CS 6316 Machine Learning

Boosting

Yangfeng Ji

Department of Computer Science
University of Virginia

UNIVERSITY of VIRGINIA | ENGINEERING

Overview

The Bias-Variance Decomposition

The expected error is decomposed as

$$E\left[\epsilon^2\right] = \underbrace{E\left[\{h(\boldsymbol{x}, S) - E\left[h(\boldsymbol{x}, S)\right]\}^2\right]}_{\text{variance}} + \underbrace{\{E\left[h(\boldsymbol{x}, S)\right] - f_{\mathscr{D}}(\boldsymbol{x})\}^2}_{\text{bias}^2}$$

- **bias**: how far the expected prediction $E\left[h(\boldsymbol{x}, S)\right]$ diverges from the optimal predictor $f_{\mathscr{D}}(\boldsymbol{x})$
- **variance**: how a hypothesis learned from a specific S diverges from the average prediction $E\left[h(\boldsymbol{x}, S)\right]$

Motivation

How can we reduce the overall error?

E.g.,

- Reduce the bias
  - Boosting: start with simple classifiers, and gradually make a powerful one
- Reduce the variance
  - Bagging: create multiple copies of data and train classifiers on each of them, then combine them together

The Idea of Boosting

Thoughts on Hypothesis Boosting.

Machine Learning class project, Dec. 1988

Michael Kearns

In this paper we present initial and modest progress on the *Hypothesis Boosting Problem*. Informally, this problem asks whether an efficient learning algorithm (in the distribution-free model of [V84]) that outputs an hypothesis whose performance is only slightly better than random guessing implies the existence of an efficient algorithm that outputs an hypothesis of arbitrary accuracy. The resolution of this question is of theoretical interest and possibly of practical importance. From the theoretical standpoint, we are interested more generally in the question of whether there is a discrete hierarchy of achievable accuracy in the model of [V84]; from a practical standpoint, the collapse of such a proposed hierarchy may yield an efficient algorithm for converting relatively poor hypotheses into very good hypotheses.

Weak Learnability

Weak Learnability

- A learning algorithm A is a γ-weak-learner for a hypothesis space, if for the PAC learning condition, the algorithm returns a hypothesis h such that, with probability of at least $1 - \delta$,

$$L_{(\mathscr{D},f)}(h) \leq \frac{1}{2} - \gamma \tag{1}$$

- A hypothesis space $\mathscr{H}$ is γ-weak-learnable if there exists a γ-weak-learner for this class

Strong vs. Weak Learnability

- Strong learnability

$$L_{(\mathscr{D},f)}(h) \leq \epsilon \tag{2}$$

  where ϵ is arbitrarily small

- Weak learnability

$$L_{(\mathscr{D},f)}(h) \leq \frac{1}{2} - \gamma \tag{3}$$

  where $\gamma > 0$. In other words, the error rate of weak learnability is slightly better than random guessing

Decision Stumps

- Let $\mathcal{X} = \mathbb{R}^d$, the hypothesis space of decision stumps is defined as

$$\mathcal{H}_{\text{DS}} = \{b \cdot \text{sign}(x_{\cdot,j} - \theta) : \theta \in \mathbb{R}, j \in [d]\} \tag{4}$$

with parameters $\theta \in \mathbb{R}$, $j \in [d]$, and $b \in \{-1, +1\}$

Decision Stumps

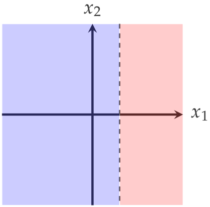
- Let $\mathcal{X} = \mathbb{R}^d$, the hypothesis space of decision stumps is defined as

$$\mathcal{H}_{\text{DS}} = \{b \cdot \text{sign}(x_{\cdot,j} - \theta) : \theta \in \mathbb{R}, j \in [d]\} \tag{4}$$

with parameters $\theta \in \mathbb{R}$, $j \in [d]$, and $b \in \{-1, +1\}$

- For each $h_{\theta,j,b} \in \mathcal{H}_{\text{DS}}$ with $j = 1$ and $b = +1$

$$h_{\theta,1,+1}(\boldsymbol{x}) = \begin{cases} +1 & \boldsymbol{x}_{\cdot,1} > \theta \\ -1 & \boldsymbol{x}_{\cdot,1} < \theta \end{cases} \tag{5}$$



Empirical Risk

- The empirical risk with a training set $S = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)\}$ is defined as

$$L_D(h_{\theta,j,b}) = \sum_{i=1}^{m} D_i \cdot \mathbf{1}[h_{\theta,j,b}(\boldsymbol{x}_i) \neq y_i] \tag{6}$$

where $\mathbf{1}[\cdot]$ is the indicator function and $\mathbf{1}[h(\boldsymbol{x}_i) \neq y_i] = 1$ when $h(\boldsymbol{x}_i) \neq y_i$ is true

