CSE 291 Scribe Notes Lecture 23

Ruiqing Qiu

May 24, 2017

1 Previous Lecture

Last time, we covered the lower bound for closeness testing of product distribution. We have shown that the number of samples needed has to be at least $O(n^{\frac{3}{4}}/\epsilon)$ and $O(n^{\frac{1}{2}}/\epsilon^2)$.

2 Upper Bounds

2.1 Introduction

Today, we will show upper bound and show that this is tight. Let product distributions P, Q has means p, q, we want to distinguish between $P = Q$ and $d_{TV}(P, Q) > \epsilon$. We will describe the algorithm. But the first to notice is that things are very different when coordinate probability can be at the extreme, very close to 0 and 1. This messes up variational distance mean for us. And these needed to be dealt with separately.

2.2 Extreme Case

First case, the extreme case, to make this work a little nicer, the first thing we are going to do is make sure p_i and $q_i < 2/3$, the point is take log(n) samples and we flip some coordinates. If more than $2/3p_i$ are 1 in the i-th coordinate, then we just flip 0 and 1 in the i-th, if more than 2/3 of p are 1 and more than 2/3 of q are 0 then we know p and q aren't the same with high probability. to ensure 2/3 gives us the correct answer all the time, we need log(n) samples, which is much smaller than the sample complexity that we are dealing with.

Consider the case where p_i and q_i are small.

$\epsilon < d_{TV}(P, Q) \leq |p - q|_1$

In case of very very far to the extreme, this is not a bad approximation. However, Like most of our tester work, L1 is a bad norm since it's not polynomial in its coefficient, it can't be approximated very easily. This implies L2 norm between the means $|p - q|_2 > \epsilon^2/n$. And this is what we want to try to detect.

As before, we take independent Poisson m many samples for each coordinates.

$X_i \sim poi(p_i \cdot m), Y_i \sim poi(p_i \cdot m)$
Let

$$Z = \sum_i (x_i - y_i)^2 - x_i - y_i$$

The expectation of Z is $E[Z] = m^2|p-q|_2^2$
The variance of Z is

$$var(Z) = O(m^3 \cdot \sum_i (p_i - q_i)^2(p_i + q_i) + m^2 \cdot \sum_i (p_i + q_i)^2)$$

and we can rewrite this in terms of E[Z] as $O(m^2 \cdot \sqrt{E[Z] \cdot |p+q|_2} + m^2 \cdot |p+q|_2^2)$

We use the similar analysis as we did when we analyzed similar tester for probability distribution over discrete set of size n.

If $m \gg n \cdot |p+q|_2/\epsilon^2$, then we should be good.

2.3 Separate light bins vs. heavy bins

However, there's a problem where p and q are not normalized unlike in the discrete case. $|p+q|_2$ could be as large as $\sqrt{n}$.

We need to restrict ourselves running this tester on the light bin and then figured out something else to do on the heavy bins. How to make this distinction? We take bunch of samples and see which one gives us the various bins.

Take Poi(k) many samples from P and Q. Define the light bins as the ones with only 0's, i-th coordinate got a_i many 1's

Now we need to make a separation between light bins and heavy bins.

$$d_{TV}(P,Q) \ll \sum_{light} |p_i - q_i| + \sqrt{\sum_{heavy} \frac{(p_i - q_i)^2}{p_i + q_i}}$$

2.4 Light Bins

if $|p-q|_{1,light} \gg \epsilon$, then we can say number of samples we need to detect it is something like $m \gg n$. Now, what's this L2 norm of p + q? We could just say it's just the L2 norm but we can do a little bit better than that. Because, on average, how often do you ended up as a light bin? The probability of being a light bin starts dropping exponentially if $p + q > 1/k$, the expected squared L2 mass is not too large, which is $n\sqrt{n/k^2}/\epsilon^2$

We need to do a balancing act somewhere between m and k, take k samples and use m samples to make comparison. To balance the total number of sample, we want to make them equal.

If m = k, then $m \gg n^{\frac{3}{2}}/(m \cdot \epsilon^2)$ and simplified to get $m \gg n^{\frac{3}{4}}/\epsilon$. And it also needs to satisfy the condition from 2.2, $m \gg n \cdot |p+q|_2/\epsilon^2$.

