Lecture 13:

Learning via Fourier Coeffs

- Some fctns & their Fourier representation

- the low degree algorithm

- applications

Fourier Representations of Important Examples

Two examples

1) $\overline{\text{AND}}$ on $T \subseteq N$ s.t. $|T| = k$

$$\overline{\text{AND}}(x_{i_1} \cdots x_{i_k}) = 1 \quad \text{if } \forall i_j \in T = \{i_1 \cdots i_k\} \quad x_{i_j} = -1$$

$$-1 \quad \text{o.w.}$$

$$\text{define} \quad \underset{\text{AND}}{f}(x) = \begin{cases} 1 & \text{if } \forall i \in T \quad x_i = -1 \\ 0 & \text{o.w.} \end{cases}$$ corresponds to AND fctn over {0,1}

$$= \frac{(1 - x_{i_1})}{2} \cdot \frac{(1 - x_{i_2})}{2} \cdot \cdots \cdot \frac{(1 - x_{i_k})}{2}$$

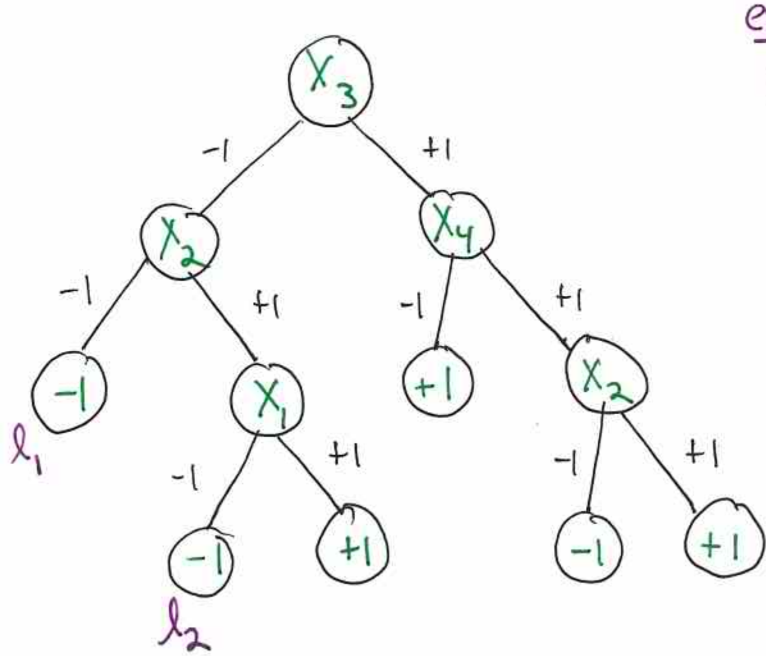
$$= \sum_{S \subseteq T} \frac{(-1)^{|S|}}{2^k} \chi_S$$

& so $\overline{\text{AND}}(x) = 2f(x) - 1$

$$= -1 + \frac{2}{2^{k}} + \sum_{\substack{S \subseteq T \\ |S| > 0}} \frac{(-1)^{|S|}}{2^{k-1}} \chi_S$$

Note: all Fourier coeffs containing vars not in T are 0

2) Decision trees



<u>examples</u>

$$f_{l_1}(x) = \frac{(1-x_3)}{2} \cdot \frac{(1-x_2)}{2}$$

$$f_{l_2}(x) = \frac{(1-x_3)}{2} \frac{(1+x_2)}{2} \frac{(1-x_1)}{2}$$

First, consider path fctns:

$$f_l(x) = \prod_{i \in V_l} \frac{(1 \pm x_i)}{2}$$

left or right

vars visited on path to leaf l

$$= \frac{1}{2^{|V_l|}} \sum_{S \subseteq V_l} (\pm 1) \chi_S$$

$(-1)^{\#\text{ left turns taken in } S}$

$$f_l(x) = \begin{cases} 1 & \text{if } x \text{ takes path } l \\ 0 & \text{o.w.} \end{cases}$$

so $f(x) = \sum_{l \in \text{leaves of } T} f_l(x) \, val(l)$

exactly one of these is 1
others are 0

Comment only coeffs corresponding to S s.t. $|S| \leq$ max path length can be nonzero.

