Topic 7: Boosting

7.1 Weak versus strong PAC learning

The problem of boosting originated in the context of PAC learning. Recall that a learning algorithm A is a PAC learner for concept class $\mathcal{C} \subseteq \{-1,1\}^{\mathcal{X}}$ if it has the following property. For $\epsilon, \delta \in (0,1)$, and any data distribution P that is $\mathcal{C}$-realizable, if A is provided as input an iid sample of size-n from P (where n is the sample complexity of A, which may depend on ϵ, δ, $\mathcal{C}$, etc.), then with probability at least $1-\delta$, it returns a hypothesis $\mathbf{h}\colon \mathcal{X} \to \{-1,1\}$ such that $\mathsf{err}(\mathbf{h}; P) \leq \epsilon$. If A is also a computationally-efficient algorithm (in a suitable sense that can been made precise), then A is an *efficient PAC learner for $\mathcal{C}$*. To distinguish PAC learning from what we are about to describe, we will also call such learning algorithms *strong (PAC) learners for $\mathcal{C}$*.

We say a learning algorithm A is a *weak (PAC) learner for $\mathcal{C}$* if it has all of the requirements of being a strong learner for $\mathcal{C}$, except for the requirement that it works for all ϵ. Instead, it is only required to succeed for a particular $\epsilon = \epsilon_{\mathrm{o}}$, where $\epsilon_{\mathrm{o}} \in (0, 1/2)$. Here, it is important that ϵ_{o} is strictly bounded away from $1/2$, since achieving error rate $1/2$ is trivial for binary label spaces.

A question asked by Kearns and Valiant (1989) is: does the existence of an efficient weak learner imply the existence of an efficient strong learner? That is, is strong learning a concept class as "easy" (in the computational complexity sense of polynomial-time equivalence) as weak learning the same concept class? Schapire (1990) answered the question affirmatively by designing a *boosting algorithm*: an algorithmic reduction from strong learning to weak learning. The boosting algorithm achieves the strong learning guarantee for a concept class $\mathcal{C}$ by using repeatedly invoking a weak learner for $\mathcal{C}$ as a subroutine. After Schapire's original proof was found, several other boosting algorithms have been developed, with the AdaBoost algorithm of Freund and Schapire (1997) probably being the most famous.

The key idea in boosting algorithms is to exploit the requirement that weak learners succeed under any marginal distribution over $\mathcal{X}$. All boosting algorithms work by invoking the weak learner under different distributions, and then use (or combine) the outputs of the weak learners (called "weak hypotheses") in some way.

Exercise 7.1. The original definition of weak learner, in fact, has two exceptions: the algorithm only needs to work for $\epsilon = \epsilon_{\mathrm{o}}$ for some $\epsilon_{\mathrm{o}} \in (0, 1/2)$, and also only for $\delta = \delta_{\mathrm{o}}$ for some $\delta_{\mathrm{o}} \in (0,1)$. Suppose you have such an efficient weak learner for $\mathcal{C}$. Explain how to construct an efficient learning algorithm that $(\epsilon_{\mathrm{o}} + \varepsilon, \delta)$-PAC learns $\mathcal{C}$ for any $\varepsilon \in (0,1)$ and $\delta \in (0,1)$. What is the sample complexity of your algorithm? $\Diamond$

7.2 In-sample boosting

7.2.1 In-sample weak learning

It is a little simpler to think about weak versus strong learning in the context of a fixed training data set—i.e., "in-sample" weak versus strong learning—and then worry only later about how to get "out-of-sample" guarantees. To facilitate this discussion, we use a modified definition of weak learners that is tailored for a fixed training data set.

Let $S := ((x_1, y_1), \ldots, (x_n, y_n)) \in (\mathcal{X} \times \{-1, 1\})^n$ be size-n training data set. (For now, we do not need to assume that it is an iid sample.) An *(in-sample) weak learner* for S is an algorithm that, given any *(re-)weighting* $p = (p_1, \ldots, p_n) \in \Delta^{n-1}$ of the training examples, returns a hypothesis $h \colon \mathcal{X} \to \{-1, 1\}$ with *p-weighted (empirical) accuracy*

$$\mathsf{acc}(h; S, p) := \sum_{i=1}^{n} p_i \, \mathbb{1}\{h(x_i) = y_i\}$$

satisfying

$$\mathsf{acc}(h; S, p) \geq \frac{1 + \gamma}{2} \tag{7.1}$$

for some $\gamma > 0$. The number γ is called the *advantage* of h over random guessing (with respect to the re-weighting p), and $(1 + \gamma)/2$ corresponds to what we previously called $1 - \epsilon_{\mathrm{o}}$. The ability to deal with any re-weighting of the training data set is analogous to the requirement that weak learners of the flavor from Section 7.1 be able to work with any marginal distribution over $\mathcal{X}$.[1] The assumption that there is a weak learner for the dataset with advantage γ is called the *γ-weak learning assumption (γ-WLA)*.

Example 7.1. Recall Exercise 3.4. For an odd number $s \in [d]$, consider the concept class $\mathsf{SUBCOM}_{d,s} \subseteq \{-1, 1\}^{\mathcal{X}}$ over the input space $\mathcal{X} := \{-1, 1\}^d$, defined by

$$\mathsf{SUBCOM}_{d,s} := \{c_T \mid T \subseteq [d], |T| = s\}$$

where

$$c_T(x) := \operatorname{sign}\left(\sum_{j \in T} x_j\right).$$

(It can be checked the $\mathsf{VCdim}(\mathsf{SUBCOM}_{d,s}) = \Theta(s \log(d/s))$ whenever $s \leq d/2$.)

