Boosting

COMS 4995-1 Spring 2020 (Daniel Hsu)

1 Weak versus strong learning

In 1989, Kearns & Valiant asked if PAC learning is equivalent to a "weaker" variant of PAC learning in which the learner is only required to return a hypothesis whose error rate is slightly better than 1/2. (Recall that in PAC learning, the learner must be able to achieve error rate ε for any $\varepsilon \in (0,1)$.) Schapire answered the question in the affirmative a year later. In particular, he provided an efficient algorithm that achieves the original PAC learning guarantees (i.e., a "Strong Learner") if provided oracle access to a subroutine that achieves the weaker variant of PAC learning (i.e., a "Weak Learner"). This kind of meta-algorithm that converts a Weak Learner and yields a Strong Learner is called a *boosting algorithm*. A year after Schapire's algorithm was published, Freund published an "optimal" boosting algorithm called "Boost-by-Majority". Even later, Freund and Schapire together invented the AdaBoost algorithm, which has been one of the crowning achievements of computational learning theory, from both practical and theoretical perspectives.

Boosting algorithms exploit that the fact that the Weak Learner is required to return a hypothesis h whose error rate $P[h(X) \neq Y]$ is better than 1/2, regardless of what the marginal distribution of X is. To do this, they change the effective distribution of X that the Weak Learner sees examples from. This way, the Weak Learner is tricked into providing hypotheses whose errors are "spread out" (loosely speaking), and a majority vote over the hypotheses produced by the Weak Learner will be (almost) always correct.

We won't worry about the formal equivalence of Weak and Strong Learning, and instead consider two aspects of boosting. First, we'll show how the main idea of boosting, the ability to find highly accurate weighted majority vote classifiers, is borne out on training data. Second, we'll discuss the "generalization" properties of weighted majority vote classifiers—specifically, what their behavior on iid training data tells us about their true error rates.

2 Boosting on training data

The main idea of boosting is borne out just on training data

$$(x_1, y_1), \dots, (x_n, y_n) \in \mathcal{X} \times \{\pm 1\}.$$

We suppose that there is a Weak Learner (i.e., a subroutine) that, if provided any weighting over the training data $\boldsymbol{p} \in \Delta([n]) = \{\boldsymbol{p} = (p_1, \dots, p_n) \in \mathbb{R}^n_+ : \sum_{i=1}^n p_i = 1\}$, returns a hypothesis $h \colon \mathcal{X} \to \{\pm 1\}$ such that the $\boldsymbol{p}$-weighted empirical error rate satisfies

$$\sum_{i=1}^n p_i \mathbb{1}\{h(x_i) \neq y\} \leq \frac{1-\theta}{2}.$$

Here, $\theta \in (0,1)$ is a fixed parameter that we call the *advantage* of the hypothesis h (quantifying how much better than random guessing is the returned hypothesis). This assumption about the Weak Learner is called the *Weak Learning Assumption.*

2.1 Existence proof

Let us assume for simplicity that the Weak Learner who always returns hypotheses from a finite class $\mathcal{H}$. The Weak Learning Assumption can be stated as

$$\min_{\boldsymbol{p} \in \Delta([n])} \max_{h \in \mathcal{H}} \sum_{i=1}^n p_i y_i h(x_i) \geq \theta.$$

Let $\boldsymbol{A}$ be the matrix whose rows are indexed by $[n]$ and columns are indexed by $\mathcal{H}$, and whose (i,h) entry is

$$A_{i,h} = y_i h(x_i) \in \{\pm 1\}.$$

Then we can write the Weak Learning Assumption in matrix notation as

$$\min_{\boldsymbol{p} \in \Delta([n])} \max_{h \in \mathcal{H}} \boldsymbol{p}^\intercal \boldsymbol{A} \boldsymbol{e}_h \geq \theta.$$

Von Neumann's Min-Max Theorem for Zero-Sum games tells us that

$$\min_{\boldsymbol{p} \in \Delta([n])} \max_{h \in \mathcal{H}} \boldsymbol{p}^\intercal \boldsymbol{A} \boldsymbol{e}_h = \max_{\boldsymbol{w} \in \Delta(\mathcal{H})} \min_{i \in [n]} \boldsymbol{e}_i^\intercal \boldsymbol{A} \boldsymbol{w}.$$

