CS260: Machine Learning Theory
Lecture 14: Generalization Error of AdaBoost

November 9, 2011

Lecturer: Jennifer Wortman Vaughan

1 A First Attempt to Bound the Generalization Error of AdaBoost

We saw last time that the training error of AdaBoost decreases exponentially as the number of rounds T grows. However, this says nothing about how well the function output by AdaBoost performs on new examples. Today we will discuss the generalization error of AdaBoost.

We know that AdaBoost gives us a consistent function quickly; the bound we derived on training error decreases exponentially, and once this bound drops below $1/m$, we know we must have a consistent function. Because of this, if we were able to find the VC dimension of the class of functions from which AdaBoost chooses, we'd be able to apply our results from the first few weeks of class to get a bound on generalization error.

Let $\mathcal{H}$ be the class of functions from which the weak learning algorithm $\mathcal{A}$ chooses, and let d be the VC dimension of this class. The class of functions $\mathcal{H}'$ from which AdaBoost chooses is the class of all functions h that can be written as

$$\mathcal{H}' = \left\{ h : h(\vec{x}) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(\vec{x}) \right) \text{ with } h_t \in \mathcal{H}, \alpha_t \in \mathbb{R}, \text{ for all } t \right\} .$$

Note that this class is infinite even if $\mathcal{H}$ is finite since each α_t is a real-valued parameter.

It is possible to show (though we will not do it here) that the VC dimension of this class $\mathcal{H}'$ is $\tilde{O}(Td)$, where the $\tilde{O}$ notation hides log terms. This implies that with high probability, for any $h \in \mathcal{H}'$,

$$err(h) \leq \widehat{err}(h) + \tilde{O}\left(\sqrt{\frac{Td}{m}} \right) .$$

Let's examine this bound. We know that as T grows, the training error of the function output by AdaBoost shrinks. However, at the same time, $\mathcal{H}'$ becomes more complex and the second term in this bound increases. We would expect to find that there is an optimal value of T that optimizes this trade-off to minimize $err(h)$.

Somewhat surprisingly, people have found that in practice, $err(h)$ continues to decrease even after $\widehat{err}(h)$ reaches 0. This is good news in the sense that overfitting does not occur as T increases, but it seems to contradict the intuition conveyed by the bound above. It turns out that this phenomenon can be explained by analyzing the "confidence" of the function h, which may continue to increase over time even after the training error has hit zero.

All CS260 lecture notes build on the scribes' notes written by UCLA students in the Fall 2010 offering of this course. Although they have been carefully reviewed, it is entirely possible that some of them contain errors. If you spot an error, please email Jenn.

2 Existence of Large Margin Classifiers

We've talked about the "confidence" of classifiers before when we talked about the margin of a linear separator. Recall that we defined the margin of a point $\vec{x}_i$ with respect to a linear separator with weights $\vec{w}$ as

$$y_i \frac{\vec{w} \cdot \vec{x}_i}{||\vec{w}||_2}.$$

We can define a notion of "margin" for the AdaBoost function too. Instead of defining it in terms of the actual input points $\vec{x}$, we will define it in terms of the T-dimensional point $\langle h_1(\vec{x}_i), \cdots, h_T(\vec{x}_i)\rangle$. Also, instead of normalizing with respect to the L2 norm of the weights as we did before, it turns out to be more convenient to normalize with respect to the L1 norm. We define the margin of a labeled point $(\vec{x}_i, y_i)$ with respect to the function h output by AdaBoost as

$$y_i \frac{\sum_{t=1}^T \alpha_t h_t(\vec{x}_i)}{\sum_{t=1}^T \alpha_t} .$$

Note that this definition does not have exactly the same geometrical interpretation as the notion of margin we discussed before since we are normalizing using the L1 norm. However, this alternative definition of margin has similar meaning, in that a larger margin implies higher confidence.

In Theorem 2, we show that if the weak learning assumption holds, then there must exist a large L1-margin classifier of the form $h(\vec{x}) = \text{sign}(\sum_{t=1}^T \alpha_t h_t(\vec{x}))$. The proof is based on the Minimax Theorem, which we proved in Lecture 12. We restate it here, in slightly different notation than the way it was originally stated. (It is easy to verify that this statement is equivalent; we've just switched around some variable names.)

