Lecture 13: Boosting

Han Shao

Sections 10.1–10.3, UML

In this lecture, we return to the PAC model. In Homework 1, we considered how to boost the confidence of a PAC learner: given an algorithm that $(\varepsilon/2, 1/2)$-PAC learns a class, we can construct an (ε, δ)-PAC learner. We now ask whether we can also boost the *accuracy*: can a learner that performs only slightly better than random guessing be converted into one with arbitrarily small error?

1 Strong and Weak Learners

We use signed labels $\mathcal{Y} = \{-1, +1\}$ and a joint distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, as in our agnostic-learning notes. In this lecture, we assume realizability with respect to the target class $\mathcal{H}$: there exists $h^\star \in \mathcal{H}$ with $\mathrm{err}_{\mathcal{D}}(h^\star) = 0$. The probability in a learning guarantee includes both the training sample and the internal randomness of the learner, written $\Pr_{S,\mathcal{A}}$.

Definition 1 (Strong learnability)**.** *A hypothesis class $\mathcal{H}$ is* strongly learnable *if there is a learning algorithm $\mathcal{A}$ such that, for every realizable distribution $\mathcal{D}$ and every $\varepsilon, \delta \in (0, 1)$, it uses*

$$\mathrm{poly}\left(\frac{1}{\varepsilon}, \frac{1}{\delta}, \mathrm{VCDim}(\mathcal{H})\right)$$

i.i.d. examples from $\mathcal{D}$ and returns a classifier h_S satisfying

$$\Pr_{S,\mathcal{A}}\left[\mathrm{err}_{\mathcal{D}}(h_S) \le \varepsilon\right] \ge 1 - \delta.$$

Such an algorithm is called a strong learner.

Definition 2 (Weak learnability)**.** *Fix $\gamma \in (0, 1/2)$. A hypothesis class $\mathcal{H}$ is γ-*weakly learnable *if there is a learning algorithm $\mathcal{A}$ such that, for every realizable distribution $\mathcal{D}$ and every $\delta \in (0, 1)$, it uses*

$$\mathrm{poly}\left(\frac{1}{\delta}, \mathrm{VCDim}(\mathcal{H})\right)$$

i.i.d. examples from $\mathcal{D}$ and returns a classifier h_S satisfying

$$\Pr_{S,\mathcal{A}}\left[\mathrm{err}_{\mathcal{D}}(h_S) \le \frac{1}{2} - \gamma\right] \ge 1 - \delta.$$

*The algorithm is called a γ-*weak learner*. Here γ is fixed; the polynomial may depend on γ.*

Predicting an independent fair label on each new example has error $1/2$. A weak learner improves on this baseline by an *edge* $\gamma > 0$. The guarantee must hold for every realizable distribution, not merely for one particular distribution.

Are there hypothesis classes that are weakly learnable but not strongly learnable?

Theorem 1 (Weak and strong learnability)**.** *A hypothesis class is weakly PAC learnable if and only if it is strongly PAC learnable.*

The nontrivial direction converts a weak learner into a strong learner. We study the AdaBoost algorithm of Freund and Schapire (1995), which repeatedly calls a weak learner on reweighted versions of the same training sample. We prove its empirical-error guarantee and bound the VC dimension of the resulting class of voting classifiers.

The AdaBoost Algorithm

Fix a labeled sample

$$S = ((x_1, y_1), \dots, (x_m, y_m)), \qquad y_i \in \{-1, +1\}.$$

AdaBoost maintains a distribution P_t on the sample indices $[m]$. Thus $P_t(i)$ is the weight of the ith sample occurrence, and

$$\operatorname{err}_{P_t}(h) := \sum_{i=1}^{m} P_t(i) \mathbb{1}\{h(x_i) \neq y_i\}.$$

Using indices keeps repeated sample occurrences distinct. Let T be the number of boosting rounds, let h_t be the weak hypothesis in round t, and write

$$\varepsilon_t = \operatorname{err}_{P_t}(h_t), \qquad \gamma_t = \frac{1}{2} - \varepsilon_t.$$

In the analysis, we assume that each call succeeds, so $\varepsilon_t \leq 1/2 - \gamma$ for some $\gamma > 0$.

Algorithm 1 AdaBoost

Require: Labeled sample S, number of rounds T, and a weak learner.