We claim that the following algorithm is an in-sample weak learner for any data set $S := ((x_1, y_1), \ldots, (x_n, y_n)) \in (\mathcal{X} \times \{-1, 1\})^n$ labeled by some $c_T \in \mathsf{SUBCOM}_{d,s}$:

> Given a weighting $p = (p_1, \ldots, p_n) \in \Delta^{n-1}$, determine
>
> $$j^* := \operatorname*{arg\,max}_{j \in [d]} \sum_{i=1}^{n} p_i \, y_i x_{i,j},$$
>
> and return $h(x) := x_{j^*}$.

Observe that for each $i \in [n]$, we have

$$y_i \sum_{j \in T} x_{i,j} \geq 1,$$

and therefore

$$\sum_{i=1}^{n} p_i y_i x_{i,j^*} = \max_{j \in T} \sum_{i=1}^{n} p_i \, y_i x_{i,j} \geq \frac{1}{s} \sum_{j \in T} \sum_{i=1}^{n} p_i \, y_i x_{i,j} \geq \frac{1}{s}.$$

[1] Given a weak learner for a concept class $\mathcal{C}$ of the type from Section 7.1, it is possible to design an efficient in-sample weak learner for any training data set whose labels are provided by some concept from $\mathcal{C}$.

This implies

$$\mathsf{acc}(h; S, p) = \sum_{i=1}^{n} p_i \, \mathbb{1}\{h(x_i) = y_i\} = \sum_{i=1}^{n} p_i \, \frac{1 + y_i x_{i,j^*}}{2} = \frac{1 + \sum_{i=1}^{n} p_i \, y_i x_{i,j^*}}{2} \geq \frac{1 + 1/s}{2}.$$

So the γ-WLA holds for S with $\gamma = 1/s$. $\Diamond$

7.2.2 An alternative characterization of the weak learning assumption

Before we describe the boosting algorithm (i.e., the algorithmic reduction from strong learning to weak learning), we first give an alternative characterization of the γ-WLA, which will in turn motivate the boosting algorithm.

Consider all possible hypotheses that the weak learner could output, and call this set $\mathcal{H}$. For simplicity, assume $\mathcal{H}$ is finite, say, $\mathcal{H} = \{h_1, \ldots, h_d\}$. Define an $n \times d$ matrix A by

$$A_{i,j} := \mathbb{1}\{h_j(x_i) = y_i\}.$$

The γ-WLA says that for any $p \in \Delta^{n-1}$, there is some $j \in [d]$ such that

$$p^\mathsf{T} A e_j \geq \frac{1 + \gamma}{2},$$

where e_j is the j-th coordinate basis vector in $\mathbb{R}^d$. (The above expression is equivalent to (7.1) for $h = h_j$.) In other words,

$$\min_{p \in \Delta^{n-1}} \max_{q \in \{e_1, \ldots, e_d\}} p^\mathsf{T} A q \geq \frac{1 + \gamma}{2}. \tag{7.2}$$

Since linear functions over the probability simplex Δ^{d-1} are extremized at the vertices of the simplex, (7.2) is equivalent to

$$\min_{p \in \Delta^{n-1}} \max_{q \in \Delta^{d-1}} p^\mathsf{T} A q \geq \frac{1 + \gamma}{2}. \tag{7.3}$$

By the Von Neumann's minimax theorem (Theorem 2.3),

$$\min_{p \in \Delta^{n-1}} \min_{q \in \Delta^{d-1}} p^\mathsf{T} A q = \max_{q \in \Delta^{d-1}} \min_{p \in \Delta^{n-1}} p^\mathsf{T} A q,$$

so (7.2) (or (7.3)) is equivalent to

$$\max_{q \in \Delta^{d-1}} \min_{p \in \Delta^{n-1}} p^\mathsf{T} A q \geq \frac{1 + \gamma}{2}. \tag{7.4}$$

Again, since linear functions over Δ^{n-1} are extremized at the vertices, (7.4) is equivalent to

$$\max_{q \in \Delta^{d-1}} \min_{i \in [n]} e_i^\mathsf{T} A q \geq \frac{1 + \gamma}{2}. \tag{7.5}$$

This means that there exists some $q = (q_1, \ldots, q_d) \in \Delta^{d-1}$ such that, for all $i \in [n]$,

$$\begin{aligned} \sum_{j=1}^{d} q_j \, \mathbb{1}\{h_j(x_i) = y_i\} &= \sum_{j=1}^{d} q_j \frac{1 + y_i h_j(x_i)}{2} \\ &= \frac{1 + y_i \sum_{j=1}^{d} q_j h_j(x_i)}{2} \\ &\geq \frac{1 + \gamma}{2}. \end{aligned}$$

This is equivalent to the existence of $q \in \Delta^{d-1}$ such that

$$y_i \sum_{j=1}^{d} q_j h_j(x_i) \geq \gamma \quad \forall i \in [n]. \tag{7.6}$$

So, in the "feature space" $\{-1,1\}^d$ where the feature vector for $x \in \mathcal{X}$ is $\phi(x) := (h_1(x), \ldots, h_d(x))$, the γ-WLA implies *linearly separability* of the training data set, i.e., the existence of a linear threshold function that classifies all of the training examples correctly.

Since γ-WLA implies linear separability, one possible approach to strong learning would be to simply apply the feature map ϕ to all of the training examples, i.e.,

$$S' := ((\phi(x_1), y_1), \ldots, (\phi(x_n), y_n)) \in (\{-1,1\}^d \times \{-1,1\})^n,$$

and then apply a strong learner for LTF_d to this transformed training data set S', such as that described in Example 6.1. This gives a strong PAC learner with sample complexity and time complexity polynomial in $1/\epsilon$, $1/\delta$, and d.

There are at least two potential issues with the above approach to strong learning.

1. The first issue is computational. It is possible that $|\mathcal{H}| = d$ is very large, and yet the weak learner is able to efficiently find a suitable hypothesis in $\mathcal{H}$ in $o(d)$ time. Yet the approach to strong learning we described seems to require $\Omega(d)$ time; for instance, it applies the feature map $\phi\colon \mathcal{X} \to \{-1,1\}^d$ to the training data.

