Boosting: a practical algorithmic tool and a statement about learning in the PAC model itself

Boosting, view #1
- Definition: Algorithm A is a weak-learner with edge $\gamma$ for class $C$ if: for any distribution $D$ over examples labeled by some target $f \in C$, whp A produces a hypothesis $h$ with $err_D(h) \leq 1/2 - \gamma$. (think of $\gamma = 0.1$)
- Note: Ignoring $\delta$ parameter throughout the lecture since it can be handled easily (hwk 2).
- Theorem: Given a weak-learner A with edge $\gamma$ for class $C$, we can produce an alg $A'$ that achieves a PAC guarantee for class $C$ (whp produces hypothesis with error $\leq \epsilon$) using $O \left( \frac{1}{\gamma^2} \log \frac{1}{\epsilon} \right)$ calls to A. $A'$ is efficient if A is.

"Weak learning $\Rightarrow$ Strong learning"

Boosting, view #2
- Imagine you want a highly accurate algorithm to predict $y$ from $x$.
- So, you publish a large dataset $S_1$ of $(x,y)$ pairs and ask if anyone can find an $h_1$ of error $\leq 40\%$. (And say we require $h_1$ to be "simple" so we know it's not overfitting)
- Now, you use $h_1$ to create a new dataset $S_2$ (by focusing more on the problematic data for $h_1$) and ask if anyone can find an $h_2$ of error $\leq 40\%$ on $S_2$.
- And so on.
- You can do this and combine the $h_i$s.t either (a) you drive your error down to 0 or else (b) you reach a hard dataset that nobody can do much better than random guessing on.

Preliminaries
- Assume we want to learn some unknown target function $f$ over distribution $D$.
- Assume we have a weak-learner A with edge $\gamma$ that uses hypotheses from some class of VC-dim $d$. (A should be able to achieve error $\leq 1/2 - \gamma$ for learning $f$ over any reweighting of $D$)
- We will end up running A for $T$ times producing hypotheses $h_1, ..., h_T$ and combining them into a single rule.
- By problem 3 on current hwk, the set of such combinations has VC-dim $O(Td \log Td)$.
- This will allow us to do all this on a sample of size $\tilde{O} \left( \frac{m^2 d}{\epsilon^2} \right)$.

($\tilde{O}$ notation hides logarithmic factors)

Preliminaries, contd.
- We will draw a training sample $S$ of size $m = \tilde{O} \left( \frac{r^2}{\epsilon \gamma} \right)$.
- Assume that given any weighting of the points in $S$, A will return a hypothesis $h$ of error at most $1/2 - \gamma$ over the distribution induced by that weighting. (ignoring $\delta$)
- Will show can produce $h$ with $err_S(h) = 0$ for $T = O \left( \frac{\log m}{\gamma^2} \right)$.
- Just need $m \gg \frac{d \log m}{\epsilon^2} \approx \frac{d \log \left( \frac{1}{\gamma^2} \right)}{\epsilon^2}$.
Boosting algo (Adaboost-light)

1. Given labeled sample $S = \{x_1, ..., x_m\}$, initialize each example $x_i$ to have weight $w_i = 1$. Let $w = (w_1, ..., w_m)$.

2. For $t = 1, ..., T$ do:
   a. Call $A$ on the distribution $D_t$ over $S$ induced by $w$.
   b. Receive hypothesis $h_t$ of error $\leq 1/2 - \gamma$ over $D_t$.
   c. Multiply the weight of each example misclassified by $h_t$ by $\alpha = \frac{0.5 + \gamma}{0.5 - \gamma}$. Leave the other weights alone.

3. Output the majority vote classifier $\text{MAJ}(h_1, ..., h_T)$.

Assume $T$ is odd so no ties.

Thm: $T = O\left(\log \frac{m}{\gamma^2}\right)$ is sufficient s.t. $\text{err}_S(\text{MAJ}(h_1, ..., h_T)) = 0$.

Example

<table>
<thead>
<tr>
<th>$h_1$</th>
<th>$h_2$</th>
<th>$h_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>+</td>
<td>-</td>
<td>+</td>
</tr>
</tbody>
</table>

$\text{err}_S(h_1) = \frac{1}{4}$
$\text{err}_S(h_2) = \frac{1}{4} 	imes 3 = \frac{3}{4}$
$\text{err}_S(h_3) = \frac{1}{4} 	imes 3 = \frac{3}{4}$

Example

<table>
<thead>
<tr>
<th>$h_1$</th>
<th>$h_2$</th>
<th>$h_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>+</td>
<td>-</td>
<td>+</td>
</tr>
</tbody>
</table>

$\text{err}_S(h_1) = \frac{1}{4}$
$\text{err}_S(h_2) = \frac{1}{4} 	imes 3 = \frac{3}{4}$
$\text{err}_S(h_3) = \frac{1}{4} 	imes 3 = \frac{3}{4}$
Proof of Boosting Theorem

Thm: $T = O\left(\frac{\log m}{\gamma^2}\right)$ is sufficient s.t. $\text{err}_5(MA(h_1, \ldots, h_T)) = 0$.

