Introduction to Machine Learning (67577)
Lecture 4

Shai Shalev-Shwartz

School of CS and Engineering,
The Hebrew University of Jerusalem

Boosting

Outline

1. Weak learnability
2. Boosting the confidence
3. Boosting the accuracy using AdaBoost
4. AdaBoost as a learner for Halfspaces++
5. AdaBoost and the Bias-Complexity Tradeoff
6. Weak Learnability and Separability with Margin
7. AdaBoost for Face Detection

Weak Learnability

Definition ((ϵ, δ)-Weak-Learnability)

A class $\mathcal{H}$ is (ϵ, δ)-weak-learnable if there exists a learning algorithm, A, and a training set size, $m \in \mathbb{N}$, such that for every distribution $\mathcal{D}$ over $\mathcal{X}$ and every $f \in \mathcal{H}$,

$$\mathcal{D}^m(\{S : L_{\mathcal{D},f}(A(S)) \leq \epsilon\}) \geq 1 - \delta \ .$$

Weak Learnability

Definition ((ϵ, δ)-Weak-Learnability)

A class $\mathcal{H}$ is (ϵ, δ)-weak-learnable if there exists a learning algorithm, A, and a training set size, $m \in \mathbb{N}$, such that for every distribution $\mathcal{D}$ over $\mathcal{X}$ and every $f \in \mathcal{H}$,

$$\mathcal{D}^m(\{S : L_{\mathcal{D},f}(A(S)) \le \epsilon\}) \ge 1 - \delta \ .$$

Remarks:

- Almost identical to (strong) PAC learning, but we only need to succeed for specific ϵ, δ

Weak Learnability

Definition ((ϵ, δ)-Weak-Learnability)

A class $\mathcal{H}$ is (ϵ, δ)-weak-learnable if there exists a learning algorithm, A, and a training set size, $m \in \mathbb{N}$, such that for every distribution $\mathcal{D}$ over $\mathcal{X}$ and every $f \in \mathcal{H}$,

$$\mathcal{D}^m(\{S : L_{\mathcal{D},f}(A(S)) \leq \epsilon\}) \geq 1 - \delta \ .$$

Remarks:

- Almost identical to (strong) PAC learning, but we only need to succeed for specific ϵ, δ
- Every class $\mathcal{H}$ is $(1/2, 0)$-weak-learnable

Weak Learnability

Definition ((ϵ, δ)-Weak-Learnability)

A class $\mathcal{H}$ is (ϵ, δ)-weak-learnable if there exists a learning algorithm, A, and a training set size, $m \in \mathbb{N}$, such that for every distribution $\mathcal{D}$ over $\mathcal{X}$ and every $f \in \mathcal{H}$,

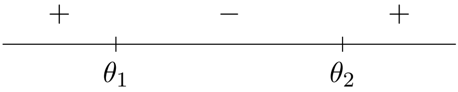
$$\mathcal{D}^m(\{S : L_{\mathcal{D},f}(A(S)) \leq \epsilon\}) \geq 1 - \delta \ .$$

Remarks:

- Almost identical to (strong) PAC learning, but we only need to succeed for specific ϵ, δ
- Every class $\mathcal{H}$ is $(1/2, 0)$-weak-learnable
- Intuitively, one can think of a weak learner as an algorithm that uses a simple 'rule of thumb' to output a hypothesis that performs just slightly better than a random guess

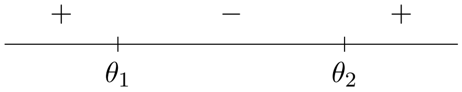
Example of weak learner

- $\mathcal{X} = \mathbb{R}$, $\mathcal{H}$ is the class of 3-piece classifiers, e.g.



Example of weak learner

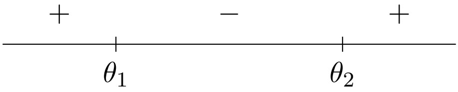
- $\mathcal{X} = \mathbb{R}$, $\mathcal{H}$ is the class of 3-piece classifiers, e.g.



- Let $B = \{x \mapsto \text{sign}(x - \theta) \cdot b : \quad \theta \in \mathbb{R}, b \in \{\pm 1\}\}$ be the class of Decision Stumps

Example of weak learner

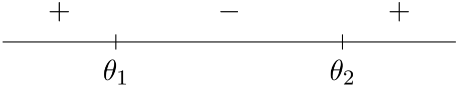
- $\mathcal{X} = \mathbb{R}$, $\mathcal{H}$ is the class of 3-piece classifiers, e.g.



