Lecture 14

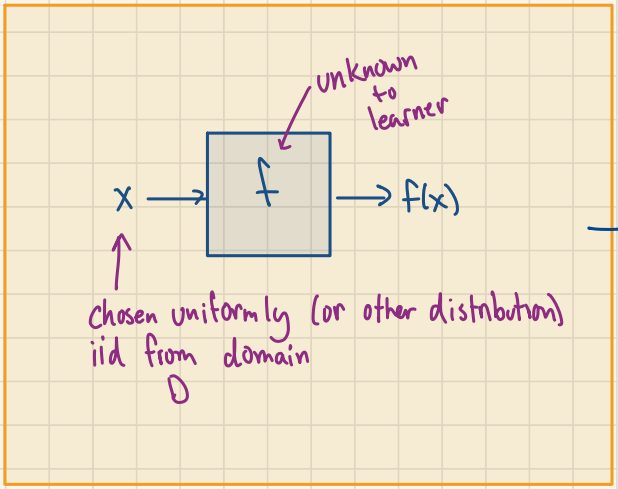
learning Boolean functions

- a model
- an example: conjunctions
- Occam's razor
- Fourier-based learning algorithms

Learning

Learn from random, uniform examples ← How to formalize?

lots of other models – will mention some others later



Example oracle $Ex(f)$

After seeing several examples, learner should output hypothesis h.

what do we hope h satisfies?

- hopefully $h = f$ ← asking too much?
- at least $\text{dist}(h,f) \leq \varepsilon$

$$\text{dist}(h,f) = \Pr_{x \in D}[h(x) \neq f(x)]$$

in what distribution on inputs do we use?

today: uniform

in general, match distribution of example oracle

Valiant's PAC model "probably approximately correct"

Common terms for same thing:

- $\text{dist}_{x \in D}(h,f)$
- $\text{error}_{x \in D}(h)$ (w.r.t. f is understood from context)
- h is ε-close to f (dist D understood from context)

Note in above:

$x \in D$ can be chosen according to uniform or any other prespecified distribution

Note if f is arbitrary, there is nothing you can do that is "efficient" in terms of sample complexity

(e.g. you can't learn a random fctn f without seeing the value of f for most inputs)

However, if you know something about f, there may be hope.

here: what if you know that f is a member of fctn family C?

e.g. C = linear fctns
k-term DNF
⋮

def uniform distribution learning algorithm for concept class $\mathcal{C}$ is algorithm $\mathcal{A}$ s.t.

- $\mathcal{A}$ given $\varepsilon, \delta > 0$
  access to $Ex(f)$ for $f \in \mathcal{C}$
- $\mathcal{A}$ outputs h s.t. with prob $\geq 1-\delta$
  error (h) w.r.t. f is $\leq \varepsilon$
  (h is ε-close to f)

Parameters of interest:

- m — # samples used by $\mathcal{A}$ "sample complexity"
- ε — accuracy parameter
- δ — confidence parameter
- runtime? — hope for poly $(\log(\text{domainsize}), \frac{1}{\varepsilon}, \frac{1}{\delta})$

- description of h?
  - should it be similar to description of fctns in $\mathcal{C}$? "proper learning"
  - at least should be relatively compact + efficient to evaluate
    compact: $O(\log|\mathcal{C}|)$

Remarks

- as before, dependence on δ needn't be more than $O(\log(1/\delta))$ why?

- uniform case is special case of PAC-model:
  given $EX_{\mathcal{D}}(f)$ for unknown $\mathcal{D}$
  output h with small error with respect to same $\mathcal{D}$
  (some $\mathcal{D}$ can be harder than others)

Efficient learning algorithm for conjunctions:

C = conjunctions over $\{0,1\}^n$

ie. $f(x) = x_i x_j \bar{x}_k$

note:

- can't hope for 0-error from subexponential # of random examples

  e.g. how to distinguish $f(x) = x_1 x_2 \cdots x_n$ from $f'(x) = 0 \ \forall x$?

