Lecture 17

Fourier-based learning algorithms

- Fourier Concentration via Noise sensitivity
- Learning heavy Fourier coeffs (with queries)

Recall Fourier Transform:

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

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

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

$$= 2 \cdot \Pr[f(x) = \chi_S(x)] - 1$$

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

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

Parseval's for Boolean $f : \{\pm 1\}^n \to \{\pm 1\}$

$$1 = \langle f, f \rangle = \sum_S \hat{f}(S)^2$$

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:

lemma for any $S \subseteq [n]$, can approx $\hat{f}(S)$ to within additive γ (i.e. $|\text{output} - \hat{f}(S)| \leq \gamma$) with prob $\geq 1-\delta$ in $O(\frac{1}{\gamma^2} \log \frac{1}{\delta})$ 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 $\sum_{\substack{S \subseteq [n] \\ \text{s.t.} \\ |S| > \alpha(\varepsilon,n)}} \hat{f}(S)^2 \leq \varepsilon \quad \forall \ 0 < \varepsilon < 1$

↑ for Boolean f, this implies

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

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

then there is a $q = O\left(\frac{n^d}{\varepsilon} \log \frac{n}{\delta}^d\right)$ sample uniform distribution learning algorithm for $\mathcal{C}$

ie. algorithm gets q samples $+$ with prob $\geq 1-\delta$ outputs h' s.t. $\Pr[f \neq h'] \leq 2\varepsilon$

Applications

1) Bounded depth decision trees

2) Const depth ckts

3) halfspaces (linear threshold fctns)

Key idea:

Noise Sensitivity

← use to bound Fourier concentration

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

$N_\varepsilon(x)$ = randomly flip each bit of x with prob ε

def "Noise sensitivity"

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

Examples

1. $f(x) = x_1$ $\quad NS_\varepsilon(f) = \varepsilon$
2. $f(x) = x_1 x_2 \dots x_k$ $\quad NS_\varepsilon(f) = \frac{2}{2^k}(1-(1-\varepsilon)^k)$
3. $f(x) = Maj(x_1 \cdots x_n)$ $\quad NS_\varepsilon(f) = O(\sqrt{\varepsilon})$
4. $f(x)$ is any LTF $\quad NS_\varepsilon(f) \leq 8.8\sqrt{\varepsilon}$
5. Parity fctns $\quad NS_\varepsilon(f) = \Pr[\text{odd \# bits flipped by noise}] = \frac{1-(1-2\varepsilon)^k}{2}$

(End review)

6. Any f

Thm $f: \{\pm 1\}^n \to \{\pm 1\}$

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

for parity fctns: χ_S $\quad \frac{1}{2} - \frac{1}{2}(1-2\varepsilon)^{|S|}$

Pf. homework?

Noise Sensitivity vs. Fourier Concentration

Thm $\forall f : \{\pm 1\}^n \to \{\pm 1\}$ $\quad 0 < \gamma < 1/2$

$$\sum_{|S| \geq \frac{1}{\gamma}} \hat{f}(S)^2 < 2.32\, NS_\gamma(f)$$

Pf $2 \cdot NS_\gamma(f) = 1 - \sum_S (1-2\gamma)^{|S|} \hat{f}(S)^2$ previous thm

$$= \sum_S \hat{f}(S)^2 - \sum_S (1-2\gamma)^{|S|} \hat{f}(S)^2$$ Boolean Parseval

$$= \sum_S \left[1 - (1-2\gamma)^{|S|}\right] \hat{f}(S)^2$$

$$\geq \sum_{S \text{ s.t. } |S| \geq 1/\gamma} \left[1 - (1-2\gamma)^{1/\gamma}\right] \hat{f}(S)^2$$

$$> \sum_{|S| \geq 1/\gamma} (1 - e^{-2})\, \hat{f}(S)^2$$

So $$\sum_{|S| \geq 1/\gamma} \hat{f}(S)^2 < \underbrace{\left(\frac{2}{1-e^{-2}}\right)}_{2.32} \cdot NS_\gamma(f)$$

∎

Corr for halfspace $h : \{\pm 1\}^n \to \{\pm 1\}$

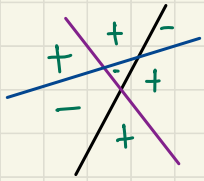
$$\sum_{|S| \geq O(1/\varepsilon^2)} \hat{f}(S) \leq \varepsilon$$

