CS 395T Computational Learning Theory	
Lecture 11: October 12, 2005	Lecturer: Adam Klivans
Scribe: Brian Kulis	

11.1 Schapire's Booster (Cont.)

Last time, we discussed Schapire's booster. We ended by showing that, in general, given error β_{old}, an iteration of Schapire's booster would reduce the error to $\beta_{new} = 3\beta_{old}^2 - 2\beta_{old}^3$. We now describe a general version of Schapire's booster and analyze it briefly.

Let $g(\beta) = 3\beta_{old}^2 - 2\beta_{old}^3$. Assume that we have a weak learning algorithm WL with a guarantee of error at most $1/2 - \gamma$. The distributions D_1, D_2, and D_3 are as in the previous lecture.

Below is the algorithm $\text{Booster}(\alpha, EX(D))$. If we want to output a final hypothesis with error less than or equal to ϵ, we call $\text{Booster}(\epsilon, EX(D))$.

1. If $\alpha \geq \frac{1}{2} - \gamma$, output hypothesis from $WL(EX(D))$
2. Otherwise,
   - $h_1 = \text{Booster}(g^{-1}(\alpha), D_1)$,
   - $h_2 = \text{Booster}(g^{-1}(\alpha), D_2)$,
   - $h_3 = \text{Booster}(g^{-1}(\alpha), D_3)$.
3. Output majority of h_1, h_2, h_3.

We ask the following question: what is the depth of the recursion in this algorithm? We want error less than or equal to ϵ, so how many levels of recursion are needed?

There are two cases to consider. First, if the error $\beta \leq \frac{1}{4}$, then it is possible to show that the number of levels of recursion is $O(\log\log(\frac{1}{\epsilon}))$ to bring the error down to ϵ (not proven in class). In the second case,

$$\frac{1}{2} - \gamma \geq \beta > \frac{1}{4}$$

If we look at $\frac{1}{2} - g(\beta)$, we get $(\frac{1}{2} - \beta)(1 + 2\beta - 2\beta^2)$, which is $\geq \frac{11}{8}$ if $\beta > \frac{1}{4}$. Initially, our distance from $\frac{1}{2}$ is γ.... we want this to go to $\frac{1}{4}$. This implies that we need $\log_{11/8}(\frac{1}{\gamma})$ depth. In the weak learning scenario, $\gamma \geq 1/poly(n)$, so after $\log(poly(n))$ levels, the error decreases to $\frac{1}{4}$. Putting this together with the first case, we have a total number of $O(\log n) + O(\log\log\frac{1}{\epsilon})$ levels of recursion.

As a corollary, the size of the final hypothesis is 3^l, where l is the number of levels of recursion in the algorithm. Using the bound above, the final size is $n^c \log(\frac{1}{\epsilon})$ (for some constant c), so Occam's Razor gives us good generalization as long as the number of examples is larger than the size of the final hypothesis.

11.2 AdaBoost

We now present a simpler boosting algorithm called AdaBoost, due to Freund and Schapire. The presentation for this algorithm is borrowed from Avrim Blum.

11.2.1 Algorithm

The algorithm works as follows:

1. Start with the uniform distribution $D_i = 1/n$ and $w_i = 1$ for all n points. Let $t = 1$.
2. Call the weak learner using the current distribution D; it returns a hypothesis h_t. Let the error of the weak learner on this distribution be E and the accuracy $A = 1 - E$.
3. Set $\beta_t = \frac{E}{A}$.
4. For every example x_i such that $h_t(x_i) = c(x_i)$, set $w_i^{new} = \beta_t w_i$. (Otherwise $w_i^{new} = w_i$.)
5. Set $D_i(x_i) = \frac{w_i^{new}}{\sum_i w_i^{new}}$.
6. Increment t and return to Step 2. Repeat this for T steps.
7. Output $\text{sign}(\sum_t \alpha_t h_t - \frac{1}{2})$, where

$$\alpha_t = \frac{\log(1/\beta_t)}{\sum_t \log(1/\beta_t)}.$$

11.2.2 Analysis

For simplification, assume that the error rate is exactly $\frac{1}{2} - \gamma$ for every iteration (in general, it is $\frac{1}{2} - \gamma_t$ on iteration t for a weak learner). Using this simplification, each β_t is the same; call this value β. Then α_t is constant as well:

$$\alpha = \frac{\log(1/\beta)}{t\log(1/\beta)} = \frac{1}{t}.$$

Hence we output $\text{sign}(\sum_t \frac{h_t}{t} - \frac{1}{2})$.

The claim is that the error of the outputted hypothesis is less than or equal to $e^{-2T\gamma^2}$. If we can prove this, then we can take $T = \frac{1}{\gamma^2}\log(1/\epsilon)$, and the error is pushed down to ϵ. In other words, AdaBoost boosts a weak learner to a learner of arbitrary accuracy.

Let us now prove the claim. The total weight after one iteration is $n(1-(1-\beta)A)$. For example, say we are 70 percent accurate. Then after one iteration, seventy percent of the weights get updated, so the total weight is

$$.7n\beta + .3n \quad = \quad n - .7n + .7n\beta$$

$$\begin{aligned} &= n - n(.7 - .7\beta) \\ &= n - nA(1-\beta) \\ &= n(1-(1-\beta)A) \end{aligned}$$

In general , after t iterations, the total weight is less than or equal to $n(1-(1-\beta)A)^t$. What is the weight of any point such that the final hypothesis still makes a mistake? More than half of the h_t hypotheses are wrong, so the weight of this point is at least $\beta^{T/2}$ based on the updates. So, if we have an ϵ-fraction of mistakes, the weight of this ϵ-fraction is at least $\epsilon n \beta^{T/2}$. Of course, this weight is less than the total weight, i.e.

$$\epsilon n \beta^{T/2} \leq n(1-(1-\beta)A)^T$$

Now we simplify this expression. Since $1-(1-\beta)A = 1 - A + E$, we have:

$$\begin{aligned} \epsilon n \beta^{T/2} &\leq n(1-A+E)^T \\ &= n(2E)^T \end{aligned}$$

Further, this implies

$$\begin{aligned} \epsilon &\leq \left(\frac{4E^2}{\beta}\right)^{T/2} \\ &= (4EA)^{T/2} \\ &= (1-4\gamma^2)^{T/2} \\ &\leq e^{-2\gamma^2 T}, \end{aligned}$$

where we have used the fact that $EA = (\frac{1}{2}-\gamma)(\frac{1}{2}+\gamma) = \frac{1}{4} - \gamma^2$ and $(1+x) \leq e^x$. This completes the proof for constant error rate $\frac{1}{2} - \gamma$.