

Lecture 3: Theoretical Analysis of Boosting Methods

Tianpei Xie

Feb. 6th., 2023

Contents

1	Boosting Algorithm	2
1.1	AdaBoost	2
1.2	Gradient Boost	2
2	Theoretical Guarantee for Boosting	2
2.1	Weak Learner	4
2.2	Training Error Bounds	5
2.3	Generalization Error Bounds for Finite Hypothesis Class	5
2.4	Generalization Error Bounds via VC Dimension	7
2.5	Generalization Error Bounds via Large Margin Theory	8
3	Fundamental Perspectives	8
3.1	Game Theory	8
3.2	Online Learning	8
3.3	Maximum Entropy Estimation	8
3.4	Iterative Projection Algorithms and Convergence Analysis	8

Algorithm 1.1
The boosting algorithm AdaBoost

Given: $(x_1, y_1), \dots, (x_m, y_m)$ where $x_i \in \mathcal{X}$, $y_i \in \{-1, +1\}$.

Initialize: $D_1(i) = 1/m$ for $i = 1, \dots, m$.

For $t = 1, \dots, T$:

- Train weak learner using distribution D_t .
- Get weak hypothesis $h_t : \mathcal{X} \rightarrow \{-1, +1\}$.
- Aim: select h_t to minimize the weighted error:

$$\epsilon_t \doteq \Pr_{i \sim D_t}[h_t(x_i) \neq y_i].$$

- Choose $\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$.
- Update, for $i = 1, \dots, m$:

$$\begin{aligned} D_{t+1}(i) &= \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases} \\ &= \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}, \end{aligned}$$

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis:

$$H(x) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(x) \right).$$

Figure 1: AdaBoost Algorithm [Schapire and Freund, 2012]

1 Boosting Algorithm

1.1 AdaBoost

-

1.2 Gradient Boost

-

2 Theoretical Guarantee for Boosting

- **Remark (*Data*)**

Define an **observation** as a d -dimensional vector x . The *unknown* nature of the observation is called a **class**, denoted as y . The domain of observation is called an **input space** or **feature space**, denoted as $\mathcal{X} \subset \mathbb{R}^d$, whereas the domain of class is called the **target space**, denoted as \mathcal{Y} . For **classification task**, $\mathcal{Y} = \{1, \dots, M\}$; and for **regression task**, $\mathcal{Y} = \mathbb{R}$.

Algorithm 10.3 *Gradient Tree Boosting Algorithm.*

1. Initialize $f_0(x) = \arg \min_{\gamma} \sum_{i=1}^N L(y_i, \gamma)$.

2. For $m = 1$ to M :

(a) For $i = 1, 2, \dots, N$ compute

$$r_{im} = - \left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} \right]_{f=f_{m-1}}.$$

(b) Fit a regression tree to the targets r_{im} giving terminal regions

$$R_{jm}, j = 1, 2, \dots, J_m.$$

(c) For $j = 1, 2, \dots, J_m$ compute

$$\gamma_{jm} = \arg \min_{\gamma} \sum_{x_i \in R_{jm}} L(y_i, f_{m-1}(x_i) + \gamma).$$

(d) Update $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$.

3. Output $\hat{f}(x) = f_M(x)$.

Figure 2: Gradient Boost Algorithm [Hastie et al., 2009]

A **concept** $c : \mathcal{X} \rightarrow \mathcal{Y}$ is the *input-output association* from the nature and is *to be learned* by a **learning algorithm**. Denote \mathcal{C} as the *set of all concepts* we wish to learn as the **concept class**. The learner is requested to output a *prediction rule*, $h : \mathcal{X} \rightarrow \mathcal{Y}$. This function is also called a **predictor**, a **hypothesis**, or a **classifier**. The predictor can be used to predict the label of new domain points. Denote a collection of n **samples** as

$$\mathcal{D} \equiv \mathcal{D}_n = ((X_1, Y_1), \dots, (X_n, Y_n)) \equiv ((X_1, c(X_1)), \dots, (X_n, c(X_n))).$$

Note that \mathcal{D}_n is a finite **sub-sequence** in $(\mathcal{X} \times \mathcal{Y})^n$.

