Lecture 18

Fourier-based learning algorithms

- the low degree algorithm
- Fourier Concentration
- Noise sensitivity

Recall Fourier Transform:

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

$$\langle f, g \rangle = \frac{1}{2^n} \sum_x f(x)\, g(x)$$

lemma

$$\hat{f}(S) = \langle f, \chi_S \rangle = 1 - 2 \cdot \Pr[f(x) \neq \chi_S(x)]$$

$$\forall f, \quad f(x) = \sum \hat{f}(S) \chi_S(x)$$

Plancherel $\langle f, g \rangle = \sum_S \hat{f}(S)\, \hat{g}(S)$

Learning via Fourier Representation

will look at learning algorithms that are based on estimating Fourier representation of fctn f (similar to polynomial interpolation)

Approximating one Fourier coefficient:

<u>lemma</u> for any $S \subseteq [n]$, can approx $\hat{f}(S)$ to within additive γ

(ie. $|\text{output} - \hat{f}(S)| \leq \gamma$)

with prob $\geq 1-\delta$ in $O\left(\frac{1}{\gamma^2} \log \frac{1}{\delta}\right)$ samples.

no queries needed!

(Proved last time)

The low degree algorithm

definition of fctns for which low degree Fourier coeffs pretty much suffice to describe fctn:

def $f:\{\pm 1\}^n \to \mathbb{R}$ has $\alpha(\varepsilon, n)$ - Fourier concentration

if $\displaystyle\sum_{\substack{S \subseteq [n] \\ \text{s.t.} \\ |S| > \alpha(\varepsilon,n)}} \hat{f}(S)^2 \le \varepsilon \quad \forall\ 0<\varepsilon<1$

for Boolean f, this implies

$$\sum_{\substack{S \subseteq [n] \\ \text{s.t.} \\ |S| \le \alpha(\varepsilon,n)}} \hat{f}(S)^2 \ge 1-\varepsilon$$

examples

1) fctn f which depends on $\le k$ vars has $\displaystyle\sum_{\substack{S \text{ s.t.} \\ |S|>k}} \hat{f}(S)^2 = 0$

if f doesn't depend on x_i then all $\hat{f}(S)$ for which $i \in S$ satisfy $\hat{f}(S)=0$

Low degree algorithm

approximates fcfns with $d \equiv d(\varepsilon, n)$ Fourier concentration:

Given:

d	degree
τ	accuracy
δ	confidence

Algorithm:

- Take $m = O\left(\frac{n^d}{\tau} \ln \frac{n^d}{\delta}\right)$ samples
- For each S s.t. $|S| \le d$:

  $C_S \leftarrow$ estimate of $\hat{f}(S)$

  $\le \binom{n}{d}$ of these

  reuse samples

- let $h(x) \equiv \sum_{|S| \le d} C_S \cdot \chi_S(x)$
- output sign(h) as hypothesis

Why does this work?

Two stages:

$\frac{L_2\text{-dist}}{2^n}$

1) Show that f has low F.C.

$$\Rightarrow \quad \mathbb{E}_x[(f(x)-h(x))^2] \text{ small}$$

2) Show that $\Pr[f(x) \neq \text{sign}(h(x))] \leq \mathbb{E}_x[(f(x)-h(x))^2]$

Hamming dist

put together:
f has low F.C.
$\Rightarrow$ sign$(h(x))$
is good
approximation
of f

First "stage":

<u>Thm 1</u> if f has $d = \alpha(\varepsilon, n)$ - F.C. then
h satisfies $\mathbb{E}_x[(f(x)-h(x))^2] \leq \varepsilon + \tau$
with prob $\geq 1-\delta$ (Proved last time)

2nd "stage":

<u>Thm 2</u> $f: \{\pm 1\}^n \to \{\pm 1\}$
$h: \{\pm 1\}^n \to \mathbb{R}$
then $\Pr[f(x) \neq \text{sign}(h(x))] \leq \mathbb{E}_{x \in U}[(f(x)-h(x))^2]$

__Proof.__

$$\underset{x \in \mathcal{U}}{E}\left[(f(x)-h(x))^2\right] = \frac{1}{2^n}\sum_x (f(x)-h(x))^2$$ defn.

compare these term-by-term to get Thm.

$$\Pr\left[f(x) \neq \operatorname{sign}(h(x))\right] = \frac{1}{2^n}\sum_x \mathbb{1}_{\{f(x)\neq \operatorname{sign}(h(x))\}}$$

Consider "$(f(x)-h(x))^2$" vs. "$\mathbb{1}_{\{f(x)\neq \operatorname{sign}(h(x))\}}$":

