Lecture 17: Lower bound on structured learning and Property testing with respect to A_k-metric

Daniel Kane
Scribe: Pinar Sen

May 15, 2017

Abstract

Last time, we showed that we are able to learn a distribution p that is ϵ approximately a t-piece wise degree d polynomial to error about ϵ in $\tilde{O}(\frac{t(d+1)}{\epsilon^2})$ samples. In this lecture, we discuss the lower bound for the same problem. Then, we move on to the problem of property testing with respect to A_k distance for a given class of distributions.

1 Structured Learning

Theorem 1.1. *Any algorithm which learns a distribution p that is t-piecewise degree-d polynomial to error ϵ requires $\Omega(\frac{t(d+1)}{\epsilon^2})$ samples.*

Proof. Recall that learning an arbitrary distribution on a set of size n requires n/ϵ^2 samples, which may be considered as an analog of Theorem 1.1. The basic point here is that the problem of learning a t-piecewise degree-d polynomial is similar in some sense to learning an arbitrary distribution on $t(d+1)$ independent bins with no structure between them.

First, we have t bins. On each bin on which the distribution is piecewise polynomial, we want d-ish independent values. One convenient way to do this is to look at the Chebyshev polynomials. Between $[-1, 1]$, we define polynomial function

$$P(cos(\theta)) = \frac{\left(\sum_{k=0}^{d} cos(k(\theta-\phi))\right)^2 + \left(\sum_{k=0}^{d} cos(k(\theta+\phi))\right)^2}{d^2},$$

which is highly concentrated on $cos(\phi)$. Notice that this function is an even trigonometric polynomial and falls of quadratically for $\theta \in (\phi \pm 1/d)$, which looks like two symmetric bumps centered at $\theta = \pm\phi$.

Let P_m denotes the normalization of the function P centered at $cos(\frac{2\pi m}{d})$ such that it has total mass 1. Now, we have d different P_m's for $m = 0, 1, \ldots, d-1$, which all have degree $O(d)$ and overlap very little.

We follow the adversarial method. Let p denote the final distribution and on each of t intervals, it can be written as

$$p = \frac{1}{d * t}\left(\sum_{m=0}^{d-1}(1 \pm 10\epsilon)P_m(x)\right)$$

The point is to learn p to error ϵ for which we need to guess $2/3$ of the $\pm$ signs correctly. The full proof is omitted. But the basic point is as follows. The distribution we have has t intervals and on each interval it is piecewise polynomial. All these piecewise polynomials are basically sums of these bumps functions and the only difference is that each of these bump functions are independently multiplied by either $(1 + 10\epsilon)$ or $(1 - 10\epsilon)$. Since most of the mass is coming from some effective support of one of the bumps and one of the intervals, on each of those bumps the mass or the probability density is dominated by the primary bump. Therefore, if we get the sign of the ϵ wrong for that bump, our probability density function is off by a reasonable multiple of ϵ. Thus, if an algorithm can learn p to error ϵ, it must guess at least $2/3$ of the $\pm$ signs for $t(d+1)$ bumps correctly.

The rest follows by a standard adversarial argument again. Let X be the string of $\pm$ signs that showed up and let N be the number of samples required. Then, the shared information between X and the samples must be much bigger than the $2/3$'s of the number of $\pm$ signs, from which we have

$$\begin{aligned} t(d+1) &\ll I(X; \text{samples}) \\ &\overset{(a)}{\leq} NI(X; \text{single sample}) \\ &\overset{(b)}{\leq} NO(\epsilon^2) \end{aligned}$$

where (a) follows by the conditional independence and (b) follows by the fact that the best information that a single sample can give about X is the ϵ bias of a single coin. Then, we have

$$N \gg \frac{t(d+1)}{\epsilon},$$

which completes the proof sketch for the theorem. □

2 Property Testing

One interesting question is how to solve the problem of property testing if we are guaranteed that the distributions lie in the same class.

A Problem definition

Let $p, q \in \mathcal{C}$ for some class $\mathcal{C}$. Given samples, distinguish $p = q$ vs. $d_{TV}(p, q) > \epsilon$.

