Learning Under Distributional Assumptions

As we've seen, many natural learning problems are hard if we make <u>no</u> distributional assumptions

<u>Today</u>: Study learning under <u>simple</u> distributions

There is no known subexponential time alg. for learning intersections of two halfspaces, but...

<u>Theorem 1</u> [Klivans, O'Donnell, Servedio]
For any constant k, there is a poly. time algorithm for learning intersections of k halfspaces under uniform distribution on hypercube

The main idea is to show:

"intersections of k halfspaces can be approximated by low degree polys."

We'll employ Fourier analysis over the hypercube

<u>Proposition</u>: For any function

$$f: \{\pm 1\}^d \to \mathbb{R}$$

we can write it as

$$f(x) = \sum_{S \subseteq [d]} \hat{f}(S)\, \chi_S(x)$$

where $\chi_S(x) = \prod_{i \in S} x_i$ and

$$\hat{f}(S) = \mathop{\mathbb{E}}_{x \sim_u \{\pm 1\}^d} [f(x)\, \chi_S(x)] \triangleq \langle f, \chi_S \rangle$$

<u>Intuition</u>: The functions χ_S are orthonormal and they span the space of functions on the hypercube

$T = \{1, 2, 3\}$ $S = \{1, 2\}$

$$\mathop{\mathbb{E}}_{x \sim_u \{\pm 1\}^d} [\chi_T(x)\, \chi_S(x)] = \mathbb{E}[x_3] = 0$$

Thus going from one representation of a function

(1) its evaluation at each point of the hypercube

to another

(2) its spectrum, i.e. the collection of it Fourier coefficients $\hat{f}(S)$

is just an orthogonal transformation

Thus we have:

<u>Fact</u> [Parseval's Identity]:

$$\mathbb{E}_{x \sim_u \{\pm 1\}^d}\left[f(x)^2\right] = \sum_{S \subseteq [d]} \hat{f}(S)^2$$

Let's prove this to make sure we are comfortable with Fourier analysis over the hypercube

Proof: Substituting in the Fourier representation, we have

$$\mathbb{E}_{x \sim_u \{\pm 1\}^d} \left[\left(\sum_{S \subseteq [d]} \hat{f}(S) \chi_S(x) \right) \left(\sum_{T \subseteq [d]} \hat{f}(T) \chi_T(x) \right) \right] \quad (*)$$

Now by orthogonality, all the cross terms that do not match are zero,

$$(*) = \mathbb{E}_{x \sim_u \{\pm 1\}^d} \Big[\sum_{S \subseteq [d]} \hat{f}(S)^2 \underbrace{\chi_S^2(x)}_{=1} \Big]$$

$$= \sum_{S \subseteq [d]} \hat{f}(S)^2$$

$\blacksquare$

The starting point for many learning algorithms is truncation

$$g_\ell(x) = \sum_{|S| < \ell} \hat{f}(S) \chi_S(x)$$

which will give us a good <u>low-degree</u> approximation

Let's make this precise:

<u>def:</u> We say that f is <u>$\alpha(\varepsilon, d)$ concentrated</u>

$$\sum_{|S| \geq \alpha(\varepsilon, d)} \hat{f}(S)^2 \leq \varepsilon$$

Or to put it another way, if we take $g_{\alpha(\varepsilon,d)}$ and let r be the residual — i.e.

$$r(x) = f(x) - g_{\alpha(\varepsilon,d)}(x)$$

we have

$$\mathbb{E}_{x \sim_u \{\pm 1\}^d}\left[r(x)^2\right] = \sum_{|S| \geq \alpha(\varepsilon,d)} \hat{f}(S)^2 \leq \varepsilon$$

The seminal work of Linial, Mansour and Nisan introduced Fourier analysis into computational learning theory

They showed the following meta-theorem

Theorem 2 If a concept class $\mathcal{H}$ has Fourier concentration $\alpha(\varepsilon, d)$ then there is a PAC learning algorithm that runs in time $d^{O(\alpha(\varepsilon,d))}$ on uniform distribution

