algorithm returns a value $Y$ such that

$$
P\left[ |Y - \|p\|_2^2| \geq \epsilon \cdot \|p\|_2^2 \right] \leq \frac{1}{3}.
$$

Proof (4/5):

- We have just shown that:

  $\text{Var} \left[ \sum_{1 \leq i < j \leq r} \sigma_{i,j} \right] = A + B + 4C$

  $$= \binom{r}{2} \cdot \|p\|_2^2 + 0 + 4 \cdot \frac{\sqrt{3}}{2} \left( \binom{r}{2} \|p\|_2^2 \right)^{3/2}$$
Approximation Analysis

Analysis

For any value \( r \geq 30 \cdot \frac{\sqrt{n}}{\epsilon^2} \), the algorithm returns a value \( Y \) such that

\[
P \left[ \left| Y - \| p \|_2^2 \right| \geq \epsilon \cdot \| p \|_2^2 \right] \leq 1/3.
\]

Proof (4/5):

- We have just shown that:

\[
\text{Var} \left[ \sum_{1 \leq i < j \leq r} \sigma_{i,j} \right] = A + B + 4C
\]

\[
= \binom{r}{2} \cdot \| p \|_2^2 + 0 + 4 \cdot \frac{\sqrt{3}}{2} \left( \binom{r}{2} \| p \|_2^2 \right)^{3/2}
\]

\[
\leq 5 \left( \binom{r}{2} \| p \|_2^2 \right)^{3/2}
\]
Approximation Analysis

Analysis

For any value \( r \geq 30 \cdot \frac{\sqrt{n}}{\epsilon^2} \), the algorithm returns a value \( Y \) such that

\[
P\left[ \left| Y - \|p\|_2^2 \right| \geq \epsilon \cdot \|p\|_2^2 \right] \leq \frac{1}{3}.
\]

Proof (5/5):
Approximation Analysis

For any value $r \geq 30 \cdot \frac{\sqrt{n}}{\epsilon^2}$, the algorithm returns a value $Y$ such that

$$P\left[ \left| Y - \|p\|_2^2 \right| \geq \epsilon \cdot \|p\|_2^2 \right] \leq 1/3.$$

Proof (5/5):

- Applying Chebyshev’s inequality to $Y := \frac{1}{\binom{r}{2}} \cdot \sum_{1 \leq i < j \leq r} \sigma_{i,j}$ yields:
Approximation Analysis

Analysis

For any value \( r \geq 30 \cdot \frac{\sqrt{n}}{\epsilon^2} \), the algorithm returns a value \( Y \) such that

\[
P\left[ \left| Y - \| p \|_2^2 \right| \geq \epsilon \cdot \| p \|_2^2 \right] \leq 1/3.
\]

Proof (5/5):

- Applying Chebyshev’s inequality to \( Y := \frac{1}{r^2} \cdot \sum_{1 \leq i < j \leq r} \sigma_{i,j} \) yields:

\[
P\left[ | Y - E[Y] | \geq \epsilon \cdot \| p \|_2^2 \right] \leq \frac{\text{Var}[Y]}{\epsilon^2 \cdot \| p \|_2^4}
\]
Approximation Analysis

For any value $r \geq 30 \cdot \frac{\sqrt{n}}{\epsilon^2}$, the algorithm returns a value $Y$ such that

$$\mathbb{P}\left[ \left| Y - \|p\|_2^2 \right| \geq \epsilon \cdot \|p\|_2^2 \right] \leq \frac{1}{3}.$$

Proof (5/5):

- Applying Chebyshev’s inequality to $Y := \frac{1}{(r)^2} \cdot \sum_{1 \leq i < j \leq r} \sigma_{i,j}$ yields:

  $$\mathbb{P}\left[ \left| Y - \mathbb{E}[Y] \right| \geq \epsilon \cdot \|p\|_2^2 \right] \leq \frac{\text{Var}[Y]}{\epsilon^2 \cdot \|p\|_2^4} \leq \frac{1}{(r)^2} \cdot 5 \left( \binom{r}{2} \cdot \|p\|_2^2 \right)^{3/2} \leq \frac{1}{(r)^2} \cdot \frac{5 \left( \binom{r}{2} \cdot \|p\|_2^2 \right)^{3/2}}{\epsilon^2 \cdot \|p\|_2^4}.$$
Approximation Analysis

For any value $r \geq 30 \cdot \frac{\sqrt{n}}{\epsilon^2}$, the algorithm returns a value $Y$ such that

$$
P\left[ \left| Y - \|p\|_2^2 \right| \geq \epsilon \cdot \|p\|_2^2 \right] \leq \frac{1}{3}.
$$

Proof (5/5):