Low degree algorithm

def $f:\{\pm1\}^n \to \mathbb{R}$ 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 $\displaystyle\sum_{\substack{S\subseteq[n] \\ \text{s.t.} \\ |S|\le\alpha(\varepsilon,n)}} \hat{f}(S)^2 \ge 1-\varepsilon$

examples

1) fcfn f which depends on $\le k$ vars

has $\displaystyle\sum_{\substack{S \text{ s.t.} \\ |S|>k}} \hat{f}(S)^2 = 0$

if f doesnt depend on x_i then all $\hat{f}(S)$ for which $i\in S$ satisfy $\hat{f}(S)=0$

2) f = AND on $T\subseteq\{1..n\}$ has $\log\left(\frac{4}{\varepsilon}\right)$-F.C.

- all $\hat{f}(S)^2=0$ for $|S|>|T|$
- if $|T|\le\log\frac{4}{\varepsilon}$ then ✓
- if $|T|\ge\log\frac{4}{\varepsilon}$ then :

$$\hat{f}(\phi)^2 = \left(1-2\Pr\left(f(x)\ne\chi_\phi(x)\right)\right)^2 = \left(1-\frac{2}{2^{|T|}}\right)^2 > 1-\varepsilon$$

so $\displaystyle\sum_{S\ne\phi}\hat{f}(S)^2\le\varepsilon$ + f has 0-F.C. ■

Now, let's approximate fctns with $d \equiv \alpha(\varepsilon, n)$ F.c.:

<u>Low Degree Algorithm</u>

Given d degree
τ accuracy
δ confidence

Algorithm
- Take $m = O\left(\frac{n^d}{\tau} \ln \frac{n^d}{\delta}\right)$ samples
- $C_s \leftarrow$ estimate of $\hat{f}(s)$ (for each S s.t. $|s| \leq d$)

  $\leq \binom{n}{d}$ of these
  Can reuse same samples for each!
- output $h(x) = \sum_{|s| \leq d} C_s \chi_s(x)$

$\boxed{\text{\& use sign}(h(x)) \text{ as hypothesis!}}$

Why does this work?

Two stages:

1) show that if f has low F.C. then $E_x[(f(x) - h(x))^2]$ small

   L_2 dist$/2^n$

2) show that $\Pr[f(x) \neq \text{sign}[h(x)]] \leq E_x[(f(x) - h(x))^2]$

   Hamming dist

Thm if f has $d = \alpha(\varepsilon, n)$ – F.c. then
h satisfies $E_x[(f(x)-h(x))^2] \le \varepsilon + \tau$
with prob $\ge 1-\delta$

Pf

Claim with prob $\ge 1-\delta$, $\forall$ S s.t. $|S| \le d$, $|C_S - \hat{f}(S)| \le \gamma$
for $\gamma \leftarrow \sqrt{\frac{\tau}{n^d}}$

Pf of claim
note, $\frac{1}{\gamma^2} = \frac{n^d}{\tau}$
Chernoff bnd $\Rightarrow O\left(\frac{n^d}{\tau} \ln \frac{n^d}{\delta}\right) = O\left(\frac{1}{\gamma^2} \ln \frac{n^d}{\delta}\right)$ samples
yields $\Pr\left[\,|C_S - \hat{f}(S)| > \gamma\right] < \frac{\delta}{n^d}$

union bnd $\Rightarrow \Pr[\exists\, S$ s.t. $|C_S - \hat{f}(S)| > \gamma] < \delta$ ∎

only $\binom{n}{d} < n^d$ such
S's of size $\le d$

Assume $\forall$ S s.t. $|S| \le d$, $|C_S - \hat{f}(S)| \le \gamma$

define $g(x) \equiv f(x) - h(x)$
Fourier transform is linear $\Rightarrow \forall S$ $\hat{g}(S) = \hat{f}(S) - \hat{h}(S)$
by defn, $\forall S$ s.t. $|S| > d$, $\hat{h}(S) = 0$ $\Rightarrow \hat{g}(S) = \hat{f}(S)$
$|S| \le d$, $\hat{h}(S) = C_S$ $\Rightarrow \hat{g}(S) = \hat{f}(S) - C_S$
so $\hat{g}(S)^2 \le \gamma^2$

so $E[(f(x)-h(x))^2] = E[g(x)^2]$

$$= \sum_S \hat{g}(S)^2 \qquad \text{Parseval}$$

$$= \underbrace{\sum_{|S|\le d} \underbrace{\hat{g}(S)^2}_{\le \gamma^2}}_{\le n^d \cdot \gamma^2} + \underbrace{\sum_{|S|>d} \hat{g}(S)^2}_{\le \varepsilon \text{ by F.C.}}$$