Theorem 1 (Minimax Theorem)**.** *For any $m \times n$ matrix M,*

$$\max_{\vec{p}\in\Delta_m} \min_{\vec{q}\in\Delta_n} \sum_{i=1}^n \sum_{j=1}^m p_i q_j M_{i,j} = \min_{\vec{q}\in\Delta_n} \max_{\vec{p}\in\Delta_m} \sum_{i=1}^n \sum_{j=1}^m p_i q_j M_{i,j} .$$

For simplicity, we prove the theorem for finite $\mathcal{H}$, but it can be extended.

Theorem 2. *Consider a finite function class $\mathcal{H} = \{h_1, \cdots, h_n\}$ and a set of labeled points $(\vec{x}_1, y_1), ..., (\vec{x}_m, y_m)$. Suppose that for some $\gamma > 0$, for every distribution $\vec{p} \in \Delta_m$, there exists an $h \in \mathcal{H}$ such that*

$$\sum_{i=1}^m p_i \mathbb{I}(h(\vec{x}_i) \neq y_i) \leq \frac{1}{2} - \gamma.$$

Then there exists $\vec{q} \in \Delta_n$ such that

$$\min_{i\in\{1,\cdots,m\}} \sum_{j=1}^n q_j h_j(\vec{x}_i) y_i \geq 2\gamma .$$

We can interpret $\vec{p}$ as a distribution over input points and $\vec{q}$ as a weighting over the functions in $\mathcal{H}$ (like $\vec{\alpha}$ will be). The result then says that if the weak learning assumption holds then there exists a function $h(\vec{x}) = \text{sign}(\sum_{j=1}^n q_j h_j(x_i))$ that has a margin of at least 2γ. Note that we are not yet showing that

AdaBoost increases the margin over time, just that a large margin function exists.

Proof of Theorem 2: We start by defining an $m \times n$ matrix M with entries

$$M_{ij} = \mathbb{I}(h_j(\vec{x}_i) \neq y_i) \; .$$

Fixing any $\vec{p} \in \Delta_m$, we have:

$$\begin{aligned}
\min_{\vec{q}\in\Delta_n} \sum_{i=1}^{m}\sum_{j=1}^{n} p_i q_j M_{ij} &= \min_{\vec{q}\in\Delta_n} \sum_{i=1}^{m}\sum_{j=1}^{n} p_i q_j \mathbb{I}(h_j(\vec{x}_i) \neq y_i) \\
&= \min_{\vec{q}\in\Delta_n} \sum_{j=1}^{n} q_j \sum_{i=1}^{m} p_i \mathbb{I}(h_j(\vec{x}_i) \neq y_i) \\
&= \min_{j\in\{1,\cdots,n\}} \sum_{i=1}^{m} p_i \mathbb{I}(h_j(\vec{x}_i) \neq y_i) \\
&\leq \frac{1}{2} - \gamma \; .
\end{aligned}$$

Since this holds for any distribution $\vec{p}$, we have

$$\max_{\vec{p}\in\Delta_m} \min_{\vec{q}\in\Delta_n} \sum_{i=1}^{m}\sum_{j=1}^{n} p_i q_j M_{ij} \leq \frac{1}{2} - \gamma \; .$$

By the Minimax Theorem, this implies that

$$\min_{\vec{q}\in\Delta_n} \max_{\vec{p}\in\Delta_m} \sum_{i=1}^{m}\sum_{j=1}^{n} p_i q_j M_{ij} \leq \frac{1}{2} - \gamma \tag{1}$$

Now, for any arbitrary $\vec{q} \in \Delta_n$, we have

$$\begin{aligned}
\max_{\vec{p}\in\Delta_m} \sum_{i=1}^{m}\sum_{j=1}^{n} p_i q_j M_{ij} &= \max_{\vec{p}\in\Delta_m} \sum_{i=1}^{m} p_i \sum_{j=1}^{n} q_j \mathbb{I}(h_j(\vec{x}_i) \neq y_i) \\
&= \max_{i\in\{1,\cdots,m\}} \sum_{j=1}^{n} q_j \mathbb{I}(h_j(\vec{x}_i) \neq y_i) \; .
\end{aligned}$$