2. The second issue statistical. The weak learner uses a hypothesis class $\mathcal{H}$ with $\mathsf{VCdim}(\mathcal{H}) \leq \log_2(d)$. However, the proposed strong learner uses a hypothesis class LTF_d, which has $\mathsf{VCdim}(\mathsf{LTF}_d) = d + 1$.

Both issues may be even more pronounced if $|\mathcal{H}| = \infty$ but $\mathsf{VCdim}(\mathcal{H}) < \infty$. Fortunately, both issues can be overcome by taking advantage of (7.6) and how we arrived at this alternative characterization.

7.2.3 Approximately solving the game using Hedge

Recall that the Von Neumann minimax theorem (Theorem 2.3) has an algorithmic proof based on executing Hedge (Algorithm 2.2) in a particular instance of the Online Allocation problem. From the execution of Hedge in this context (Proposition 2.4), there is a way to obtain (in the terminology of Theorem 2.3) $p \in \Delta^{n-1}$ and $q \in \Delta^{d-1}$ that approximately achieve the optimal value of the game (which is at least $(1+\gamma)/2$ in our case). This suggests a simple algorithm based on Hedge (Freund and Schapire, 1997), given as Algorithm 7.1 below. The difference between Algorithm 7.1 and the algorithm described in Proposition 2.4 is that in Algorithm 7.1, WL does not necessarily provide a "best response" in each round; it may only give a "good enough response" for the purposes of satisfying the γ-WLA. Nevertheless, we can still follow the analysis of Hedge to analyze Algorithm 7.1.

Algorithm 7.1: EtaBoost for in-sample boosting.

Hyperparameters: $\eta > 0$ and $T \in \mathbb{N}$.
Input: data set $S := ((x_1, y_1), \ldots, (x_n, y_n)) \in (\mathcal{X} \times \{-1, 1\})^n$; in-sample weak learner WL for S.

- Initially, set $p_1 := (1/n, \ldots, 1/n) \in \Delta^{n-1}$.
- For $t = 1, 2, \ldots, T$:
  - Invoke WL for data set S with weighting $p_t := (p_{t,1}, \ldots, p_{t,n}) \in \Delta^{n-1}$ to obtain hypothesis $h_t \colon \mathcal{X} \to \{-1, 1\}$.
  - Define loss vector $\ell_t := (\ell_{t,1}, \ldots, \ell_{t,n}) \in \{0, 1\}^n$ by
  $$\ell_{t,i} := \mathbb{1}\{h_t(x_i) = y_i\}.$$
  - Compute new weighting $p_{t+1} := (p_{t+1,1}, \ldots, p_{t+1,n}) \in \Delta^{n-1}$:
  $$p_{t+1,i} := \frac{p_{t,i} \exp(-\eta \ell_{t,i})}{Z_{t+1}} \quad \text{for all } i \in [n],$$
  where
  $$Z_{t+1} := \sum_{i=1}^{n} p_{t,i} \exp(-\eta \ell_{t,i}).$$
- Return: final hypothesis $\hat{h}(x) := \operatorname{sign}(\hat{g}(x))$ where $\hat{g}(x) := \frac{1}{T} \sum_{t=1}^{T} h_t(x)$.

The analysis of Algorithm 7.1 is much like the proof of Proposition 2.4; we simply have to interpret the guarantees for Hedge (Algorithm 2.2) in this setting.

Theorem 7.1. *Suppose the γ-Weak Learning Assumption for $S := ((x_1, y_1), \ldots, (x_n, y_n)) \in (\mathcal{X} \times \{-1, 1\})^n$ holds, and $\hat{h} \colon \mathcal{X} \to \{-1, 1\}$ is the output of Algorithm 7.1 when executed with S and an in-sample weak learner for S. If the hyperparameters satisfy $\eta := 2\gamma$ and $T > 2\ln(n)/\gamma^2$, then $\hat{h}(x_i) = y_i$ for all $i \in [n]$.*

Proof. The weightings p_t are computed as in Hedge, with loss vectors ℓ_t determined by the hypotheses h_t returned by the weak learner. Theorem 2.2 implies the following: for any $u \in \Delta^{n-1}$,

$$\frac{1}{T} \sum_{t=1}^{T} \langle p_t, \ell_t \rangle \leq \frac{1}{T} \sum_{t=1}^{T} \langle u, \ell_t \rangle + \frac{\ln(n)}{\eta T} + \frac{\eta}{8}. \tag{7.7}$$

Let us interpret the inequality in (7.7). First, the left-hand side satisfies

$$\begin{aligned} \frac{1}{T} \sum_{t=1}^{T} \langle p_t, \ell_t \rangle &= \frac{1}{T} \sum_{t=1}^{T} \left(\sum_{i=1}^{n} p_{t,i} \, \mathbb{1}\{h_t(x_i) = y_i\} \right) \\ &= \frac{1}{T} \sum_{t=1}^{T} \mathsf{acc}(h_t; S, p_t) \\ &\geq \frac{1 + \gamma}{2} \end{aligned}$$

where the inequality follows from the γ-WLA. Now suppose $u = e_i$ for some $i \in [n]$. Then

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\langle e_i, \ell_t\rangle &= \frac{1}{T}\sum_{t=1}^{T}\mathbb{1}\{h_t(x_i) = y_i\} \\ &= \frac{1}{T}\sum_{t=1}^{T}\frac{1 + y_i h_t(x_i)}{2} \\ &= \frac{1 + y_i \hat{g}(x_i)}{2}.\end{aligned}$$

Therefore, via (7.7), we have

$$\frac{1 + y_i \hat{g}(x_i)}{2} \geq \frac{1+\gamma}{2} - \frac{\ln(n)}{\eta T} - \frac{\eta}{8} \quad \text{for all } i \in [n].$$

This is equivalent to

$$y_i \hat{g}(x_i) \geq \gamma - \frac{2\ln(n)}{\eta T} - \frac{\eta}{4} \quad \text{for all } i \in [n].$$

Thus, the choice of the hyperparameters η and T ensure that $y_i \hat{g}(x_i) > 0$ for all $i \in [n]$, which means that $\hat{h}(x_i) = y_i$ for all $i \in [n]$. □

A slight variation of the proof of Theorem 7.1 shows how the error rate on S decreases with T.

