Lecture 17

Boosting
Weak Learning

def. algorithm A weakly PAC learns concept class \( C \) if \( \forall \delta > 0 \) st.

\( \forall c \in C \quad \forall \text{ dists } D \),
given examples of \( c \) according to \( D \)

A outputs \( h \) st.

\[ \Pr_D [ h(x) \neq c(x) ] \leq \frac{1}{2} - \frac{\delta}{2} \quad \text{advantage} \]

Thm: if \( C \) can be weakly PAC learned (on any \( D \)) then

\( C^* \) can be (strongly) PAC learned.
Weak vs. Strong Learning

Def. Algorithm $A$ weakly "learns" concept class $C$ if

$\forall c \in C \land \forall$ dists $D \exists \delta > 0$

$\forall \epsilon, \delta > 0 \quad (\delta \leq \frac{1}{4} \text{ or } \frac{1}{2} \text{ doesn't affect})$

with prob $= 1 - \delta$

given examples of $c$

$A$ outputs $h$ s.t. $Pr_D[h(x) \neq c(x)] \leq \frac{\epsilon}{2} - \frac{\delta}{2}$

It was conjectured that distribution free weak learning

was really weaker but surprise!

Can "boost" a weak learner

Then if $C$ can be weakly learned on

any dist $D$, then $C$ can be

(strongly) learned.
Applications

1) "Theoretical"
   - Unit dist Algorithms for poly term DNF
     weight w-poly threshold ftms
     (Boosting + KM)
   - Ave case vs. worst case

2) practical - Boosting
   Freund-Schapire

Good & Bad Ideas

1) simulate weak learner several times on
   same distribution & take majority answer
   or best answer
   gives better confidence
   but doesn't reduce error, what if always get same answer?

2) filter out examples on which current hypotheses
does well & run weak learner on part where you
do badly.

Problem: given a new example, how do you
know which section it is in?
3) Keep some samples on which you are ok
always use majority vote on all previous hypotheses
to predict value of new samples

text:

history: Schapire, Freund-Schapire, Impagliazzo-Servedio-Klivans

---

Filtering Procedures

- A procedure to decide which samples to keep, which to throw out
- Decide which samples on which so far you guess correctly ← need for checking future hypotheses incorrectly ← need to improve on these

---

The setting

- Given labelled examples
  
  \[(x_1, f(x_1)), (x_2, f(x_2)), \ldots\]

  \[x_i \in \mathbb{R}^d\]

  \[f \in \mathcal{C}\]

- Given weak learning alg (WL) which weakly learns (advantage \(\frac{1}{2}\)) on any dist \(\mathcal{D}\)
Boosting Algorithm

Stage 0 (Initialize)

\[ D_0 \leftarrow \emptyset \]

run WL on \( D_0 \) to generate (whp)

\[ C, \text{ s.t. } \Pr_{D_0} [ f(x) = C(x) ] \geq \frac{1}{2} + \frac{1}{2} \]

* For \( i = 1 \cdots T = O(\frac{1}{\epsilon^2}) \) stages, stage \( i \): (can stop if
  Majority(\( C_1 \cdots C_i \)) correct
  on \( \geq 1 - \epsilon \) input)

  1. Construct \( D_i \) via "filtering procedure":
     * favor pts on which maj of \( C_1 \cdots C_i \) don't do well
     * but also keep some other points 3

     Will specify soon

  2. run WL on examples from \( D_i \) to output

     \[ C_{i+1}, \text{ s.t. } \Pr_{D_i} [ f(x) = C_{i+1}(x) ] \geq \frac{1}{2} + \frac{\epsilon}{2} \]

* output \( C = \text{MAJ}(C_1 \cdots C_T) \)
Filtering procedure

Given new example $x_j, f(x)$ from example oracle

- If majority of $c_1 \ldots c_i$ wrong, keep it
  \[ i \epsilon = \frac{i}{2} \]

- If large majority right, then discard
  \[ i \epsilon \text{ right} - i \epsilon \text{ wrong} > \frac{1}{8 \epsilon} \]
  or
  \[ i \epsilon \text{ wrong} \leq \frac{i}{2} - \frac{1}{2 \epsilon} \]

