6.889 Sublinear Time Algorithms March 18, 2019

Lecture 12

Lecturer: Ronitt Rubinfeld *Scribe: Sandeep Silwal*

1 Distribution Testing Model

So far in the class we have mainly discussed graph property testing. In this setting, the models that we used were adjacency list queries for sparse graphs and adjacency matrix queries for dense graphs. We will now transition to a different model when discussing *distribution* property testing where we assume there is an oracle (or a stream) that gives us an independent copy of random variable from some distribution $\mathcal{P}$ in every query.

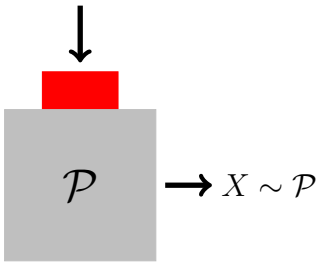

Figure 1: We can imagine querying the oracle as pushing a button on a black box which spits out independent copies of a random variable from some distribution $\mathcal{P}$

Throughout the next few lectures, we will assume that the domain of our probability distributions are discrete and over $[n] = \{1, \cdots, n\}$. We are interested in testing properties of our distributions such as: Is the distribution uniform, monotone, bimodal, $\cdots$ and does the distribution have high entropy, large support, $\cdots$. Similar to graph property testing, we measure the complexity of algorithms in this model in terms of the number of queries that we make to our oracle/stream. Our goal is to get this sample complexity to be sublinear in the size of the domain. In this lecture, we focus on testing uniformity. We expand upon this in the next section.

2 Uniform Testing

2.1 Setup

Given an oracle or steram to some unknown distribution $\mathcal{P}$, our goal is to design an algorithm that tests if $\mathcal{P}$ is close to $\mathcal{U}$, the uniform distribution over $[n]$. More specifically, we want an algorithm that does the following:

- If $\mathcal{P} = \mathcal{U}$, pass with probability at least $\frac{2}{3}$.
- If $dist(\mathcal{P}, \mathcal{U}) \geq \epsilon$, fail with probability at least $\frac{2}{3}$.

This raises two questions. First, why do we insist on our algorithm having two sided error and second, what is our definition of distance. To answer the first question, we note that our algorithm necessarily needs to have two sided error since there is some (small) probability that even if $\mathcal{P} = \mathcal{U}$, our oracle only spits out the same value in every query. For the second question, we will use two notions of distance: ℓ_1 and ℓ_2. For two distributions $\mathcal{P}$ and $\mathcal{Q}$, the ℓ_1 distance is defined as

$$\|\mathcal{P} - \mathcal{Q}\|_1 = \sum_{i \in [n]} |p(i) - q(i)|$$

where $p(i) = \mathbb{P}(\mathcal{P} = i)$ and similarly $q(i) = \mathbb{P}(\mathcal{Q} = i)$. The ℓ_2 distance is defined as

$$\|\mathcal{P} - \mathcal{Q}\|_2 = \left(\sum_{i \in [n]} (p(i) - q(i))^2 \right)^2 .$$

Note that we have the following relationship between these two distances

$$\|\mathcal{P} - \mathcal{Q}\|_2 \leq \|\mathcal{P} - \mathcal{Q}\|_1 \leq \sqrt{n} \|\mathcal{P} - \mathcal{Q}\|_2 \tag{1}$$

where the left inequality follows by expansion and the right inequality is just Cauchy Schwarz.

2.2 Understanding the Metrics

At first glance one might expect both ℓ_1 and ℓ_2 to behave similarly. However, it turns out that these two metrics are quite different. This is illustrated by the following example. Let $\mathcal{P}_1$ be the distribution that assigns all of its mass to $i = 1$ and let $\mathcal{Q}_1 = \mathcal{U}$. Then

$$\|\mathcal{P}_1 - \mathcal{Q}_1\|_1 = 1 \cdot \left(1 - \frac{1}{n}\right) + (n-1) \cdot \frac{1}{n} \approx 2$$
$$\|\mathcal{P}_1 - \mathcal{Q}_1\|_2 = \sqrt{1 \cdot \left(1 - \frac{1}{n}\right)^2 + (n-1) \cdot \frac{1}{n^2}} \approx 1.$$

In this case, both the ℓ_1 and ℓ_2 distances between $\mathcal{P}_1$ and $\mathcal{Q}_1$ are large. Now consider the example where $\mathcal{P}_2$ is uniform over the first $n/2$ elements of $[n]$ while $\mathcal{Q}_2$ is uniform over the last $n/2$ elements of $[n]$. We can compute that

$$\|\mathcal{P}_2 - \mathcal{Q}_2\|_1 = n \cdot \frac{2}{n} = 2$$
$$\|\mathcal{P}_2 - \mathcal{Q}_2\|_2 = \sqrt{n \cdot \frac{4}{n^2}} = \frac{2}{\sqrt{n}}.$$

