CS6781 - Theoretical Foundations of Machine Learning

Lecture 19: Adaptive Boosting

April 16, 2020

Lecturer: Nika Haghtalab *Readings: N/A*
Scribe: Oliver Richardson and Cole Miles

1 Empirical Error of AdaBoost

In this section, we prove that the empirical error of AdaBoost converges to 0 quickly as $T \to \infty$. Recall from last lecture that we are aiming to prove the following results.

Theorem 1.1. *Let h_{final} be the output of AdaBoost. Then* $\mathrm{err}_S(h_{final}) \leq \exp\left(-2\sum_{t=1}^{T}\gamma_t^2\right)$.

Under the assumption that for all t, $\gamma_t \geq \gamma$, setting $T = \mathcal{O}\left(\frac{1}{\gamma^2}\ln\left(\frac{1}{\varepsilon}\right)\right)$ implies that `AdaBoost` generates a hypothesis h_{final} that achieves error at most ϵ. That is, $\mathrm{err}_S(h_{\text{final}}) \leq \epsilon$.

Let us recall the proof outline we discussed on April 14th.

1. If h_{final} is wrong on x_i, i.e., the weighted majority of classifiers gave the wrong answer, which means x_i's weight was increased many times. We will show in Lemma 1.2 that this is at least $\frac{1}{m}\prod_{t=1}^{T}\frac{1}{Z_t}$.
2. This means that the number of points where h_{final} is wrong is sat most $m\prod_{t=1}^{T}Z_t$ that implies that $\mathrm{err}_S(h_{\text{final}}) \leq \prod_{t=1}^{T}Z_t$
3. This quantity, $\prod_{t=1}^{T}Z_t$ goes to zero quickly.

We now formally go through the steps of this proof to show that AdaBoost converges to zero sample error.

Lemma 1.2. *The distribution at the end of our algorithm, P_{T+1} can be written as*

$$P_{T+1}(x_i) = \frac{\exp\left(-y_i\bar{h}_{final}\right)}{m\prod Z_t}$$

where $\bar{h}_{final} = \sum_t \alpha_t h_t(x_i)$ is the weighted sum before its sign is taken; $h_{final} = \mathrm{sign}(\bar{h}_{final})$

Proof. Recall the update rule, given by:

$$P_{T+1}(x_i) = P_T(x_i)\cdot\frac{1}{Z_T}\cdot\exp(-y_i\cdot\alpha_T\cdot h_T(x_i))$$

we can keep unrolling this:

$$= P_{T-1}(x_i)\cdot\frac{1}{Z_{T-1}}\cdot\exp(-y_i\cdot\alpha_{T-1}\cdot h_{T-1}(x_i))\cdot\frac{1}{Z_T}\cdot\exp(-y_i\cdot\alpha_T\cdot h_T(x_i))$$
$$= P_1(x_i)\prod_{t=1}^{T}\frac{1}{Z_t}\exp(-y_i\alpha_t h_t(x_i))$$

Because the initial distribution P_1 is uniform, $P_1(x_i) = \frac{1}{m}$, so bringing out the normalization factors and

$$= \frac{\exp\left(-y_i\sum_t\alpha_t h_t(x_i)\right)}{m\prod Z_t}$$
$$= \frac{\exp\left(-y_i\bar{h}_{\text{final}}\right)}{m\prod Z_t}$$

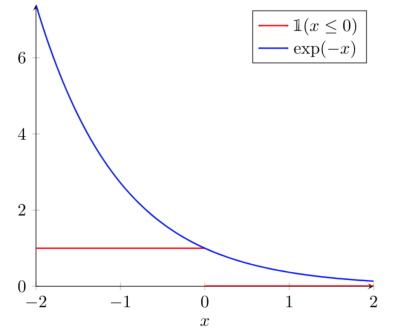
□

This says that we can describe the probability assigned to each point by the magnitude of the prediction on that point. How do we connect the magnitude to the sign of the prediction, which determines the error? This is given by a simple fact.

Fact 1.3. $\mathbf{1}(y_i \neq h_{final}(x_i)) \leq \exp(-y_i \cdot \bar{h}_{final}(x_i)))$

Proof. $\mathbf{1}(y_i \neq h_{\text{final}}(x_i)) = \mathbf{1}(y_i \neq \text{sign}(\bar{h}_{\text{final}}(x_i))) = \mathbf{1}(y_i \bar{h}_{\text{final}}(x_i) \leq 0) \leq \exp(-y_i \bar{h}_{\text{final}}(x_i))$

since multiplying by the magnitude will not change the sign, and using the fact $\mathbf{1}(x \leq 0) \leq \exp(-x)$.



□

Using this, we can prove:

Lemma 1.4. $\text{err}_S(h_{final}) \leq \prod_{t=1}^{T} Z_t$.