__Case 1__ if $f(x) = \operatorname{sign}(h(x))$:

$$\mathbb{1}_{f(x)\neq \operatorname{sign}(h(x))} = 0$$

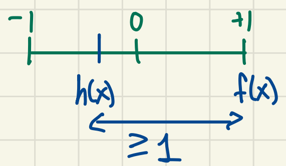
$$(f(x)-h(x))^2 \geq 0$$

__Case 2__ if $f(x) \neq \operatorname{sign}(h(x))$:

$$\mathbb{1}_{f(x)\neq \operatorname{sign}(h(x))} = 1$$

$$(f(x)-h(x))^2 \geq 1$$

why? e.g. if $f(x) = +1$ then in this case $h(x) < 0$:



(other case is analogous)

So, $\forall x$

$$(f(x)-h(x))^2 \geq \mathbb{1}_{f(x)\neq \operatorname{sign}(h(x))}$$ ∎

Correctness of learning algorithm

<u>Thm</u> if $\mathcal{C}$ has Fourier concentration $d = \alpha(\varepsilon, n)$

then there is a $q = O\left(\frac{n^d}{\varepsilon} \log \frac{n^d}{\delta}\right)$ sample

uniform distribution learning algorithm for $\mathcal{C}$

i.e. algorithm gets q samples & with prob $\geq 1-\delta$

outputs h' s.t. $\Pr[f \neq h'] \leq 2 \cdot \varepsilon$

<u>Pf</u>.

run low degree alg with $\gamma = \varepsilon$

thm 1 $\Rightarrow$ get h s.t. $E[(f-h)^2] \leq \varepsilon + \varepsilon = 2\varepsilon$

output $h' = \text{sign}(h)$

$\Uparrow$

thm 2 $\Rightarrow$ h' has error $\leq 2 \cdot \varepsilon$

$\blacksquare$

Applications

1) Bounded depth decision trees

$$f(x) = \sum_{\substack{\ell \in \text{leaves} \\ \text{of } T}} \underbrace{f_\ell(x)}_{\substack{\text{fctn which} \\ \text{depends on} \\ \le \text{depth many} \\ \text{vars}}} \cdot \underbrace{val(\ell)}_{\text{const}}$$

$$\hat{f}(S) = \sum val(\ell)\, \underbrace{\hat{f}_\ell(S)}_{\substack{0 \text{ for} \\ |S| > \text{depth}}} \qquad \text{linearity}$$

$\Longrightarrow \quad \forall\, S \text{ s.t. } |S| > \text{depth}, \quad \hat{f}(S) = 0$

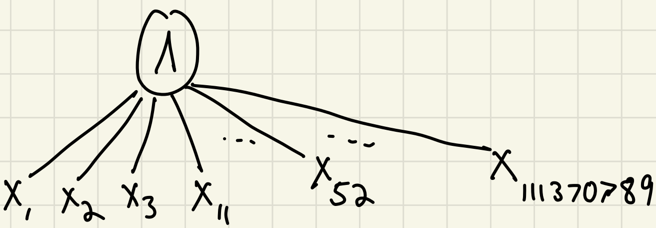
so $O\left(\frac{n^{\text{depth}}}{\varepsilon} \log \frac{n^{\text{depth}}}{\delta}\right)$ suffices

2) Constant depth ckts

<u>def</u>. "Boolean Ckt C" is DAG

gates: $\underbrace{\wedge, \vee, \neg}_{\text{operations}}, \underbrace{1, 0}_{\text{consts}}, \underbrace{x_1 \cdots x_n}_{\text{vars}}$

how many inputs? const, poly, unbounded?



Can we compute parity (xor) of n bits

in const depth?

yes! can compute any fctn on n bits

in const depth "Karnaugh maps"

parity in const depth, poly size ?

no! [Furst Saxe Sipser] & lemon

switching lemma

lemons $\Rightarrow$ lemonade:

<u>Thm</u> [Hastad, Linial Mansour Nisan]

$\forall f$ computable via size s depth d ckts

$$\sum_{|S|>t} \hat{f}^2(S) \le \alpha \quad \text{for} \quad t = O\left(\log \frac{s}{\alpha}\right)^{d-1}$$

take $s = \text{poly}(n)$, $d = \text{const}$, $\alpha = O(\varepsilon)$ $\Rightarrow t = O\left(\log^d\left(\frac{n}{\varepsilon}\right)\right)$

yields $n^{O\left(\log^d\left(\frac{n}{\varepsilon}\right)\right)}$ sample algorithm

(can improve to $n^{O(\log\log n)}$ [Jackson])

(recall parity of S will have 1 large Fourier coeff of degree $|S|$)