(pf omitted - some calculations + bound on NS)

$\Rightarrow$ can learn any halfspace from $n^{O(1/\varepsilon^2)}$ random examples

(actually can do a lot better)

Corr any function of k halfspaces can be learned with $n^{O(k^2/\varepsilon^2)}$ samples

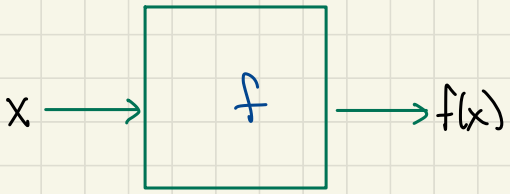


e.g. parity of k vars, $\wedge$ of $\frac{1}{2}$ spaces

Pf idea noise sensitivity $\leq 8.8k\sqrt{\varepsilon}$ by union bound.

Learning Heavy Fourier Coeffs

[Goldreich Levin]
[Kushilevitz Mansour]



not just low degree S
↓
all close linear fctns

Given f, θ

- Output all coeffs S s.t. $|\hat{f}(S)| \geq \theta$
- Only output S s.t. $|\hat{f}(S)| \geq \frac{\theta}{2}$ ← no junk

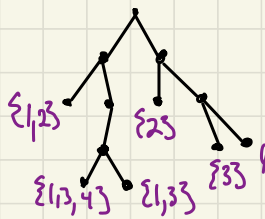
Probably can't do it with only random examples

What if can query f at any input?

Observation if there is a really big $\hat{f}(S)$, (say $> 3/4$) can use self-correction to find it

Main Idea: "exhaustive search with good pruning"

leaves ~ Fourier coeffs



Var
x_1
x_2
x_3
x_4

left $\Rightarrow x_i \in S$
right $\Rightarrow x_i \notin S$

ONLY OUTPUT THOSE THAT REACH BOTTOM LEVEL

recursive algorithm:

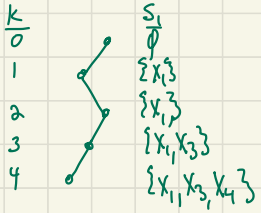
- each node ~ setting of $x_1 \cdots x_i$
- estimate "total energy" of subtrees $x_1 \cdots x_i \circ (x_{i+1} = +1)$ & $x_1 \cdots x_i \circ (x_{i+1} = -1)$
- only go down paths with high enough energy

How to prune?

Define quantity:

Fix $0 \le k \le n$ — current "level" of search

$S_1 \subseteq [k]$ — current "node" of search



doesn't depend on 1st K bits

$f_{K,S_1} : \{\pm 1\}^{n-K} \to \mathbb{R}$

2^K such fctns (for each S_1)

all Fourier coeffs which agree on first K elements

s.t. $f_{K,S_1}(x) = \sum_{T_2 \subseteq \{K+1..n\}} \hat{f}(S_1 \cup T_2)\, \chi_{T_2}(x)$

all extensions of S_1 to indices in $\{k+1 \dots n\}$

could be $S_1 \cup T_2$ but no need since $\chi_{S_1 \cup T_2} = \chi_{S_1} \cdot \chi_{T_2}$ (χ_{S_1}: same for all)

notation: index 1 → prefix, 2 → suffix

Sanity Checks!

1) $K=0$

$$f_{0,\emptyset}(x) = \sum_{T_2 \subseteq [n]} \hat{f}(T_2)\, \chi_{T_2}(x) = f(x)$$

since $k=0$; since $S_1 = \emptyset$

2) $K=n$

$$f_{n,S_1}(x) = \hat{f}(S_1) \cdot \chi_\emptyset(x)$$

$\chi_\emptyset(x) = 1$; since $T_2 = \emptyset$; sum over $T_2 = \emptyset$

Plan Only go down paths with $E_{K,S_1}[f^2(x)] \geq \theta^2$

1. can we compute it?
2. does it bring us to right leaves?
   - do we get to all heavy leaves?
   - do we get junk? (light leaves)
3. how many paths do we take?

   lots of dead ends?
   is runtime good?