Here, we have regarded $\boldsymbol{A}$ as the payoff matrix for a two-player zero-sum game between the "Booster" (row player) and "Weak Learner" (column player). We conclude that

$$\max_{\boldsymbol{w} \in \Delta(\mathcal{H})} \min_{i \in [n]} \boldsymbol{e}_i^\intercal \boldsymbol{A} \boldsymbol{w} \geq \theta.$$

This shows that the Weak Learning Assumption is equivalent to the existence of a weighted majority vote over $\mathcal{H}$, given by

$$x \mapsto \operatorname{sign}\left(\sum_{h \in \mathcal{H}} w_h h(x) \right),$$

such that for every training example (x_i, y_i),

$$y_i \sum_{h \in \mathcal{H}} w_h h(x_i) \geq \theta.$$

In particular, since $\theta > 0$, this weighted majority vote hypothesis correctly classifies every training example.

The above argument, however, does not tell us how to find a weighted majority vote hypothesis when the Weak Learning Assumption holds. In principle, if one could enumerate all of $\mathcal{H}$, then one could find the desired $\boldsymbol{w}$ using linear programming. But we are interested in algorithms that only access hypotheses from $\mathcal{H}$ via the Weak Learner.

2.2 Boosting game

Our aim is to construct an algorithm that finds a majority vote hypothesis that correctly classifies all of the training examples, accessing hypotheses in $\mathcal{H}$ only through the Weak Learner. To do this, we consider designing an algorithm—a Booster—for playing the following game against the Weak Learner that takes place over T rounds.

- Initialize: $\boldsymbol{s}_0 := \boldsymbol{0} \in \mathbb{R}^n$
- For $t = 1, 2, \dots, T$:
  - Booster choose $\boldsymbol{p}_t \in \Delta([n])$
  - Weak Learner picks $\boldsymbol{z}_t \in \{\pm 1\}^n$ such that $\boldsymbol{p}_t^\intercal \boldsymbol{z}_t \geq \theta$.
  - $\boldsymbol{s}_t := \boldsymbol{s}_{t-1} + \boldsymbol{z}_t$

In this game, when the Weak Learner picks $\boldsymbol{z}_t$, we really mean that it picks $h_t \in \mathcal{H}$ such that $z_{t,i} = y_i h_t(x_i)$ for $i = 1, \ldots, n$ satisfies

$$\boldsymbol{p}_t^\mathsf{T} \boldsymbol{z}_t = \sum_{i=1}^n p_{t,i} y_i h_t(x_i) \geq \theta.$$

In other words, it must have $\boldsymbol{p}_t$-weighted empirical error rate at most $(1-\theta)/2$.

Observe that the state vector $\boldsymbol{s}_t$ satisfies

$$s_{t,i} = y_i(h_1(x_i) + \cdots + h_t(x_i))$$

for all $i = 1, \ldots, n$. Therefore, if $s_{T,i} > 0$ for all $i = 1, \ldots, n$, then the majority vote classifier

$$x \mapsto \operatorname{sign}\left(\sum_{t=1}^T h_t(x)\right)$$

correctly classifies all of the training examples. That is the goal of the Booster. The Weak Learner's goal is to ensure that at least one training example remains incorrectly classified.

Let us define the "total loss" incurred by the Booster to be

$$L(\boldsymbol{s}) := \sum_{i=1}^n \mathbb{1}\{s_i \leq 0\}$$

if the final state is $\boldsymbol{s} \in \mathbb{R}^n$. In this notation, the Booster wins the game if and only if

$$L(\boldsymbol{s}_T) < 1.$$

Is there a T such that the Booster has a winning strategy? If so, can it be computed efficiently?

2.3 Last round of the game

To build intuition for the boosting algorithm, let us consider what can happen in the very last round of the game. Suppose $\boldsymbol{s}$ is the state after first $T-1$ rounds. How should the Booster choose $\boldsymbol{p}_T$ in round T?

Define

$$\Lambda_T(\boldsymbol{s}) := L(\boldsymbol{s})$$

for all $\boldsymbol{s} \in \mathbb{R}^n$. If the Booster chooses $\boldsymbol{p}_T$, then the Weak Learner can choose any $\boldsymbol{z}_T \in \{\pm 1\}^n$ subject to $\boldsymbol{p}_T^\mathsf{T} \boldsymbol{z}_T \geq \theta$ so that the state becomes $\boldsymbol{s} + \boldsymbol{z}_T$ and the total loss incurred by the Booster is $\Lambda_T(\boldsymbol{s} + \boldsymbol{z}_T)$. Let us assume that the Weak Learner is going to choose the worst possible $\boldsymbol{z}_T$. In that case, the Booster should choose $\boldsymbol{p}_T$ to achieve the min below:

$$\Lambda_{T-1}(\boldsymbol{s}) := \min_{\boldsymbol{p} \in \Delta([n])} \max_{\substack{\boldsymbol{z} \in \{\pm 1\}^n : \\ \boldsymbol{p}^\mathsf{T} \boldsymbol{z} \geq \theta}} \Lambda_T(\boldsymbol{s} + \boldsymbol{z}),$$

so that whatever the Weak Learner does, the total loss is as small as possible. Here, the semantic of $\Lambda_{T-1}(\boldsymbol{s})$ is the total loss of the Booster if the state after round $T-1$ is $\boldsymbol{s}$, and both the Booster and Weak Learner play optimally in round T.

In general, we can define $\Lambda_t(\boldsymbol{s})$ to be the total loss of the Booster if the state after round t is $\boldsymbol{s}$, and both the Booster and Weak Learner play optimally in subsequent rounds $t+1, \ldots, T$. By the same argument as above, we have

$$\Lambda_{t-1}(\boldsymbol{s}) = \min_{\boldsymbol{p} \in \Delta([n])} \max_{\substack{\boldsymbol{z} \in \{\pm 1\}^n : \\ \boldsymbol{p}^\mathsf{T} \boldsymbol{z} \geq \theta}} \Lambda_t(\boldsymbol{s} + \boldsymbol{z}).$$

2.4 Minimax value

Since the entire game starts in state $\boldsymbol{s}_0 = \boldsymbol{0}$, the minimum possible total loss achievable by the Booster playing against an optimal Weak Learner is given by the following expression:

$$\min_{\boldsymbol{p}_1 \in \Delta([n])} \max_{\substack{\boldsymbol{z}_1 \in \{\pm 1\}^n : \\ \boldsymbol{p}_1^\mathsf{T} \boldsymbol{z}_1 \geq \theta}} \min_{\boldsymbol{p}_2 \in \Delta([n])} \max_{\substack{\boldsymbol{z}_2 \in \{\pm 1\}^n : \\ \boldsymbol{p}_2^\mathsf{T} \boldsymbol{z}_2 \geq \theta}} \cdots \min_{\boldsymbol{p}_T \in \Delta([n])} \max_{\substack{\boldsymbol{z}_T \in \{\pm 1\}^n : \\ \boldsymbol{p}_T^\mathsf{T} \boldsymbol{z}_T \geq \theta}} \Lambda_T \left(\sum_{t=1}^{T} \boldsymbol{z}_t \right).$$

This is the minimax value of this sequential game. The Booster has a winning strategy if and only if the minimax value is less than one.

Unfortunately, the above expression is rather unwieldy and does not shed light on when the value is less than one. It is also not clear if there is an efficient algorithm to compute the winning strategy even when one exists.

Instead, we will develop a tractable upper bound on Λ_t that (amazingly) decomposes over $i = 1, \ldots, n$. Not only will this facilitate analysis, it will also suggest an efficient algorithm—precisely the "Boost-by-Majority" (BBM) algorithm of Freund. Our derivation and analysis will follow the "Drifting Games" formulation of Schapire.

2.5 The upper bound

Let us define $\phi_T(s_i) := \mathbb{1}\{s_i \leq 0\}$, so

$$\Lambda_T(\boldsymbol{s}) = \sum_{i=1}^{n} \phi_T(s_i).$$

Then, assuming ϕ_t has been defined for some $t \geq 1$, define

$$\phi_{t-1}(s_i) := \min_{q_i \geq 0} \max_{z_i \in \{\pm 1\}} \phi_t(s_i + z_i) + q_i(z_i - \theta).$$

We claim that the sum of $\phi_t(s_i)$ over $i = 1, \ldots, n$ provides an upper bound on $\Lambda_t(\boldsymbol{s})$.