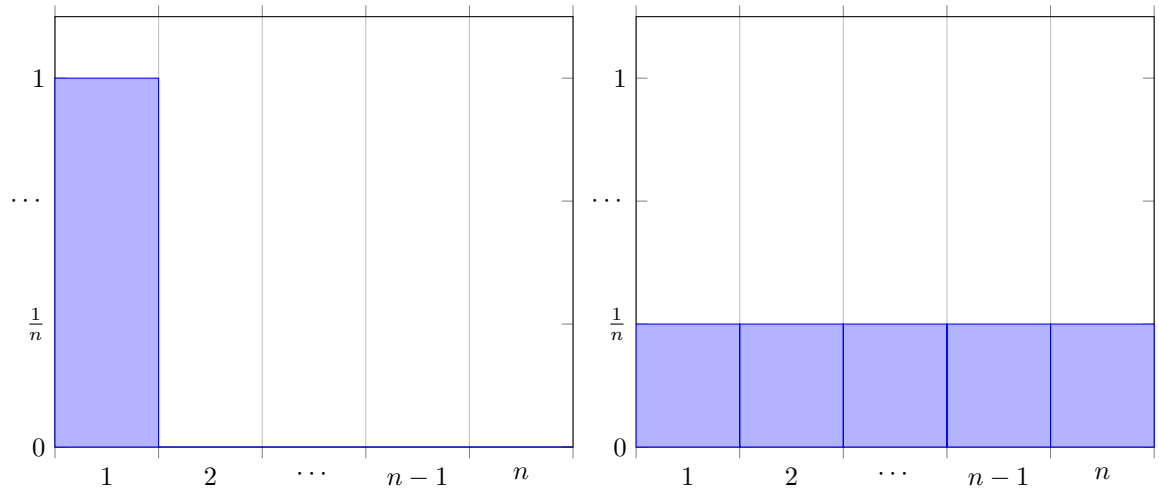

Figure 2: Right: Distribution $\mathcal{P}_1$. Left: Distribution $\mathcal{Q}_1$.

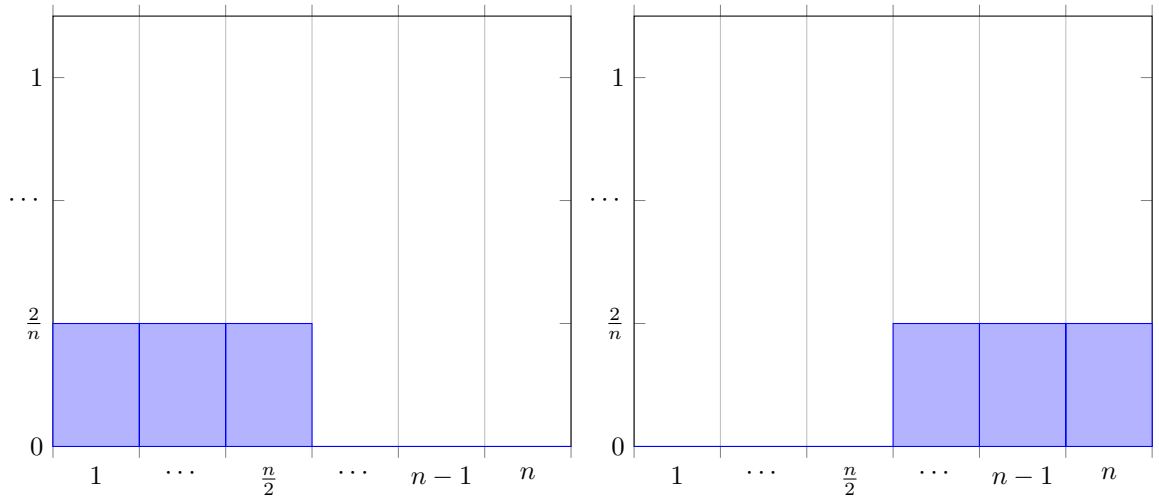

Figure 3: Right: Distribution $\mathcal{P}_2$. Left: Distribution $\mathcal{Q}_2$.