Let $\vec{q^*}$ denote the distribution $\vec{q}$ that minimizes this expression. Equation 1 implies that

$$\max_{i\in\{1,\cdots,m\}} \sum_{j=1}^{n} q_j^* \mathbb{I}(h_j(\vec{x}_i) \neq y_i) \leq \frac{1}{2} - \gamma \; .$$

This implies that for all i,

$$\begin{aligned}
\sum_{j=1}^{n} q_j^* h_j(\vec{x}_i) y_i &= \sum_{j=1}^{n} q_j^* (1 - 2\mathbb{I}(h_j(\vec{x}_i) \neq y_i)) \\
&= 1 - 2\sum_{j=1}^{n} q_j^* \mathbb{I}(h_j(\vec{x}_i) \neq y_i) \geq 1 - 2\left(\frac{1}{2} - \gamma\right) = 2\gamma.
\end{aligned}$$

□

3 Bounding the Margin on the Training Data

Let's now turn our attention back to AdaBoost. Define

$$f(\vec{x}_i) = \frac{\sum_{t=1}^{T} \alpha_t h_t(\vec{x}_i)}{\sum_{t=1}^{T} \alpha_t} ,$$

so the L1-margin of $\vec{x}_i$ is $y_i f(\vec{x}_i)$. In the last class, we showed that it is possible to bound

$$\widehat{err}(h) \leq \prod_{t=1}^{T} 2\sqrt{\epsilon_t(1-\epsilon_t)} .$$

This bound can actually be generalized to

$$\frac{1}{m}\sum_{i=1}^{m} \mathbb{I}(y_i f(\vec{x}_i) \leq \theta) \leq \prod_{t=1}^{T} 2\sqrt{\epsilon_t^{1-\theta}\,(1-\epsilon_t)^{1+\theta}}$$

for any value of $\theta \geq 0$. When $\theta = 0$ we get back the bound from the last class. The proof is very similar to the proof of the $\theta = 0$ case so we won't go through it here.

If the weak learning assumption holds, then for all t, $\epsilon_t \leq 1/2 - \gamma$, and so

$$\frac{1}{m}\sum_{i=1}^{m} \mathbb{I}(y_i f(\vec{x}_i) \leq \theta) \leq \left(\sqrt{(1-2\gamma)^{1-\theta}(1+2\gamma)^{1+\theta}}\right)^T .$$

If $\theta < \gamma$, then the quantity inside the square root is less than 1, and the quantity on the right will go to 0 as T gets large. (Exercise: Verify this.) This implies that as T grows, we will eventually get to a point where the margin of every training example is at least γ.

4 Back to Generalization Error

It turns out that it's possible to use these results to bound the generalization error of AdaBoost too. We state this bound for finite weak hypothesis spaces, $\mathcal{H}$. Let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \{-1, 1\}$, and let $(\vec{x_1}, y_1), \cdots, (\vec{x_m}, y_m)$ be sampled i.i.d. from this distribution. For any δ, with probability at lest $1-\delta$, for all f of the form $f(\vec{x}) = \sum_{i=1}^{T} \alpha_t h_t(\vec{x})$ with $T > 0$, and $h(\vec{x}) = \text{sign}(f(\vec{x}))$,

$$\text{err}(h) \leq \frac{1}{m}\sum_{i=1}^{m} \mathbb{I}(y_i f(\vec{x}_i) \leq \theta) + O\left(\sqrt{\frac{\log m \log|\mathcal{H}|}{m\theta^2}} + \log\frac{1}{\delta}\right) .$$

Let's examine this theorem. The first term on the right hand side is precisely the fraction of points in our sample that have margin less than θ, which we know goes to 0 as T gets big. Therefore, this theorem provides a bound on $\text{err}(h)$ with one term that goes to 0 as T gets big, and one term that doesn't depend on T at all. This confirms the empirical observations that the generalization error of AdaBoost tends to decrease as T gets big.