- Else
  \[ i \epsilon \text{ right} - i \epsilon \text{ wrong} = \frac{i \epsilon}{8 \epsilon} \text{ for } 0 < i < 1 \]

\[ i \epsilon \text{ wrong} - i \epsilon \text{ right} = \frac{-i \epsilon}{8 \epsilon} \]

So keep with prob $= 1 - \alpha$

Prob of keeping

slope $= 2 \sqrt{3}$
Need to show:

1) Output is has nontrivial agreement with f

2) # samples needed not too bad

   why could it be bad?
   - if throw out lots of samples, might need to wait a long time before WL can give an output
   - but if throw out too many samples then you already have a good hypothesis!

\[ \text{will stop if } \text{Maj}(C_i) \text{ correct on } \frac{1}{2} \varepsilon \text{ fraction of inputs} \]

\[ \text{or } \text{Maj}(C_i) \text{ incorrect on } > \varepsilon \text{ fraction} \]

\[ \text{so filtering procedure outputs} \]

\[ \text{sample with prob } \geq \varepsilon \]

\[ (+ \text{ in expectation, every } \frac{1}{2} \varepsilon \text{ samples of } \delta \text{ at least one makes it thru the filtering system}) \]

\[ \Rightarrow \text{ filtering slows down sample collection by } O(\delta \varepsilon) \]

So let's focus on 1.
Notation

\[ R_c(x) = \begin{cases} 1 & \text{if } f(x) = c(x) \\ -1 & \text{if } f(x) \neq c(x) \end{cases} \]

\[ N_x^i(x) = \sum_{1 \leq j \leq i} R_c(x) \]

\[ M_x^i(x) = \begin{cases} 1 & \text{if } N_x^i(x) \leq 0 \\ 0 & \text{if } N_x^i(x) \geq \frac{1}{\delta} \\ 1 - 3x \cdot N_x^i(x) & \text{o.w.} \end{cases} \]

Note that new distribution on samples is proportional to \( M_x^i \):

\[ D_{M_x^i}(x) = \frac{M_x^i(x)}{\sum_{x \in \mathbb{X}} M_x^i(x)} \]

\[ \sum_{x \in \mathbb{X}} M_x^i(x) \text{ includes all "wrong" } x \text{ but also } x \text{ for which maj on } M \text{ is overwhelming correct} \]

How correct are we with \( D_{M_x^i} \)?

\[ \text{Adv}_c(M_x^i) = \sum_{x \in \mathbb{X}} R_c(x) M_x^i(x) \]

\[ \text{Pr}_{x \in D_{M_x^i}} [c(x) = f(x)] = \frac{1}{\delta} + \frac{\text{Adv}_c(M_x^i)}{2 \cdot \sum_{x \in \mathbb{X}} M_x^i(x)} \]

"Advantage" of \( c \) on \( M_x^i \)

\[ \sim \text{Pr} \{ \text{correct} \} - \text{Pr} \{ \text{incorrect} \} \]

\[ = 2 \cdot \text{Pr} \{ \text{correct} \} - 1 \]
Note:
If $\sum_{i} x_{i} \geq \varepsilon \cdot 2^{n}$

$$\text{Adv}_{c}(M_{i}) \equiv \gamma \cdot 2^{n}$$

Convert claim about WL $\Rightarrow$ claim about advantage:
- if a high advantage on output of WL
  - $\gamma \approx \varepsilon$ on lots of inputs
  - then new advantage is pretty good
- if not, then you are done.

Begin Proof

For input $x$

Let $A_{x}(i) \leq \sum_{0 \leq j \leq i-1} R_{c_{j+1}}(x) M_{j}(x)$

Claim $A_{x}(i) \leq \frac{1}{\varepsilon} + \frac{3 \cdot 2}{\varepsilon} \cdot i$

- bounds advantage per input
- only helps after $\frac{1}{\varepsilon} \cdot i$ rounds

Plan for use of claim:

Consider large matrix:

The $j^{th}$ entry: $R_{c_{j+1}}(x) \cdot M_{j}(x)$

$x$'s row sum $= \sum_{0 \leq j \leq i} R_{c_{j+1}}(x) M_{j}(x)$

$j$'th col sum $= \sum_{x} R_{c_{j+1}}(x) M_{j}(x)$

$$= \text{Adv}_{c_{j+1}}(M_{j}) \leq 2^{\frac{1}{\varepsilon} \cdot M_{j}(x)}$$

else algorithm stops