Claim. For all $t = 0, \ldots, T$,

$$\Lambda_t(\boldsymbol{s}) \leq \sum_{i=1}^{n} \phi_t(s_i), \quad \text{for all } \boldsymbol{s} \in \mathbb{R}^n.$$

Proof. The proof is by backwards induction. The base case of $t = T$ is trivial, by definition of ϕ_T. Now assume as the inductive hypothesis that the inequality holds for some $t \geq 1$. Then

$$\begin{aligned}
\Lambda_{t-1}(\boldsymbol{s}) &= \min_{\boldsymbol{p} \in \Delta([n])} \max_{\boldsymbol{z} \in \{\pm 1\}^n : \boldsymbol{p}^\intercal \boldsymbol{z} \geq \theta} \Lambda_t(\boldsymbol{s} + \boldsymbol{z}) \\
&= \min_{\boldsymbol{p} \in \Delta([n])} \max_{\boldsymbol{z} \in \{\pm 1\}^n} \min_{\lambda \geq 0} \Lambda_t(\boldsymbol{s} + \boldsymbol{z}) + \lambda(\boldsymbol{p}^\intercal \boldsymbol{z} - \theta) \\
&\leq \min_{\boldsymbol{p} \in \Delta([n])} \min_{\lambda \geq 0} \max_{\boldsymbol{z} \in \{\pm 1\}^n} \Lambda_t(\boldsymbol{s} + \boldsymbol{z}) + \lambda(\boldsymbol{p}^\intercal \boldsymbol{z} - \theta) \\
&= \min_{\boldsymbol{q} \in \mathbb{R}_+^n} \max_{\boldsymbol{z} \in \{\pm 1\}^n} \Lambda_t(\boldsymbol{s} + \boldsymbol{z}) + \sum_{i=1}^{n} q_i(z_i - \theta) \\
&\leq \min_{\boldsymbol{q} \in \mathbb{R}_+^n} \max_{\boldsymbol{z} \in \{\pm 1\}^n} \sum_{i=1}^{n} \phi_t(s_i + z_i) + q_i(z_i - \theta) \\
&= \sum_{i=1}^{n} \min_{q_i \geq 0} \max_{z_i \in \{\pm 1\}} \phi_t(s_i + z_i) + q_i(z_i - \theta) \\
&= \sum_{i=1}^{n} \phi_{t-1}(s_i).
\end{aligned}$$

The first inequality follows from switching the order of $\min_{\lambda 0}$ and $\max_{\boldsymbol{z}\in\{\pm1\}^n}$, which can only increase the value. In the subsequent line, we switch notation to $\boldsymbol{q} = (q_1, \ldots, q_n)$ with $q_i = \lambda p_i$ for $i = 1, \ldots, n$. The second inequality follows from the inductive hypothesis. The subsequent equality holds because the objective decomposes over $i = 1, \ldots, n$. The final equality holds by definition of $\phi_{t-1}(s_i)$. ▮

2.6 Achieving the bound

Since the upper bound on Λ_t decomposes over i, we can separately find the q_i that achieves the min in the i-th component of the upper bound. To do this, we first observe one more useful property of the ϕ_t's.

Claim. $\phi_t(s_i + 1) \le \phi_t(s_i - 1)$

Proof. The proof is by backwards induction. ▮

In the definition of ϕ_{t-1},

$$\phi_{t-1}(s_i) = \min_{q_i \ge 0} \max_{z_i \in \{\pm1\}} \phi_t(s_i + z_i) + q_i(z_i - \theta),$$

we observe that the minimization objective is the maximum of two linear functions, one with a positive slope and the other with a negative slope. Since $\phi_t(s_i + 1) \le \phi_t(s_i - 1)$, it follows that the minimizing q_i is the point at which these two linear functions intersect, namely

$$\frac{\phi_t(s_i - 1) - \phi_t(s_i + 1)}{2},$$

and the value achieved at that point is

$$\phi_{t-1}(s_i) = \frac{1+\theta}{2}\phi_t(s_i + 1) + \frac{1-\theta}{2}\phi_t(s_i - 1).$$

This gives the strategy employed by the BBM algorithm:

- In round t, let

$$q_{t,i} := \frac{\phi_t(s_{t-1,i} - 1) - \phi_t(s_{t-1,i} + 1)}{2} \quad \text{for all } i = 1, \ldots, n,$$

and

$$p_{t,i} \propto q_{t,i} \quad \text{for all } i = 1, \ldots, n.$$

The argument above also provides a (backwards) recurrence relation, $\phi_{t-1}(s_i) = \frac{1+\theta}{2}\phi_t(s_i + 1) + \frac{1-\theta}{2}\phi_t(s_i - 1)$, that is satisfied by the upper bound functions, starting with $\phi_T(s_i) = \mathbb{1}\{s_i \leq 0\}$. The solution to the recurrence is

$$\begin{aligned}
\phi_t(s_i) &= \mathrm{BinomialCDF}\left(\frac{T-t-s_i}{2} \,\middle|\, T-t, \frac{1+\theta}{2} \right) \\
&= \sum_{k=0}^{\left\lfloor \frac{T-t-s_i}{2} \right\rfloor} \binom{T-t}{k} \left(\frac{1+\theta}{2}\right)^k \left(\frac{1-\theta}{2}\right)^{T-t-k}.
\end{aligned}$$

