1 Overview

In this lecture, we will overview the three areas of computer science which we will cover in this course.

1. Property Testing: There is a large amount of data which we can ‘query’ locally and we wish to test whether the data satisfies some property using $o(n)$ queries.

2. Streaming Algorithms: The data arrives in a stream and we wish to evaluate some statistic of the data while using $o(n)$ space.

3. Compressed Sensing: We wish to make a $o(n)$ ‘measurements’ of the data and then compute functions of the data.

2 Property Testing

2.1 Introduction

Some well known property testing problems are:

1. Bipartiteness Testing: Is the graph bipartite?
2. Connectedness Testing: Is the graph connected?
3. Uniformity Testing: Is the given distribution uniform?

One might argue that the problems stated above cannot be solved by making sublinear queries. For example, there may be a bipartite graph with just one edge within a bipartition.

So, we relax the problem slightly and phrase it as follows:

**Distinguish between**

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

The uniformity testing problem may be phrased as follows:

**Distinguish between**

1. $X$ is the uniform distribution on $[n]$: Accept with high probability
2. $X$ is $\epsilon$-far in TV distance from uniform on $[n]$: Reject with high probability.

In this case, the distance metric used is Total Variation distance: $\|X - \mathcal{U}_n\|_{TV} = \frac{1}{2} \sum_{i=0}^{n} |X_i - \frac{1}{n}|$

**Toy Problem: Distinguish between**

1. $X$ is the uniform distribution on $[n]$: Accept with high probability
2. $X$ is uniform over $S \subseteq [n], |S| = \frac{n}{2}$: Reject with high probability.

In order to distinguish between these two cases, we may sample until we see more than $n/2$ distinct elements.

If our distribution is from the first case, observe that if we have sampled less than $n/2$ elements, the probability of sampling a new element is more than $\frac{1}{2}$. So, we will need $\approx n$ draws until we see more than $n/2$ elements. So, we use $O(n)$ samples.

If our distribution lies in the second case, we never see more than $n/2$ elements and always reject.

In this course, we will deal with more sophisticated methods that will give us better results.

If we wish to accept/reject with constant probability, we know that $O\left(\sqrt{n}\frac{\epsilon}{\epsilon^2}\right)$ samples suffice for Uniformity Testing problem.

If we wish to accept/reject with $\delta$ probability, we knew that $O\left(\sqrt{n}\log\left(\frac{1}{\delta}\right)\right)$. A recent result improves this to $O\left(\frac{\sqrt{n\log\left(\frac{1}{\delta}\right)}}{\epsilon^2} + \frac{\log\left(\frac{1}{\delta}\right)}{\epsilon^2}\right)$

3 Streaming Algorithms

3.1 Introduction

We assume we have a “stream” of data passing by, that is too large to keep in main storage.

- Orders being processed.
- Connections through a router.
- Scanning off disk into RAM.
We can divide these problems into two cases.

**Single Pass** We only get to see the data once ever. (e.g. router connections)

**Multiple Pass** Accessing the data stream is expensive, but we can do it multiple times, so we can make a small (usually constant or logarithmic) number of passes over the stream.

Our goal is to compute some function of the data, while using sublinear space, often $O(n^\epsilon)$ or $O(\log^c n)^1$.

### 3.2 Distinct Elements

In the distinct elements problem, we receive a stream of elements from a “universe” $[U]$. The stream contains $n$ elements, with $k$ distinct elements.

The naïve algorithm takes $O(k \log U)$ bits, by storing every unique element seen so far. If we want to compute $k$ with total accuracy, even with only, say, 9/10 success probability, this many bits is necessary. (see problem set 1)

Instead, we will attempt to find an $(\epsilon, \delta)$ approximation—with probability $1 - \delta$, our estimate should be between $(1 - \epsilon)k$ and $(1 + \epsilon)k$. A series of algorithms have steadily improved the space bounds for this problem.

**Probabilistic Counting** $O(\frac{1}{\epsilon^2} \log U)$ space.

**LogLog** $O(\frac{1}{\epsilon^2} \log \log U)$ space.

**HyperLogLog** As above, but with better constants. Widely used in practice.

**Theoretical Result** $O(\frac{1}{\epsilon^2} + \log U)$. Not a practical algorithm.

### 4 Compressed Sensing

#### 4.1 Introduction

We want to estimate some vector $X \in \mathbb{R}^n$.

- Image
- Audio
- Geological Strata (seismic sensing)

But we can’t observe the data directly. Instead, we have some ability to take “measurements” of it.

---

1This is typically referred to as “polylog” space
In an MRI scan, we get samples from the Fourier transform of brain waves.

In a single-pixel camera, there is a single photosensitive element with a mask in front of it, as opposed to the many distinct photosensitive elements in a standard camera.

How can we recover $X$ from these measurements? Consider the related problem of compression. There, we use the structure of a signal to compress the signal into a smaller representation, then recover the signal from that representation. (approximately, in the case of lossy compression methods such as mp3s)

Perhaps with the right measurements we can recover $X$ without ever observing the whole thing?

### 4.2 General Formulation

We need to define what we mean by “measurements”. You may choose $A \in \mathbb{R}^{m \times n}$. You may then sense $AX \in \mathbb{R}^m$. (we can think of $AX$ as being $m$ measurements of $X$)

But what do we mean by “structure”? Here there are many options. One is sparsity.

**Exact Sparsity** $X$ has at most $k$ non-zero values.

**Approximate Sparsity** $\min ||X - X_k||$ is “small”, with the minimum taken over $k$-sparse vectors $X_k$.

So in this case, given $Y = AX$ and $A$, we will attempt to find an estimate $\hat{X}$ of $X$ with error bounded by the lack of sparsity in $X$, i.e.

$$||\hat{X} - X|| \leq C \min_{\text{k-sparse } X_k} ||X - X_k||$$

### 4.3 Example

If $A \sim \mathcal{N}(0, I_{m \times n})$, with $m \geq O(k \log \frac{n}{k})$ then, with high probability, $A$ will satisfy the restricted isometry property (the RIP)

If $A$ satisfies the RIP then

$$\forall X, \hat{X} := \arg\min_{||A\hat{X} - Y|| \leq \epsilon} ||\hat{X}||$$

satisfies $||\hat{X} - X||_2 \leq C||X - X_k||_2$ with high probability.