Theorem 7.2. *In the same setting as Theorem 7.1, if $\eta := 2\gamma$ and $M := \{i \in [n] \mid y_i \hat{g}(x_i) \leq 0\}$ (where $\hat{g}$ is from Algorithm 7.1), then*

$$|M| \leq n \exp\left(-\frac{\gamma^2 T}{2}\right).$$

Proof. By Theorem 2.2, we have

$$\frac{1}{T}\sum_{t=1}^{T}\langle p_t, \ell_t\rangle \leq \frac{1}{T}\sum_{t=1}^{T}\langle u, \ell_t\rangle + \frac{\mathsf{RE}(u, p_1)}{\eta T} + \frac{\eta}{8}. \tag{7.8}$$

for all $u \in \Delta^{n-1}$. Let u be the uniform distribution over M. Then

$$\mathsf{RE}(u, p_1) = \sum_{i=1}^{n} u_i \ln\left(\frac{u_i}{1/n}\right) = -\ln(|M|/n).$$

Then, as in the proof of Theorem 7.1 (using the γ-WLA),

$$\frac{1}{T}\sum_{t=1}^{T}\langle p_t, \ell_t\rangle \geq \frac{1+\gamma}{2},$$

and

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\langle u, \ell_t\rangle &= \sum_{i=1}^{n} u_i \cdot \frac{1}{T}\sum_{t=1}^{T}\langle e_i, \ell_t\rangle \\ &= \sum_{i=1}^{n} u_i \cdot \frac{1 + y_i\hat{g}(x_i)}{2} \\ &= \frac{1}{2} + \frac{1}{2}\sum_{i=1}^{n} u_i \cdot y_i\hat{g}(x_i) \\ &\leq \frac{1}{2}.\end{aligned}$$

So, from (7.8), we obtain

$$\frac{1+\gamma}{2} \leq \frac{1}{2} - \frac{\ln(|M|/n)}{\eta T} + \frac{\eta}{8}.$$

Plugging in the choice $\eta = 2\gamma$ and re-arranging gives the claimed upper bound on $|M|$. □

Theorems 7.1 and 7.2 show that the γ-WLA for a data set S not only implies the linear separability of S, but also the existence of an efficient algorithm for finding a LTF that correctly classifies all (or approximately all) of the examples in S. This addresses the computational issue raised at the end of Section 7.2.2.

The AdaBoost algorithm of Freund and Schapire (1997) is an improved version of Algorithm 7.1 that removes the hyperparameter η. Instead, it performs an update in each round t similar to that of Algorithm 7.1, except that the update is based on the actual advantage γ_t of h_t, rather than the lower bound γ guaranteed by the γ-WLA. Another difference is that the final hypothesis given by AdaBoost is a weighted majority vote, where the weight of h_t is based on the actual advantage of h_t. These are important modifications that make AdaBoost a very practical and successful machine learning algorithm.

7.3 Out-of-sample performance of boosting

7.3.1 VC dimension of majority votes over hypotheses

Having shown how the γ-WLA leads to an efficient algorithm for finding a hypothesis that classifies all (or approximately all) of the examples "in-sample", it remains to understand how that hypothesis performs "out-of-sample". That is, if Algorithm 7.1 is executed on an iid sample $\mathbf{S} = ((\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n)) \sim P^n$ for some $\mathcal{C}$-realizable distribution P, what is the error rate of the final hypothesis on P?

A first step towards answering this question is to consider the VC dimension of the hypothesis class being used by the boosting algorithm. If the weak learner uses hypothesis class $\mathcal{H} \subseteq \{-1, 1\}^{\mathcal{X}}$, then the boosting algorithm uses the class of *majority votes over T hypotheses from $\mathcal{H}$*:

$$\mathsf{MAJ}_{\mathcal{H},T} := \left\{ x \mapsto \operatorname{sign}\left(\frac{1}{T}\sum_{t=1}^{T} h_t(x)\right) \;\middle|\; (h_1, \ldots, h_T) \in \mathcal{H}^T \right\}.$$

Proposition 7.1. $\mathsf{VCdim}(\mathsf{MAJ}_{\mathcal{H},T}) \leq 2d_{\mathsf{WL}}T(\log_2(eT) + 0.1)$ *where* $d_{\mathsf{WL}} := \mathsf{VCdim}(\mathcal{H})$.

Proof. Suppose $S := \{x_1, \ldots, x_m\}$ is a set of cardinality m shattered by $\mathsf{MAJ}_{\mathcal{H},T}$. Then it suffices to show that $m \leq 2d_{\mathsf{WL}}T(\log_2(eT) + 0.1)$. Every $v \in \mathsf{MAJ}_{\mathcal{H},T}[S]$ is determined by a choice of T behaviors from $\mathcal{H}[S]$, and hence

$$|\mathsf{MAJ}_{\mathcal{H},T}[S]| \leq |\mathcal{H}[S]|^T.$$

By Sauer's lemma (Lemma 5.2), we have

$$|\mathcal{H}[S]|^T \leq \binom{m}{\leq d}^T \leq \left(\frac{em}{d_{\mathsf{WL}}}\right)^{d_{\mathsf{WL}}T}.$$