Here, $\mathrm{BinomialCDF}(x \mid n, p)$ is the cumulative distribution function at x for the binomial distribution with n trials and success probability p. Having this explicit solution for the upper bound functions makes it clear that the BBM algorithm can be implemented efficiently.

2.7 Monotonicity of the upper bound

There is one last property we need for the upper bound to complete the analysis of BBM.

Claim. Pick any $t = 1, \ldots, T$. Let $\boldsymbol{s} \in \mathbb{R}^n$ be the state after $t-1$ rounds. Assume the Booster chooses $\boldsymbol{p}_t$ (through $\boldsymbol{q}_t$) to achieve the minimum value in the definition of $\phi_{t-1}(s_i)$. Then for any $\boldsymbol{z}_t \in \{\pm 1\}^n$ such that $\boldsymbol{p}_t^\intercal \boldsymbol{z}_t \geq \theta$,

$$\sum_{i=1}^n \phi_t(s_i + z_{t,i}) \leq \sum_{i=1}^n \phi_{t-1}(s_i).$$

Proof. Recall that

$$\phi_{t-1}(s_i) = \min_{q_i \geq 0} \max_{z_i \in \{\pm 1\}} \phi_t(s_i + z_i) + q_i(z_i - \theta).$$

Since $\boldsymbol{p}_t$ is chosen (via choice of $\boldsymbol{q}_t$) to achieve the min,

$$\begin{aligned}\sum_{i=1}^{n} \phi_{t-1}(s_i) &= \sum_{i=1}^{n} \max_{z_i \in \{\pm 1\}} \phi_t(s_i + z_i) + q_{t,i}(z_i - \theta) \\ &\geq \sum_{i=1}^{n} \phi_t(s_i + z_{t,i}) + q_{t,i}(z_{t,i} - \theta) \\ &\geq \sum_{i=1}^{n} \phi_t(s_i + z_{t,i}),\end{aligned}$$

where the last inequality uses the constraint $\boldsymbol{p}_t^\intercal \boldsymbol{z}_t \geq \theta$. ▮

2.8 Conclusion of the analysis

Recall that $L(\boldsymbol{s}) = \Lambda_T(\boldsymbol{s}) = \sum_{i=1}^{n} \phi_T(s_i)$. By monotonicity property, in the sequence of states $\boldsymbol{s}_0, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_T$ actually encountered by BBM, we have

$$L(\boldsymbol{s}_T) = \sum_{i=1}^{n} \phi_T(s_{T,i}) \leq \sum_{i=1}^{n} \phi_{T-1}(s_{T-1,i}) \leq \cdots \leq \sum_{i=1}^{n} \phi_0(s_{0,i}) = n \cdot \phi_0(0).$$

So,

$$\sum_{i=1}^{n} \mathbb{1}\{s_{T,i} \leq 0\} \leq n \cdot \text{BinomialCDF}\left(\frac{T}{2} \middle| T, \frac{1+\theta}{2} \right).$$

Therefore, if T is large enough so that the right-hand side is less than one, then BBM wins the game, i.e., the majority vote classifier $x \mapsto \operatorname{sign}(\sum_{t=1}^{T} h_t(x))$ correctly classifies all training examples. (By Hoeffding's inequality, the right-hand side is at most $ne^{-\theta^2 T/2}$, so $T = \frac{2 \ln n}{\theta^2}$ suffices.)

2.9 Exponential weights variant of BBM

The choice of the $\boldsymbol{p}_t$ by BBM in each round depends both on T and θ. It is possible to obtain a boosting algorithm that does away with these dependences. We first describe how to remove the dependence on T. The main idea is to redefine $\Lambda_T(\boldsymbol{s})$ to be a particular upper bound on the total loss $L(\boldsymbol{s})$ based on a surrogate loss function. In particular, we let $\Lambda_T(\boldsymbol{s}) := \sum_{i=1}^{n} \phi_T(s_i)$ where

$$\phi_T(s_i) := e^{-\eta s_i}$$

for an appropriate choice of $\eta > 0$. Note that $e^{-\eta s_i} \geq \mathbb{1}\{s_i \leq 0\}$, so ensuring that $\Lambda_T(\boldsymbol{s}) < 1$ is sufficient to guarantee that every training example is correctly classified by the majority vote classifier.