- Applying Chebyshev’s inequality to $Y := \frac{1}{\binom{r}{2}} \cdot \sum_{1 \leq i < j \leq r} \sigma_{i,j}$ yields:

$$
P\left[ \left| Y - E[Y] \right| \geq \epsilon \cdot \|p\|_2^2 \right] \leq \frac{\text{Var}[Y]}{\epsilon^2 \cdot \|p\|_2^4}
$$

$$
\leq \frac{\frac{1}{\binom{r}{2}} \cdot 5 \left( \binom{r}{2} \cdot \|p\|_2^2 \right)^{3/2}}{\epsilon^2 \cdot \|p\|_2^4}
$$

$$
\leq \frac{10}{r \cdot \|p\|_2 \cdot \epsilon^2}
$$
Approximation Analysis

For any value $r \geq 30 \cdot \frac{\sqrt{n}}{\epsilon^2}$, the algorithm returns a value $Y$ such that

$$P\left[|Y - \|p\|_2^2| \geq \epsilon \cdot \|p\|_2^2\right] \leq 1/3.$$

Proof (5/5):

- Applying Chebyshev’s inequality to $Y := \frac{1}{(r)} \cdot \sum_{1 \leq i < j \leq r} \sigma_{i,j}$ yields:

  $$P\left[|Y - E[Y]| \geq \epsilon \cdot \|p\|_2^2\right] \leq \frac{\text{Var}[Y]}{\epsilon^2 \cdot \|p\|_2^4} \leq \frac{\frac{1}{(r)^2} \cdot 5 \left(\frac{r}{2}\right) \cdot \|p\|_2^3}{\epsilon^2 \cdot \|p\|_2^4} \leq \frac{10}{r \cdot \|p\|_2 \cdot \epsilon^2} \leq \frac{10}{r \cdot (1/\sqrt{n}) \cdot \epsilon^2} \square$$
UNIFORM-TEST

1. Run **APPROXIMATE** $\|p\|_2^2$ with $r = 30 \cdot \frac{\sqrt{n}}{(\epsilon/2)^2} = O\left(\frac{\sqrt{n}}{\epsilon^4}\right)$ samples to get a value $Y$ such that

$$P\left[|Y - E[Y]| \geq \frac{\epsilon^2}{4} \cdot \|p\|_2^2\right] \leq 1/3.$$

2. If $Y \geq \frac{1+\epsilon^2/2}{n}$, then REJECT.
3. Otherwise, ACCEPT.
Approximation of $\|p - u\|_1$ using $\|p\|_2^2$

**UNIFORM-TEST**

1. Run **APPROXIMATE** $\|p\|_2^2$ with $r = 30 \cdot \frac{\sqrt{n}}{(\epsilon^2/4)^2} = \mathcal{O}(\frac{\sqrt{n}}{\epsilon^4})$ samples to get a value $Y$ such that

   $$\Pr \left[ \left| Y - \mathbb{E}[Y] \right| \geq \frac{\epsilon^2}{4} \cdot \|p\|_2^2 \right] \leq \frac{1}{3}.$$

2. If $Y \geq \frac{1 + \epsilon^2/2}{n}$, then REJECT.
3. Otherwise, ACCEPT.

**Correctness Analysis**

- If $p = u$, then $\Pr[\text{ACCEPT}] \geq 2/3$.
- If $p$ is $\epsilon$-far from $u$, i.e., $\sum_{i=1}^n |p_i - \frac{1}{n}| \geq \epsilon$), then $\Pr[\text{REJECT}] \geq 2/3$. 

Approximation of $\|p - u\|_1$ using $\|p\|_2^2$

**UNIFORM-TEST**

1. Run **APPROXIMATE** $\|p\|_2^2$ with $r = 30 \cdot \frac{\sqrt{n}}{(\epsilon^2/4)^2} = O(\frac{\sqrt{n}}{\epsilon^4})$ samples to get a value $Y$ such that

$$P\left[|Y - E[Y]| \geq \frac{\epsilon^2}{4} \cdot \|p\|_2^2\right] \leq 1/3.$$

2. If $Y \geq \frac{1 + \epsilon^2/2}{n}$, then REJECT.
3. Otherwise, ACCEPT.

**Correctness Analysis**

- If $p = u$, then $P[\text{ACCEPT}] \geq 2/3$.
- If $p$ is $\epsilon$-far from $u$, i.e., $\sum_{i=1}^{n} |p_i - \frac{1}{n}| \geq \epsilon$, then $P[\text{REJECT}] \geq 2/3$.

**Exercise**: Prove that any testing algorithm in this model will have a two-sided error!
Case 1: $p$ is uniform.

In this case

$$\|p\|_2^2 = \frac{1}{n},$$
Case 1: $p$ is uniform.
In this case

$$\|p\|_2^2 = \frac{1}{n},$$

and the approximation guarantee on $Y$ implies

$$\mathbf{P}\left[Y \geq \|p\|_2^2 \cdot (1 + \epsilon^2 / 4)\right] \leq 1/3,$$
Case 1: $p$ is uniform.
In this case

$$
\|p\|_2^2 = \frac{1}{n},
$$

and the approximation guarantee on $Y$ implies

$$
P\left[ Y \geq \|p\|_2^2 \cdot (1 + \epsilon^2/4) \right] \leq 1/3,
$$

which means that the algorithm will ACCEPT with probability at least $2/3$. 
Case 2: $p$ is $\epsilon$-far from $u$.
We will show that if $P[\text{REJECT}] \leq 2/3$, then $p$ is $\epsilon$-close to $u$. 
Case 2: \( p \) is \( \epsilon \)-far from \( u \).