Proof.

$$\begin{aligned}
\text{err}_S(h_{\text{final}}) &= \frac{1}{m} \sum \mathbf{1}(y_i \neq h_{\text{final}}(x_i)) \\
&\leq \frac{1}{m} \sum_i \exp(-y_i \bar{h}_{\text{final}}(x_i)) && \text{by Fact 1.3} \\
&= \sum_{i=1} m P_{T+1}(x_i) \prod_{t=1}^{T} Z_t && \text{by Lemma 1.2} \\
&= \prod_{t=1}^{T} Z_t. && \text{by definition of } P_{T+1} \text{ being a distribution}
\end{aligned}$$

□

This shows that the sample error is bounded by the product of our normalization factors over all iterations. The next step is to show that this product goes to 0 quickly as T gets large; in particular, we will show that $\prod_t Z_t \leq \exp(-2\sum_t \gamma_t^2)$. Recall that γ_t is the amount we do better than random predictions ($\epsilon_t = \frac{1}{2} - \gamma_t$), and so even if γ_t is small, summing all T of them will get large. We now show this bound.

Lemma 1.5. $\prod_{t=1}^{T} Z_t \leq \exp(-2\sum_t \gamma_t^2)$

Proof.

$$\begin{aligned}
\prod_{t=1}^{T} Z_t &= \prod_{t=1}^{T} \overbrace{2\sqrt{\epsilon_t(1-\epsilon_t)}}^{\text{Corollary 3.4 from April 14 lecture}} \\
&= \prod_{t=1}^{T} \sqrt{(1-2\gamma_t)(1+2\gamma_t)} \\
&= \prod_{t=1}^{T} \sqrt{1-4\gamma_t^2} \\
&\leq \prod_{t=1}^{T} \sqrt{\exp(-4\gamma_t^2)} \qquad \text{since } 1-x \leq \exp(x) \\
&= \exp\left(-2\sum_t \gamma_t^2\right).
\end{aligned}$$

□

Putting this all together, we find that after T timesteps, AdaBoost achieves sample error $\text{err}_S(h_{\text{final}}) \leq \exp(-2\sum_t \gamma_t^2)$. If we want to bound this to be less than ϵ, we require

$$\exp(-2T\gamma^2) \leq \epsilon \implies T = O\left(\frac{1}{\gamma^2}\ln\left(\frac{1}{\epsilon}\right)\right). \tag{1}$$

2 Generalization Error of AdaBoost

We have shown that we can get arbitrarily small sample error if we run AdaBoost for long enough. However, at the end of the day, what we really care about is the true distribution error. To show that we don't overfit, we'd like to obtain a similar bound on the true error: $\text{err}_{\mathcal{D}}(h_{\text{final}}) \leq \epsilon$.

One technique we have previously used to bound true distribution errors is to find the VC dimension of the hypothesis class, which is a measure of how "complex" a hypothesis class can be. We would like to relate how "complex" these boosted classifiers are in terms of the base classifiers used to construct them.

For simplicity, we will assume that the boosting is not adaptive. That is, we will use a fixed $\alpha_t = \alpha > 0$, which will reduce the boosting process to just a majority vote ($f = \text{sign}(\alpha\sum_t h_t) = \text{sign}(\sum_t h_t)$). This is not exactly what happens with AdaBoost, but it will make the ideas more clear.

Define $\mathcal{H}$ to be the class of hypotheses of our base learners, and $\mathcal{H}^k$ to be the hypothesis class of all boosted classifiers obtained by using k hypotheses from $\mathcal{H}^k$:

$$\mathcal{H}^k := \left\{ f = \text{sign}\left(\sum_{t=1}^{k} h_t\right),\ \forall t \in [k], h_t \in \mathcal{H} \right\}$$

To bound the complexity of $\mathcal{H}^k$, we would like to relate VCDim($\mathcal{H}^k$) to VCDim($\mathcal{H}$).

Recall that the growth function $\Pi_{\mathcal{H}}(m)$ is defined to be the number of possible labelings of m points that functions in $\mathcal{H}$ can produce, and that the VC dimension is defined as the maximum m at which $\Pi_{\mathcal{H}}(m) = 2^m$.

We can think of hypotheses in $\mathcal{H}^k$ as being obtained by combining (through majority vote) k hypotheses in $\mathcal{H}$. Since we know there are only $\Pi_{\mathcal{H}}(m)$ hypotheses in $\mathcal{H}$ that produce distinct labelings, we really can reduce this to only thinking about combining these $\Pi_{\mathcal{H}}(m)$ hypotheses. In the worst case, each combination will produce a new, unseen labeling

of the data. Since a combination is formed by picking k hypotheses, and we can pick the same hypothesis multiple times, we can see that in the worst case we will be able to produce $(\Pi_{\mathcal{H}}(m))^k)$ labelings with these combinations.

Hence, letting $d = \text{VCDim}(\mathcal{H})$:

$$\begin{aligned}
\Pi_{\mathcal{H}^k}(m) &\leq (\Pi_{\mathcal{H}}(m))^k \\
&\leq \left(\sum_{i=0}^{d} \binom{m}{i}\right)^k && \text{Sauer's Lemma} \\
&\leq \left(\frac{em}{d}\right)^{kd} && \text{Fact proved in 1/30 lecture}
\end{aligned}$$

Let $d_k = \text{VCDim}(\mathcal{H}^k)$. We know then that $\mathcal{H}^k$ can shatter a set of size 2^{d_k}, so we have

$$\begin{aligned}
2^{d_k} &\leq \left(\frac{ed_k}{d}\right)^{kd} \\
d_k &\leq kd \log_2\left(\frac{ed_k}{d}\right) \\
\rightarrow d_k &< 2kd\log(ke)
\end{aligned}$$

where the last line uses the fact from HW#2 that for $x > 1, a, b > 2$ we have that $x \leq a\log_2(bx) \implies x < 2a\log_2(ab)$.