1: Initialize $P_1(i) = 1/m$ for every $i \in [m]$.
2: **for** $t = 1, \dots, T$ **do**
3: Run the weak learner on the distribution induced by P_t to obtain h_t.
4: Compute $\varepsilon_t = \sum_{i=1}^{m} P_t(i) \mathbb{1}\{h_t(x_i) \neq y_i\}$.
5: **if** $\varepsilon_t = 0$ **then**
6: Return $h_{\text{final}} = h_t$.
7: **end if**
8: Set $\alpha_t = \frac{1}{2} \ln \frac{1 - \varepsilon_t}{\varepsilon_t}$.
9: Compute $Z_t = \sum_{j=1}^{m} P_t(j) \exp\bigl(-\alpha_t y_j h_t(x_j)\bigr)$.
10: For every $i \in [m]$, update $P_{t+1}(i) = \dfrac{P_t(i) \exp\bigl(-\alpha_t y_i h_t(x_i)\bigr)}{Z_t}$.
11: **end for**
12: Return $h_{\text{final}}(x) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x) \right)$.

We set $\operatorname{sign}(u) = +1$ for $u \geq 0$ and -1 otherwise. For the formulas below, assume $0 < \varepsilon_t < 1/2$ in every round; then α_t is finite and positive.

2 Empirical Error of AdaBoost

Theorem 2 (AdaBoost training-error bound)**.** *Suppose that AdaBoost completes T rounds and that $\varepsilon_t = 1/2 - \gamma_t$ with $\gamma_t > 0$ in every round. Then*

$$\mathrm{err}_S(h_{\mathrm{final}}) \le \exp\left(-2\sum_{t=1}^{T}\gamma_t^2\right).$$

In particular, if $\gamma_t \ge \gamma > 0$ for every t, then

$$\mathrm{err}_S(h_{\mathrm{final}}) \le e^{-2\gamma^2 T}.$$

Consequently, taking

$$T \ge \frac{1}{2\gamma^2}\ln\frac{1}{\varepsilon}$$

ensures $\mathrm{err}_S(h_{\mathrm{final}}) \le \varepsilon$.

Why Reweight the Sample?

The next distribution is chosen to make the preceding hypothesis no better than random guessing. Thus a successful weak learner cannot simply return the same hypothesis in the next round: it must find a hypothesis that captures additional information.

Lemma 1 (The previous hypothesis has error one half)**.** *For every round with $0 < \varepsilon_t < 1/2$,*

$$Z_t = 2\sqrt{\varepsilon_t(1-\varepsilon_t)}, \qquad \mathrm{err}_{P_{t+1}}(h_t) = \frac{1}{2}.$$

Proof. Let

$$C_t = \{i \in [m] : h_t(x_i) = y_i\}, \qquad I_t = \{i \in [m] : h_t(x_i) \ne y_i\}.$$

Under P_t, these sets have masses $1-\varepsilon_t$ and ε_t, respectively. The update multiplies the unnormalized weights of correct predictions by $e^{-\alpha_t}$ and those of incorrect predictions by e^{α_t}. Hence

$$\begin{aligned} P_{t+1}(C_t) &= \frac{(1-\varepsilon_t)e^{-\alpha_t}}{Z_t} \\ &= \frac{1-\varepsilon_t}{Z_t}\sqrt{\frac{\varepsilon_t}{1-\varepsilon_t}} = \frac{\sqrt{\varepsilon_t(1-\varepsilon_t)}}{Z_t}, \\ P_{t+1}(I_t) &= \frac{\varepsilon_t e^{\alpha_t}}{Z_t} \\ &= \frac{\varepsilon_t}{Z_t}\sqrt{\frac{1-\varepsilon_t}{\varepsilon_t}} = \frac{\sqrt{\varepsilon_t(1-\varepsilon_t)}}{Z_t}. \end{aligned}$$

The two masses are equal and sum to one, proving both claims. □

Proof of the Training-Error Bound

The proof has three steps. First, express the final sample weights in terms of the weighted vote. Second, use these weights to bound the training error by $\prod_{t=1}^{T} Z_t$. Third, show that this product is exponentially small.