We will show that if \( P[\text{REJECT}] \leq 2/3 \), then \( p \) is \( \epsilon \)-close to \( u \).

\( P[\text{REJECT}] \leq 2/3 \) implies
Case 2: $p$ is $\epsilon$-far from $u$.

We will show that if $\Pr[\text{REJECT}] \leq 2/3$, then $p$ is $\epsilon$-close to $u$.

$\Pr[\text{REJECT}] \leq 2/3$ implies

$$
\Pr \left[ Y > \frac{1 + \epsilon^2/2}{n} \right] < 2/3.
$$

(1)
Case 2: $p$ is $\epsilon$-far from $u$. We will show that if $P[\text{REJECT}] \leq 2/3$, then $p$ is $\epsilon$-close to $u$. $P[\text{REJECT}] \leq 2/3$ implies

$$P\left[Y > \frac{1 + \epsilon^2/2}{n}\right] < 2/3. \quad (1)$$

From line 1 of the algorithm we know that

$$P\left[Y > (1 - \epsilon^2/4) \cdot \|p\|_2^2\right] \geq 2/3. \quad (2)$$
Case 2: $p$ is $\epsilon$-far from $u$.

We will show that if $\mathbb{P}[\text{REJECT}] \leq 2/3$, then $p$ is $\epsilon$-close to $u$.

$\mathbb{P}[\text{REJECT}] \leq 2/3$ implies

$$\mathbb{P}\left[Y > \frac{1 + \epsilon^2/2}{n}\right] < 2/3. \quad (1)$$

From line 1 of the algorithm we know that

$$\mathbb{P}\left[Y > (1 - \epsilon^2/4) \cdot \|p\|_2^2\right] \geq 2/3. \quad (2)$$

Combining (1) and (2) yields, and rearranging yields
Case 2: \( p \) is \( \epsilon \)-far from \( u \).

We will show that if \( P[\text{REJECT}] \leq 2/3 \), then \( p \) is \( \epsilon \)-close to \( u \).

\( P[\text{REJECT}] \leq 2/3 \) implies

\[
P \left[ Y > \frac{1 + \epsilon^2/2}{n} \right] < 2/3. \tag{1}
\]

From line 1 of the algorithm we know that

\[
P \left[ Y > (1 - \epsilon^2/4) \cdot \|p\|_2^2 \right] \geq 2/3. \tag{2}
\]

Combining (1) and (2) yields, and rearranging yields

\[
\|p\|_2^2 < \frac{1}{n} \cdot \left(1 + \frac{\epsilon^2}{2}\right) \cdot \frac{1}{1 - \frac{\epsilon^2}{4}}
\]
Case 2: $p$ is $\epsilon$-far from $u$. We will show that if $\Pr[\text{REJECT}] \leq 2/3$, then $p$ is $\epsilon$-close to $u$.

$\Pr[\text{REJECT}] \leq 2/3$ implies

$$
\Pr \left[ Y > \frac{1 + \epsilon^2/2}{n} \right] < 2/3.
$$

(1)

From line 1 of the algorithm we know that

$$
\Pr \left[ Y > (1 - \epsilon^2/4) \cdot \|p\|_2^2 \right] \geq 2/3.
$$

(2)

Combining (1) and (2) yields, and rearranging yields

$$
\|p\|_2^2 < \frac{1}{n} \cdot (1 + \epsilon^2/2) \cdot \frac{1}{1 - \epsilon^2/4}
$$

$$
1 \leq (1 + \epsilon^2/3) \cdot (1 - \epsilon^2/4)
$$
Case 2: $p$ is $\epsilon$-far from $u$.

We will show that if $\Pr[\text{REJECT}] \leq 2/3$, then $p$ is $\epsilon$-close to $u$.

$\Pr[\text{REJECT}] \leq 2/3$ implies

$$
\Pr \left[ Y > \frac{1 + \epsilon^2/2}{n} \right] < 2/3.
$$

(1)

From line 1 of the algorithm we know that

$$
\Pr \left[ Y > (1 - \epsilon^2/4) \cdot \|p\|_2^2 \right] \geq 2/3.
$$

(2)

Combining (1) and (2) yields, and rearranging yields

$$
\|p\|_2^2 < \frac{1}{n} \cdot (1 + \epsilon^2/2) \cdot \frac{1}{1 - \epsilon^2/4} \leq \frac{1 + \epsilon^2}{n}.
$$

$$
1 \leq (1 + \epsilon^2/3) \cdot (1 - \epsilon^2/4)
$$
Case 2: \( p \) is \( \epsilon \)-far from \( u \).