Not too many paths! (answer to 3)

Lemma "not too many" ← at any stage in algorithm

$f : \{\pm 1\}^n \to \{\pm 1\}$

(1) $\leq \frac{1}{\theta^2}$ S's satisfy $|\hat{f}(S)| \geq \theta$

(2) $\forall\ 0 \leq k \leq n$, $\leq \frac{1}{\theta^2}$ fctns f_{k,S_1}

have $E_x[f^2_{k,S_1}(x)] \geq \theta^2$

Pf

(1) Boolean Parseval's $1 = \sum_S \hat{f}(S)^2$

so if $> \frac{1}{\theta^2}$ S's satisfy $|\hat{f}(S)| \geq \theta$

then $\sum_S \hat{f}(S)^2 > \frac{1}{\theta^2} \cdot \theta^2 > 1$

$\to\leftarrow$

(2) For given k:

Claim: $\forall\ k,\ S_1 \subseteq k$

$$E_x\left[f^2_{k,S_1}(x)\right] = \sum_{T_2 \subseteq \{k+1,\dots,n\}} \hat{f}(S_1 \cup T_2)^2$$

pf of claim:

$$E_x\left[f_{k,S_1}(x)^2\right] = E_x\left[\left(\sum_{T_2} \hat{f}(S_1 \cup T_2)\chi_{T_2}(x)\right)^2\right] \quad \text{def.}$$

$$= E_x\left[\sum_{\substack{T_2, T_2' \\ \subseteq \{k+1\dots n\}}} \hat{f}(S_1 \cup T_2)\cdot\hat{f}(S_1 \cup T_2')\,\chi_{T_2}(x)\chi_{T_2'}(x)\right]$$

$$= \sum_{T_2, T_2'} \hat{f}(S_1 \cup T_2)\hat{f}(S_1 \cup T_2')\, E\left[\chi_{T_2}(x)\cdot\chi_{T_2'}(x)\right]$$

$= 1$ if $T_2 = T_2'$

$= 0$ o.w.

$$= \sum_{T_2} \hat{f}(S_1 \cup T_2)^2$$

∎

Using Claim:

Boolean Parseval's ← $1 = \sum_S \hat{f}(S)^2 =$ ← expanding $\sum_{S_1 \subseteq K} \sum_{T_2 \subseteq \{K+1,\dots,n\}} \hat{f}(S_1 \cup T_2)^2$

$$= \sum_{S_1} E_x\left[f^2_{K,S_1}(x)\right] \qquad \text{claim}$$

so $\leq \frac{1}{\theta^2}$ S_1's can have $E_x\left[f^2_{K,S_1}(x)\right] > \theta^2$

∎

Does algorithm bring us to good leaves?

(answer to 2)

Fact: "not missing out" $\Rightarrow$ find all big Fourier coeffs

For any s_1, if $\exists\ T_2$ s.t.

$$\left|\hat{f}(s_1 \cup T_2)\right| > \theta$$

then $E_x\left[f^2_{k,s_1}(x)\right] = \sum_{T_2} \hat{f}(s_1 \cup T_2)^2 \geq \theta^2$ via claim

$\Rightarrow$ $E_x\left[f^2_{k,s_1}(x)\right]$ is a good measure to use when deciding whether to investigate subtree!

So we find all good leaves
we don't spend too much time

but do we output junk?

No junk (answer to 2)

Simple fix:

For each "candidate" to S,

estimate it's Fourier coeff & make sure it is big enough before outputting it

recall:

$f: \{\pm 1\}^n \to \{\pm 1\}$

$0 \le k \le n$

$S_1 \subseteq [k]$

Can we estimate $f_{k,S_1}(x)$?

(answer to 1)

$$f_{k,S_1}(x) = \sum_{T_2 \subseteq \{k+1..n\}} \hat{f}(S_1 \cup T_2)\, \chi_{T_2}(x)$$

$$E_x[f_{k,S_1}(x)^2] = \sum_{T_2 \subseteq \{k+1,\dots,n\}} \hat{f}(S_1 \cup T_2)^2$$

Bad idea: estimate each $\hat{f}(S_1 \cup T_2)$ $\forall T_2$ ← too much time

Another challenge:

heavy $\hat{f}(S_1 \cup T_2)$ can differ greatly depending on how you fix vars in S_1

e.g. $S_1 = \{x_1\}$

$$f(x) = \begin{cases} 1 & \text{if } x_1 = 1 \\ \prod_i x_i & \text{if } x_1 = -1 \end{cases}$$

recall:

$f: \{\pm 1\}^n \to \{\pm 1\}$

$0 \le k \le n$

$S_1 \subseteq [k]$

Can we estimate $f_{k,S_1}(x)$?