The Low Degree Algorithm

For each $S \subseteq [d]$ with $|S| < \alpha(\varepsilon, d)$

Take m samples to estimate

$$\hat{g}(S) = \frac{\sum_{i=1}^{m} f(x) \chi_S(x)}{m} \approx \hat{f}(S)$$

Output the estimate

$$g(x) = \sum_{|S| < \alpha(\varepsilon, d)} \hat{g}(S) \chi_S(x)$$

Proof: If each estimate satisfies

$$|\hat{g}(S) - \hat{f}(S)|^2 \leq \frac{\varepsilon}{d^{O(\alpha(\varepsilon,d))}} \qquad (\Delta)$$

then we have

$$\mathbb{E}\left[|g(x)-f(x)|^2\right] \le 2\varepsilon$$

And now, given x, if you output the label $y = \mathrm{sgn}(g(x))$ we can check

$$(f(x)-g(x))^2 \ge \mathbb{1}_{\mathrm{sgn}(g(x)) \neq f(x)}$$

Thus we have

$$\mathbb{P}_{x \sim_u \{\pm 1\}^d}\left[\mathrm{sgn}(g(x)) \neq f(x)\right] \le 2\varepsilon$$

Finally, standard tail bounds imply we can choose

$$m = \frac{d^{O(\alpha(\varepsilon,d))}}{\varepsilon^2} \log \frac{d^{O(\alpha(\varepsilon,d))}}{\delta}$$

s.t. (Δ) holds for all S with $|S| < \alpha(\varepsilon, d)$ with probability $\ge 1-\delta$. $\blacksquare$

Main Question: what kinds of functions have Fourier Concentration?

We will see that stable functions have Fourier concentration, and are thus learnable

def: Let $T_\eta(x)$ denote the noise operator which flips each bit independently with prob. η

def: Let $NS_\eta(x)$ denote the noise sensitivity of f, defined as

$$NS_\eta(x) \triangleq \mathbb{P}_{x \sim_u \{\pm 1\}^d}\left[f(x) \neq f(T_\eta(x))\right]$$

The noise sensitivity has a closed-form expression:

Proposition 2: For $\eta < \frac{1}{2}$

$$NS_\eta(x) = \frac{1}{2} - \frac{1}{2}\sum_S (1-2\eta)^{|S|}\hat{f}(S)^2$$

<u>Proof:</u> First we claim

$$NS_\eta(f) = \frac{1}{2} - \frac{1}{2}\overbrace{\mathbb{E}[f(x)f(T_\eta(x))]}^{(\square)}$$

Again, we can substitute in the Fourier representation:

$$(\square) = \mathbb{E}\left[\left(\sum_S \hat{f}(S)\chi_S(x)\right)\left(\sum_U \hat{f}(U)\chi_U(T_\eta(x))\right)\right]$$

And again, only the terms where $S=U$ contribute. Thus we have:

$$(\square) = \sum_S \hat{f}(S)^2\,\mathbb{E}\underbrace{[\chi_S(x)\,\chi_S(T_\eta(x))]}_{(\circ)}$$

Now we can think about the noise operator equivalently as:

1. Select bits independently with probability 2η

(2) set them u.a.r. to ± 1

Thus we have

$$(0) = \mathbb{P}\left[\substack{\text{no bit in} \\ S \text{ is selected}}\right] = (1-2\eta)^{|S|}$$

Putting it all together

$$NS_\eta(f) = \frac{1}{2} - \frac{1}{2} \sum_S \hat{f}(S)^2 (1-2\eta)^{|S|}$$

$\blacksquare$

Note: There are many manifestations of these same principles in other learning settings

gaussian noise sensitivity $\rightarrow$ learnability under gaussian distributions

which applies for functions on $\mathbb{R}^d$

With these tools in hand, the key ideas are

1. halfspaces have low noise sensitivity
2. combining functions with low noise sensitivity, does not increase noise sensitivity too much
3. low noise sensitivity $\Rightarrow$ Fourier concentration