On the other hand, since S is shattered by $\mathsf{MAJ}_{\mathcal{H},T}$,

$$|\mathsf{MAJ}_{\mathcal{H},T}[S]| = 2^m.$$

So we have

$$m \leq d_{\mathsf{WL}}T\log_2\left(\frac{em}{d_{\mathsf{WL}}}\right).$$

Letting $u := m/(d_{\mathsf{WL}}T)$, the inequality above becomes

$$u \leq \log_2(eT) + \log_2(u).$$

It can be checked that $\log_2(u) \leq u/2 + 0.1$ for all $u > 0$, so

$$u \leq 2(\log_2(eT) + 0.1).$$

Hence $m \leq 2d_{\mathsf{WL}}T(\log_2(eT) + 0.1)$. □

7.3.2 Generalization guarantee for boosting

We can now give a generalization guarantee for boosting under the γ-WLA. The main idea is to leverage uniform convergence of training error rates for $\mathsf{MAJ}_{\mathcal{H},T}$ (which makes use of Proposition 7.1), along with the training error rate guarantee of boosting the γ-WLA from Theorem 7.2.

Theorem 7.3. *Let P be a probability distribution over $\mathcal{X} \times \{-1, 1\}$ such that $\mathbf{S} \sim P^n$ always satisfies the γ-WLA, and let $\delta \in (0, 1)$. If $\hat{\mathbf{h}}$ is the output of Algorithm 7.1 (with hyperparameters $\eta := \gamma$ and T) on input $\mathbf{S}$ and an in-sample weak learner for $\mathbf{S}$ with advantage γ that always outputs hypotheses from $\mathcal{H}$, then with probability at least $1 - \delta$,*

$$\mathsf{err}(\hat{\mathbf{h}}; P) \leq \exp\left(-\frac{\gamma^2 T}{2}\right) + O\left(\sqrt{\frac{V_{d_{\mathsf{WL}},T}\log(n/V_{d_{\mathsf{WL}},T}) + \log(1/\delta)}{n}}\right),$$

where $d_{\mathsf{WL}} = \mathsf{VCdim}(\mathcal{H})$ and $V_{d_{\mathsf{WL}},T} = O(d_{\mathsf{WL}}T\log(T))$.

Using $T := 2\ln(2/\epsilon)/\gamma^2$, it follows that the error rate bound is at most ϵ provided that the number of training data n is at least

$$O\left(\frac{d_{\mathsf{WL}}}{\epsilon^2\gamma^2}\log^2\left(\frac{1}{\epsilon}\right)\log\left(\frac{1}{\gamma^2}\log\frac{1}{\epsilon}\right) + \frac{1}{\epsilon^2}\log\left(\frac{1}{\delta}\right)\right).$$

Each appearance of ϵ^2 can be be replaced by ϵ by using the improved version of the VC theorem alluded to in the context of Theorem 6.1. Also, some logarithmic factors can be removed by directly using the growth function for $\mathsf{MAJ}_{\mathcal{H},T}$ instead of its VC dimension. But even without these improvements, the quantity is at most a polynomial in d_{WL}, $1/\gamma$, $1/\epsilon$, and $1/\delta$.

7.4 Margins theory

7.4.1 An empirical puzzle

In Theorem 7.3, the error rate bound has two terms: one that decreases with T, and another that increases with T. This suggests a trade-off that should be optimized by carefully choosing T (as done in the previous section). However, in practice, it is occasionally observed with boosting algorithms (like AdaBoost, for which T is the only hyperparameter) that the population error rate only decreases with T, even beyond the point at which the training error rate is zero. To provide a theoretical explanation of this phenomenon, Schapire et al. (1998) gave a different analysis of boosting based on what they call the margins theory.

Recall that the output of Algorithm 7.1 is $\hat{h} = \operatorname{sign} \circ \hat{g}$ where $\hat{g}(x) = \frac{1}{T} \sum_{t=1}^{T} h_t(x)$. In particular, $\hat{g}$ is a *convex combination* of hypotheses from $\mathcal{H}$; the set of such convex combintations is called the *convex hull* of $\mathcal{H}$, written

$$\operatorname{conv}(\mathcal{H}) := \left\{ \sum_{i=1}^{d} \alpha_i h_i \;\middle|\; d \in \mathbb{N},\ \alpha = (\alpha_1, \ldots, \alpha_d) \in \Delta^{d-1},\ (h_1, \ldots, h_d) \in \mathcal{H}^d \right\}.$$

Since $h(x) \in \{-1, 1\}$ for each $h \in \mathcal{H}$, it follows that $\hat{g}(x) \in [-1, 1]$. Although only the sign of $\hat{g}(x)$ is used for prediction, the magnitude of $\hat{g}(x)$ can be regarded as a measure of confidence in the prediction. Let us call $y\hat{g}(x)$ the *margin* of $\hat{g}$ on $(x, y) \in \mathcal{X} \times \{-1, 1\}$.

Schapire et al. (1998) found that, in several experiments, running AdaBoost with T larger than that sufficient to achieve zero training error rate resulted in $\hat{g}$ with larger margins on larger fractions of the training data set. For example, the following was observed in one such experiment using AdaBoost with a decision tree learner as the Weak Learning algorithm:

	T		
	5	100	1000
Training error rate	0.0%	0.0%	0.0%
Test error rate	8.4%	3.3%	3.1%
% margins below 0.5	7.7%	0.0%	0.0%
Minimum margin	0.14	0.52	0.55

Moreover, in these experiments, it appeared that larger margins on the training data set correlated with better out-of-sample performance. Schapire et al. (1998) sought to explain this phenomenon with their margins theory.

