Today:

Weak vs. Strong Learning

via "Boosting"

Def. Algorithm A weakly-"PAC learns" concept class \( C \) if \( \forall c \in C \)

\( \exists \delta > 0 \)

\( \forall x \in \mathcal{X} \)

with \( \text{prob} \geq 1 - \delta \)

given labelled examples of \( C \leftarrow (x, c(x)) \)

A outputs \( h \) s.t. \( \Pr_{x \in \mathcal{X}} [h(x) \neq c(x)] \leq \frac{x}{2} - \frac{\delta}{2} \)

\( \delta \) is "advantage"
is weak learning easier than strong-learning?

surprisingly "no"!

Thus if $C$ can be weakly learned on any dist $D$ then $C$ can be (strongly) $\text{PAC}$-learned.

Applications:

"Theoretical":

- Boosting + KM $\Rightarrow$ unit dist learning algys
  for poly term DNF
  (better than low deg alg)

- insights into average case & worst case complexity
2) Practical -
   many boosting algs
   Freund Schapire ... (many many)

   history:
   Schapire
   Freund.Schapire
   lots more

   Good & Bad ideas

1) simulate weak learner many times on
   same distribution & take \{majority answer
   best answer

   \Rightarrow better confidence (5)
   but doesn't reduce error

2) filter out examples on which current
   hypothesis does well & rerun weak learner
   on part where we did badly
3) Keep some "good" samples in filter.
   Use majority vote on all hypothesis $h_1, h_2, h_3$.
   (Don't need to know "color" of $x$)

Filtering Procedures

- Decide which samples to keep/throw.
- Need keep some samples on which you did well.
  But weight more on those on which you need to improve.

Training phase:

$x, f(x)$

Later:

Given $x$, predict $f(x)$.

Here, we don't know if $h_1, h_2, h_3$ correct.
Which section are we in?
The setting:

- Given labelled examples \((x_1, f(x_1)) (x_2, f(x_2))\)
  \[ x_i \in \mathcal{X} \]
  \[ f \in \mathcal{F} \]
  \[ \text{target} \]
  \[ \text{train} \rightarrow f \in \mathcal{C} \]

- Given weak learning alg. \(WL\) which weakly learns \(\{\text{advantage} \leq 0\}\) on any dist.

\[ \text{error} (f) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{f(x_i) \neq y_i} \]

- Given weak learning alg. \(WL\) which weakly learns \(\{\text{advantage} \leq 0\}\) on any dist.

\[ \text{error} (f) = \mathbb{E}_{x \sim \mathcal{D}} [f(x) \neq y] \]

The plan:

- Simple "modest" boosting procedure
  (error slightly improves)

- Recursively use \(\Upsilon\) to drive down error
Part I: Modest improvement

Algorithm: Given oracle to \(f\), \(D\) & WL

\(h_1 \leftarrow \text{run WL on } D \text{ for func } f\)

create an example oracle \(D_2\)

flip coin:

\[
\begin{align*}
\text{Heads:} & \quad \text{draw examples from } D \text{ until find } x \text{ s.t. } h_1(x) = f(x) \quad \text{"} h_1 \text{ correct}\text{"} \\
\text{output } x \\
\text{Tails:} & \quad \text{"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""\end{align*}
\]

\(h_2 \leftarrow \text{run WL on } D_2\)

so \(h_1, h_2\) create example oracle \(D_3\)

draw examples until find \(x\) s.t. \(h_1(x) \neq f(x)\)

output \(x\)

\(h_3 \leftarrow \text{run WL on } D_3\)

output \(h \equiv \text{maj}(h_1, h_2, h_3)\)

i.e., evaluate \(h_1, h_2, h_3\) on \(x\) & output most common answer
Error analysis:

$$\text{error} \leq \beta \text{ promised by WL}$$

$$\beta_1 = \Pr_{xy}[h_1(x) \neq f(x)]$$
$$\beta_2 = \Pr_{xy}[h_2(x) \neq f(x)]$$
$$\beta_3 = \Pr_{xy}[h_3(x) = f(x)]$$

For simplicity (worst case) assume

$$\beta = \beta_1 = \beta_2 = \beta_3$$

Observation

If $h_1(x) = f(x)$ then

$$D(x) = 2(1 - \beta) D_2(x)$$

$$= 2 \beta_1 D_2(x)$$

why?

\[\text{Diagram showing different cases and implications for error.}\]
total wt of $X$ st: $h(x) = f(x)$ goes from $1-\beta$ to $\frac{1}{2}$

t rel wts of these $X$'s stay same

$$\sum_{x \text{ st } h(x) = f(x)} D(x) = 1 - \beta$$

$$\sum_{x \text{ st } h(x) = f(x)} (D(x) \cdot x) = \frac{1}{2}$$

so

$$D(x) = D(x) \cdot x = \frac{1}{2(1-\beta)} D(x)$$

$$D(x) = 2(1-\beta) D_2(x)$$

$\beta = \text{err of output of WL}$

Main Lemma

$$\text{err}_\theta (h) \leq \frac{3\beta^2 - 2\beta^3}{\beta} \equiv g(\beta)$$

$g(\beta) \ll \beta$

idea of pf:
\( \text{err}_0(h) \) has 2 types:

1) \( x \in \mathbb{S}_1 \), \( h_1(x) = h_2(x) \neq f(x) \) (both wrong)
   
   so \( h_3 \) can't fix

2) \( x \in \mathbb{S}_2 \), \( h_1(x) \neq h_2(x) \)

   here \( h_3 \) decides if \( h \) correct

\[
\text{err}_0(h) = \Pr_{x \in \mathbb{D}_0} \left[ h_1(x) \neq f(x) \right. \\
+ \Pr_{x \in \mathbb{D}_0} \left[ h_3(x) \neq f(x) \mid h_1(x) \neq h_2(x) \right] \cdot \Pr_{x \in \mathbb{D}_0} [h(x) \neq h_2(x)]
\]

\( \text{def of } \beta_3 = \text{error of WL on } \mathbb{D}_3 \)

\( \alpha_1 = \Pr_{x \in \mathbb{D}_2} [h_1(x) = f(x) \hspace{1em} \& \hspace{1em} h_2(x) \neq f(x)] \hspace{1em} \text{region 2} \)

\( \alpha_2 = \Pr_{x \in \mathbb{D}_2} [h_1(x) \neq f(x) \hspace{1em} \& \hspace{1em} h_2(x) \neq f(x)] \hspace{1em} \text{region 3} \)

\( \alpha_1 + \alpha_2 = \beta_3 \)
Then
\[
Pr_{x \in \mathbb{D}_0} \left[ h_1(x) = f(x) \land h_2(x) \neq f(x) \right] = 2 \cdot (1 - \beta'_1) \cdot Pr_{x \in \mathbb{D}_2} \left[ h_1(x) = f(x) \land h_2(x) \neq f(x) \right]
\]
\[
= 2(1 - \beta'_1) \alpha_1
\]

\[
Pr_{x \in \mathbb{D}_2} \left[ h_1(x) \neq f(x) \land h_2(x) = f(x) \right] = \frac{1}{2} - \alpha_2
\]

So
\[
Pr_{x \in \mathbb{D}_0} \left[ h_1(x) \neq f(x) \land h_2(x) = f(x) \right] = 2 \beta_1 \left( \frac{1}{2} - \alpha_2 \right)
\]

(1)

Putting together:

\[
Pr_{x \in \mathbb{D}_0} \left[ h_1(x) = h_2(x) \right] = 2(1 - \beta'_1) \alpha_1 + 2 \beta_1 \left( \frac{1}{2} - \alpha_2 \right)
\]

Also
\[
Pr_{x \in \mathbb{D}_0} \left[ \text{both } h_1, h_2 \text{ wrong} \right] = \beta_1 \alpha_2
\]

(2)

(1) + (2) put into previous

\[
\mathcal{E}_{\beta}(h) \leq 2 \beta_1 \alpha_2 + \beta \left[ 2(1 - \beta'_1) \alpha_1 + 2 \beta_1 \left( \frac{1}{2} - \alpha_2 \right) \right]
\]
\[ g(y) = 3y^2 - 2y^3 \]

**Part II** Recursive Accuracy Boosting

Strong learning Algorithm:

given $p$, $D$,

if $p$ can be achieved directly from $WL$
  then just call $WL$ & return result

$\beta \leftarrow g^{-1}(p)$ error from level below required to get error $p$

get $D_2'$ & $D_3'$ as in "modest boost"^2

$h_1 \leftarrow $ strong learn ($\beta$, $Ex(f, D_2')$)

$h_2 \leftarrow $ strong learn ($\beta$, $Ex(f, D_2')$)

$h_3 \leftarrow $ strong learn ($\beta$, $Ex(f, D_3')$)

$h \leftarrow $ maj $(h_1, h_2, h_3)$

return $h$
Sample Complexity:

- How many recursive calls?
  - Depth + size of recursion tree
- How many samples to construct filters?
  - Depth of recursion!

Also:

- If \( \beta \leq \frac{1}{4} \) then \( g(\beta) = 2\beta^2 - 2\beta^3 \leq 3\beta^2 \)
- In \( k \) steps:
  \[
  \frac{1}{3} (3\beta)^k \leq \left(\frac{3}{4}\right)^k
  \]
- \( \beta \approx \frac{1}{2} - \frac{\text{const}}{n^2} \Rightarrow K = \Theta(\log \beta \frac{1}{\epsilon}) \) depth suffices
- Given poly depth: size is \( O(3^k \log \frac{1}{\epsilon^2}) \sim \Theta(\log \frac{1}{\epsilon}) \)