Today's lecture:

The PAC learning model
motivation-definition
Occam's razor
Learning conjunctions
(if time: begin learning via Fourier representation)

Learning how to formalize?

Example oracle $\text{Ex}(f)$

Goal: output $f$ is too hard?
output $h$ s.t. $f$ is $\epsilon$-close to $h$

$\Pr_{x \in \Omega} \left[ f(x) = h(x) \right] \geq 3 - \epsilon$
def given hypothesis $h$, error of $h$ with respect to $f$ is $\text{error}(h) = \Pr_{x \sim D} [f(x) \neq h(x)]$

$f$ is $\varepsilon$-close to $h$ wrt. uniform on $D$

Observe if $f$ arbitrary then nontrivial learning is impossible

What if $f$ is in a class of functions $C$?

def uniform distribution learning algorithm for concept class $C$ is algorithm $A$ st.

$A$ is given $\varepsilon, \delta \Rightarrow$ access to $E_x(f)$ for $f \in C$

$A$ outputs $h$ st. with prob $\geq 1 - \delta$

error($h$) wrt. $f$ is $\leq \varepsilon$ according to $f$

$h$ is $\varepsilon$-close to $f$
Parameters of interest

- $m$ # samples used by $A$ "sample complexity"
- $\epsilon$ accuracy parameter
- $\delta$ confidence parameter
- runtime hope for $\text{poly} \left( \log \text{ (domain size)}, \frac{\epsilon}{\delta} \right)$
- description of $h$: $|C|$
  - similar to description of all $f \in \mathcal{C}$?
    - (proper learning)
  - at least should be "compact"
    - $O(\log |C|)$ efficient to evaluate

Remarks

- dependence on $\delta$ needn't be more than
  - $O \left( \log \left( \frac{1}{\delta} \right) \right)$
- uniform dist is a special case
Occam's Razor

learning is easy!

wrt sample complexity

not runtime

brute force algorithm

* draw \( M = \frac{1}{\varepsilon} \left( \ln |C| + \ln \frac{1}{\delta} \right) \) samples

* search over all \( h \in C \) until

  find one that labels all examples correctly. Output \( h \).

  (choose arbitrarily if \( \varepsilon > 1 \))

behavior:

examples come from \( f \in C \)

good to output \( f \)

bad to output \( h \) s.t.

\( h \neq f \) not \( \varepsilon \)-close
$h$ is "bad" if error $(h)$ wrt $f \geq \varepsilon$

\[
\Pr[\text{bad } h \text{ consistent with examples}] 
\leq (1-\varepsilon)^M
\]

\[
\Pr[\text{any bad } h \text{ consistent with examples}]
\leq |C|^M (1-\varepsilon)^M \quad \text{union bound}
\]

\[
\leq |C|^M \cdot \frac{1}{e} \left( \ln |C| + \frac{1}{8} \right)
\]

\[
\leq 8 
\]

\[
\Rightarrow \text{ unlikely to output any bad } h
\]

Proof applies to learning under any distribution
Once we have a good hypothesis \( h \):

1) can predict values of \( f \) on new random inputs \( \Pr_{x \in \mathcal{X}}[f(x) = h(x)] \geq 1 - \varepsilon \)

2) can compress description of samples

\[(x_1, f(x_1)), (x_2, f(x_2)), \ldots, (x_m, f(x_m)) \mid (\log |D| - \log |R|)
\]

\[\downarrow\]

\[x_1 \ldots x_m, \text{description of } h \mid m \log |D| + \log |C|\]

\[\text{learning } \Rightarrow \text{ prediction } \Rightarrow \text{ compression}\]

Occam's Razor: simplest explanation is best
An efficient learning algorithm

\[ C = \text{conjunctions over } \{0,1\}^n \]

i.e. \( f(x) = x_1 \land \neg x_3 \land (x_1 \lor x_3) \)

Observe: how to distinguish

\[ f(x) = x_1 \land \cdots \land x_n \]

from \( f(x) = 0 \)

satisfying \( \sum \text{need } \approx 2^n \text{ samples} \)

\( \Rightarrow \) can't hope for poly time \( \pm \) 0-error

Brute force algorithm: (i.e., alg in Occam's razor)

try each \( f \in C \)

\( |C| \approx 2^n \)

union bound \( \Rightarrow \) need \( \Omega \left( \frac{1}{\varepsilon^2} \ln 2^n + \ln \frac{1}{\delta} \right) \) samples

Poly time algorithm

Simplifying assumption:

Assume \( \Pr_{x \in \{0,1\}^n}[f(x) = 1] > \varepsilon \) in expectation

\( \Rightarrow \) in a sample of size \( m \)

\( \geq \varepsilon m \) many "positive" examples
Algorithm:

Take $N$ examples, $K$ of which are "positive" $\mathcal{P}(W) = 1$

let $V = \exists$ vars set same way in each positive example $3$

$V = \{1, 2, 3\}$

output $h(x) = \bigwedge_{i \in V} x_i$

$h(x) = \overline{x_1 \lor x_2}$

Behavior:

$f(x) = \overline{x}$

for $i$ in conjunction:

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

for $i$ in conjunction:

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

$\leq \frac{1}{2^K} + \frac{1}{2^K} = \frac{1}{2^{K-1}}$

$Pr[\text{any } i \text{ not in conjunction survives}]$

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

$\leq 8$ if pick $K = \log_2 \frac{n}{8}$
\[ N \left( \log \frac{1}{\delta} \right) \] positive examples
\[ \Rightarrow \quad N \left( \frac{1}{2} \log \frac{1}{\delta} \right) \] total examples suffice.

More general algorithm:

Using \( \text{poly}(\frac{1}{\delta}) \) samples

- estimate \( \Pr[f(\omega) = 1] \) to additive error \( \pm \frac{\epsilon}{q} \)
- if estimate \( \leq \frac{\epsilon}{2} \), output \( h = 0 \)

\[ \Rightarrow \quad \Pr[f(\omega) = 1] \geq \frac{\epsilon}{2} + \frac{\epsilon}{q} > \frac{3}{q} \]

... good answer

0.o., continue

\[ \Rightarrow \quad \Pr[f(\omega) = 1] \geq \frac{\epsilon}{2} - \frac{\epsilon}{q} \geq \frac{\epsilon}{q} \]

\[ \Rightarrow \quad \text{see positive example every} \frac{1}{\delta} \text{ samples} \]

\[ \Rightarrow \quad \text{above algorithm is efficient} \]