For 1, we have:

<u>Theorem 3</u> [Peres]: Let f be any halfspace. Then $NS_n(f) \leq O(\sqrt{n})$

We will not prove this. But the intuition is related to the <u>central limit theorem</u>:

In particular, consider

$$\mathrm{MAJ} \triangleq \operatorname{sgn}\left(\sum_{i=1}^{d} x_i\right)$$

Now if we consider the quantity

$$X \triangleq \frac{\sum_{i=1}^{d} x_i}{\sqrt{d}}$$

it behaves like $N(0,1)$ - a mean zero and variance one Gaussian

Now let $y = T_\eta(x)$ and $Y = \frac{\sum_{i=1}^{d} y_i}{\sqrt{d}}$

Again by the central limit theorem

$$Y - X \sim N(0, 4\eta)$$

because we flip a η fraction of bits in expectation

And the dominant contribution to the event $\operatorname{sgn}(Y) \neq \operatorname{sgn}(X)$ comes from:

1. $-c\sqrt{n} \le X \le c\sqrt{n}$

2. $|Y - X| > c\sqrt{n}$ and the sign goes the right way

You can check that the probability all these things happen is about $\sqrt{n}$

Next we will translate this into a bound on the Fourier concentration

<u>Lemma 1</u>: For any $0 < \eta < \frac{1}{2}$, we have

$$\sum_{|S| \ge \frac{1}{\eta}} \hat{f}(S)^2 \le \left(\frac{2}{1 - \frac{1}{e^2}}\right) NS_\eta(f)$$

<u>Proof</u>: From Proposition 2, we have

$$2\, NS_\eta(f) = 1 - \sum_S (1-2\eta)^{|S|} \hat{f}(S)^2$$

Applying Parseval's identity

$$= \sum_{S} \hat{f}(S)^2 - \sum_{S} (1-2\eta)^{|S|} \hat{f}(S)^2$$

$$= \sum_{S} \hat{f}(S)^2 \left(1 - (1-2\eta)^{|S|}\right)$$

$$\geq \sum_{|S| \geq \frac{1}{\eta}} \hat{f}(S)^2 \left(1 - (1-2\eta)^{|S|}\right)$$

$$\geq \sum_{|S| \geq \frac{1}{\eta}} \hat{f}(S)^2 \left(1 - (1-2\eta)^{\frac{1}{\eta}}\right)$$

$$\geq \sum_{|S| \geq \frac{1}{\eta}} \hat{f}(S)^2 \left(1 - \frac{1}{e^2}\right)$$

Rearranging completes the proof. $\blacksquare$

Finally, we need an elementary fact

Fact 2: Suppose we consider composite functions $h(x) = g(f_1(x), \ldots, f_k(x))$

Then $NS_n(h) \leq \sum_{i=1}^{k} NS_n(f_i)$

Proof: This follows from the definition of noise sensitivity + union bound

$\blacksquare$

Now we are ready to prove Theorem 1

Proof: Let f be an intersection of k halfspaces. Using Lemma 1:

$$\sum_{|S| \geq \frac{1}{n}} \hat{f}(S)^2 \leq 2.32\, NS_n(f)$$

And applying Fact 2 and Theorem 3:

$$\leq C k \sqrt{n}$$

Now set $n = \frac{\varepsilon^2}{Ck^2}$, which implies

$$\sum_{|S| \geq \frac{Ck^2}{\varepsilon^2} = \alpha(\varepsilon, d)} \hat{f}(S)^2 \leq \varepsilon$$

Finally, using the low degree algorithm we get a $d^{O(\frac{k^2}{\varepsilon^2})}$ time algorithm, as desired. $\boxtimes$

Is this bound tight?