Poly time algorithm:

- draw $poly(\frac{1}{\varepsilon})$ random examples
- estimate $Pr[f(x)=1]$ to additive error $\pm \frac{\varepsilon}{4}$
- if estimate $< \varepsilon/2$, output "$h(x)=0$" & halt

  else estimate $\geq \varepsilon/2$ so $Pr[f(x)=1] > \frac{\varepsilon}{4}$

  $\Rightarrow$ see new random positive example every (at most) $O(1/\varepsilon)$ examples
- collect $\underline{?}$ more positive examples
- let $V = \{$vars set same way in each positive example$\}$
- Output $h(x) = \bigwedge_{i \in V} x_i^{b_i}$ ← b_i tells us if x_i complemented or not

Behavior of poly time algorithm:

for i in conjunction:

must be set same way in each positive example $\Rightarrow$ in V

for i <u>not</u> in conjunction:

$$\Pr[i \in V] \leq \Pr[i \text{ set same way in each of } k \text{ positive examples}]$$

$$\leq \frac{1}{2^{k-1}}$$

$$\Pr[\text{any } i \text{ } \underline{\text{not}} \text{ in conjunction manages to survive}]$$

$$\leq \frac{n}{2^{k-1}}$$

$$\leq \delta \text{ if pick } k = \log \frac{n}{\delta}$$

So if use $\Omega(\log \frac{n}{\delta})$ positive examples

or $\Omega(\frac{1}{\varepsilon} \log \frac{n}{\delta})$ total examples, will suffice to rule out all $i \notin$ conjunction.

Occam's Razor

"high level claim":

if ignore runtime, then learning is easy

with respect to sample complexity

Brute force algorithm

- draw $M = \frac{1}{\varepsilon}\left(\ln|\mathcal{C}| + \ln\frac{1}{\delta}\right)$ uniform examples
- search over all $h \in \mathcal{C}$ until find one consistent with all examples.

  Output it. (choose arbitrarily if ≥ 1 such h works)

Representation of $\mathcal{C}$: (if computation not issue)

$c \in \mathcal{C}$ needs $\sim \ln|\mathcal{C}|$ bits

Behavior of brute force algorithm

what should behavior be?

- f is a good thing to output ✓
- what is a bad thing to output?

h is "bad" if error(h) w.r.t. $f \geq \varepsilon$

$\Pr[\text{bad } h \text{ consistent with examples}]$

$$\leq (1-\varepsilon)^M$$

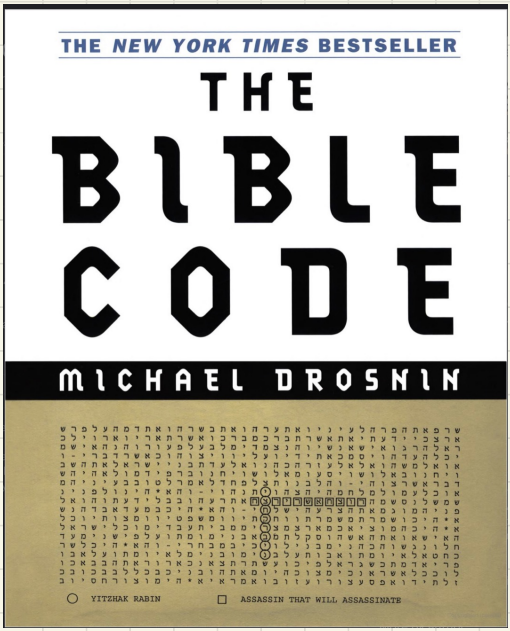
$\Pr[\underline{\text{any}} \text{ bad } h \text{ consistent with examples}]$

$$\leq |\mathcal{C}| \cdot (1-\varepsilon)^M \qquad \text{union bnd}$$

$$\leq |\mathcal{C}| \, (1-\varepsilon)^{\frac{1}{\varepsilon}\left(\ln |\mathcal{C}| + \ln \frac{1}{\delta}\right)}$$

$$\leq \delta$$

$\therefore$ unlikely to output any bad h



divine inspiration?

Coincidence?

not enough samples to kill off union bound?

Comments:

- proof didn't use anything special about uniform distribution

  actually works for any dist $\mathcal{D}$ as long as error defined w.r.t. same $\mathcal{D}$ as sample generator

- Once have good h

  1) can predict values of f on new random (according to $\mathcal{D}$) inputs since

  $$\Pr_{x \in \mathcal{D}}[f(x) = h(x)] \geq 1 - \delta$$

  2) can compress description of samples

  $(x_1, f(x_1))\ (x_2, f(x_2)), \ldots (x_m, f(x_m))$ $\quad$ $m(\log|D| + \log|R|)$ (range of f)

  $\Downarrow$ $\quad$ $\Downarrow$

  $x_1 \ldots x_m$, description of h $\quad$ $m \cdot \log|D| + \log|C|$

So learning, prediction & compression are related

learning $\Rightarrow$ prediction & compression

formal relations in other direction too

Occam's razor:

Simplest explanation is best

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!