- **Definition (Generalization Error in Deterministic Scenario)** [Mohri et al., 2018]
Under a *deterministic scenario*, generalization error or the risk or simply error for the classifier $h \in \mathcal{H}$ is defined as

$$L(h) \equiv L_{\mathcal{P}, c}(h) = \mathcal{P} \{h(X) \neq c(X)\} \equiv \mathbb{E}_X [\mathbb{1} \{h(X) \neq c(X)\}] \quad (1)$$

with respect to the concept $c \in \mathcal{C}$ and the *feature distribution* $\mathcal{P} \equiv \mathcal{P}_X$.

- **Definition (Empirical Error or Training Error)**
Given the data \mathcal{D} , the **training error** or the empirical error/risk of a hypothesis $h \in \mathcal{H}$ is defined as

$$\hat{L}(h) \equiv \hat{L}_{\mathcal{D}}(h) = \frac{1}{n} \sum_{i=1}^n \mathbb{1} \{h(X_i) \neq Y_i\} = \frac{1}{n} |\{i : h(X_i) \neq Y_i\}| := \hat{\mathbb{E}} [\mathbb{1} \{h(X) \neq Y\}]$$

where either $Y = c(X)$ or Y is a random variable associated with X .

- **Definition (*The Realizable Assumption*)**

There exists $h^* \in \mathcal{H}$ s.t. $L_{\mathcal{P},c}(h^*) = 0$.

- **Definition (*PAC Learnability*)**

A hypothesis class \mathcal{H} is **PAC learnable** if there exist a function $m_{\mathcal{H}} : (0, 1)^2 \rightarrow \mathbb{N}$ and a learning algorithm with the following property: For every $\epsilon, \delta \in (0, 1)$, for every distribution \mathcal{P} over \mathcal{X} , and for every labeling function $c : \mathcal{X} \rightarrow \{0, 1\}$, if the realizable assumption holds with respect to \mathcal{H} , \mathcal{P} , c , then when running the learning algorithm on $m \geq m_{\mathcal{H}}(\epsilon, \delta)$ i.i.d. examples generated by \mathcal{P} and labeled by c , the algorithm returns a hypothesis h such that, with probability of at least $1 - \delta$ (over the choice of the examples),

$$L_{\mathcal{P},c}(h) \leq \epsilon.$$

2.1 Weak Learner

- **Definition (γ -Weak Learnability)** [Schapire and Freund, 2012, Shalev-Shwartz and Ben-David, 2014]

A learning algorithm, \mathcal{A} , is a **γ -weak-learner** for a class \mathcal{H} if there exists a function $m_{\mathcal{H}} : (0, 1) \rightarrow \mathbb{N}$ such that for **every** $\delta \in (0, 1)$, **for every distribution** \mathcal{P} over \mathcal{X} , and **for every labeling function** $c : \mathcal{X} \rightarrow \{-1, +1\}$, if the realizable assumption holds with respect to \mathcal{H} , \mathcal{P} , c , then when running the learning algorithm on $m \geq m_{\mathcal{H}}(\delta)$ i.i.d. examples generated by \mathcal{P} and labeled by c , the algorithm returns a hypothesis h such that, with probability of at least $1 - \delta$,

$$L_{\mathcal{P},c}(h) \leq \frac{1}{2} - \gamma.$$

A hypothesis class \mathcal{H} is **γ -weak-learnable** if there exists a γ -weak-learner for that class.

- **Remark** We call PAC learnable **the strong learnable**.

- **Remark (*Weak Learner Without Accuracy Guarantee*)**

Unlike the *PAC learner*, who guarantees that *with high probability* the generalization error rate is less than ϵ **for all** ϵ , a **γ -weak-learner** guarantees that with high probability, the error rate is less than ϵ **for some** $\epsilon = 1/2 - \gamma$, i.e. *less than half with a margin γ* .

In other word, *under the realizability assumption*, it is expected that **with more data, a PAC learner** can learn the “true” labeling function behind the data, (i.e. **zero generalization error** with high probability). While a **γ -weak-learner** can only get **slightly better than random guess** and it is **not expected to have lower error rate** even if more data are available.