Since it has worked so well for learning results, a lot of studies discussed A_k distance. The idea is to find a large enough k such that for any p and $q \in \mathcal{C}$,

$$d_{TV}(p,q) = |p-q|_{A_k} \pm \epsilon/2. \tag{1}$$

Therefore, if (1) holds, it is enough to test $p = q$ vs $|p-q|_{A_k} > \epsilon/2$.

Here is an analogy that works moderately well. Recall that learning an arbitrary distribution on a set of size n takes about n/ϵ^2 sample. If we want to learn a distribution on $\mathbb{R}$ with respect to variational distance, it takes infinite number of samples (i.e., we can not do it). On the other hand, learning a distribution on $\mathbb{R}$ with respect to the A_k metric takes only k/ϵ^2 samples. In some sense, the point is that there are only essentially k bins that we need to deal with. It is easy to show that this problem requires at least k/ϵ^2 samples: You could assume that this distribution was a histogram on these k pieces, which turns out to be exactly the same problem as learning an arbitrary unstructured distribution on k pieces. Surprisingly, we could show that k/ϵ^2 samples are indeed enough.

On the other hand, identity testing on $[n]$ takes $\sqrt{n}/\epsilon^2$ samples. By analogy, we expect that identity testing with respect to the A_k-metric should require $\sqrt{k}/\epsilon^2$ samples. Considering the case of histogram argument above, it requires at least $\sqrt{k}/\epsilon^2$ many samples. But, we are going to show over the course of the next couple of lectures, $\sqrt{k}/\epsilon^2$ samples are indeed sufficient. It will immediately imply that for a whole bunch of testing problems restricted to various classes in which the normal variational distance is approximated by A_k we will also get effectively optimal algorithms.

B Algorithm

Let p and q be two distributions where q is known. We want to distinguish $p = q$ from $|p-q|_{A_k} > \epsilon$.

There is a simple reduction that we can perform. We can assume $q = U(0,1)$. The point is to make a change of variables. The scaling of x axis does not change the A_k distance. As long as we preserve the ordering on the x axis, all of the partitions into intervals is also going to be preserved. So, we can make any order preserving change of variables. Assuming q is a continuous distribution, there is a unique order preserving change of variables that turns it into $U(0,1)$. We use the CDF of q as the new variable to make that reduction.

Secondly, we are going to look at the special case where p is a k-flat distribution. The point is that for this special case A_k distance is exactly the variational distance and the intervals are the intervals on which p is flat. For this case, we know that there is a partition into k intervals such that the restriction to that partition gives the A_k distance. If S is a partition into intervals, and p^S and q^S be the distributions of which interval we land in. Then, by definition we have

$$|p-q|_{A_k} = \sup_{|S|=k} d_{TV}(p^S, q^S).$$

If we know S for any given S, it is not hard to distinguish $p^S = q^S$ vs. $|p^S - q^S| > \epsilon$, which requires $\sqrt{k}/\epsilon^2$ samples. Now, the problem is that for such an algorithm to work we have to

know what partition to look at. If you consider S as the partition into k *equal* intervals, it may have the following problem: p may have zigzags on $k/2$ intervals and agree with q on the rest of the intervals. Then, the A_k distance between p and q will be pretty substantial. What do we do to detect these zigzags? We can use finer partition. In general, we have the following algorithm.

Algorithm: Let S_m be the partition into $k2^m$ equal parts for $m = 0, 1, \ldots, \log(1/\epsilon)$. Run testers

$$p^{S_m} = q^{S_m} \quad \text{vs.} \quad |p^{S_m} - q^{S_m}|_2^2 > \epsilon^2 \frac{2^{2m/3}}{k}$$

with error probability $\leq \frac{1}{10*2^m}$. Return same if and only if $p^{S_m} = q^{S_m}$ for all m.

Apparently, there is a trade-off between the number of bins that algorithm looks at and the L^2 error. We will discuss the details of analysis of the algorithm next lecture .