- Let $B = \{x \mapsto \text{sign}(x - \theta) \cdot b : \quad \theta \in \mathbb{R}, b \in \{\pm 1\}\}$ be the class of Decision Stumps
- Claim: There is a constant m, such that ERM_B over m examples is a $(5/12, 1/2)$-weak learner for $\mathcal{H}$

Example of weak learner

- $\mathcal{X} = \mathbb{R}$, $\mathcal{H}$ is the class of 3-piece classifiers, e.g.



- Let $B = \{x \mapsto \text{sign}(x - \theta) \cdot b : \quad \theta \in \mathbb{R}, b \in \{\pm 1\}\}$ be the class of Decision Stumps
- Claim: There is a constant m, such that ERM_B over m examples is a $(5/12, 1/2)$-weak learner for $\mathcal{H}$
- Proof:
  - Observe that there's always a decision stump with $L_{\mathcal{D},f}(h) \leq 1/3$
  - Apply VC bound for the class of decision stumps

The problem of boosting

- Suppose we have an (ϵ_0, δ_0)-weak-learner algorithm, A, for some class $\mathcal{H}$

The problem of boosting

- Suppose we have an (ϵ_0, δ_0)-weak-learner algorithm, A, for some class $\mathcal{H}$
- Can we use A to construct a strong learner ?

The problem of boosting

- Suppose we have an (ϵ_0, δ_0)-weak-learner algorithm, A, for some class $\mathcal{H}$
- Can we use A to construct a strong learner ?
- If A is computationally efficient, can we boost it efficiently ?

The problem of boosting

- Suppose we have an (ϵ_0, δ_0)-weak-learner algorithm, A, for some class $\mathcal{H}$
- Can we use A to construct a strong learner ?
- If A is computationally efficient, can we boost it efficiently ?
- Two questions:

The problem of boosting

- Suppose we have an (ϵ_0, δ_0)-weak-learner algorithm, A, for some class $\mathcal{H}$
- Can we use A to construct a strong learner ?
- If A is computationally efficient, can we boost it efficiently ?
- Two questions:
  - Boosting the confidence

The problem of boosting

- Suppose we have an (ϵ_0, δ_0)-weak-learner algorithm, A, for some class $\mathcal{H}$
- Can we use A to construct a strong learner ?
- If A is computationally efficient, can we boost it efficiently ?
- Two questions:
  - Boosting the confidence
  - Boosting the accuracy

Outline

1 Weak learnability
2 Boosting the confidence
3 Boosting the accuracy using AdaBoost
4 AdaBoost as a learner for Halfspaces++
5 AdaBoost and the Bias-Complexity Tradeoff
6 Weak Learnability and Separability with Margin
7 AdaBoost for Face Detection

Boosting the confidence

- Suppose A is an (ϵ_0, δ_0)-weak learner for $\mathcal{H}$ that requires m_0 examples

Boosting the confidence

- Suppose A is an (ϵ_0, δ_0)-weak learner for $\mathcal{H}$ that requires m_0 examples
- For any $\delta, \epsilon \in (0, 1)$ we show how to learn $\mathcal{H}$ to accuracy $\epsilon_0 + \epsilon$ with confidence δ

Boosting the confidence

- Suppose A is an (ϵ_0, δ_0)-weak learner for $\mathcal{H}$ that requires m_0 examples
- For any $\delta, \epsilon \in (0,1)$ we show how to learn $\mathcal{H}$ to accuracy $\epsilon_0 + \epsilon$ with confidence δ
- Step 1: Apply A on $k = \left\lceil \frac{\log(2/\delta)}{\log(1/\delta_0)} \right\rceil$ i.i.d. samples, each of which of m_0 examples, to obtain $h_1, \ldots, h_k$

