Lecture 12: Learning Structured Distributions

Daniel Kane
Scribe: Alankrita Bhatt

May 22, 2017

Abstract

We study methods of learning distributions in the case when the distribution p has infinite support. We introduce the idea of an ϵ-cover and use this to construct a learning algorithm.

1 Introduction

All the learning algorithms we have examined so far have a sample complexity that depends on the support size n. Hence if we take a distribution with infinite support (for example, any continuous distribution), these algorithms will not work. For example, given N samples, it is impossible to distinguish whether p came from $U(0,1)$ or some discrete distribution with N^2 spikes (since it'll take us approximately $\sqrt{N^2}$ samples to start seeing collisions, and thus it's highly unlikely any collisions will be observed in the first N samples).

This makes it seem that learning continuous distributions is impossible. However, we will see that for some "nice" classes of distributions p, we can still learn a substantial amount about the distribution from a finite number of samples. Some of these "nice" classes $\mathcal{C}$ include

- Gaussians or Mixtures of Gaussians
- Log-Concave distributions
- Distributions with bounded Lipschitz Constants
- k-piecewise constant distributions

2 Distinguishing between two distributions

We now consider a very simple case, where our class $\mathcal{C}$ consists only of two distributions. Thus $\mathcal{C} = \{q_1, q_2\}$. We get samples from a distribution p and we want to figure out which one of q_1 or q_2 it came from.

A Lower bound

A simple lower bound on the sample complexity is $\Omega\left(1/(d_{TV}(q_1, q_2)\right)$. The intuition behind this is as follows. Let $d_{TV}(q_1, q_2) = \epsilon$. We can then see q_2 as a copy of q_1 that has an error with probability ϵ. If we take less than around $1/\epsilon$ samples, we are highly unlikely to see any of these "errors" and hence it's impossible to distinguish between q_1 and q_2. Another way of looking at this is to note that $d_{TV}(q_1^N, q_2^N) \leq N d_{TV}(q_1, q_2)$. If $N \ll 1/\epsilon$, then we cannot distinguish independent N copies of q_1 and q_2.

This lower bound may or may not be tight. We illustrate one example of each case.

- $q_1 = \delta_0$ and $q_2 = (1-\epsilon)\delta_0 + \epsilon\delta_1$. This only requires some multiple of $1/\epsilon$ samples to distinguish between the two (again, by thinking of q_2 as a version of q_1 with error probability ϵ).
- If q_1 is a fair coin and q_2 is a coin with ϵ bias, i.e. $q_1 = \frac{1}{2}\delta_0 + \frac{1}{2}\delta_1$ and $q_2 = (\frac{1}{2} + \epsilon)\delta_0 + (\frac{1}{2} - \epsilon)\delta_1$, we need at least $1/\epsilon^2$ samples. Fortunately, this is the worst possible case and the sample complexity cannot get any worse than this.

The following lemma gives us a very useful characterization of the total variation distance d_{TV}.

Lemma 2.1.

$$d_{TV}(q_1, q_2) = \sup_A |q_1(A) - q_2(A)|.$$

Proof. Let $A = \{x : q_1(x) > q_2(x)\}$.

$$\begin{aligned} d_{TV}(q_1, q_2) &= \frac{1}{2}\int |dq_1 - dq_2| \\ &= \frac{1}{2}\left(\int_{q_1 > q_2} (dq_1 - dq_2) + \int_{q_1 < q_2} (dq_2 - dq_1)\right) \\ &= \frac{1}{2} \cdot 2(q_1(A) - q_2(A)) = q_1(A) - q_2(A). \end{aligned}$$

□

Consider $\mathbb{1}_{\{p \in A\}}$, where $A = \{x : q_1(x) > q_2(x)\}$. This is distributed exactly as a biased coin, with two possible biases (depending on whether $p = q_1$ or $p = q_2$). Since $d_{TV}(q_1, q_2) = \epsilon$ the difference in biases in the two cases is ϵ. Following the arguments given above, we can learn the bias of this coin to error $\epsilon/3$ in $O(1/\epsilon^2)$ samples.

We can do this procedure algorithmically if we can

- determine the set A where $q_1(x) > q_2(x)$ in polynomial time;
- approximate $p(A)$ using samples from p efficiently.

We can then compare whether $p(A)$ is closer to $q_1(A)$ or $q_2(A)$. Since by definition A is the set on which $q_1(A)$ and $q_2(A)$ differ the most, it is the easiest to test whether $p = q_1/q_2$ by comparing probabilities on this set. Hence we have an algorithm distinguishing $p = q_1$ vs. $p = q_2$.

3 Distinguishing more than two distributions

We now want to extend the above problem to more than two distributions. Thus, we have $p \in \{q_1, q_2, \cdots q_n\}$ and want to know which one of them it is equal to.

One way of doing this is running a tournament - given any pair q_i, q_j test whether $p = q_i$ or $p = q_j$. Keep eliminating possibilities unless there's only one distribution left. If we define $\epsilon = \min_{i,j} d_{TV}(p_i, p_j)$ where $i \neq j \in [n]$, the sample complexity of the above would be $O(n/\epsilon^2)$.

One possible improvement over this is by using the same samples for all the tests. In this case, we need to ensure that the probability of failure of each test is $\ll 1/n$ so that our tester doesn't fail an important test and give the wrong answer. We know that we can boost the probability of success of a test to $1 - \delta$ by running it in parallel $O(1/\delta)$ times. Hence, using this idea, we can bring down the sample complexity to $O(\log n/\epsilon^2)$. However it seems slightly unrealistic that the distributions are explicitly known.