7.4.2 Margin errors

For a data set $S := ((x_1, y_1), \ldots, (x_n, y_n)) \in (\mathcal{X} \times \{-1, 1\})^n$, $f \colon \mathcal{X} \to [-1, 1]$, and $\theta > 0$, define the *θ-margin error rate* by

$$\mathsf{err}_\theta(f; S) := \frac{1}{n} \sum_{i=1}^{n} \ell_\theta(y f(x)),$$

where $\ell_\theta \colon \mathbb{R} \to \mathbb{R}_+$ is the *θ-margin error (or loss) function*, defined by

$$\ell_\theta(s) := \mathbb{1}\{s \leq \theta\}.$$

Observe that

$$\mathbb{1}\{\operatorname{sign}(f(x)) \neq y\} \leq \ell_\theta(yf(x)),$$

so

$$\mathsf{err}(\operatorname{sign} \circ f; S) \leq \mathsf{err}_\theta(f; S).$$

Similarly, for a probability distribution P on $\mathcal{X} \times \{-1, 1\}$, define

$$\mathsf{err}_\theta(f; P) := \mathrm{E}[\ell_\theta(\mathbf{y}f(\mathbf{x}))]$$

for $(\mathbf{x}, \mathbf{y}) \sim P$, so we also have

$$\mathsf{err}(\operatorname{sign} \circ f; P) \leq \mathsf{err}_\theta(f; P).$$

One can show that Algorithm 7.1, with appropriate choice of hyperparameter η, finds $\hat{g}$ with $\mathsf{err}_\theta(\hat{g}; S) = 0$ for $\theta = \Omega(\gamma)$ under the γ-WLA. (Schapire et al. (1998) do the same for AdaBoost.) The main question is how this can be used to bound $\mathsf{err}(\operatorname{sign} \circ \hat{g}; P)$.

Exercise 7.2. How should the hyperparameters η and T be set so that Algorithm 7.1 achieves $\mathsf{err}_{\gamma/3}(\hat{g}; S) = 0$ under the γ-WLA? ◊

7.4.3 Uniform covers

For any $K \in \mathbb{N}$, define

$$\operatorname{avg}_K(\mathcal{H}) := \left\{ \frac{1}{K} \sum_{t=1}^{K} h_t \;\middle|\; (h_1, \ldots, h_K) \in \mathcal{H}^K \right\}$$

to be the set of all *uniform averages* of K hypotheses from $\mathcal{H}$. We can now state the centerpiece of the margins theory.

Theorem 7.4 (Schapire, Freund, Bartlett, and Lee, 1998)**.** *Let $\mathcal{H} \subseteq \{-1, 1\}^{\mathcal{X}}$ be a hypothesis class over $\mathcal{X}$, let P be a probability distribution over $\mathcal{X} \times \{-1, 1\}$, and let $S \in (\mathcal{X} \times \{-1, 1\})^n$. For every $K \in \mathbb{N}$, $\theta > 0$, and $f \in \operatorname{conv}(\mathcal{H})$,*

$$\mathsf{err}(\operatorname{sign} \circ f; P) \leq \mathsf{err}_\theta(f; S) + \max_{g \in \operatorname{avg}_K(\mathcal{H})} \big(\mathsf{err}_{\theta/2}(g; P) - \mathsf{err}_{\theta/2}(g; S)\big) + 4 \exp\left(-\frac{\theta^2 K}{8}\right).$$

The last term on the right-hand side in Theorem 7.4 is $O(1/n)$ for $K = \Theta(\log(n)/\theta^2)$. (The $O(1/n)$ is not special but rather just a conveniently small quantity when n is large; we could work harder to find the optimal trade-off in terms of K, but the analysis will be messier.) When S is replaced by $\mathbf{S} \sim P^n$, the max term on the right-hand side can be bounded (in expectation, or with high probability) by proving uniform convergence of $\mathsf{err}_{\theta/2}(g; \mathbf{S})$ to $\mathsf{err}_{\theta/2}(g; P)$ for all $g \in \operatorname{avg}_K(\mathcal{H})$.

Corollary 7.1. *Let $\mathcal{H} \subseteq \{-1, 1\}^{\mathcal{X}}$ be a finite hypothesis class over $\mathcal{X}$, let P be a probability distribution over $\mathcal{X} \times \{-1, 1\}$, and let $\mathbf{S} \sim P^n$. For any $\theta \in (0, 1]$ and $\delta \in (0, 1)$, with probability at least $1 - \delta$, every $f \in \operatorname{conv}(\mathcal{H})$ satisfies*

$$\mathsf{err}(\operatorname{sign} \circ f; P) \leq \mathsf{err}_\theta(f; S) + \sqrt{\frac{4 \ln(4n) \ln(|\mathcal{H}|)}{\theta^2 n} + \frac{\ln(1/\delta)}{2n}} + \frac{1}{n}.$$

The key to proving Theorem 7.4 is to show that every $f \in \text{conv}(\mathcal{H})$ can be approximated by some $g_f \in \text{avg}_K(\mathcal{H})$ in the following sense.

- The functions in $\text{conv}(\mathcal{H})$ and $\text{avg}_K(\mathcal{H})$ are $[-1,1]$-valued, so the sort of approximation we care about is in terms of the output values being close—say, within some small value $\varepsilon > 0$.
- Since Theorem 7.4 compares 0-margin errors (on the left-hand side) to θ-margin errors (on the right-hand side), it will be enough to ensure that every $f \in \text{conv}(\mathcal{H})$ is approximated in output value by some $g_f \in \text{avg}_K(\mathcal{H})$ up to error $\varepsilon := \theta/2$.
- We will allow the $\theta/2$-closeness in output values between f and g_f to fail for some portion of the data distribution (or training data set).

This is done in the following proposition.