We will show that if \( P[\text{REJECT}] \leq 2/3 \), then \( p \) is \( \epsilon \)-close to \( u \).

\( P[\text{REJECT}] \leq 2/3 \) implies

\[
\Pr \left[ Y > \frac{1 + \epsilon^2/2}{n} \right] < 2/3. \quad (1)
\]

From line 1 of the algorithm we know that

\[
\Pr \left[ Y > \left(1 - \frac{\epsilon^2/4}{1}\right) \cdot \|p\|_2^2 \right] \geq 2/3. \quad (2)
\]

Combining (1) and (2) yields, and rearranging yields

\[
\|p\|_2^2 < \frac{1}{n} \cdot \left(1 + \frac{\epsilon^2/2}{1}\right) \cdot \frac{1}{\frac{1}{1 - \epsilon^2/4}} \leq \frac{1 + \epsilon^2}{n}.
\]

Hence,

\[
\|p - u\|_2^2 = \|p\|_2^2 - \frac{1}{n} < \frac{\epsilon^2}{n} \quad \Rightarrow \quad \|p - u\|_2 < \frac{\epsilon}{\sqrt{n}}.
\]
Case 2: $p$ is $\epsilon$-far from $u$.

We will show that if $P[\text{REJECT}] \leq 2/3$, then $p$ is $\epsilon$-close to $u$.

$P[\text{REJECT}] \leq 2/3$ implies

$$P\left[ Y > \frac{1 + \epsilon^2/2}{n} \right] < 2/3. \quad (1)$$

From line 1 of the algorithm we know that

$$P\left[ Y > (1 - \epsilon^2/4) \cdot \|p\|_2^2 \right] \geq 2/3. \quad (2)$$

Combining (1) and (2) yields, and rearranging yields

$$\|p\|_2^2 < \frac{1}{n} \cdot (1 + \epsilon^2/2) \cdot \frac{1}{1 - \epsilon^2/4} \leq \frac{1 + \epsilon^2}{n}. \quad (3)$$

Hence,

$$\|p - u\|_2^2 = \|p\|_2^2 - \frac{1}{n} < \frac{\epsilon^2}{n} \quad \Rightarrow \quad \|p - u\|_2 < \frac{\epsilon}{\sqrt{n}}.$$
Analysis of UNIFORM-TEST (2/2)

Case 2: $p$ is $\epsilon$-far from $u$.
We will show that if $P[\text{REJECT}] \leq 2/3$, then $p$ is $\epsilon$-close to $u$.

$P[\text{REJECT}] \leq 2/3$ implies

$$P \left[ Y > \frac{1 + \epsilon^2 / 2}{n} \right] < 2/3. \quad (1)$$

From line 1 of the algorithm we know that

$$P \left[ Y > (1 - \epsilon^2 /4) \cdot \|p\|_2^2 \right] \geq 2/3. \quad (2)$$

Combining (1) and (2) yields, and rearranging yields

$$\|p\|_2^2 < \frac{1}{n} \cdot (1 + \epsilon^2 / 2) \cdot \frac{1}{1 - \epsilon^2 /4} \leq \frac{1 + \epsilon^2}{n}.$$ 

Hence,

$$\|p - u\|_2^2 = \|p\|_2^2 - \frac{1}{n} < \frac{\epsilon^2}{n} \quad \Rightarrow \quad \|p - u\|_2 < \frac{\epsilon}{\sqrt{n}}.$$ 

Since $\|\cdot\|_2 \geq \frac{1}{\sqrt{n}} \cdot \|\cdot\|_1$,

$$\|p - u\|_1 \leq \sqrt{n} \cdot \|p - u\|_2 < \epsilon.$$ 

\[ \square \]
Outline

Introduction

Upper Bounds on Testing Uniformity

Lower Bounds on Testing Uniformity

Extensions
Lower Bound

Theorem
Let $0 < \epsilon < 1$. There is no algorithm with the following three properties:
1. The algorithm samples at most $r := \frac{1}{64} \sqrt{n/\epsilon}$ times from $p$,
2. If $p = u$, then $\mathbb{P} [ \text{ACCEPT} ] \geq \frac{2}{3}$,
3. If $\|p - u\|_1 \geq \epsilon$, then $\mathbb{P} [ \text{REJECT} ] \geq \frac{2}{3}$.

Exercise:
Can you see why is it important to choose $I$ randomly?
Idea is that algorithm needs enough samples of the first $\epsilon \cdot n$ elements to see any collisions!
Lower Bound

**Theorem**

Let $0 < \epsilon < 1$. There is no algorithm with the following three properties:

1. The algorithm samples at most $r := \frac{1}{64} \sqrt{n/\epsilon}$ times from $p$,
2. If $p = u$, then $P[\text{ACCEPT}] \geq \frac{2}{3}$,
3. If $\|p - u\|_1 \geq \epsilon$, then $P[\text{REJECT}] \geq \frac{2}{3}$.