4 Learning using ϵ-cover

Let $p \in \mathcal{C}$. Our goal is to learn p in d_{TV} up to $O(\epsilon)$. We will now define an ϵ-cover which will be further used to learn this distribution.

A finite set S is called an *ϵ-cover* of $\mathcal{C}$ if for any $p \in \mathcal{C}$, there exists $q \in S$ such that $d_{TV}(p, q) < \epsilon$. We know that p is within ϵ of some elements of S. Hence we hope to find some elements of S within $O(\epsilon)$ of p. Suppose we explicitly know two distributions q_1 and q_2 and have sample access to p. We wish to distinguish between

- $d_{TV}(p, q_1) \leq \epsilon$ and $d_{TV}(p, q_2) \geq 10\epsilon$
- $d_{TV}(p, q_2) \leq \epsilon$ and $d_{TV}(p, q_1) \geq 10\epsilon$

Define $A := q_1 > q_2$. Since we know q_1 and q_2 we also know the set A. In both of these cases, by the triangle inequality, $|q_1(A) - q_2(A)| = d_{TV}(q_1, q_2) \geq 9\epsilon$. But, in the first case, $|p(A) - q_1(A)| < \epsilon$. In the second case $|p(A) - q_2(A)| < \epsilon$. Hence, the two cases are separated by a large gap.

So if we can approximate $q_1(A), q_2(A)$ and $p(A)$ within ϵ, we can ask whether $\hat{p}(A)$ is closer to $\hat{q}_1(A)$ or $\hat{q}_2(A)$. (Note : Initially we assumed that q_1 and q_2 are known, but that need not be the case always. We'll be looking at the case when we have, say, an ϵ-cover of $\mathcal{C}$ and not $\mathcal{C}$ itself.)

Now suppose we have an ϵ-cover of $\{q_1, \cdots q_n\}$. Thus, $\exists$ a q_i such that $d_{TV}(p, q_i) \leq \epsilon$. Our algorithm will be as follows.

Run the following tester on all q_i, q_j pairs. Test for

- $d_{TV}(p, q_1) \leq \epsilon$ and $d_{TV}(p, q_2) \geq 10\epsilon$ OR
- $d_{TV}(p, q_2) \leq \epsilon$ and $d_{TV}(p, q_1) \geq 10\epsilon$

Find an i such that q_i doesn't fail to any q_j with $d_{TV}(q_i, q_j) \geq 20\epsilon$.

Lemma 4.1. $\exists$ *such a* q_i *and it is close to* p *in total variation distance.*

Proof. Existence of q_i -
If $d_{TV}(p, q_i) \leq \epsilon$, then for all q_j such that $d_{TV}(q_i, q_j) > 20\epsilon$, we get that $d_{TV}(p, q_j) > 19\epsilon$. By the construction of our algorithm, we get that q_i beats q_j in its pairwise comparision. Thus there is always such a q_i.
Closeness to p -
If $d_{TV}(p, q_i) > 30\epsilon$, if we take q_j with $d_{TV}(p, q_j) \leq \epsilon$ (and there always exists such a q_j), q_j beats q_i in pairwise comparision, and $d_{TV}(q_i, q_j) > 20\epsilon$. □

This lemma gives us an algorithm for getting an approximation of p.

- Run all pairwise comparisons.
- Find q_i with the property q_i doesn't fail to any q_j with $d_{TV}(q_i, q_j) > 20\epsilon$.
- Return q_i.

Assuming all comparisons worked, $d_{TV}(p, q_i) = O(\epsilon)$.

Runtime of this algorithm is poly$(|S|/\epsilon)$. This runtime is usually bad, because a lot of the times $|S|$ can be exponentially large.
Sample complexity is $O(\log |S|/\epsilon^2)$ by reusing the samples and using the boosting argument mentioned before. This sample complexity is often near optimal.

5 Examples

Gaussians in one dimension Assume $\mu = O(1)$, $\sigma = \Theta(1)$. Note that $d_{TV}(N(\mu, \sigma^2), N(\mu', \sigma'^2)) = O(|\mu - \mu'| + \|\sigma - \sigma'|)$ and hence we can get a small cover. If we take the ϵ-grid, $|S| = O(1/\epsilon^2)$. Now we have sample complexity $O(\epsilon^{-2} \log(1/\epsilon))$ and runtime is poly$(1/\epsilon)$.
This is not the best way of learning a Gaussian - as we know, the optimal way is to compute the sample mean and variance. This is a lot faster, and uses only $1/\epsilon^2$ samples.

Mixture of k Gaussians Let $p = \sum_{i=1}^{k} w_i G_i$, $\sum_i w_i = 1$, where $G_i = N(\mu_i, \sigma_i^2)$. Again we assume $\mu_i = O(1)$ and $\sigma_i = \Theta(1)$. We cannot use the sample mean and variance approach in this case to learn this distribution. If our only requirement is that of a decent sample complexity, then we can use the following method to learn - first find an $\epsilon/2$-cover of each G_i, and find an ϵ-cover of k dimensional simplex $O_k(\{w_1, \ldots, w_k\})$ by approximating w_i, μ_i, σ_i to error ϵ/k. This gives $|S| = (k/\epsilon)^{3k}$. This gives runtime $\epsilon^{-O(k)}$, which is large and impractical, but sample complexity $O(\epsilon^{-2} k \log(k/\epsilon))$, which is a near optimal other than the log term.