- **Remark (*Weak Learner is as Hard as PAC Learner*)**

The fundamental theorem of learning states that if a hypothesis class \mathcal{H} has a VC dimension d , then the sample complexity of PAC learning \mathcal{H} satisfies $m_{\mathcal{H}}(\epsilon, \delta) \geq C_1(d + \log(1/\delta))/\epsilon$, where C_1 is a constant. Applying this with $\epsilon = 1/2 - \gamma$ we immediately obtain that **if $d = \infty$ then \mathcal{H} is not γ -weak-learnable**.

This implies that from **the statistical perspective** (i.e., if we ignore *computational complexity*), **weak learnability is also characterized by the VC dimension of \mathcal{H} and therefore is just as hard as PAC (strong) learning**. However, when we do consider **computational**

complexity, the potential advantage of weak learning is that maybe there is *an algorithm that satisfies the requirements of weak learning and can be implemented efficiently*.

2.2 Training Error Bounds

- **Remark** Recall that $h_t \in \mathcal{H}$ are base learners for $t \in [1, T]$, and $(\alpha_1, \dots, \alpha_T) \in \Sigma_T$. The combined learner is

$$H(x) := \text{sgn} \left(\sum_{t=1}^T \alpha_t h_t(x) \right)$$

- The space of all such combined classifiers is defined as below:

Definition (Class of Linear Combination of Base Hypotheses)

Define the class of T **linear combinations of base hypotheses** from \mathcal{H} as

$$L(\mathcal{H}, T) := \left\{ \text{sgn} \left(\sum_{t=1}^T \alpha_t h_t(\cdot) \right) : \alpha \in \mathbb{R}^T, h_t \in \mathcal{H}, t = 1, \dots, T \right\} \quad (2)$$

- **Definition** Define Σ_n as the space of all **linear threshold functions**

$$\Sigma_n := \{ \text{sgn}(\langle w, x \rangle) : w \in \mathbb{R}^n \}.$$

Thus $L(\mathcal{H}, T) = \{ \sigma(h_1(x), \dots, h_T(x)) : \sigma \in \Sigma_T \}$

- **Proposition 2.1 (Training Error Bound for AdaBoost)** [Schapire and Freund, 2012]
Given the notation of Adaboost algorithm, let $\gamma_t = 1/2 - \epsilon_t$, and let \mathcal{D}_1 be an arbitrary initial distribution over the training set. Then **the weighted training error** of the combined classifier \mathcal{H} with respect to \mathcal{D}_1 is bounded as

$$\hat{L}_{\mathcal{D}_1}(H) \leq \prod_{t=1}^T \sqrt{1 - 4\gamma_t^2} \leq \exp \left(-2 \sum_{t=1}^T \gamma_t^2 \right). \quad (3)$$

2.3 Generalization Error Bounds for Finite Hypothesis Class

- **Definition (Restriction of \mathcal{H} to \mathcal{D}).**

Let \mathcal{H} be a class of functions from \mathcal{X} to $\{0, 1\}$ and let $\mathcal{D} = \{x_1, \dots, x_m\} \subset \mathcal{X}$.

The restriction of \mathcal{H} to \mathcal{D} is the set of functions from \mathcal{D} to $\{0, 1\}$ that can be derived from \mathcal{H} . That is,

$$\mathcal{H}_{\mathcal{D}} := \{ (h(x_1), \dots, h(x_m)) : h \in \mathcal{H} \},$$

where we **represent** each function from \mathcal{X} to $\{0, 1\}$ as a **vector** in $\{0, 1\}^{|\mathcal{D}|}$.

- **Definition (Shattering).**

A hypothesis class \mathcal{H} **shatters** a finite set $\mathcal{D} \subset \mathcal{X}$ if **the restriction of \mathcal{H} to \mathcal{D}** is the set of **all functions** from \mathcal{D} to $\{0, 1\}$. That is,

$$|\mathcal{H}_{\mathcal{D}}| = 2^{|\mathcal{D}|}.$$

- **Definition (Growth Function).**

Let \mathcal{H} be a hypothesis class. Then the growth function of \mathcal{H} , denoted $\tau_{\mathcal{H}} : \mathbb{N} \rightarrow \mathcal{N}$, is defined as

$$\tau_{\mathcal{H}}(m) := \max_{\mathcal{D} \subset \mathcal{X} : |\mathcal{D}|=m} |\mathcal{H}_{\mathcal{D}}|.$$

In words, $\tau_{\mathcal{H}}(m)$ is **the number of different functions** from a set \mathcal{D} of **size m** to $\{0, 1\}$ that can be obtained by **restricting \mathcal{H} to \mathcal{D}** .