Proposition 7.2. *For any $\mathcal{H} \subseteq \{-1,1\}^{\mathcal{X}}$, any $\mathcal{X}$-valued random variable $\mathbf{z}$, any $K \in \mathbb{N}$, any $\varepsilon > 0$, and any $f \in \text{conv}(\mathcal{H})$, there exists $g_f \in \text{avg}_K(\mathcal{H})$ such that*

$$\Pr(|f(\mathbf{z}) - g_f(\mathbf{z})| > \varepsilon) \leq 2\exp\left(-\frac{\varepsilon^2 K}{2}\right).$$

The proof of Proposition 7.2 is based on a technique attributed to B. Maurey (Pisier, 1981) and uses the probabilistic method. We define a random function $\mathbf{G}$ from $\text{avg}_K(\mathcal{H})$ that is meant to approximate f. We first show that, for any fixed $x \in \mathcal{X}$, $\mathbf{G}(x)$ is close to $f(x)$ for most realizations of the random function $\mathbf{G}$. Then, we replace x with $\mathbf{z}$ and take expectation, so $\mathbf{G}(\mathbf{z})$ is close to $f(\mathbf{z})$ in an average-case sense. Switching the order of expectations, we conclude that there is some $g_f \in \text{avg}_K(\mathcal{H})$ such that $g_f(\mathbf{z})$ is also close to $f(\mathbf{z})$ in an average-case sense.

Proof of Proposition 7.2. Since $f \in \text{conv}(\mathcal{H})$, we can write $f = \sum_{i=1}^{d} \alpha_i h_i$ for some $d \in \mathbb{N}$, $\alpha = (\alpha_1, \ldots, \alpha_d) \in \Delta^{d-1}$, and $(h_1, \ldots, h_d) \in \mathcal{H}^d$. Let $\mathbf{J}_1, \ldots, \mathbf{J}_K$ be iid random variables (independent of $\mathbf{z}$) with

$$\Pr(\mathbf{J}_1 = j) = \alpha_j \quad \forall j \in [d],$$

and define the random function $\mathbf{G}$ taking values in $\text{avg}_K(\mathcal{H})$ by

$$\mathbf{G} := \frac{1}{K}\sum_{t=1}^{K} h_{\mathbf{J}_t}.$$

Fix $x \in \mathcal{X}$. Then $K(\mathbf{G}(x) - f(x)) = \sum_{t=1}^{K}(h_{\mathbf{J}_t}(x) - f(x))$ is the sum of K iid random variables, each taking values in $\{-1 - f(x), 1 - f(x)\}$ and each with mean zero. By Hoeffding's inequality (Corollary 4.4),

$$\Pr(\mathbf{G}(x) - f(x) > \varepsilon) \leq \exp\left(-\frac{\varepsilon^2 K}{2}\right).$$

Similarly, the same probability bound applies to the event $f(x) - \mathbf{G}(x) > \varepsilon$. By a union bound,

$$\Pr(|f(x) - \mathbf{G}(x)| > \varepsilon) \leq 2\exp\left(-\frac{\varepsilon^2 K}{2}\right).$$

Since the above inequality holds for every $x \in \mathcal{X}$, we can replace x with the random variable $\mathbf{z}$, and the same inequality holds in expectation:

$$\Pr\Big(|f(\mathbf{z}) - \mathbf{G}(\mathbf{z})| > \varepsilon\Big) = \mathrm{E}\Big[\mathrm{E}\big[\mathbb{1}\{|f(\mathbf{z}) - \mathbf{G}(\mathbf{z})| > \varepsilon\} \;\big|\; \mathbf{z}\big]\Big] \le 2\exp\bigg(-\frac{\varepsilon^2 K}{2}\bigg).$$

The order of the expectations can be reversed:

$$\Pr\Big(|f(\mathbf{z}) - \mathbf{G}(\mathbf{z})| > \varepsilon\Big) = \mathrm{E}\Big[\mathrm{E}\big[\mathbb{1}\{|f(\mathbf{z}) - \mathbf{G}(\mathbf{z})| > \varepsilon\} \;\big|\; \mathbf{G}\big]\Big].$$

So there exists $g_f \in \mathrm{avg}_K(\mathcal{H})$ such that

$$\Pr\Big(|f(\mathbf{z}) - g_f(\mathbf{z})| > \varepsilon\Big) = \mathrm{E}\big[\mathbb{1}\{|f(\mathbf{z}) - \mathbf{G}(\mathbf{z})| > \varepsilon\} \;\big|\; \mathbf{G} = g_f\big] \le 2\exp\bigg(-\frac{\varepsilon^2 K}{2}\bigg). \qquad \square$$

Proof of Theorem 7.4. Let $(\mathbf{x}, \mathbf{y}) \sim P$. Let $S := ((x_1, y_1), \ldots, (x_n, y_n))$. Define the $\mathcal{X}$-valued random variable $\mathbf{z}$ as follows.

- Toss a fair coin.
- If the coin comes up heads, set $\mathbf{z} := \mathbf{x}$.
- If the coin comes up tails, randomly pick $\mathbf{I} \sim \mathrm{Uniform}([n])$, and set $\mathbf{z} := x_{\mathbf{I}}$.

Applying Proposition 7.2 to $\mathbf{z}$ (with $\varepsilon = \theta/2$) gives the following: for every $f \in \mathrm{conv}(\mathcal{H})$, there exists $g_f \in \mathrm{avg}_K(\mathcal{H})$ such that

$$\frac{1}{2}\left(\Pr\Big(|f(\mathbf{x}) - g_f(\mathbf{x})| > \theta/2\Big) + \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{|f(x_i) - g_f(x_i)| > \theta/2\}\right) \le 2\exp\bigg(-\frac{\theta^2 K}{8}\bigg). \tag{7.9}$$

We next reason about how classification errors relate to $\theta/2$-margin errors, and how $\theta/2$-margin errors relate to θ-margin errors. If $yf(x) \le 0$ and $|f(x) - g_f(x)| \not> \theta/2$, then $yg_f(x) \le \theta/2$. This can be expressed using indicator functions as follows:

$$\mathbb{1}\{yf(x) \le 0\} \le \mathbb{1}\{yg_f(x) \le \theta/2\} + \mathbb{1}\{|f(x) - g_f(x)| > \theta/2\}.$$