Empirical Risk

- The empirical risk with a training set $S = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)\}$ is defined as

$$L_D(h_{\theta,j,b}) = \sum_{i=1}^{m} D_i \cdot \mathbf{1}[h_{\theta,j,b}(\boldsymbol{x}_i) \neq y_i] \tag{6}$$

  where $\mathbf{1}[\cdot]$ is the indicator function and $\mathbf{1}[h(\boldsymbol{x}_i) \neq y_i] = 1$ when $h(\boldsymbol{x}_i) \neq y_i$ is true
- A special case with $D_i = \frac{1}{m}$, then

$$L_D(h) = L_S(h) = \frac{\sum_{i=1}^{m} \mathbf{1}[h(\boldsymbol{x}_i) \neq y_i]}{m} \tag{7}$$

Learning a Decision Stump

- For each $j \in [d]$
  - Sort training examples, such that

$$x_{1,j} \leq x_{2,j} \leq \cdots \leq x_{m,j} \tag{8}$$

Learning a Decision Stump

- For each $j \in [d]$
  - Sort training examples, such that

$$\boldsymbol{x}_{1,j} \leq \boldsymbol{x}_{2,j} \leq \cdots \leq \boldsymbol{x}_{m,j} \tag{8}$$

  - Define
    $\Theta_j = \{\frac{\boldsymbol{x}_{i,j}+\boldsymbol{x}_{i+1,j}}{2} : i \in [m-1]\} \cup \{(\boldsymbol{x}_{1,j} - 1), (\boldsymbol{x}_{m,j} + 1)\}$

Learning a Decision Stump

- For each $j \in [d]$
  - Sort training examples, such that
    $$x_{1,j} \leq x_{2,j} \leq \cdots \leq x_{m,j} \tag{8}$$
  - Define
    $\Theta_j = \{\frac{x_{i,j}+x_{i+1,j}}{2} : i \in [m-1]\} \cup \{(x_{1,j}-1), (x_{m,j}+1)\}$
  - Try each $\theta' \in \Theta_j$ and find the minimal risk with j
    $$L_D(h_{\theta',j,b}) = \sum_{i=1}^{m} D_i \cdot \mathbf{1}[h_{\theta',j}(x_i) \neq y_i] \tag{9}$$

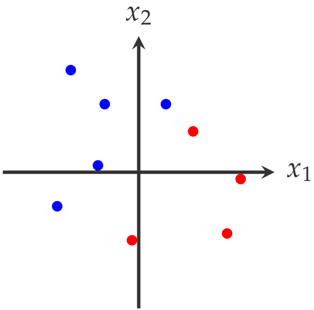
Learning a Decision Stump

- For each $j \in [d]$
  - Sort training examples, such that
$$\boldsymbol{x}_{1,j} \leq \boldsymbol{x}_{2,j} \leq \cdots \leq \boldsymbol{x}_{m,j} \tag{8}$$
  - Define
    $\Theta_j = \{\frac{\boldsymbol{x}_{i,j}+\boldsymbol{x}_{i+1,j}}{2} : i \in [m-1]\} \cup \{(\boldsymbol{x}_{1,j}-1), (\boldsymbol{x}_{m,j}+1)\}$
  - Try each $\theta' \in \Theta_j$ and find the minimal risk with j
$$L_D(h_{\theta',j,b}) = \sum_{i=1}^{m} D_i \cdot \mathbf{1}[h_{\theta',j}(\boldsymbol{x}_i) \neq y_i] \tag{9}$$
- Find the minimal risk for all $j \in [d]$

Example

Build a decision stump for the following classification task with the assumption that

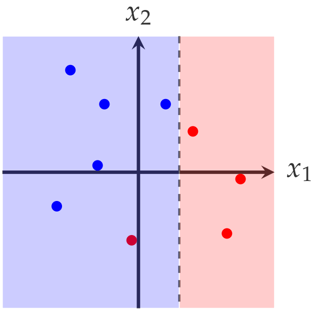
$$D = (\frac{1}{9}, \ldots, \frac{1}{9}) \tag{10}$$



Example

Build a decision stump for the following classification task with the assumption that

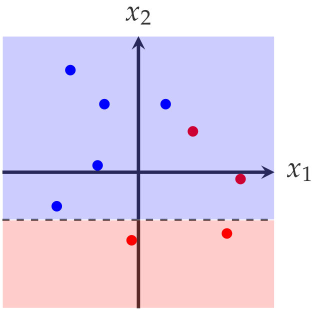
$$D = (\frac{1}{9}, \dots, \frac{1}{9}) \tag{10}$$