- **Lemma 2.2 (Sauer's Lemma).** [Shalev-Shwartz and Ben-David, 2014, Mohri et al., 2018]
Let \mathcal{H} be a hypothesis class with $VCdim(\mathcal{H}) \leq d < \infty$. Then, for all $m \geq d + 1$,

$$\tau_{\mathcal{H}}(m) \leq \sum_{i=0}^d \binom{m}{i} \leq \left(\frac{em}{d}\right)^d. \quad (4)$$

- **Proposition 2.3 (Growth Function Generalization Bound)** [Mohri et al., 2018]
Let \mathcal{H} be a family of functions taking values in $\{-1, +1\}$. Then, for any $\delta > 0$, with probability at least $1 - \delta$, for any $h \in \mathcal{H}$,

$$L(h) \leq \hat{L}_m(h) + \sqrt{\frac{2 \log \tau_{\mathcal{H}}(m)}{m}} + \sqrt{\frac{\log(1/\delta)}{2m}} \quad (5)$$

Growth function bounds can be also derived directly (without using Rademacher complexity bounds first). The resulting bound is then the following:

$$\mathcal{P} \left\{ \exists h \in \mathcal{H}, \left| L(h) - \hat{L}_m(h) \right| > \epsilon \right\} \leq 4\tau_{\mathcal{H}}(2m) \exp \left(-\frac{m\epsilon^2}{8} \right) \quad (6)$$

which only differs from (??) by constants.

- The following lemma shows that the VC dimension of Σ_T is T .

Lemma 2.4 [Schapire and Freund, 2012]

The space Σ_n of **linear threshold functions** over \mathbb{R}^n has **VC-dimension n** .

- **Lemma 2.5 (Growth Number of Combined Hypothesis Class, Finite Hypothesis Class)** [Schapire and Freund, 2012, Shalev-Shwartz and Ben-David, 2014]
Assume \mathcal{H} is **finite**. Let $m \geq T \geq 1$. For any set \mathcal{D} of m points, the number of dichotomies realizable by $L(\mathcal{H}, T)$ is bounded as follows:

$$|L(\mathcal{H}, T)| \leq \tau_{L(\mathcal{H}, T)}(m) \leq \left(\frac{em}{T}\right)^T |\mathcal{H}|^T. \quad (7)$$

- **Theorem 2.6 (Growth Function Generalization Bound for AdaBoost, Finite Hypothesis)** [Schapire and Freund, 2012]

Suppose **AdaBoost** is run for T rounds on $m \geq T$ random examples, using base classifiers from a **finite space \mathcal{H}** . Then, with probability at least $1 - \delta$, the combined classifier H satisfies

$$L_{\mathcal{P},c}(H) \leq \hat{L}_m(H) + \sqrt{\frac{2T(\log |\mathcal{H}| + \log(em/T))}{m}} + \sqrt{\frac{\log(1/\delta)}{2m}} \quad (8)$$

Furthermore, with probability at least $1 - \delta$, if \mathcal{H} is realizable with the training set (i.e. $\hat{L}_m(h) \equiv 0$), then

$$L_{\mathcal{P},c}(H) \leq \frac{2T(\log |\mathcal{H}| + \log(2em/T)) + 2 \log(2/\delta)}{m}. \quad (9)$$

2.4 Generalization Error Bounds via VC Dimension

- **Lemma 2.7** (*Growth Number of Combined Hypothesis Class, VC Class*). [Schapire and Freund, 2012]

Assume \mathcal{H} has **finite VC-dimension** $d \geq 1$. Let $m \geq \max\{T, d\} \geq 1$. For any set \mathcal{D} of m points, the number of dichotomies realizable by $L(\mathcal{H}, T)$ is bounded as follows:

$$|L(\mathcal{H}, T)| \leq \tau_{L(\mathcal{H}, T)}(m) \leq \left(\frac{em}{T}\right)^T \left(\frac{em}{d}\right)^{dT}. \quad (10)$$