The analysis of this variant of BBM is entirely the same as before, except for the final conclusion. The solution to the backwards recurrence defining ϕ_t is also different, and in fact very simple:

$$\phi_t(s_i) = \left(\frac{1+\theta}{2}e^{-\eta} + \frac{1-\theta}{2}e^{\eta}\right)^{T-t} e^{-\eta s_i}.$$

The Booster's distribution over examples in round t is given by

$$p_{t,i} \propto e^{-\eta s_{t-1,i}}.$$

The final conclusion of the analysis gives the following chain of inequalities (from the monotonicity of the upper bounds):

$$\underbrace{\sum_{i=1}^{n} \mathbb{1}\{s_{T,i} \leq 0\}}_{L_T(\boldsymbol{s}_T)} \leq \underbrace{\sum_{i=1}^{n} e^{-\eta s_{T,i}}}_{\Lambda_T(\boldsymbol{s}_T)=\sum_{i=1}^{n}\phi_T(s_{T,i})} \leq \cdots \leq \underbrace{n\left(\frac{1+\theta}{2}e^{-\eta} + \frac{1-\theta}{2}e^{\eta}\right)^{T}}_{\sum_{i=1}^{n}\phi_0(s_{0,i})}.$$

To minimize the bound, we choose $\eta = \frac{1}{2}\ln\frac{1+\theta}{1-\theta}$. With this choice of η (which has no dependence on T, but does depend on θ), we have

$$\sum_{i=1}^{n} \mathbb{1}\{s_{T,i} \leq 0\} \leq n(1-\theta^2)^{T/2}.$$

This can be upper bounded by $ne^{-\theta^2 T/2}$, and hence $T = \frac{2\ln n}{\theta^2}$ rounds is sufficient to guarantee that the Booster wins the game.

Finally, removing the dependence on both T and θ is achieved by the AdaBoost algorithm of Freund and Schapire. The probability distribution chosen by AdaBoost in round t is

$$p_{t,i} \propto \exp\left(-\sum_{\tau=1}^{t-1} \eta_\tau z_{\tau,i}\right),$$

where $\eta_t = \frac{1}{2}\ln\frac{1+\theta_t}{1-\theta_t}$ and θ_t is the advantage of the hypothesis h_t. AdaBoost returns a *weighted* majority vote hypothesis of the form

$$x \mapsto \operatorname{sign}\left(\sum_{t=1}^{T} \eta_t h_t(x)\right).$$

The final bound on the number of examples misclassified by this hypothesis is

$$n \prod_{t=1}^{T} \sqrt{1-\theta_t^2},$$

which is at most one for

$$T = \frac{2\ln n}{\bar{\theta}^2},$$

where $\bar{\theta}^2 = \frac{1}{T}\sum_{t=1}^{T}\theta_t^2$.

3 Generalization properties of weighted majority vote hypotheses

In the previous section, we have shown how to find (weighted) majority vote hypotheses that, under the Weak Learning Assumption, achieve zero error rate on training data. Of course, we are generally interested in the behavior of such hypotheses on new data. To study this, we assume that the training data are drawn iid from a probability distribution P over $\mathcal{X} \times \{\pm 1\}$. We would like to relate the error rate on the training data (i.e., the empirical error rate) to the "true" error rate with respect to P.

In more detail, we are considering hypotheses f of the form

$$f(x) = \operatorname{sign}(g(x))$$

where $g \in \operatorname{conv}(\mathcal{H})$, i.e., $f \in \mathcal{F} := \operatorname{sign}(\operatorname{conv}(\mathcal{H}))$. Here, $\operatorname{conv}(\mathcal{H})$ is the set of convex combinations of hypotheses from $\mathcal{H}$. The difference between the empirical error rate $P_n[f(X) \neq Y]$ on the n iid training examples and the true error rate $P[f(X) \neq Y]$ of hypotheses from $\mathcal{F}$ is, in the worst case, controlled by the VC dimension of $\mathcal{F}$, which can be as large as $|\mathcal{H}|$. In contrast, the VC dimension of $\mathcal{H}$ is at most $\log|\mathcal{H}|$. This means that to reliably control the deviations between empirical and true error rates of hypotheses from $\mathcal{F}$, the sample size may need to be exponentially larger than what is required for hypotheses from $\mathcal{H}$. This poor dependence on $|\mathcal{H}|$ is highly undesirable.