Pf. Chernoff + $\hat{f}(S) = 2 \Pr_x [f(x) = \chi_S(x)] - 1$

estimate this

■

Can we find any or all heavy coefficients?

there are exponentially many coeffs.

Can use same samples to estimate each coeff, but must union bound prob of error (error = bad approx) on any of them.

Need $\delta \ll \frac{1}{2^n}$, which needs $O(\frac{1}{\gamma^2} \cdot n)$ samples, but exponential runtime. ← turns out that queries help a lot

What if we "know where to look" for heavy coeffs?

e.g. all heavy coeffs are in "low degree" coeffs? If so, can search!

Fourier Representations of Important Examples

$\underset{T}{AND}(x)$ = "all x_i for $i \in T$ are set to -1"

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

$$\overline{AND}(x) = \begin{cases} 1 & \text{if } \forall i_j \in T = \{i_1 \cdots i_k\} \\ & \quad x_{i_j} = -1 \\ -1 & \text{o.w.} \end{cases}$$

$$f(x) = \begin{cases} 1 & \text{if } \forall i \in T \; x_i = -1 \\ 0 & \text{o.w.} \end{cases}$$ } AND over range $\{0,1\}$

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

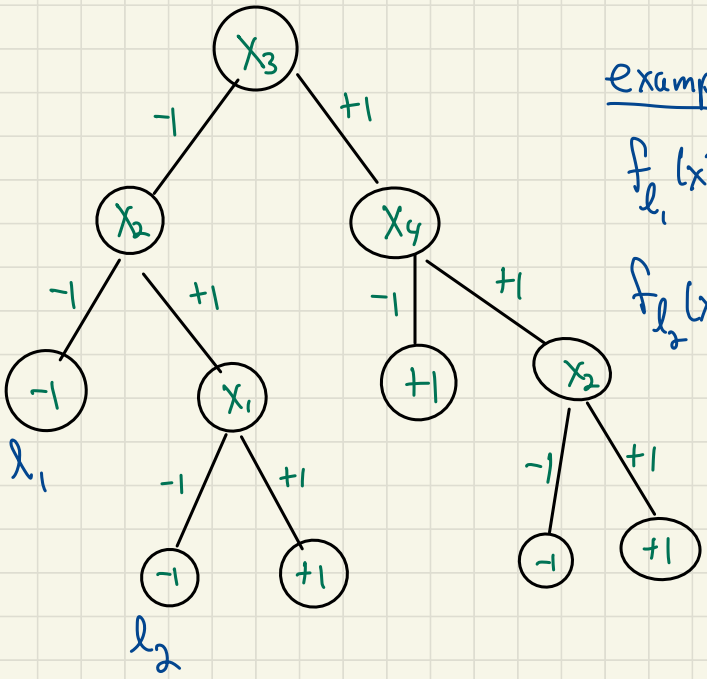
$$= \sum_{S \subseteq T} \frac{(-1)^{|S|}}{2^k} \chi_S$$

$$\overline{AND} = 2f(x) - 1 = -1 + \frac{2}{2^{|T|}} + \sum_{\substack{S \subseteq T \\ S \neq \emptyset}} \frac{(-1)^{|S|}}{2^{k-1}} \chi_S$$

($\chi_\emptyset$)

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

2) Decision trees



examples

$$f_{\ell_1}(x) = \frac{(1-x_3)}{2} \cdot \frac{(1-x_2)}{2}$$

$$f_{\ell_2}(x) = \frac{(1-x_3)}{2} \cdot \frac{(1+x_2)}{2} \cdot \frac{(1-x_1)}{2}$$

First consider path functions:

$$f_\ell(x) = \prod_{i \in V_\ell} \frac{(1 \pm x_i)}{2}$$

(left or right)

(V_ℓ: vars visited on path to leaf ℓ)

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

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

$$= \begin{cases} 1 & \text{if } x \text{ takes path to } \ell \\ 0 & \text{o.w.} \end{cases}$$

exactly one of these is 1. all others are 0.

So $f(x) = \sum_{\ell \in \text{leaves of } T} f_\ell(x) \cdot \text{val}(\ell)$

Comment only coeffs corresponding to S st. $|S| \leq$ max path length have a hope of being non-zero.