Note that T plays the role of k here, since after every timestep we add a new base classifier to our boosted classifier. From our previously proven VCDim bounds on AdaBoost, we then have that if $\text{err}_S(h_{\text{final}}) \leq \epsilon$, that (up to some logarithmic terms)

$$\begin{aligned}
\text{err}_{\mathcal{D}}(h_{\text{final}}) &\leq \text{err}_{\mathcal{S}}(h_{\text{final}}) + O\left(\sqrt{\frac{\text{VCDim}(\mathcal{H}^T)}{m}}\right) \\
&\leq \epsilon + \tilde{O}\left(\sqrt{\frac{T \cdot \text{VCDim}(\mathcal{H})}{m}}\right)
\end{aligned}$$

Previously, to get the sample error below some threshold ϵ, we used Eq. 1 to choose T. Using that choice here gives us

$$\text{err}_{\mathcal{D}}(h_{\text{final}}) \leq \epsilon + \tilde{\mathcal{O}}\left(\sqrt{\frac{\frac{1}{\gamma^2}\ln(1/\epsilon) \cdot \text{VCDim}(\mathcal{H})}{m}}\right)$$

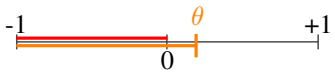
As long as our number of samples is large enough, $m \geq \frac{\ln(1/\epsilon)}{\gamma^2\epsilon^2}$, we get that the generalization error is $\mathcal{O}(\epsilon)$, which is exactly what we were after.

However, what if we don't early stop T? This bound tells us that the generalization error could get arbitrarily large for as $T \rightarrow \infty$. Does this mean that we heavily overfit if we run AdaBoost for long times? It turns out that in practice, that this is not the case, and that AdaBoost only improves with time.

3 Intuition for why Adaboost does not overfit.

The result of our work in the above theorem is to bound $\mathrm{err}_S(h_{\text{final}})$, which is a sum of terms of the form $\mathbf{1}(y_i \neq h_{\text{final}}(x_i))$. However, because our analysis provides guarantees for $\bar{h}_{\text{final}} = \sum \alpha_t h_t(x_i)$ we have shown something stronger.

For some intuition, if the α's are scaled so that they lie in a unit interval, then $g_{\text{final}} = \sum_t \alpha_t h_t / \sum_t \alpha_t$ is a convex combination of predictions, and hence after multiplication by a label y is just some number in [-1, 1], which encodes both whether we were correct, and its confidence.



Any point (x_i, y_i) such that $y_i \cdot g_{\text{final}}(x_i)$ lands in the red region is judged to be incorrect; our analysis so far has shown that the number of samples falling in this region goes to zero quickly, but with a small change, we can extend the right endpoint of the bound (shown above in orange) to a small positive threshold θ, and show that the number of samples in $[-1, \theta]$ goes to zero quickly as well.

So, it is not just that $\{\#\text{ of points } y_i g_{\text{final}}(x_i) \leq 0\}$ goes to 0 as T increases; so does $\{\#\text{ of points } y_i g_{\text{final}}(x_i) \leq \theta\}$. In the case that there are no such points, then for every point x_i, $y_i g_{\text{final}}(x_i) \geq \theta$ — making the prediction not only correct, but also with certainty at least θ.

We can think of g_{final} as a distribution over predictions of the T functions $\{h_t\}_{t=1}^T$. Using this intuition, we can pretend to take draws from $g_{\text{final}}(x_i)$, the i^{th} of which we will call h_i. Even if there are infinitely many hypotheses in the "support" of g_{final}, viewed as a distribution, we could have chosen just a few, and then use a Hoeffding bound to show that a sampled hypothesis is unlikely to be wrong, as such an event would be far away from the mean outcome, which must be a hypothesis which is correct with certainty at least θ. More precisely, taking

$$N = \Theta\left(\frac{1}{\theta^2} \ln \frac{1}{\delta}\right)$$

draws, and realizing the mean is at least theta, for any (x_i, y_i)

$$\mathbb{E}_{h_j \sim g_{\text{final}}}[y_i h_j(x_i)] = y_i g_{\text{final}}(x) \geq \theta \qquad \Longrightarrow \qquad \Pr_{h_1,\ldots,h_N}\left[\frac{1}{N}\sum_{k=1}^{N} y_i h_k(x_i) < 0\right] < \exp\left(-\frac{\theta^2 N}{2}\right) \leq \delta.$$

In effect, a small amount of certainty allows us to approximate g_{final} which potentially is made up off infinitely many base classifiers using a classifier $\bar{g}_{\text{final}}$ which only has N classifiers in it. Since the latter has a small VC dimension, the former cannot overfit either.