(answer to 1)

$$f_{k,S_1}(x) = \sum_{T_2 \subseteq \{k+1..n\}} \hat{f}(S_1 \cup T_2)\, \chi_{T_2}(x)$$

$$E_x\left[f_{k,S_1}(x)^2\right] = \sum_{T_2 \subseteq \{k+1,\dots,n\}} \hat{f}(S_1 \cup T_2)^2$$

Lemma "$f_{k,S_1}(x)$ Estimation lemma"

for $x \in \{\pm 1\}^{n-k}$

$$f_{k,S_1}(x) = E_{y \in \{\pm 1\}^k}\left[f(yx)\chi_{S_1}(y)\right]$$

("agreement"; yx: concatenation)

Use this to estimate $f_{k,S_1}(x)$

Pf

Fourier representation $\Rightarrow$ $f(yx) = \sum_T \hat{f}(T)\chi_T(yx)$

think of as fctn of y since x is fixed throughout

$T = T_1 \cup T_2 \quad T_1 \subseteq [k] \quad T_2 \subseteq \{k+1..n\}$

so $\chi_T(yx) = \chi_{T_1}(y) \cdot \chi_{T_2}(x)$

$$E_y[f(y|x)\chi_{S_1}(y)]$$

$$= E_y\Big[\sum_{T_1}\sum_{T_2}\hat{f}(T_1 \cup T_2)\chi_{T_1}(y)\chi_{T_2}(x)\cdot\chi_{S_1}(y)\Big] \quad \text{above}$$

$$= \sum_{T_1}\sum_{T_2}\hat{f}(T_1 \cup T_2)\chi_{T_2}(x)\underbrace{E_y[\chi_{T_1}(y)\chi_{S_1}(y)]}_{= \begin{cases} 0 & \text{if } T_1 \neq S_1 \\ 1 & \text{if } T_1 = S_1 \end{cases}}$$

$$= \sum_{T_2}\hat{f}(S_1 \cup T_2)\chi_{T_2}(x)$$

$$= f_{\to S_1}(x)$$

∎

Overall Algorithm:

Pick random x's $\in \{\pm 1\}^{n-k}$

For each x, pick random y's $\in \{\pm 1\}^k$

Estimate $E_y[f(yx)\chi_{S_1}(y)]$

which gives estimate of

$f_{k,S_1}(x)$

Estimate $E_x[f_{k,S_1}(x)^2]$

Chernoff + samples

$\Rightarrow$ Can get γ-additive estimate with prob $\geq 1-\delta$ in $O(\frac{1}{\gamma^2}\log\frac{1}{\delta}$ queries)

$\Rightarrow$ Can get $\frac{\gamma^2}{2}$-additive est of $f^2_{k,S_1}(x)$

Thm $\forall \theta > 0$, KM-alg outputs

$S = \{s_1 .. s_\ell\}$ s.t. $\ell = O(1/\theta^2)$ & with prob $\geq 1-\delta$

$$\forall s_i \in S \quad |\hat{f}(s_i)| \geq \frac{\theta}{2} \qquad \text{no junk}$$

$$\forall s \notin S \quad |\hat{f}(s)| \leq \theta \qquad \text{no misses}$$

& query time is poly $(n, \frac{1}{\theta}, \log \frac{1}{\delta})$

Pf. if $\hat{f}(s) < \frac{\theta}{2}$, $\hat{f}(s)^2 \leq \frac{\theta^2}{4}$

& test * prevents it from being output

if $\hat{f}(s) > \theta$, $\forall k$ if s_1 & s agree on $[1..k]$

"not missing out" fact $\Rightarrow$

$$E_x[f^2_{k,s_1}(x)] \geq \theta^2$$

total # nodes explored at level $k \leq \frac{1}{\theta^2}$

$\Rightarrow$ total nodes explored $\leq \frac{n}{\theta^2}$

■