Boosting the confidence

- Suppose A is an (ϵ_0, δ_0)-weak learner for $\mathcal{H}$ that requires m_0 examples
- For any $\delta, \epsilon \in (0,1)$ we show how to learn $\mathcal{H}$ to accuracy $\epsilon_0 + \epsilon$ with confidence δ
- Step 1: Apply A on $k = \left\lceil \frac{\log(2/\delta)}{\log(1/\delta_0)} \right\rceil$ i.i.d. samples, each of which of m_0 examples, to obtain $h_1, \ldots, h_k$
- Step 2: Take additional validation sample of size $|V| \geq \frac{2\log(4k/\delta)}{\epsilon^2}$ and output $\hat{h} \in \mathrm{argmin}_{h_i} L_V(h_i)$

Boosting the confidence

- Suppose A is an (ϵ_0, δ_0)-weak learner for $\mathcal{H}$ that requires m_0 examples
- For any $\delta, \epsilon \in (0,1)$ we show how to learn $\mathcal{H}$ to accuracy $\epsilon_0 + \epsilon$ with confidence δ
- Step 1: Apply A on $k = \left\lceil \frac{\log(2/\delta)}{\log(1/\delta_0)} \right\rceil$ i.i.d. samples, each of which of m_0 examples, to obtain $h_1, \ldots, h_k$
- Step 2: Take additional validation sample of size $|V| \geq \frac{2\log(4k/\delta)}{\epsilon^2}$ and output $\hat{h} \in \mathrm{argmin}_{h_i} L_V(h_i)$
- Claim: W.p. at least $1 - \delta$, we have $L_{\mathcal{D}}(\hat{h}) \leq \epsilon_0 + \epsilon$

Proof

- First, by the validation procedure guarantees

$$\mathbb{P}[L_{\mathcal{D}}(\hat{h}) > \min_i L_{\mathcal{D}}(h_i) + \epsilon] \leq \delta/2 \ .$$

Proof

- First, by the validation procedure guarantees

$$\mathbb{P}[L_{\mathcal{D}}(\hat{h}) > \min_i L_{\mathcal{D}}(h_i) + \epsilon] \le \delta/2 \ .$$

- Second,

$$\begin{aligned}
\mathbb{P}[\min_i L_{\mathcal{D}}(h_i) > \epsilon_0] &= \mathbb{P}[\forall_i \, L_{\mathcal{D}}(h_i) > \epsilon_0] \\
&= \prod_{i=1}^{k} \mathbb{P}[L_{\mathcal{D}}(h_i) > \epsilon_0] \\
&\le \delta_0^k \le \delta/2 \ .
\end{aligned}$$

Proof

- First, by the validation procedure guarantees

$$\mathbb{P}[L_{\mathcal{D}}(\hat{h}) > \min_i L_{\mathcal{D}}(h_i) + \epsilon] \le \delta/2 \ .$$

- Second,

$$\begin{aligned}\mathbb{P}[\min_i L_{\mathcal{D}}(h_i) > \epsilon_0] &= \mathbb{P}[\forall_i \, L_{\mathcal{D}}(h_i) > \epsilon_0] \\ &= \prod_{i=1}^{k} \mathbb{P}[L_{\mathcal{D}}(h_i) > \epsilon_0] \\ &\le \delta_0^k \le \delta/2 \ .\end{aligned}$$

- Apply the union bound to conclude the proof.

Boosting a learner that succeeds on expectation

- Suppose that A is a learner that guarantees:

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(A(S))] \le \min_{h\in\mathcal{H}} L_{\mathcal{D}}(h) + \epsilon \ .$$

Boosting a learner that succeeds on expectation

- Suppose that A is a learner that guarantees:

$$\mathop{\mathbb{E}}_{S \sim \mathcal{D}^m}[L_{\mathcal{D}}(A(S))] \le \min_{h \in \mathcal{H}} L_{\mathcal{D}}(h) + \epsilon \ .$$

- Denote $\theta = L_{\mathcal{D}}(A(S)) - \min_{h \in \mathcal{H}} L_{\mathcal{D}}(h)$, so we obtain

$$\mathop{\mathbb{E}}_{S \sim \mathcal{D}^m}[\theta] \le \epsilon \ .$$

Boosting a learner that succeeds on expectation

- Suppose that A is a learner that guarantees:

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(A(S))] \le \min_{h\in\mathcal{H}} L_{\mathcal{D}}(h) + \epsilon \ .$$

- Denote $\theta = L_{\mathcal{D}}(A(S)) - \min_{h\in\mathcal{H}} L_{\mathcal{D}}(h)$, so we obtain