One saving grace is that boosting algorithms like BBM and AdaBoost may return a weighted majority vote over just a small number of hypotheses from

$\mathcal{H}$—in fact, just $K = \frac{2 \log n}{\theta^2}$ of them (under the Weak Learning Assumption). Thus, they are *sparse* weighted majority vote hypotheses, and the family of such hypotheses has a much smaller VC dimension, roughly $K \log |\mathcal{H}|$.

However, practitioners using AdaBoost observed that in many cases, weighted majority vote hypotheses produced by running AdaBoost for many rounds—many more than was required to achieve zero training error rate—could have lower test error rates than those produced after just a small number of rounds. It turns out that what seemed to be improving is the *margin* achieved by these hypotheses on the training examples.

Indeed, recall that the Weak Learning Assumption not only guaranteed the existence of a weighted majority vote hypothesis that correct classifies all of the training examples, but it also guarantees the existence of $g \in \operatorname{conv}(\mathcal{H})$ such that $y_i g(x_i) \geq \theta$ for all $i = 1, \ldots, n$. It can be shown that the "exponential weights" variant of BBM (as well as AdaBoost) achieve something like this guarantee as well. Do hypotheses that achieve large margins on training examples have better generalization behavior?

3.1 Margin bound

Schapire, Freund, Bartlett, and Lee proved a generalization bound for weighted majority vote hypotheses in terms of the margins achieved on training examples. The following is a simplified version of their main result.

Theorem. Let $(X_1, Y_1), \ldots, (X_n, Y_n)$ be iid sample from P over $\mathcal{X} \times \{\pm 1\}$, and fix any $\delta, \theta \in (0, 1)$. With probability at least $1 - \delta$, for all $g \in \operatorname{conv}(\mathcal{H})$,

$$P[Y g(X) \leq 0] \leq P_n[Y g(X) \leq \theta] + O\left(\sqrt{\frac{(\log n)(\log |\mathcal{H}|)}{\theta^2 n}} + \sqrt{\frac{\log(1/\delta)}{n}}\right).$$

There are two aspects to notice about this bound. The first is that the first term on the right-hand side is the fraction of training examples on which g does not achieve a margin of θ. The second is that the second term on the right-hand side is, if θ is regarded as a constant, the same as the deviation term that one gets just with hypotheses from $\mathcal{H}$ (up to log factors).

The main idea of the proof is as follows. We shall approximate each $g \in C := \operatorname{conv}(\mathcal{H})$ by a convex combination of $K \approx \frac{\log n}{\theta^2}$ hypotheses.

Let $C_K := \{\frac{1}{K}\sum_{i=1}^K h_i : h_1, \dots, h_K \in \mathcal{H}\}$ be the set of *simple* averages of K hypotheses from $\mathcal{H}$ (allowing repeats). Note $|C_K| \leq |\mathcal{H}|^K$. We'll use the probabilistic method to ensure existence of good approximations for $g \in C$. For any $g \in C$, we randomly pick $G_g \in C_K$ in a natural way. Specifically, if $g = \sum_{h \in \mathcal{H}} \alpha_h h$, we pick K hypotheses $h_1, \dots, h_K$ iid from $\mathcal{H}$ according to the distribution $(\alpha_h : h \in \mathcal{H})$, and let $G_g := \frac{1}{K}\sum_{i=1}^K h_i$. Now, for any fixed $(x, y) \in \mathcal{X} \times \in \{\pm 1\}$, we have

$$\mathbb{E}_{G_g}[yG_g(x)] = yg(x).$$

Moreover, by Hoeffding's inequality, for any $t > 0$, we have

$$\Pr_{G_g}[|yG_g(x) - yg(x)| > t] \leq e^{-Kt^2/2}.$$

Therefore, G_g can be regarded as a random "sparsification" of g.

3.2 Proof of the margin bound

The proof begins with three main steps.