$$\le \tau + \varepsilon \qquad \blacksquare$$

<u>Thm</u> $f : \{\pm 1\}^n \to \{\pm 1\}$

$h : \{\pm 1\}^n \to \mathbb{R}$

then $\Pr[f(x) \ne \text{sign}(h(x))] \le E[(f(x)-h(x))^2]$

<u>Pf.</u> $E[(f(x)-h(x))^2] = \frac{1}{2^n} \sum_x (f(x)-h(x))^2$ defn

$\Pr[f(x) \ne \text{sign}(h(x))] = \frac{1}{2^n} \sum_x \mathbb{1}_{\{f(x) \ne \text{sign}(h(x))\}}$

show term by term

But if $f(x) = \text{sign}(h(x))$

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

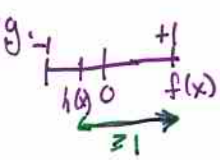
$\mathbb{1}_{f(x) \ne \text{sign}(h(x))} = 0$

if $f(x) \ne \text{sign}(h(x))$

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

$\mathbb{1}_{f(x) \ne \text{sign}(h(x))} = 1$

eg.



So $\forall x$, $(f(x)-h(x))^2 \ge \mathbb{1}_{f(x) \ne \text{sign}(h(x))}$ $\blacksquare$

Correctness of learning algorithm:

<u>Thm.</u> if $\mathcal{C}$ has Fourier concentration $d = \alpha(\varepsilon, n)$
then there is a $q = O(\frac{n^d}{\varepsilon} \log \frac{n^d}{\delta})$ sample
uniform distribution learning algorithm for $\mathcal{C}$
i.e. algorithm gets q samples & with prob $\geq 1-\delta$
outputs h' s.t. $\Pr[f \neq h'] \leq 2\varepsilon$

<u>Pf.</u> run low degree alg with $\gamma = \varepsilon$
get h s.t. $E[(f-h)^2] \leq \varepsilon + \varepsilon = 2\varepsilon$
output $\text{sign}(h)$ ∎

<u>Applications</u>

1) Bounded depth decision trees

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

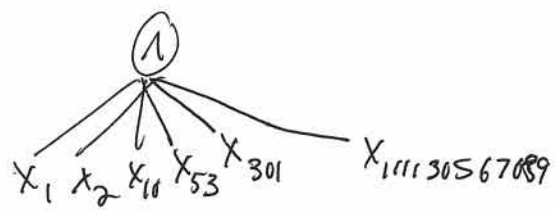
by linearity, $\hat{f}(S) = \sum \text{val}(\ell) \cdot \hat{f}_\ell(S)$ which is 0 if $|S| > \text{depth}$

2) Constant depth ckts:

<u>Def.</u> "Boolean Ckt C" is a DAG

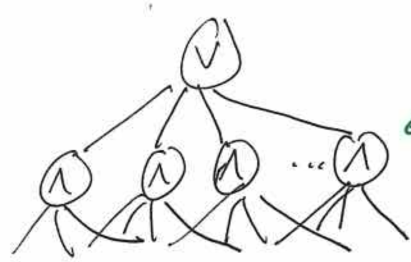
gates: $\underbrace{\wedge, \vee, \neg}_{\text{and, or, not}}$, $\underbrace{1, 0}_{\text{or } \pm 1}$, $\underbrace{x_1 \cdots x_n}_{\text{vars}}$

how many inputs?
const? poly? unbounded?



can we compute parity of n bits in const depth?

yes! can compute any n-bit fctn in constdepth



each "∧" picks an arbitrary sat setting & checks if input matches

can we compute parity of n bits in const depth, poly size?

No! [Furst Saxe Sipser] } lemon
Switching lemma

Lemons ⇒ Lemonade

Thm [Hastad, Linial Mansour Nisan]

prove via random restrictions as in [FSS] parity result

Take Advanced Complexity!

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

$$\sum_{|S|>t} \hat{f}^2(S) \le \alpha \quad \text{for} \quad t = O\left(14 \log \frac{2s}{\alpha}\right)^{d-1}$$

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

Gives $n^{O(\log^d(\frac{n}{\varepsilon}))}$ sample query algorithm

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