Write the real-valued voting score as

$$\overline{h}_{\mathrm{final}}(x) = \sum_{t=1}^{T}\alpha_t h_t(x), \qquad h_{\mathrm{final}}(x) = \mathrm{sign}\big(\overline{h}_{\mathrm{final}}(x)\big).$$

Lemma 2 (Unrolling the weight updates)**.** *For every $i \in [m]$,*

$$P_{T+1}(i) = \frac{\exp\bigl(-y_i \overline{h}_{\text{final}}(x_i)\bigr)}{m \prod_{t=1}^{T} Z_t}.$$

Proof. Repeatedly applying the update rule gives

$$\begin{aligned} P_{T+1}(i) &= P_1(i) \prod_{t=1}^{T} \frac{\exp\bigl(-\alpha_t y_i h_t(x_i)\bigr)}{Z_t} \\ &= \frac{1}{m} \frac{\exp\Bigl(-y_i \sum_{t=1}^{T} \alpha_t h_t(x_i)\Bigr)}{\prod_{t=1}^{T} Z_t} \\ &= \frac{\exp\bigl(-y_i \overline{h}_{\text{final}}(x_i)\bigr)}{m \prod_{t=1}^{T} Z_t}. \end{aligned}$$

□

If the final classifier is wrong at x_i, then $y_i \overline{h}_{\text{final}}(x_i) \le 0$. Since $\mathbb{1}\{z \le 0\} \le e^{-z}$, we have

$$\mathbb{1}\{h_{\text{final}}(x_i) \neq y_i\} \le \mathbb{1}\{y_i \overline{h}_{\text{final}}(x_i) \le 0\} \le \exp\bigl(-y_i \overline{h}_{\text{final}}(x_i)\bigr). \tag{1}$$

The first inequality accommodates a zero voting score.

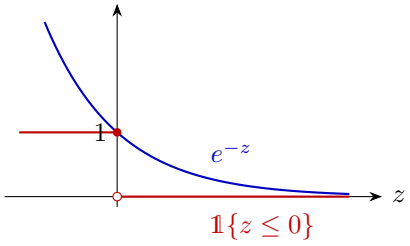


Figure 1: Exponential loss upper-bounds the zero-margin indicator, which in turn upper-bounds the classification error.

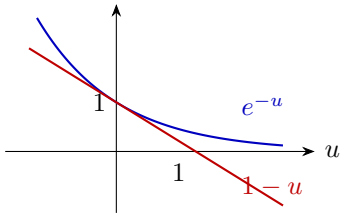


Figure 2: The inequality $1-u \le e^{-u}$ bounds each normalization factor by $e^{-2\gamma_t^2}$.

Lemma 3 (Bounding training error by normalization factors)**.** *The final classifier satisfies*

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

Proof. Using (1) and lemma 2,

$$\begin{aligned} \text{err}_S(h_{\text{final}}) &= \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}\{h_{\text{final}}(x_i) \neq y_i\} \\ &\le \frac{1}{m} \sum_{i=1}^{m} \exp\bigl(-y_i \overline{h}_{\text{final}}(x_i)\bigr) \\ &= \sum_{i=1}^{m} P_{T+1}(i) \prod_{t=1}^{T} Z_t \end{aligned}$$

$$= \prod_{t=1}^{T} Z_t.$$

The last equality uses $\sum_{i=1}^{m} P_{T+1}(i) = 1$. □

Lemma 4 (Exponential decay of the product)**.** *If $\varepsilon_t = 1/2 - \gamma_t$, then*

$$\prod_{t=1}^{T} Z_t \le \exp\left(-2\sum_{t=1}^{T} \gamma_t^2\right).$$

Proof. By lemma 1,

$$\begin{aligned} Z_t &= 2\sqrt{\varepsilon_t(1-\varepsilon_t)} \\ &= 2\sqrt{\left(\frac{1}{2} - \gamma_t\right)\left(\frac{1}{2} + \gamma_t\right)} = \sqrt{1 - 4\gamma_t^2}. \end{aligned}$$

Using $1 - u \le e^{-u}$,

$$\begin{aligned} \prod_{t=1}^{T} Z_t &= \prod_{t=1}^{T} \sqrt{1 - 4\gamma_t^2} \\ &\le \prod_{t=1}^{T} \sqrt{\exp(-4\gamma_t^2)} \\ &= \exp\left(-2\sum_{t=1}^{T} \gamma_t^2\right). \end{aligned}$$

□

Proof of theorem 2. Combine lemmas 3 and 4. If $\gamma_t \ge \gamma$ for every t, then $\sum_{t=1}^{T} \gamma_t^2 \ge T\gamma^2$, giving the stated round bound. □

3 Generalization Error of AdaBoost

The training-error bound shows that AdaBoost can achieve arbitrarily small empirical error. Our ultimate goal, however, is small true error $\mathrm{err}_{\mathcal{D}}(h_{\mathrm{final}})$.