1. For any $g \in C$,

$$P[Yg(X) \leq 0] \leq \mathbb{E}_{G_g}[P[YG_g(X) \leq \theta/2]] + e^{-K\theta^2/8}.$$

2. With probability at least $1 - \delta$, for all $\tilde{g} \in C_K$,

$$P[Y\tilde{g}(X) \leq \theta/2] \leq P_n[Y\tilde{g}(X) \leq \theta/2] + \sqrt{\frac{\log(|C_K|/\delta)}{2n}}.$$

3. For any $g \in C$,

$$\mathbb{E}_{G_g}\left[P_n[YG_g(X) \leq \theta/2]\right] \leq P_n[Yg(X) \leq \theta] + e^{-K\theta^2/8}.$$

Steps 1 and 3 are used to ensure existence of good approximating $\tilde{g} \in C_K$ for each $g \in C$. Step 2 just uses Hoeffding's inequality and a union bound over a finite class.

To prove Step 1, observe that for any $(x, y) \in \mathcal{X} \times \{\pm 1\}$ and any $g \in C$,

$$\mathbb{1}\{yg(x) \leq 0\} \leq \mathbb{1}\{yG_g(x) \leq \theta/2\} + \mathbb{1}\{yG_g(x) > \theta/2 \wedge yg(x) \leq 0\}.$$

Now we take expectation with respect to G_g:

$$\mathbb{1}\{yg(x) \leq 0\} \leq \mathbb{E}_{G_g}[\mathbb{1}\{yG_g(x) \leq \theta/2\}] + \Pr_{G_g}[yG_g(x) > \theta/2 \wedge yg(x) \leq 0].$$

Consider $\mathbb{E}_{G_g}[\mathbb{1}\{yG_g(x) > \theta/2 \wedge yg(x) \leq 0\}]$. For any (x, y) such that $yg(x) \leq 0$, we have

$$\Pr_{G_g}[yG_g(x) > \theta/2 \wedge yg(x) \leq 0] \leq e^{-K(\theta/2)^2/2} = e^{-K\theta^2/8};$$

this is a consequence of Hoeffding's inequality. For any (x, y) such that $yg(x) > 0$,

$$\Pr_{G_g}[yG_g(x) > \theta/2 \wedge yg(x) \leq 0] = 0.$$

Therefore,

$$\mathbb{1}\{yg(x) \leq 0\} \leq \mathbb{E}_{G_g}[\mathbb{1}\{yG_g(x) \leq \theta/2\}] + e^{-K\theta^2/8}.$$

Now we replace (x, y) with (X, Y), and then take expectation with respect to $(X, Y) \sim P$:

$$\begin{aligned} P[Yg(X) \leq 0] &\leq \mathbb{E}_{(X,Y)\sim P}[\mathbb{E}_{G_g}[\mathbb{1}\{YG_g(X) \leq \theta/2\}]] + e^{-K\theta^2/8} \\ &= \mathbb{E}_{G_g}[P[YG_g(X) \leq \theta/2]] + e^{-K\theta^2/8}, \end{aligned}$$

which finishes the proof of Step 1.

The proof of Step 3 is similar, except it concludes with taking expectation with respect to $(X, Y) \sim P_n$.

Now we put everything together. By Steps 1 and 3, for every $g \in C$, we have

$$\mathbb{E}_{G_g}\Big[P[Yg(X) \leq 0] - P[YG_g(X) \leq \theta/2] \\ + P_n[YG_g(X) \leq \theta/2] - P_n[Yg(X) \leq \theta]\Big] \leq 2e^{-K\theta^2/8}.$$

Therefore, for every $g \in C$, there exists $\tilde{g} \in C_K$ such that

$$P[Yg(X) \leq 0] - P[Y\tilde{g}(X) \leq \theta/2] \\ + P_n[Y\tilde{g}(X) \leq \theta/2] - P_n[Yg(X) \leq \theta] \leq 2e^{-K\theta^2/8},$$

or in other words,

$$\begin{aligned}&P[Yg(X) \leq 0] - P_n[Yg(X) \leq \theta] \\ &\qquad\qquad\qquad \leq P[Y\tilde{g}(X) \leq \theta/2] - P_n[Y\tilde{g}(X) \leq \theta/2] + 2e^{-K\theta^2/8}.\end{aligned}$$

Now applying the Step 2 result gives, with probability at least $1-\delta$, for all $g \in C$,

$$P[Yg(X) \leq 0] - P_n[Yg(X) \leq \theta] \leq \sqrt{\frac{\log(|C_K|/\delta)}{2n}} + 2e^{-K\theta^2/8}.$$

Choosing $K = \frac{8\ln(2n)}{\theta^2}$ gives the desired conclusion. ▮

We note that the margin bound can also be proved using Rademacher complexity (and it gives a somewhat tighter bound).