- **Lemma 2.8** (*VC-Dimension of Combined Hypothesis Class, VC Class*). [Schapire and Freund, 2012, Shalev-Shwartz and Ben-David, 2014]

Assume \mathcal{H} has **finite VC-dimension** $\nu(\mathcal{H}) = d$ and $\min\{T, d\} \geq 3$. Then the VC dimension of combined hypothesis class is bounded by

$$\nu(L(\mathcal{H}, T)) \leq T(d+1)(3\log(T(d+1)) + 2) = \mathcal{O}(dT \log(dT)). \quad (11)$$

- **Remark** (*Lower Bound on VC Dimension*). [Shalev-Shwartz and Ben-David, 2014]

For some base hypothesis class \mathcal{H} , the VC-dimension of ensemble is at least Td . For instance, for \mathcal{H}_n be the class of *decision stumps* over \mathbb{R}^n , we can show that $\log(n) \leq d = \nu(\mathcal{H}) \leq 2\log(n) + 5$. In this example, for all $T \geq 1$,

$$\nu(L(\mathcal{H}_n, T)) \geq 0.5T \log(n) \asymp \Omega(Td).$$

- **Theorem 2.9** (*Growth Function Generalization Bound for AdaBoost*). [Schapire and Freund, 2012]

Suppose **AdaBoost** is run for T rounds on $m \geq \max\{T, d\}$ random examples, using base classifiers from a **finite space** \mathcal{H} . Then, with probability at least $1 - \delta$, the combined classifier H satisfies

$$L_{\mathcal{P}, c}(H) \leq \hat{L}_m(H) + \sqrt{\frac{2T(d \log(em/d) + \log(em/T))}{m}} + \sqrt{\frac{\log(1/\delta)}{2m}} \quad (12)$$

Furthermore, with probability at least $1 - \delta$, if \mathcal{H} is realizable with the training set (i.e. $\hat{L}_m(h) \equiv 0, \forall h \in \mathcal{H}$), then

$$L_{\mathcal{P}, c}(H) \leq \frac{2T(d \log(2em/d) + \log(2em/T)) + 2\log(2/\delta)}{m}. \quad (13)$$

- **Corollary 2.10** [Schapire and Freund, 2012]

Assume, in addition to the assumptions of theorem 2.9, that each base classifier has weighted error $\epsilon_t \leq 1/2 - \gamma$ for some $\gamma > 0$. Let the number of rounds T be equal to

$$\inf \left\{ t \in \mathbb{N} : t \geq \frac{\log(m)}{2\gamma^2} \right\}$$

Then, with probability at least $1 - \delta$, the generalization error of the combined classifier H will be at most

$$\mathcal{O} \left(\frac{1}{m} \left[\frac{\log(m)}{\gamma^2} \left(\log(m) + d \log \left(\frac{m}{d} \right) \right) + \frac{1}{\delta} \right] \right)$$

- **Remark** Ignoring the log factor, the generalization error bound (12) can be summarized as

$$L_{\mathcal{P},c}(H) \leq \hat{L}_m(H) + \mathcal{O}\left(\sqrt{\frac{TC_{\mathcal{H}}}{m}}\right)$$

where $C_{\mathcal{H}}$ is some complexity measure of base class \mathcal{H} .

- **Theorem 2.11** (***Strong Learnable = Weak Learnable***) [Schapire and Freund, 2012]
*A target class \mathcal{H} is (efficiently) **weakly** PAC learnable **if and only if** it is (efficiently) **strongly** PAC learnable.*

2.5 Generalization Error Bounds via Large Margin Theory

3 Fundamental Perspectives

3.1 Game Theory

3.2 Online Learning

3.3 Maximum Entropy Estimation

3.4 Iterative Projection Algorithms and Convergence Analysis

References

- Trevor Hastie, Robert Tibshirani, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.
- Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.
- Robert E. Schapire and Yoav Freund. *Boosting: Foundations and Algorithms*. The MIT Press, 2012. ISBN 0262017180.
- Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.