In this example. the ℓ_1 distance between $\mathcal{P}_2$ and $\mathcal{Q}_2$ is large while the ℓ_2 distance is very small even though the distributions are disjoint! This suggests that testing ℓ_2 distance is a lot easier than testing ℓ_1 distance since we have to distinguish fewer things! Our strategy then is to first get an efficient uniform tester for ℓ_2 distance and use (1) to get a (less)efficient tester for ℓ_1 distance that incurs an extra multiplicative factor of $O(\sqrt{n})$. This will be explored more in the later sections and the next lecture. We first begin with a naive algorithm to test ℓ_1 distance to $\mathcal{U}$, the uniform distribution.

3 Naive ℓ_1 Tester

The following naive algorithm is also called "Plug in estimate."

Algorithm 1: Plug-In Estimate

Input : Oracle access to $\mathcal{P}, \epsilon$
Output: Accept or Reject
1 Take m samples of $\mathcal{P}$ from the oracle
2 For all $i \in [n]$, estimate $p(i) = \mathbb{P}(\mathcal{P} = i)$ by $\widehat{p}(i) = \frac{\#\text{ of times } i \text{ is in sample}}{m}$
3 Reject if $\sum_{i\in[n]} |\widehat{p}(i) - 1/n| > \epsilon/2$ and accept otherwise

Intuitively, the **Plug-In Estimate** tries to get a good estimate of $p(i)$ for all i using m samples. Now suppose that we pick m such that

$$|\widehat{p}(i) - p(i)| \leq \frac{\epsilon}{2n} \tag{2}$$

for all i. Then if $\mathcal{P} = \mathcal{U}$, we clearly have $\|\mathcal{P} - \mathcal{U}\|_1 \leq \epsilon$ so **Plug-In Estimate** will accept $\mathcal{P}$. On the

other hand, if $\|\mathcal{P} - \mathcal{U}\|_1 \geq \epsilon$, then by the reverse triangle inequality,

$$\begin{aligned}\sum_{i\in[n]} \left|\widehat{p}(i) - \frac{1}{n}\right| &= \sum_{i\in[n]} \left|\widehat{p}(i) - p(i) + p(i) - \frac{1}{n}\right| \\ &\geq \|\mathcal{P} - \mathcal{U}\|_1 - \sum_{i\in[n]} |\widehat{p}(i) - p(i)| \\ &\geq \epsilon - \frac{\epsilon}{2} = \frac{\epsilon}{2}\end{aligned}$$

so **Plug-In Estimate** will reject $\mathcal{P}$. Now the question is how pick should we take m such that (2) holds. This ties into why **Plug-In Estimate** is a naive algorithm. Note that we want (2) to hold for *all* i which means that we are actually learning the *whole* distribution $\mathcal{P}$, a harder task than learning just the distance to $\mathcal{U}$. For (2) to hold, its reasonable to assume that we need $\Omega(n \log n)$ samples due to coupon collection. We now do a tight analysis to show that $m = O(n)$ suffices. Note that

$$\mathbb{E}\left[\sum_{i\in[n]} |\widehat{p}(i) - p(i)|\right] = \sum_{i\in[n]} \mathbb{E}[|\widehat{p}(i) - p(i)|] \leq \sum_{i\in[n]} \sqrt{\mathbb{E}(\widehat{p}(i) - p(i))^2}$$

Note that $\mathbb{E}\widehat{p}(i) = p(i)$ for all i since we can write $\widehat{p}(i)$ as the sum of m Bernoulli random variables with parameter $p(i)$. Hence, $\mathbb{E}(\widehat{p}(i) - p(i))^2 = \mathrm{Var}(\widehat{p}(i))$. Again using the fact that $\widehat{p}(i)$ is the sum of m independent Bernoulli random variables with parameter $p(i)$, we have that

$$\mathrm{Var}(\widehat{p}(i)) = \frac{1}{m} p(i)(1 - p(i)) \leq \frac{p(i)}{m}.$$

Therefore,

$$\mathbb{E}\left[\sum_{i\in[n]} |\widehat{p}(i) - p(i)|\right] \leq \frac{1}{\sqrt{m}} \sum_{i\in[n]} \sqrt{p(i)} \leq \sqrt{\frac{n}{m}}$$

where we have used Cauchy-Schwarz for the last inequality. Hence, if $m = O(n/\epsilon^2)$, we have

$$\mathbb{E}\left[\sum_{i\in[n]} |\widehat{p}(i) - p(i)|\right] \leq O(\epsilon)$$

and our analysis above goes through (with sufficiently large probability from Markov's inequality). Leaving ℓ_1 aside, we now move onto testing closeness to $\mathcal{U}$ using the ℓ_2 metric.