**Proof Outline.**

Exercise:
Can you see why is it important to choose $I$ randomly?

Idea is that algorithm needs enough samples of the first $\epsilon \cdot n$ elements to see any collisions!
Lower Bound

**Theorem**

Let $0 < \epsilon < 1$. There is no algorithm with the following three properties:

1. The algorithm samples at most $r := \frac{1}{64} \sqrt{n/\epsilon}$ times from $p$,

2. If $p = u$, then $\mathbb{P}[\text{ACCEPT}] \geq \frac{2}{3}$,

3. If $\|p - u\|_1 \geq \epsilon$, then $\mathbb{P}[\text{REJECT}] \geq \frac{2}{3}$.

**Proof Outline.**

- Generate a distribution $p$ randomly as follows:
Lower Bound

**Theorem**

Let $0 < \epsilon < 1$. There is no algorithm with the following three properties:

1. The algorithm samples at most $r := \frac{1}{64} \sqrt{n/\epsilon}$ times from $p$,
2. If $p = u$, then $\mathbb{P}[\text{ACCEPT}] \geq \frac{2}{3}$,
3. If $\|p - u\|_1 \geq \epsilon$, then $\mathbb{P}[\text{REJECT}] \geq \frac{2}{3}$.

**Proof Outline.**

- Generate a distribution $p$ randomly as follows:
  - Pick a set $\mathcal{I} \subseteq \{1, \ldots, \epsilon \cdot n\}$ of size $\epsilon \cdot n/2$ uniformly at random.
**Theorem**

Let \( 0 < \epsilon < 1 \). There is no algorithm with the following three properties:

1. The algorithm samples at most \( r := \frac{1}{64} \sqrt{n/\epsilon} \) times from \( p \),
2. If \( p = u \), then \( \mathbf{P}[\text{ACCEPT}] \geq \frac{2}{3} \),
3. If \( \|p - u\|_1 \geq \epsilon \), then \( \mathbf{P}[\text{REJECT}] \geq \frac{2}{3} \).

**Proof Outline.**

- Generate a distribution \( p \) randomly as follows:
  - Pick a set \( \mathcal{I} \subseteq \{1, \ldots, \epsilon \cdot n\} \) of size \( \epsilon \cdot n/2 \) uniformly at random.
  - Then define:

\[
p_i = \begin{cases} 
\frac{2}{n} & \text{if } i \in \mathcal{I}, \\
0 & \text{if } i \in \{1, \ldots, \epsilon \cdot n\} \setminus \mathcal{I}, \\
\frac{1}{n} & \text{if } \epsilon \cdot n < i < n.
\end{cases}
\]

- Idea is that algorithm needs enough samples of the first \( \epsilon \cdot n/2 \) elements to see any collisions!
Lower Bound

Let $0 < \epsilon < 1$. There is no algorithm with the following three properties:

1. The algorithm samples at most $r := \frac{1}{64} \sqrt{n/\epsilon}$ times from $p$,
2. If $p = u$, then $\mathbb{P}[\text{ACCEPT}] \geq \frac{2}{3}$,
3. If $\|p - u\|_1 \geq \epsilon$, then $\mathbb{P}[\text{REJECT}] \geq \frac{2}{3}$.

Proof Outline.
- Generate a distribution $p$ randomly as follows:
  - Pick a set $\mathcal{I} \subseteq \{1, \ldots, \epsilon \cdot n\}$ of size $\epsilon \cdot n/2$ uniformly at random.
  - Then define:
    $$p_i = \begin{cases} \frac{2}{n} & \text{if } i \in \mathcal{I}, \\ 0 & \text{if } i \in \{1, \ldots, \epsilon \cdot n\} \setminus \mathcal{I}, \\ \frac{1}{n} & \text{if } \epsilon \cdot n < i < n. \end{cases}$$
  - Then $\|p - u\|_1$
Lower Bound

**Theorem**

Let $0 < \epsilon < 1$. There is no algorithm with the following three properties:

1. The algorithm samples at most $r := \frac{1}{64} \sqrt{n/\epsilon}$ times from $p$,
2. If $p = u$, then $\mathbb{P}[\text{ACCEPT}] \geq \frac{2}{3}$,
3. If $\|p - u\|_1 \geq \epsilon$, then $\mathbb{P}[\text{REJECT}] \geq \frac{2}{3}$.

**Proof Outline.**

- Generate a distribution $p$ randomly as follows:
  - Pick a set $I \subseteq \{1, \ldots, \epsilon \cdot n\}$ of size $\epsilon \cdot n/2$ uniformly at random.
  - Then define:
    $$p_i = \begin{cases} 
      \frac{2}{n} & \text{if } i \in I, \\
      0 & \text{if } i \in \{1, \ldots, \epsilon \cdot n\} \setminus I, \\
      \frac{1}{n} & \text{if } \epsilon \cdot n < i < n.
    \end{cases}$$
  - Then $\|p - u\|_1 = \epsilon \cdot n \cdot 1/n = \epsilon$.