For intersections of halfspaces, can beat the union bound substantially

<u>Theorem</u> [Kane]: For an intersection of k halfspaces

$$NS_n(f) \leq O(\sqrt{n \log k})$$

Corollary: There is a $d^{O(\frac{\log k}{\varepsilon^2})}$ time alg. for learning intersections of k halfspaces over the uniform distribution on $\{\pm 1\}^d$

Note: All of these algorithms even work in the agnostic setting and get error $opt + \varepsilon$

In Gaussian space, can do even better

Theorem [Vempala]: There is a

$$\text{poly}(d, k, \tfrac{1}{\varepsilon}) + k^{O(\frac{\log k}{\varepsilon^4})}$$

time algorithm for learning intersections of k halfspaces over a Gaussian

Intuition: If you restrict to just the positive/negative examples, you get

span of normals $\Longleftrightarrow$ directions of smallest variance

Statistical Query Lower Bounds

Recall, we discussed cryptographic lower bounds for learning

Main Question: But what about lower bounds for "nice" distributions?

Today, we'll introduce a powerful framework

def [Kearns]: In the statistical query model, in each step

(1) The algorithm specifies a query

$$q : \mathbb{R}^d \times Y \to [-1, 1]$$

and a tolerance τ

(2) It gets a response r s.t.

$$|r - \mathbb{E}_{(x,y)\sim D}[q(x,y)]| \leq \tau$$

This is a restriction on what you are allowed to do algorithmically

And you can turn SQ algorithms into actually learning algorithms in the following sense

SQ Algorithm

Assumptions

(1) at most polynomially many queries

(2) each tolerance is at least inverse polynomially big

③ can compute the next query, and evaluate it on samples efficiently

$\Downarrow$

There is a polynomial time learning algorithm

The important point is SQ algorithms do not look directly at the data, but rather use inexact statistics/summaries

Meta Claim: By dropping ③, can prove many tight upper/lower bounds

Let's start with a key example:

Learning Parity Functions

1. There is an unknown $T \subseteq [d]$
2. $X \sim_u \{\pm 1\}^d$, and $y = \chi_T(x)$

Observe that parities are not at all Fourier concentrated / stable

Theorem 3 [Blum, Furst, Jackson, Kearns, Mansour, Rudich] Any SQ algorithm for learning parities must make at least $2^{\Omega(d)}$ queries or have $\tau = 2^{-\Omega(d)}$

Proof (sketch): It turns out that it suffices to allow only correlational queries:

$$q(x,y) \triangleq y\, q(x)$$

Now, once again, we use the Fourier representation

$$q(x) = \sum_S \hat{q}(S)\, \chi_S(x)$$

Thus we have

$$\mathbb{E}[y\, q(x)] = \mathbb{E}\Big[\chi_T(x) \sum_S \hat{q}(S)\, \chi_S(x)\Big]$$
$$= \hat{q}(T)$$

Now by Parseval's identity

$$\sum_S \hat{q}(S)^2 = \mathbb{E}[q(x)^2] \leq 1$$

Thus the intuition is

(i) we can only have $|\hat{q}(S)| \geq \tau$ for at most $\frac{1}{\tau^2}$ parities

(2) If we have $|\hat{q}(T)| < \tau$ the oracle can just reply "zero"

Thus it will take us about $\tau^2 2^d$ queries to find T. $\blacksquare$

Is there a non-SQ algorithm for learning parities?

<u>Fact 3</u>: There is a polynomial time algorithm for learning parities

<u>Main Idea</u>: Solve for T by setting up a linear system over $\mathbb{F}_2$

$$\underset{m \times d}{A}\,\overset{1_T}{c} = y$$

$$\begin{matrix} - x_1 - \\ - x_2 - \\ \vdots \end{matrix}$$

But this algorithm is <u>highly non-robust</u>

In fact:

i.e. flip label with prob. $\frac{1}{3}$

Theorem [Blum, Kalai, Wasserman]
There is a $2^{d/\log d}$ time algorithm for learning noisy parities

This is the best known algorithm

Are there other natural learning problems that separate SQ and general learning algorithms?

Next time: SGD as an SQ algorithm, and SQ lower and upper bounds for learning deep nets on Gaussian inputs