Example

Build a decision stump for the following classification task with the assumption that

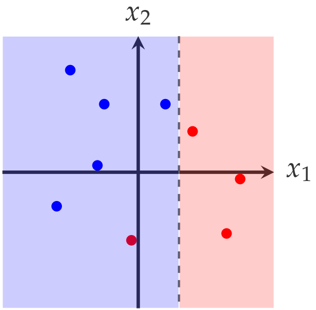
$$D = (\frac{1}{9}, \dots, \frac{1}{9}) \tag{10}$$



Example

Build a decision stump for the following classification task with the assumption that

$$D = (\frac{1}{9}, \dots, \frac{1}{9}) \tag{10}$$



The best decision stump is $\boldsymbol{x}_{\cdot,1} = 0.6$

Boosting

Boosting

Q: Cen we boost a set of weak classifiers and make a strong classifier?

Boosting

Q: Cen we boost a set of weak classifiers and make a strong classifier?

A: Yes. It looks like

$$h_S(\boldsymbol{x}) = \text{sign}(\sum_{t=1}^{T} w_t h_t(\boldsymbol{x})) \tag{11}$$

Boosting

Q: Cen we boost a set of weak classifiers and make a strong classifier?

A: Yes. It looks like

$$h_S(\boldsymbol{x}) = \text{sign}(\sum_{t=1}^{T} w_t h_t(\boldsymbol{x})) \tag{11}$$

Three questions

- How to find each weak classifier $h_t(\boldsymbol{x})$?
- How to compute w_t?
- How large the T is?

AdaBoost

Input: $S = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m))\}$, weak learner A, number of rounds T

Initialize $D^{(1)} = (\frac{1}{m}, \ldots, \frac{1}{m})$

for $t = 1, \ldots, T$ **do**

end for

Output: the hypothesis $h_S(\boldsymbol{x}) = \text{sign}(\sum_{t=1}^{T} w_t h_t(\boldsymbol{x}))$

AdaBoost

Input: $S = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m))\}$, weak learner A, number of rounds T

Initialize $D^{(1)} = (\frac{1}{m}, \ldots, \frac{1}{m})$

for $t = 1, \ldots, T$ **do**

  Learn a weak classifier $h_t = A(D^{(t)}, S)$

end for

Output: the hypothesis $h_S(\boldsymbol{x}) = \mathrm{sign}(\sum_{t=1}^{T} w_t h_t(\boldsymbol{x}))$

AdaBoost

Input: $S = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m))\}$, weak learner A, number of rounds T

Initialize $D^{(1)} = (\frac{1}{m}, \ldots, \frac{1}{m})$

for $t = 1, \ldots, T$ **do**

    Learn a weak classifier $h_t = A(D^{(t)}, S)$

    Compute error $\epsilon_t = \sum_{i=1}^{m} D_i^{(t)} \mathbf{1}[h_t(\boldsymbol{x}_i) \neq y_i]$

end for

Output: the hypothesis $h_S(\boldsymbol{x}) = \text{sign}(\sum_{t=1}^{T} w_t h_t(\boldsymbol{x}))$

AdaBoost

Input: $S = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m))\}$, weak learner A, number of rounds T

Initialize $D^{(1)} = (\frac{1}{m}, \ldots, \frac{1}{m})$

for $t = 1, \ldots, T$ **do**

  Learn a weak classifier $h_t = A(D^{(t)}, S)$

  Compute error $\epsilon_t = \sum_{i=1}^m D_i^{(t)} \mathbf{1}[h_t(\boldsymbol{x}_i) \neq y_i]$

  Let $w_t = \frac{1}{2} \log(\frac{1}{\epsilon_t} - 1)$

end for

Output: the hypothesis $h_S(\boldsymbol{x}) = \text{sign}(\sum_{t=1}^T w_t h_t(\boldsymbol{x}))$

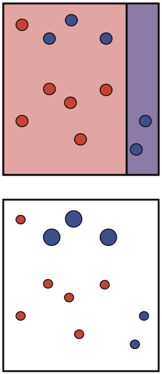
AdaBoost

Input: $S = \{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_m, y_m))\}$, weak learner A, number of rounds T
Initialize $D^{(1)} = (\frac{1}{m}, \dots, \frac{1}{m})$
for $t = 1, \dots, T$ **do**
    Learn a weak classifier $h_t = A(D^{(t)}, S)$
    Compute error $\epsilon_t = \sum_{i=1}^{m} D_i^{(t)} \mathbf{1}[h_t(\boldsymbol{x}_i) \neq y_i]$
    Let $w_t = \frac{1}{2} \log(\frac{1}{\epsilon_t} - 1)$
    Update, for all $i = 1, \dots, m$

$$D_i^{(t+1)} = \frac{D_i^{(t)} \exp(-w_t y_i h_t(\boldsymbol{x}))}{\sum_{j=1}^{m} D_j^{(t)} \exp(-w_t y_j h_t(\boldsymbol{x}_j))}$$

end for
Output: the hypothesis $h_S(\boldsymbol{x}) = \mathrm{sign}(\sum_{t=1}^{T} w_t h_t(\boldsymbol{x}))$

