1 Overview

In this lecture we will introduce 3 different but related areas of computer science.

1. Streaming Algorithms: There is a lot of data coming in, but there is a constraint on the amount of storage i.e. $o(n)$ space.
2. Compressed Sensing: We are allowed to make $o(n)$ observations on the data and compute functions of the data.
3. Property Testing: Testing properties of objects e.g. graphs with randomized algorithms that run in very less time and succeed with high probability.

1.1 Property Testing

Let $G$ be a graph. Some properties you might want to test for are:

1. Is $G$ bipartite?
2. Is $G$ connected?

Similarly, for a distribution $D$ we can ask if $D$ is uniform.

It turns out that for exactly testing of a property is a hard problem in some cases. Thus we use a relaxed definition of property testing. We will be interested in the following task:

Distinguish

1. $X$ has property $P$: Accept with high probability.
2. $X$ is a $\epsilon$-far from having $P$: Reject with high probability.

Thus for testing a graph property, we can test for:

1. $G$ has property $P$.
2. Need to change at least $\epsilon n$ vertices of $G$ to have $P$.

It turns out that testing for testing if a graph $G$ is bipartite (using the above definition), there is a known algorithm that takes $\text{poly}(\frac{1}{\epsilon})$ samples and $\text{poly}(\frac{1}{\epsilon})$ time.

For testing if $G$ is connected, there is an algorithm that take $\text{poly}(\frac{1}{\epsilon})$ samples.
2 Testing if a distribution is uniform

We now present and analyze an algorithm for testing if a distribution is uniform. We note that the naive way would require $O(n)$ samples.

**Distribution** Consider a distribution over $\{1, 2, \cdots, n\}$ with pdf $P$. We need to distinguish between the following possibilities.

1. $\forall i, p_i = \frac{1}{n}$ (then $P = U_n$).
2. $\sum_{i=1}^{n} |p_i - \frac{1}{n}| \geq \epsilon$.

Let $x_1, x_2, \cdots, x_m$ be independent samples from $P$. Our algorithm works by counting the number of collisions in the samples. We define the random variable $A$ as:

$$A = \sum_{1 \leq i < j \leq m} 1(x_i = x_j)$$

Thus,

$$E[A] = \sum_{i=1}^{n} p_i^2 = \|P\|_2^2$$

We note the following simple claims.

**Claim 2.1.** $\|U_n\|_2^2 = \frac{1}{n}$.

**Claim 2.2.** For any pdf $P$, if $\|P - U_n\|_1 \geq \epsilon$, then $\|P\|_2^2 \geq \frac{1}{n} + \frac{\epsilon^2}{n}$.

Algorithm: Compute $A$:

1. output YES, if $A \leq \frac{1}{n} + \frac{\epsilon^2}{2n}$;
2. No, otherwise.

To prove correctness, we need to show that $Var[A]$ small.

**Claim 2.3.**

$$Var[A] < \frac{\epsilon^4}{8n^2} \quad if \quad m > \frac{\sqrt{n}}{\epsilon^2}$$

**Proof.** Define: $Z_{ij} = 1(x_i = x_j) - \|P\|_2^2$.

Thus, we have:
\[ \text{Var}[A] = E[ A - E[ A] ]^2 \\
= E \left[ \frac{\sum_{1 \leq i < j \leq m} 1( x_i = x_j)}{\binom{m}{2}} - \| P \|_2^2 \right]^2 \\
= E \left[ \frac{\sum_{1 \leq i < j \leq m} Z_{ij}}{\binom{m}{2}} \right]^2 \\
= \frac{1}{\binom{m}{2}} \left( \sum_{1 \leq i < j \leq m} E[Z_{ij}^2] + \sum_{1 \leq i < j \leq m, k < l, i, j \neq k, l} E[Z_{ij}Z_{kl}] + \sum_{1 \leq i < j \leq m, k \notin \{i, j\}} E[Z_{ij}Z_{jk}] \right) \\
\]

Consider the first term:
\[ E[Z_{ij}^2] \leq E[(Z_{ij} + \| P \|_2)^2] = \| P \|_2^2 \]

Consider the second term:
\[ E[Z_{ij}Z_{kl}] = 0 \]

Consider the third term:
\[ E[Z_{ij}Z_{jk}] = E[1(x_i = x_j = x_k)] - \| P \|_2^2 \cdot (1(x_i = x_j) + 1(x_j = x_k)) + \| P \|_4^2 \\
= \sum_{i=1}^n p_i^3 - \| P \|_2^2 \cdot 2p_i^2 + \| P \|_2^4 = \left( \sum_{i=1}^n p_i^3 \right) - \| P \|_2^4 \]

Now, we have that
\[ \sum_{i=1}^n p_i^3 \leq \sqrt{n} \left( \sum_{i} p_i^2 \right)^2 \]

because for each \( i \),

if \( p_i \leq 1/\sqrt{n} \), then \( p_i^3 \leq p_i^2 \frac{\sqrt{n}}{n} \leq \sqrt{n}p_i^2 \left( \sum_{i} p_i^2 \right) \)

if \( p_i \geq 1/\sqrt{n} \), then \( p_i^3 \leq \sqrt{n}p_i^2 \cdot p_i^2 \leq \sqrt{n}p_i^2 \left( \sum_{i} p_i^2 \right) \).
Therefore,
\[
\sum E[Z_{ij}Z_{jk}]
= \left( \frac{m}{3} \right) \cdot (\sqrt{n} \cdot (\sum p_i^2)^2 - \|P\|_4^4)
\leq \frac{m^3}{6} \sqrt{n} \|P\|_2^4
\]

Thus,
\[
Var[A] \leq \frac{4}{m^4} \left( \frac{m^2}{2} \|P\|_2^2 + \frac{m^3}{6} \sqrt{n} \|P\|_2^4 \right)
= \frac{2}{m^2} \|P\|_2^2 + \frac{2}{3m} \sqrt{n} \|P\|_2^4
< \frac{2}{n} \|P\|_2^2 + \frac{2}{3} \epsilon^4 \|P\|_2^4 \quad (\text{since } m > \frac{\sqrt{n}}{\epsilon^4})
\approx \frac{2}{3} \epsilon^4 \|P\|_2^4
\]

We can conclude that \(|A - E[A]| \leq \epsilon^2 \|P\|_2^2 \) with probability > 3/4.

3 Streaming Algorithm

1. orders coming by
2. connection pass through router
3. scanning disk

Ex. Distinct elements.
1,7,3,997,1,1,1,5,7, \cdots \in [U]. Estimate number of distinct values \( n \) to \( (1 \pm \epsilon) \) factor.

1. Hash table \( O(n) \) space
2. today \( O(\frac{1}{\epsilon^4} \log |U|) \) space
3. Next class: \( O(\frac{1}{\epsilon^2} \log \log |U|) \) space

A simpler problem: Is \( n > (1 + \epsilon)T \) or \( n < (1 - \epsilon)T \) in space \( S \).

We show how a solution for the above problem can be used to solve the general problem.

Algorithm: Choose random set \( S \subset [U] \), each \( i \in S \) with \( p = \frac{1}{T} \). Record any element if stream lies in \( S \).

We run the algorithm in parallel for the following values of \( T \): 1, \((1 + \epsilon),(1 + \epsilon)^2, \cdots, (1 + \epsilon)^X = U\).