E.g., $n = 16$, $\epsilon = 1/4$, $I = \{1, 4\}$:

```latex
\begin{bmatrix}
2n, & 0, & 0, & 2n \\
\frac{1}{n}, & \frac{1}{n}, & \frac{1}{n}, & \frac{1}{n}, & \frac{1}{n}, & \frac{1}{n}, & \frac{1}{n}, & \frac{1}{n}
\end{bmatrix}
```

**Exercise:** Can you see why is it important to choose $I$ randomly? Idea is that algorithm needs enough samples of the first $\epsilon \cdot n$ elements to see any collisions!
Lower Bound

**Theorem**

Let $0 < \epsilon < 1$. There is no algorithm with the following three properties:

1. The algorithm samples at most $r := \frac{1}{64} \sqrt{\frac{n}{\epsilon}}$ times from $p$.
2. If $p = u$, then $P[\text{ACCEPT}] \geq \frac{2}{3}$,
3. If $\|p - u\|_1 \geq \epsilon$, then $P[\text{REJECT}] \geq \frac{2}{3}$.

**Proof Outline.**

- Generate a distribution $p$ randomly as follows:
  - Pick a set $\mathcal{I} \subseteq \{1, \ldots, \epsilon \cdot n\}$ of size $\epsilon \cdot n/2$ uniformly at random.
  - Then define:
    $$ p_i = \begin{cases} 
    \frac{2}{n} & \text{if } i \in \mathcal{I}, \\
    0 & \text{if } i \in \{1, \ldots, \epsilon \cdot n\} \setminus \mathcal{I}, \\
    \frac{1}{n} & \text{if } \epsilon \cdot n < i < n.
    \end{cases} $$
  - Then $\|p - u\|_1 = \epsilon \cdot n \cdot \frac{1}{n} = \epsilon$.
- E.g., $n = 16$, $\epsilon = 1/4$, $\mathcal{I} = \{1, 4\}$:
Lower Bound

Theorem

Let $0 < \epsilon < 1$. There is no algorithm with the following three properties:

1. The algorithm samples at most $r := \frac{1}{64} \sqrt{n/\epsilon}$ times from $p$,
2. If $p = u$, then $\mathbb{P}[\text{ACCEPT}] \geq \frac{2}{3}$,
3. If $\|p - u\|_1 \geq \epsilon$, then $\mathbb{P}[\text{REJECT}] \geq \frac{2}{3}$.

Proof Outline.

- Generate a distribution $p$ randomly as follows:
  - Pick a set $\mathcal{I} \subseteq \{1, \ldots, \epsilon \cdot n\}$ of size $\epsilon \cdot n/2$ uniformly at random.
  - Then define:
    \[
    p_i = \begin{cases} 
    \frac{2}{n} & \text{if } i \in \mathcal{I}, \\
    0 & \text{if } i \in \{1, \ldots, \epsilon \cdot n\} \setminus \mathcal{I}, \\
    \frac{1}{n} & \text{if } \epsilon \cdot n < i < n.
    \end{cases}
    \]
  - Then $\|p - u\|_1 = \epsilon \cdot n \cdot 1/n = \epsilon$.
- E.g., $n = 16$, $\epsilon = 1/4$, $\mathcal{I} = \{1, 4\}$:

\[
\begin{pmatrix}
\frac{2}{n}, & 0, & 0, & \frac{2}{n}, & \frac{1}{n}, & \frac{1}{n}, & \frac{1}{n}, & \frac{1}{n}, & \frac{1}{n}, & \frac{1}{n}, & \frac{1}{n}, & \frac{1}{n}, & \frac{1}{n}, & \frac{1}{n}, & \frac{1}{n}, & \frac{1}{n}
\end{pmatrix}
\]

$\epsilon n = 4$ elements \hspace{1cm} 12 elements
Lower Bound

Theorem

Let $0 < \epsilon < 1$. There is no algorithm with the following three properties:

1. The algorithm samples at most $r := \frac{1}{64} \sqrt{n/\epsilon}$ times from $p$,
2. If $p = u$, then $P[\text{ACCEPT}] \geq \frac{2}{3}$,
3. If $\|p - u\|_1 \geq \epsilon$, then $P[\text{REJECT}] \geq \frac{2}{3}$.

Proof Outline.