Similarly, if $yg_f(x) \le \theta/2$ and $|f(x) - g_f(x)| \not> \theta/2$, then $yf(x) \le \theta$:

$$\mathbb{1}\{yg_f(x) \le \theta/2\} \le \mathbb{1}\{yf(x) \le \theta\} + \mathbb{1}\{|f(x) - g_f(x)| > \theta/2\}.$$

Therefore

$$\Pr(\mathbf{y}f(\mathbf{x}) \le 0) \le \Pr(\mathbf{y}g_f(\mathbf{x}) \le \theta/2) + \Pr(|f(\mathbf{x}) - g_f(\mathbf{x})| > \theta/2) \tag{7.10}$$

and

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{y_i g_f(x_i) \le \theta/2\} \le \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{y_i f(x_i) \le \theta\} + \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{|f(x_i) - g_f(x_i)| > \theta/2\}. \tag{7.11}$$

Putting (7.9)–(7.11) together gives

$$\begin{aligned}
\Pr(\mathbf{y}f(\mathbf{x}) \le 0) &\le \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{y_i f(x_i) \le \theta\} \\
&\quad + \Pr(\mathbf{y}g_f(\mathbf{x}) \le \theta/2) - \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{y_i g_f(x_i) \le \theta/2\} \\
&\quad + \Pr(|f(\mathbf{x}) - g_f(\mathbf{x})| > \theta/2) + \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{|f(x_i) - g_f(x_i)| > \theta/2\} \\
&\le \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{y_i f(x_i) \le \theta\} \\
&\quad + \max_{g \in \mathrm{avg}_K(\mathcal{H})} \left(\Pr(\mathbf{y}g(\mathbf{x}) \le \theta/2) - \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{y_i g(x_i) \le \theta/2\} \right) \\
&\quad + 4\exp\left(-\frac{\theta^2 K}{8}\right).
\end{aligned}$$

□

Exercise 7.3. Let $\mathcal{H} \subseteq \{-1,1\}^{\mathcal{X}}$ be a hypothesis class over $\mathcal{X}$ with VC dimension $d < \infty$. Suppose $\mathbf{S} \sim P^n$ for a probability distribution P over $\mathcal{X} \times \{-1,1\}$. Prove a bound on

$$\mathrm{E}\left[\sup_{g \in \mathrm{avg}_K(\mathcal{H})} \mathsf{err}_{\theta/2}(g; P) - \mathsf{err}_{\theta/2}(g; \mathbf{S}) \right]$$

in terms of n, d, and $K \in \mathbb{N}$. The bound should hold for every choice of $\theta \ge 0$. ◊

Bibliography

Noga Alon. On the density of sets of vectors. *Discrete Mathematics*, 46(2):199–202, 1983.

Dana Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1988.

Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.

Jānis M Barzdin and Rūsiņš V Freivald. On the prediction of general recursive functions. *Soviet Math. Doklady*, 13:1224–1228, 1972.

Thomas M Cover. Behavior of sequential predictors of binary sequences. *Transactions on Prague Conference on Information Theory Statistical Decision Functions, Random Processes*, pages 263–272, 1965.

Peter Frankl. On the trace of finite sets. *Journal of Combinatorial Theory, Series A*, 34(1):41–45, 1983.

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

Yoav Freund and Robert E Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1-2):79–103, 1999.

Steve Hanneke. The optimal sample complexity of PAC learning. *Journal of Machine Learning Research*, 17(38):1–15, 2016.

David Haussler, Nick Littlestone, and Manfred K Warmuth. Predicting {0, 1}-functions on randomly drawn points. *Information and Computation*, 115(2):248–292, 1994.

David P Helmbold and Manfred K Warmuth. On weak learning. *Journal of Computer and System Sciences*, 50(3):551–573, 1995.

Michael Kearns and Leslie Valiant. Cryptographic limitations on learning boolean formulae and finite automata. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, pages 433–444, 1989.

Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1988.

Nick Littlestone. From on-line to batch learning. In *Second Annual Workshop on Computational Learning Theory*, pages 269–284, 1989.

Nick Littlestone and Manfred K Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.

Wolfgang Maass and György Turán. How fast can a threshold gate learn? In Stephen José Hanson, George A Drastal, and Ronald L Rivest, editors, *Computational Learning Theory and Natural Learning System: Constraints and Prospects*, pages 381–414. MIT Press, 1994.

Marvin Minsky and Seymour Papert. *Perceptrons: An introduction to computational geometry*. MIT Press, 1969.

Saburo Muroga. *Threshold logic and its applications*. Wiley-Interscience, 1971.

Albert Novikoff. On convergence proofs on perceptrons. In *Proceedings of the Symposium on the Mathematical Theory of Automata*, volume 12, pages 615–622, 1962.

Gilles Pisier. Remarques sur un résultat non publié de b. maurey. *Séminaire Analyse fonctionnelle*, pages 1–12, 1981.

Leonard Pitt and Leslie G Valiant. Computational limitations on learning from examples. *Journal of the ACM (JACM)*, 35(4):965–984, 1988.

Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, pages 65–386, 1958.

Norbert Sauer. On the density of families of sets. *Journal of Combinatorial Theory, Series A*, 13(1): 145–147, 1972.

Robert E Schapire. The strength of weak learnability. *Machine Learning*, 5:197–227, 1990.

Robert E Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):1651–1686, 1998.

Saharon Shelah. A combinatorial problem; stability and order for models and theories in infinitary languages. *Pacific Journal of Mathematics*, 41(1):247–261, 1972.

Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Vladimir N Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1985.

Vladimir N Vapnik and Alexey Ya Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16:264–280, 1971.

John von Neumann. Zur theorie der gesellschaftsspiele. *Mathematische annalen*, 100(1):295–320, 1928.

Yurii A Zuev. Asymptotics of the logarithm of the number of boolean threshold functions. *Dokl. Akad. Nauk SSSR*, 306:528–530, 1989.