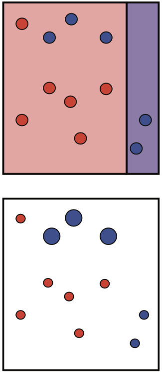
Example



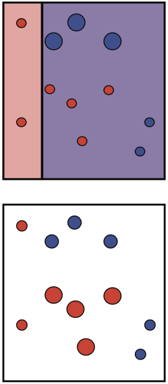
(a) $t = 1$

[Mohri et al., 2018, Page 147]

Example



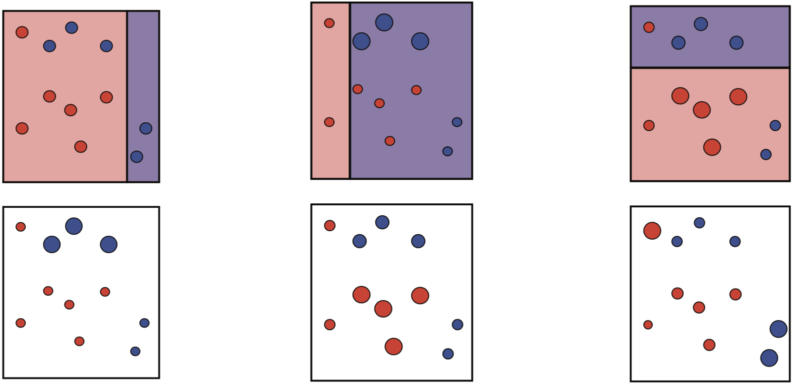
(a) $t = 1$



(b) $t = 2$

[Mohri et al., 2018, Page 147]

Example

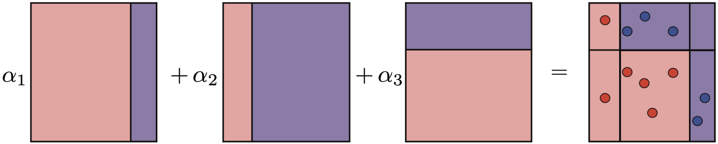


(a) $t = 1$ (b) $t = 2$ (c) $t = 3$

[Mohri et al., 2018, Page 147]

Example (Cont.)

$$\text{sign}\left(\sum_{t=1}^{T} w_t h_t(\boldsymbol{x})\right) = h(\boldsymbol{x}) \tag{12}$$



[Mohri et al., 2018, Page 147]

Theortical Analysis

Let S be a training set and assume that at each iteration of AdaBoost, the weak learner returns a hypothesis for which

$$\epsilon_t \leq \frac{1}{2} - \gamma.$$

[Shalev-Shwartz and Ben-David, 2014, Page 135 – 137]

Theortical Analysis

Let S be a training set and assume that at each iteration of AdaBoost, the weak learner returns a hypothesis for which

$$\epsilon_t \le \frac{1}{2} - \gamma.$$

Then, the training error of the output hypothesis of AdaBoost is at most

$$L_S(h_S) = \frac{1}{m}\mathbf{1}[h_S(\boldsymbol{x}_i) \neq y_i] \le \exp(-2\gamma^2 T) \tag{13}$$

[Shalev-Shwartz and Ben-David, 2014, Page 135 – 137]

VC Dimension

Let

- B be a base hypothesis space (e.g., decision stumps)
- $L(B, T)$ be the hypothesis space produced by the AdaBoost algorithm

[Shalev-Shwartz and Ben-David, 2014, Page 139]

VC Dimension

Let

- B be a base hypothesis space (e.g., decision stumps)
- $L(B, T)$ be the hypothesis space produced by the AdaBoost algorithm

Assume that both T and VC-dim(B) are at least 3. Then,

$$\text{VC-dim}(L(B, T)) \leq \mathcal{O}\{T \cdot \text{VC-dim}(B) \cdot \log(T \cdot \text{VC-dim}(B))\}$$

[Shalev-Shwartz and Ben-David, 2014, Page 139]

Reference

Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2018).
Foundations of machine learning.
MIT press.

Shalev-Shwartz, S. and Ben-David, S. (2014).
Understanding machine learning: From theory to algorithms.
Cambridge university press.