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[\theta] \le \epsilon \ .$$

- Since θ is a non-negative random variable, we can apply Markov's inequality to obtain

$$\mathbb{P}[\theta \ge 2\epsilon] \le \frac{\mathbb{E}[\theta]}{2\epsilon} \le \frac{1}{2} \ .$$

Boosting a learner that succeeds on expectation

- Suppose that A is a learner that guarantees:

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(A(S))] \le \min_{h\in\mathcal{H}} L_{\mathcal{D}}(h) + \epsilon \ .$$

- Denote $\theta = L_{\mathcal{D}}(A(S)) - \min_{h\in\mathcal{H}} L_{\mathcal{D}}(h)$, so we obtain

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[\theta] \le \epsilon \ .$$

- Since θ is a non-negative random variable, we can apply Markov's inequality to obtain

$$\mathbb{P}[\theta \ge 2\epsilon] \le \frac{\mathbb{E}[\theta]}{2\epsilon} \le \frac{1}{2} \ .$$

- Corollary: A is $(2\epsilon, 1/2)$-weak learner.

Outline

1 Weak learnability

2 Boosting the confidence

3 Boosting the accuracy using AdaBoost

4 AdaBoost as a learner for Halfspaces++

5 AdaBoost and the Bias-Complexity Tradeoff

6 Weak Learnability and Separability with Margin

7 AdaBoost for Face Detection

Boosting the accuracy

Problem raised in 1988 by Kearns and Valiant

Solved in 1990 by Robert Schapire, then a graduate student at MIT

In 1995, Schapire & Freund proposed the AdaBoost algorithm

AdaBoost ('Adaptive Boosting')

- Input: $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$, where for each i, $y_i = f(\mathbf{x}_i)$

AdaBoost ('Adaptive Boosting')

- Input: $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$, where for each i, $y_i = f(\mathbf{x}_i)$
- Output: hypothesis h with small error on S

AdaBoost ('Adaptive Boosting')

- Input: $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$, where for each i, $y_i = f(\mathbf{x}_i)$
- Output: hypothesis h with small error on S
- We'll later analyze $L_{(\mathcal{D},f)}(h)$ as well

AdaBoost ('Adaptive Boosting')

- Input: $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$, where for each i, $y_i = f(\mathbf{x}_i)$
- Output: hypothesis h with small error on S
- We'll later analyze $L_{(\mathcal{D},f)}(h)$ as well
- AdaBoost calls the weak learner on distributions over S

The AdaBoost Algorithm

- **input:** training set $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$, weak learner WL, number of rounds T

The AdaBoost Algorithm

- **input:** training set $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$, weak learner WL, number of rounds T
- **initialize** $\mathbf{D}^{(1)} = (\frac{1}{m}, \ldots, \frac{1}{m})$

The AdaBoost Algorithm

- **input:** training set $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$, weak learner WL, number of rounds T
- **initialize** $\mathbf{D}^{(1)} = (\frac{1}{m}, \ldots, \frac{1}{m})$
- **for** $t = 1, \ldots, T$:

The AdaBoost Algorithm

- **input:** training set $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$, weak learner WL, number of rounds T
- **initialize** $\mathbf{D}^{(1)} = (\frac{1}{m}, \ldots, \frac{1}{m})$
- **for** $t = 1, \ldots, T$:
  - invoke weak learner $h_t = \mathrm{WL}(\mathbf{D}^{(t)}, S)$

The AdaBoost Algorithm

- **input:** training set $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$, weak learner WL, number of rounds T
- **initialize** $\mathbf{D}^{(1)} = (\frac{1}{m}, \ldots, \frac{1}{m})$
- **for** $t = 1, \ldots, T$:
  - invoke weak learner $h_t = \mathrm{WL}(\mathbf{D}^{(t)}, S)$
  - compute $\epsilon_t = L_{\mathbf{D}^{(t)}}(h_t) = \sum_{i=1}^{m} D_i^{(t)} \mathbb{1}_{[y_i \neq h_t(\mathbf{x}_i)]}$