- Generate a distribution $p$ randomly as follows:
  - Pick a set $I \subseteq \{1, \ldots, \epsilon \cdot n\}$ of size $\epsilon \cdot n/2$ uniformly at random.
  - Then define:
    \[
    p_i = \begin{cases} 
    \frac{2}{n} & \text{if } i \in I, \\
    0 & \text{if } i \in \{1, \ldots, \epsilon \cdot n\} \setminus I, \\
    \frac{1}{n} & \text{if } \epsilon \cdot n < i < n.
    \end{cases}
    \]
  - Then $\|p - u\|_1 = \epsilon \cdot n \cdot 1/n = \epsilon$.
- E.g., $n = 16$, $\epsilon = 1/4$, $I = \{1, 4\}$:
  \[
  p = \left(\begin{array}{cccccccccccccccc} 
  \frac{2}{n} & 0 & 0 & \frac{2}{n} & \frac{1}{n} & \frac{1}{n} & \frac{1}{n} & \frac{1}{n} & \frac{1}{n} & \frac{1}{n} & \frac{1}{n} & \frac{1}{n} & \frac{1}{n} & \frac{1}{n} & \frac{1}{n} & \frac{1}{n} \\
  \frac{\epsilon n = 4 \text{ elements}}{} & \frac{12 \text{ elements}}{}
  \end{array}\right)
  \]

Idea is that algorithm needs enough samples of the first $\epsilon \cdot n$ elements to see any collisions!
Lower Bound

Theorem

Let $0 < \epsilon < 1$. There is no algorithm with the following three properties:

1. The algorithm samples at most $r := \frac{1}{64} \sqrt{n/\epsilon}$ times from $p$,
2. If $p = u$, then $\Pr[\text{ACCEPT}] \geq \frac{2}{3}$,
3. If $\|p - u\|_1 \geq \epsilon$, then $\Pr[\text{REJECT}] \geq \frac{2}{3}$.

Proof Outline.

- Generate a distribution $p$ randomly as follows:
  - Pick a set $\mathcal{I} \subseteq \{1, \ldots, \epsilon \cdot n\}$ of size $\epsilon \cdot n/2$ uniformly at random.
  - Then define:
    $$p_i = \begin{cases} \frac{2}{n} & \text{if } i \in \mathcal{I}, \\ 0 & \text{if } i \in \{1, \ldots, \epsilon \cdot n\} \setminus \mathcal{I}, \\ \frac{1}{n} & \text{if } \epsilon \cdot n < i < n. \end{cases}$$
  - Then $\|p - u\|_1 = \epsilon \cdot n \cdot 1/n = \epsilon$.

- E.g., $n = 16$, $\epsilon = 1/4$, $\mathcal{I} = \{1, 4\}$:
  $$p = \begin{pmatrix} \frac{2}{n}, 0, 0, \frac{2}{n}, 1, 1, 1, \frac{1}{n}, \frac{1}{n}, \frac{1}{n}, \frac{1}{n}, \frac{1}{n}, \frac{1}{n}, \frac{1}{n}, \frac{1}{n}, \frac{1}{n}, \frac{1}{n}, \frac{1}{n} \end{pmatrix}$$

  - $\epsilon n = 4$ elements
  - 12 elements

Exercise: Can you see why is it important to choose $\mathcal{I}$ randomly?

Idea is that algorithm needs enough samples of the first $\epsilon \cdot n$ elements to see any collisions!
Outline

Introduction

Upper Bounds on Testing Uniformity

Lower Bounds on Testing Uniformity

Extensions
\[ \|p - q\|_2^2 = \sum_{i=1}^{n} (p_i - q_i)^2 = \sum_{i=1}^{n} p_i^2 + \sum_{i=1}^{n} q_i^2 - 2 \cdot \sum_{i=1}^{n} p_i \cdot q_i = \|p\|_2^2 + \|q\|_2^2 - 2 \cdot \langle p, q \rangle \]

We already know how to estimate \( \|p\|_2^2 \) and \( \|q\|_2^2 \)!

1. Sample \( r \) elements from \( p \), \( x_1, x_2, \ldots, x_r \in [n] \), and sample \( r \) elements from \( q \), \( y_1, y_2, \ldots, y_r \in [n] \).

2. For each \( 1 \leq i < j \leq r \), \( \tau_{i,j} := \begin{cases} 1 & \text{if } x_i = y_j, \\ 0 & \text{otherwise}. \end{cases} \)

3. Output \( Y := \frac{1}{r^2} \sum_{1 \leq i, j \leq r} \tau_{i,j} \).

\( \text{APPROXIMATE } \langle p, q \rangle \)
\[ \| p - q \|_2^2 = \sum_{i=1}^{n} (p_i - q_i)^2 \]
\[ \|p - q\|_2^2 = \sum_{i=1}^{n} (p_i - q_i)^2 = \sum_{i=1}^{n} p_i^2 + \sum_{i=1}^{n} q_i^2 - 2 \cdot \sum_{i=1}^{n} p_i \cdot q_i \]
∥p − q∥^2 = \sum_{i=1}^{n} (p_i - q_i)^2 = \sum_{i=1}^{n} p_i^2 + \sum_{i=1}^{n} q_i^2 - 2 \cdot \sum_{i=1}^{n} p_i \cdot q_i