Proof:
- First, if $MA(h_1, \ldots, h_T)$ makes a mistake on any $x_i$, then its final weight must be greater than $aT/2$.
- Let $W_i$ be total weight after update $i$. $W_0 = m$.
- By the weak-learning assumption, $h_i$ has error $\leq 1/2 - \gamma$ on $D_i$. So, at most $1/2 - \gamma$ fraction of weight multiplied by $a$.
- So, $W_{i+1} \leq \left(\alpha\left(\frac{1}{2} - \gamma\right) + \frac{1}{2} + \gamma\right)W_i = (1 + 2\gamma)W_i$.
- So if $\text{err}_5(...) > 0$ then $aT/2 \leq W_T \leq (1 + 2\gamma)^Tm$.
  
  So, $1 \leq a^{-T/2}(1 + 2\gamma)^Tm$.

More Reflections

- Consider a zero-sum game with examples as columns and hypotheses in $H$ as rows.

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
<th>$x_6$</th>
<th>$x_{11}$</th>
<th>$x_{12}$</th>
<th>$x_{13}$</th>
<th>$x_{14}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_1$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
</tr>
<tr>
<td>$h_2$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
</tr>
<tr>
<td>$h_3$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
</tr>
<tr>
<td>...</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
</tr>
</tbody>
</table>

  Rows represent all $h$ in the class used by $A$.

- If row plays $h_i$ and column plays $x_j$, then row wins if $h_i(x_j)$ is correct, and column wins if $h_i(x_j)$ is incorrect.
More Reflections

- Consider a zero-sum game with examples as columns and hypotheses in $H$ as rows.

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>...</th>
<th>$x_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_1$</td>
<td>$X$</td>
<td>$X$</td>
<td>$X$</td>
<td>$X$</td>
</tr>
<tr>
<td>$h_2$</td>
<td>$X$</td>
<td>$X$</td>
<td>$X$</td>
<td>$X$</td>
</tr>
<tr>
<td>$h_3$</td>
<td>$X$</td>
<td>$X$</td>
<td>$X$</td>
<td>$X$</td>
</tr>
<tr>
<td>...</td>
<td>$X$</td>
<td>$X$</td>
<td>$X$</td>
<td>$X$</td>
</tr>
<tr>
<td>$X$</td>
<td>$X$</td>
<td>$X$</td>
<td>$X$</td>
<td>$X$</td>
</tr>
</tbody>
</table>

- We are given that for any distrib $D$ over columns (mixed strategy for the column player) there exists a row that wins with prob $\geq 1/2 + \gamma$ (payoff $\geq 1/2 + \gamma$).

- By Minimax Thm, there exists a distribution $P$ over $h_i$ that wins with prob $\geq 1/2 + \gamma$ for any $x_j$.
- So, whp a large random sample from $P$ will give correct majority vote on all $x_j$. (One way to see boosting is possible in principle)

In fact, this is just like RWM versus a best-response oracle, except our focus is on properties of the majority vote over the choices of the best-response oracle.

Margin Analysis

- Empirically noticed that you can keep running the booster past the point of perfect classification of $S$, and generalization doesn't get worse.
- One way to explain: "$L_1$ margins" or "margin of the vote"

Argument sketch:

- As $T \to \infty$, row player's strategy approaches minimax optimal (for all $x_j \in S$, $1/2 + \gamma$ of $h_i$ vote correctly).

- Define $h'$ as the randomized predictor: "given $x$, select $\theta \left(\frac{1}{T} \log \frac{1}{\epsilon} \right) h_i$ at random from $h$ and take their maj vote"

- $\text{So, } err_S(h') \leq \epsilon/2$.

- Also, $err_D(h') \geq err_D(h)/2$. (If $h(x)$ is wrong, then at least 50% chance that $h'(x)$ is wrong too)

- But $h'$ isn't overfitting since whp no small majority-votes are overfitting and this is just a randomization over them. So $h$ isn't overfitting by much either.