For this section, let $\mathcal{H}$ be a fixed class containing all hypotheses returned by the weak learner.[1] Define the class of weighted votes of T base hypotheses by

$$\mathcal{L}(\mathcal{H}, T) := \left\{ x \mapsto \operatorname{sign}\left(\sum_{t=1}^{T} w_t h_t(x)\right) : w \in \mathbb{R}^T,\ h_1, \ldots, h_T \in \mathcal{H} \right\}.$$

The output of AdaBoost belongs to this class, with $w = (\alpha_1, \ldots, \alpha_T)$. Although AdaBoost uses nonnegative weights, allowing arbitrary real weights only enlarges the class.

Each such classifier first maps x to the prediction vector $(h_1(x), \ldots, h_T(x)) \in \mathbb{R}^T$ and then applies a homogeneous linear separator to that vector. Thus, bounding the VC dimension of $\mathcal{L}(\mathcal{H}, T)$ lets us apply the generalization bounds developed in previous lectures.

[1] The handwriting uses $\mathcal{H}$ both for the target class and for the class of weak hypotheses. The VC-dimension argument requires that all h_t belong to the fixed class whose VC dimension is being bounded. Weak learnability as defined above does not by itself require the learner to be proper.

Theorem 3 (VC dimension of weighted votes)**.** *Let $d = \mathrm{VCDim}(\mathcal{H}) \geq 1$, and let $T \geq 1$ be an integer. There is a universal constant $C > 0$ such that*

$$\mathrm{VCDim}\big(\mathcal{L}(\mathcal{H}, T)\big) \leq C\, dT \ln(2dT).$$

Proof. Let $U = \{u_1, \ldots, u_n\}$ be a set shattered by $\mathcal{L}(\mathcal{H}, T)$. We count the labelings induced on U. If $n < \max\{d, T\}$, then $n \leq dT$, which already gives the desired bound after adjusting C. Assume henceforth that $n \geq \max\{d, T\}$.

Choosing the base-hypothesis labelings. By Sauer's lemma, the number of labelings of U induced by $\mathcal{H}$ is at most

$$\Pi_{\mathcal{H}}(n) \leq \Big(\frac{en}{d}\Big)^{d}.$$

Choosing an ordered T-tuple of such labelings therefore gives at most

$$\Big(\frac{en}{d}\Big)^{dT}$$

possible prediction matrices. Repetitions are allowed.

Choosing the linear separator. Fix the predictions of $h_1, \ldots, h_T$ on U. Each point u_i now corresponds to a fixed vector

$$v_i = (h_1(u_i), \ldots, h_T(u_i)) \in \mathbb{R}^T.$$

The remaining choice is a homogeneous halfspace on these vectors. Since homogeneous halfspaces in $\mathbb{R}^T$ have VC dimension T, Sauer's lemma bounds the number of resulting labelings by

$$\Big(\frac{en}{T}\Big)^{T}.$$

Combining the two choices, and using that U is shattered, yields

$$2^n \leq \Big(\frac{en}{d}\Big)^{dT} \Big(\frac{en}{T}\Big)^{T} \leq (en)^{(d+1)T}.$$

Let $a = (d+1)T$. Taking logarithms gives

$$n \ln 2 \leq a \ln(en).$$

To make the last step explicit, set $q = n \ln 2 / a$. Then

$$q \leq \ln \frac{ea}{\ln 2} + \ln q \leq \ln \frac{ea}{\ln 2} + \frac{q}{2},$$

where $\ln q \leq q/2$ for every $q > 0$. Hence

$$n \leq \frac{2a}{\ln 2} \ln \frac{ea}{\ln 2} = O\big(dT \ln(2dT)\big),$$

since $a = (d+1)T \leq 2dT$. This bounds the size of every shattered set and proves the theorem. □

Thus the small empirical error from theorem 2 can be combined with a generalization bound for the fixed class $\mathcal{L}(\mathcal{H}, T)$. The VC bound controls all possible choices of the weak hypotheses and their voting weights, including choices made adaptively from the sample.