= ∥p∥^2 + ∥q∥^2 - 2 \cdot ⟨p, q⟩
\[ \|p - q\|_2^2 = \sum_{i=1}^{n} (p_i - q_i)^2 = \sum_{i=1}^{n} p_i^2 + \sum_{i=1}^{n} q_i^2 - 2 \cdot \sum_{i=1}^{n} p_i \cdot q_i \]

\[ = \|p\|_2^2 + \|q\|_2^2 - 2 \cdot \langle p, q \rangle \]

We already know how to estimate \(\|p\|_2^2\) and \(\|q\|_2^2\)!
\[ \|p - q\|_2^2 = \sum_{i=1}^{n} (p_i - q_i)^2 = \sum_{i=1}^{n} p_i^2 + \sum_{i=1}^{n} q_i^2 - 2 \cdot \sum_{i=1}^{n} p_i \cdot q_i \]

\[ = \|p\|_2^2 + \|q\|_2^2 - 2 \cdot \langle p, q \rangle \]

We already know how to estimate \(\|p\|_2^2\) and \(\|q\|_2^2\)!

**APPROXIMATE \(\langle p, q \rangle\)**

1. Sample \(r\) elements from \(p\), \(x_1, x_2, \ldots, x_r \in [n]\), and sample \(r\) elements from \(q\), \(y_1, y_2, \ldots, y_r \in [n]\)

2. For each \(1 \leq i < j \leq r\),

\[ \tau_{i,j} := \begin{cases} 1 & \text{if } x_i = y_j, \\ 0 & \text{otherwise.} \end{cases} \]

3. Output \(Y := \frac{1}{r^2} \sum_{1 \leq i, j \leq r} \tau_{i,j}\).
There exists an algorithm using $O(1/\epsilon^4)$ samples such that if the distributions $p$ and $q$ satisfy $\|p - q\|_2 \leq \epsilon/2$, then the algorithm accepts with probability at least 2/3. If $\|p - q\|_2 \geq \epsilon$, then the algorithm rejects with probability at least 2/3.
There exists an algorithm using $O(1/\epsilon^4)$ samples such that if the distributions $p$ and $q$ satisfy $\|p - q\|_2 \leq \epsilon/2$, then the algorithm accepts with probability at least $2/3$. If $\|p - q\|_2 \geq \epsilon$, then the algorithm rejects with probability at least $2/3$.

There exists an algorithm using $O(1/\epsilon^4 \cdot n^{2/3} \log n)$ samples such that if the distributions $p$ and $q$ satisfy $\|p - q\|_1 \leq \max\{\frac{\epsilon^2}{32\sqrt{n}}, \frac{\epsilon}{4\sqrt{n}}\}$, then the algorithm accepts with probability at least $2/3$. If $\|p - q\|_1 \geq \epsilon$, then the algorithm rejects with probability at least $2/3$. 
Extension 1: Testing Closeness of Arbitrary Distributions (2/2)

There exists an algorithm using $O(1/\epsilon^4)$ samples such that if the distributions $p$ and $q$ satisfy $\|p - q\|_2 \leq \epsilon/2$, then the algorithm accepts with probability at least 2/3. If $\|p - q\|_2 \geq \epsilon$, then the algorithm rejects with probability at least 2/3.

Theorem (Batu, Fortnow, Rubinfeld, Smith, White; JACM 60(1), 2013)

There exists an algorithm using $O(1/\epsilon^4 \cdot n^{2/3} \log n)$ samples such that if the distributions $p$ and $q$ satisfy $\|p - q\|_1 \leq \max\{\frac{\epsilon^2}{32 \sqrt{n}}, \frac{\epsilon}{4 \sqrt{n}}\}$, then the algorithm accepts with probability at least 2/3. If $\|p - q\|_1 \geq \epsilon$, then the algorithm rejects with probability at least 2/3.

<table>
<thead>
<tr>
<th></th>
<th>$L_2$-distance</th>
<th>$L_1$-distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Testing uniformity $|p - u|$</td>
<td>$\Theta(1)$</td>
<td>$\Theta(\sqrt{n})$</td>
</tr>
<tr>
<td>Testing closeness $|p - q|$</td>
<td>$\Theta(1)$</td>
<td>$\in [\Omega(n^{2/3}), O(n^{2/3} \log n)]$</td>
</tr>
</tbody>
</table>

Figure: Overview of the known sampling complexities for constant $\epsilon \in (0, 1)$.
Testing Conductance of Graphs

- **Idea:** Start several random walks from the same vertex
Extension 2: Testing Conductance of Graphs

- **Idea:** Start several random walks from the same vertex
  - Count the number of pairwise collisions
Extension 2: Testing Conductance of Graphs

Testing Conductance of Graphs

- **Idea:** Start several random walks from the same vertex
- **Count the number of pairwise collisions**
  - If the number of collisions high, graphs is not an expander
  - If the number of collisions is sufficiently small, graph is an expander