3) Learning halfspaces

<u>def</u>. $h(x) = \text{sign}(w \cdot x - \theta)$ is "halfspace function"

$\text{sign}(y) = \begin{cases} +1 & \text{if } y \geq 0 \\ -1 & \text{o.w.} \end{cases}$

<u>Thm</u> Let h be halfspace over $\{\pm 1\}^n$ then h has f.c. $\alpha(\varepsilon) = \frac{c}{\varepsilon^2}$

$$\left(\text{i.e. } \sum_{|S| \geq \frac{c}{\varepsilon^2}} \hat{h}(S)^2 \leq \varepsilon\right)$$

(will prove soon)

<u>Corr</u> low degree alg learns halfspaces under unif dist with $n^{O(1/\varepsilon^2)}$ unif. samples.

(actually $O(n^5)$ sample algorithms exist, but this approach will have "big win" soon)

Key idea:

Noise Sensitivity

← use to bound Fourier Concentration

def. "Noise operator" $0 < \varepsilon < 1/2$

$$N_\varepsilon(x) = \text{randomly flip each bit of } x \text{ with prob } \varepsilon$$

def "Noise sensitivity"

$$NS_\varepsilon(f) = \Pr_{\substack{x \in \{\pm 1\}^n \\ + \\ \text{noise}}} [f(x) \neq f(N_\varepsilon(x))]$$

Examples

1. $f(x) = x_1$ $\qquad ns_\varepsilon(f) = \varepsilon$

2. $f(x) = x_1 x_2 \ldots x_k$ $\qquad ns_\varepsilon(f) = \Pr[f(x) = F \;\&\; f(N_\varepsilon(x)) = T]$
$+ \Pr[f(x) = T \;\&\; f(N_\varepsilon(x)) = F]$

by symmetry → $= 2 \cdot \Pr[f(x) = T \;\&\; f(N_\varepsilon(x)) = F]$

$= \frac{2}{2^k} (1 - (1-\varepsilon)^k)$

if $\varepsilon \ll 1/k$, $\approx \frac{1}{2^{k-1}}(\varepsilon k)$

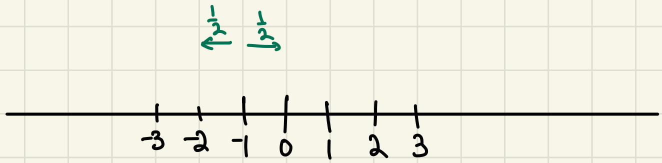
if $\varepsilon \gg 1/k$, $\approx (1 - e^{-k\varepsilon}) / 2^{k-1}$

3. $f(x) = \text{Maj}(x_1 \cdots x_n)$

$$ns_\varepsilon(f) = O(\sqrt{\varepsilon})$$

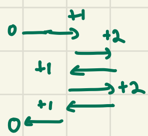
Sketch:

Maj(x) ~ random walk on line starting at 0



e.g. $x = (1\,1\;-1\;1\;-1\,-1)$

$\quad\quad\;\; 1\,2\;\;\;1\;\,2\;\;\,1\;\,0$



well known fact:

$$E[|x_1 + x_2 + \ldots + x_n|] = \sqrt{n}$$

+ likely to be close to $\sqrt{n}$

$N_\varepsilon(x) \sim$ random walk on $\underbrace{\varepsilon n}_{\text{each flip displaces by } \pm 2}$ bits

$(-1 \to +1$ or $+1 \to -1)$

$$E[\text{displacement}] = 2\sqrt{\varepsilon n}$$

Process: take walk specified by x + continue walk according to $N_\varepsilon(x) \cdot 2$

heuristic argument:

pretend first walk leaves us at $\sqrt{n}$

$Pr[2^{nd}$ walk takes us across $0]$

$$= \frac{1}{2} Pr[2^{nd} \text{ displacement} > \underbrace{\sqrt{n}}_{\frac{1}{2\sqrt{\varepsilon}} \cdot 2\sqrt{\varepsilon n}}]$$

$\leq 2\sqrt{\varepsilon}$ by Markov's $\neq$

4. any LTF ($\frac{1}{2}$ space)

<u>Thm</u> (Peres) $NS_\varepsilon(LTF) < 8.8\sqrt{\varepsilon}$

best possible since $NS_\varepsilon(Maj) = \Theta(\sqrt{\varepsilon})$

5. Parity fctns $\chi_S(x)$ for $|S| = k$

$$ns_\varepsilon(f) = Pr[\text{odd \# bits in } S \text{ flipped by } N_\varepsilon]$$
$$= \frac{1-(1-2\varepsilon)^k}{2}$$

for $|S|=1$: ε