4 ℓ_2 Tester

We first reduce testing $\|\mathcal{P} - \mathcal{U}\|_2$ to a simpler property about $\mathcal{P}$. We compute that

$$\begin{aligned}\|\mathcal{P} - \mathcal{U}\|_2^2 &= \sum_{i\in[n]} \left(p(i) - \frac{1}{n}\right)^2 \\ &= \sum_{i\in[n]} p(i)^2 - \frac{2}{n} \sum_{i\in[n]} p(i) + \frac{1}{n^2} \sum_{i\in[n]} 1 \\ &= \left(\sum_{i\in[n]} p(i)^2\right) - \frac{1}{n}.\end{aligned}$$

Now note that $\sum_{i\in[n]} p(i)^2 = \|\mathcal{P}\|_2^2$ is the collision probability, that is, it is the probability that two elements drawn from $\mathcal{P}$ are the same. This tells us that testing the distance from $\mathcal{P}$ to $\mathcal{U}$ using ℓ_2 is equivalent to testing the collision probability of $\mathcal{P}$. (It also tells us that the uniform distribution has the smallest collision probability which intuitively makes sense.) This suggests the following algorithm for testing ℓ_2 distance to $\mathcal{U}$:

Algorithm 2: $\ell_2 -$ **Attempt 1**

Input : Oracle access to $\mathcal{P}, \epsilon$
Output: Accept or Reject
1 Take m samples of $\mathcal{P}$ from the oracle
2 Let $\widehat{c}$ be our estimate for $\|\mathcal{P}\|_2^2 = \sum_{i\in[n]} p(i)^2$
3 If $\widehat{c} < \frac{1}{n} + \delta$ pass, else fail

(We are purposely leaving out a lot of important details in the description of $\ell_2 -$ **Attempt 1** since we will propose another algorithm later.) If we take m samples and group them into $m/2$ groups of size 2, we get to see $O(m)$ possible collisions that are all independent. However, this does not quite work. To estimate the collision probability, we need to *see* a collision which can take upto $\Omega(n)$ samples. The better idea is to *recycle* the samples we have seen. This means that we can see $O(m^2)$ potential collisions from m samples at a cost of messier analysis since these collisions are not independent. This leads us to our final algorithm for testing ℓ_2 distance to $\mathcal{U}$.

Algorithm 3: $\ell_2 -$ **Recycling**

Input : Oracle access to $\mathcal{P}, \epsilon$
Output: Accept or Reject
1 $\delta \leftarrow \epsilon^2/2$
2 Take m samples of $\mathcal{P}$ from the oracle
3 Let σ_{ij} be indicator variable if sample i is equal to sample j
4 $\widehat{c} \leftarrow \sum_{i<j} \sigma_{ij} / \binom{m}{2}$
5 If $\widehat{c} < \frac{1}{n} + \delta$ pass, else fail

Note that

$$\mathbb{E}[\widehat{c}] = \binom{m}{2} \frac{1}{\binom{m}{2}} \mathbb{E}[\sigma_{ij}] = \mathbb{P}(\text{sample } i = \text{sample } j) = \sum_{k\in[n]} p(k)^2 = \|\mathcal{P}\|_2^2$$

so $\widehat{c}$ is an unbiased estiamtor for $\|\mathcal{P}\|_2^2$. Now suppose that

$$|\widehat{c} - \mathbb{E}[\widehat{c}]| = |\widehat{c} - \|\mathcal{P}\|_2^2| < \delta = \frac{\epsilon^2}{2}. \tag{3}$$

We will analyze $\ell_2 -$ **Recycling** under assumption (3) and show its correctness. In the next lecture, we will show that we need $m = \text{poly}(1/\epsilon)$ for 3 to hold which will complete the analysis. Now suppose that $\mathcal{P} = \mathcal{U}$. Then under assumption (3),

$$\widehat{c} < \|\mathcal{P}\|_2^2 + \delta = \frac{1}{n} + \frac{\epsilon^2}{2}$$

so $\ell_2 -$ **Recycling** will accpet $\mathcal{P}$. Now if $\|\mathcal{P} - \mathcal{U}\|_2 > \epsilon$, we have

$$\|\mathcal{P} - \mathcal{U}\|_2^2 = \|\mathcal{P}\|_2^2 - \frac{1}{n} = \mathbb{E}[\widehat{c}] - \frac{1}{n} > \epsilon^2.$$

Therefore, under assumption (3),

$$\widehat{c} > \frac{1}{n} + \epsilon^2 - \frac{\epsilon^2}{2} = \frac{1}{n} + \delta$$

so $\ell_2 -$ **Recycling** will reject $\mathcal{P}$. Now the challenge is to pick m in $\ell_2 -$ **Recycling** such that (3) holds. This will involve calculating the variance of $\widehat{c}$ which will be done in the next lecture.