2.5 Heavy Bins

Next, how do we deal with the heavy bins? What happens if

$$\sum_{i,heavy} \frac{(p_i - q_i)^2}{p_i + q_i} \gg \epsilon^2$$

The basic idea is that $a_i \approx k(p_i + q_i)$, which is $a_i = poi(k(p_i + q_i))$.
We need some statistics to give us some good approximation.
Let Z' =

$$\sum_{i,heavy} \frac{((x_i - y_i)^2 - x_i - y_i)}{(a_i/k)}$$

.

The expectation of Z' is

$$E[Z'] = \sum_{i,heavy} \frac{(p_i - q_i)^2}{a_i/k}$$

The point is

$$E[Z'] \gg \sum_{i,p_i+q_i>1/k} \frac{(p_i - q_i)^2}{p_i + q_i}$$

. As long as $p_i + q_i > 1/k$, this is a pretty good approximation.

$$var(Z') = O(m^3 \cdot \sum_i (p_i - q_i)^2(p_i + q_i)/(a_i/k)^2 + m^2 \cdot \sum_i (p_i + q_i)^2/(a_i/k)^2)$$

The first term is approximately $m \cdot E[Z']$. The second term is with some reasonable probability is less than $m^2 \cdot n$
$E[Z'] \gg \sqrt{var(Z')}$
$m^2\epsilon^2 \gg \sqrt{m^2 \cdot n}$
$m \gg \sqrt{n}/\epsilon^2$
Now, we put two regimes together and run their corresponding testers. If they passes both, then they are probably the same. If they failed one or the other, then they are probably different.

3 Robust Statistics

The algorithm we deal with so far work if they come from Gaussian or product distribution. But in real world, you can't expect it to be exactly correct. It could be the case that small fraction of the samples are bad due to human error or random answers.

3.1 Different types of error

The basic problem is that ϵ-fraction of samples are bad. There are different ways to talk about these errors.

3.1.1 Additive error (Huber model)

ϵ-fraction of samples come from some other distributions. See samples from $(1-\epsilon)\cdot G + \epsilon \cdot E$

3.1.2 General L1 error

See samples from G', $d_{TV}(G', G) < \epsilon$ for some $G \in C$

3.1.3 Strong adversary

Take N independent samples from G. Adversary change $\epsilon \cdot N$ of them arbitrarily.

3.2 Problem

We want to learn an unknown $N(\mu, I)$ Gaussian in R^n with ϵ-error.

Information theoretically : Cannot recover the initial G to error better than ϵ. $G, G' \geq 2\epsilon$ far, and we see samples from the mixture $\frac{1}{2}\cdot G + \frac{1}{2}\cdot G'$. The best thing we can do is to return the Gaussian in the middle, which results in ϵ error.

3.3 Cover Argument

ϵ-cover by picking ϵ cover of R^n.

And that doesn't quite work because cover of all of R^n is infinite. We need to restrict to a finite domain, if we take a random sample probably be within $\approx \sqrt{n}$ of true mean. $|Cover| = poly(n/\epsilon)^n$

Learn to error $O(\epsilon)$ with $\frac{nlog(n/\epsilon)}{\epsilon^2}$ samples.

However, this algorithm takes exponential time.

3.4 Sample mean

If we didn't have noise, then we can look at the sample mean. But this doesn't work now since a single bad sample can corrupt sample mean arbitrarily.

In one dimension, we can consider the median. And the median gives true mean to within $O(\epsilon)$ error.

In higher dimension, return sample median of each coordinate. However, there's ϵ error in each coordinate. And we are looking at L2 metric, which results in $\sqrt{n}\cdot\epsilon$ error.

There's a fix for this, using the Tukey median.

3.5 Tukey Median

For point set X, find $\hat{\mu}$ that minimized $Sup(\#\{x : v(x-\hat{\mu}) > 0\} - \#\{x : v(x-\hat{\mu}) < 0\})$ where $v \in$ cover of unit sphere. However, the size of the cover is exponential size.

We want to find v that minimized the maximum discrepancy over all distances. True mean $\#\{x : v(x-\hat{\mu}) > 0\} = O(\epsilon)$ with high probability.

If $|\hat{\mu} - \mu|_2 \gg \epsilon$, $|v|_2 = 1$ such that $v(\hat{\mu} - \mu) \gg \epsilon$, with high probability, $\#\{x : v(x - \hat{\mu}) < 0\}) \gg \epsilon$

with high probability, $|\hat{\mu} - \mu|_2 = O(\epsilon)$ Is $sup < \delta$? This is a linear program. However, Tukey median is NP-hard for arbitrary point set.

In conclusion, you can get optimal error but the algorithm run in exponential time.