The AdaBoost Algorithm

- **input:** training set $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$, weak learner WL, number of rounds T
- **initialize** $\mathbf{D}^{(1)} = (\frac{1}{m}, \ldots, \frac{1}{m})$
- **for** $t = 1, \ldots, T$:
  - invoke weak learner $h_t = \mathrm{WL}(\mathbf{D}^{(t)}, S)$
  - compute $\epsilon_t = L_{\mathbf{D}^{(t)}}(h_t) = \sum_{i=1}^m D_i^{(t)} \mathbb{1}_{[y_i \neq h_t(\mathbf{x}_i)]}$
  - let $w_t = \frac{1}{2} \log \left(\frac{1}{\epsilon_t} - 1 \right)$

The AdaBoost Algorithm

- **input:** training set $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$, weak learner WL, number of rounds T
- **initialize** $\mathbf{D}^{(1)} = (\frac{1}{m}, \ldots, \frac{1}{m})$
- **for** $t = 1, \ldots, T$:
  - invoke weak learner $h_t = \mathrm{WL}(\mathbf{D}^{(t)}, S)$
  - compute $\epsilon_t = L_{\mathbf{D}^{(t)}}(h_t) = \sum_{i=1}^{m} D_i^{(t)} \mathbb{1}_{[y_i \neq h_t(\mathbf{x}_i)]}$
  - let $w_t = \frac{1}{2} \log\left(\frac{1}{\epsilon_t} - 1\right)$
  - update $D_i^{(t+1)} = \frac{D_i^{(t)} \exp(-w_t y_i h_t(\mathbf{x}_i))}{\sum_{j=1}^{m} D_j^{(t)} \exp(-w_t y_j h_t(\mathbf{x}_j))}$ for all $i = 1, \ldots, m$

The AdaBoost Algorithm

- **input:** training set $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$, weak learner WL, number of rounds T
- **initialize** $\mathbf{D}^{(1)} = (\frac{1}{m}, \ldots, \frac{1}{m})$
- **for** $t = 1, \ldots, T$:
  - invoke weak learner $h_t = \mathrm{WL}(\mathbf{D}^{(t)}, S)$
  - compute $\epsilon_t = L_{\mathbf{D}^{(t)}}(h_t) = \sum_{i=1}^m D_i^{(t)} \, \mathbb{1}_{[y_i \neq h_t(\mathbf{x}_i)]}$
  - let $w_t = \frac{1}{2} \log \left(\frac{1}{\epsilon_t} - 1 \right)$
  - update $D_i^{(t+1)} = \frac{D_i^{(t)} \exp(-w_t y_i h_t(\mathbf{x}_i))}{\sum_{j=1}^m D_j^{(t)} \exp(-w_t y_j h_t(\mathbf{x}_j))}$ for all $i = 1, \ldots, m$
- **output** the hypothesis $h_s(\mathbf{x}) = \mathrm{sign} \left(\sum_{t=1}^T w_t h_t(\mathbf{x}) \right)$.

Intuition: AdaBoost forces WL to focus on problematic examples

- Claim: The error of h_t w.r.t. $\mathbf{D}^{(t+1)}$ is exactly $1/2$
- Proof:

$$\sum_{i=1}^{m} D_i^{(t+1)} \mathbb{1}_{[y_i \neq h_t(\mathbf{x}_i)]} = \frac{\sum_{i=1}^{m} D_i^{(t)} e^{-w_t y_i h_t(\mathbf{x}_i)} \mathbb{1}_{[y_i \neq h_t(\mathbf{x}_i)]}}{\sum_{j=1}^{m} D_j^{(t)} e^{-w_t y_j h_t(\mathbf{x}_j)}}$$

Intuition: AdaBoost forces WL to focus on problematic examples

- Claim: The error of h_t w.r.t. $\mathbf{D}^{(t+1)}$ is exactly $1/2$
- Proof:

$$\sum_{i=1}^{m} D_i^{(t+1)} \mathbb{1}_{[y_i \neq h_t(\mathbf{x}_i)]} = \frac{\sum_{i=1}^{m} D_i^{(t)} e^{-w_t y_i h_t(\mathbf{x}_i)} \mathbb{1}_{[y_i \neq h_t(\mathbf{x}_i)]}}{\sum_{j=1}^{m} D_j^{(t)} e^{-w_t y_j h_t(\mathbf{x}_j)}}$$

$$= \frac{e^{w_t} \epsilon_t}{e^{w_t} \epsilon_t + e^{-w_t}(1 - \epsilon_t)} = \frac{\epsilon_t}{\epsilon_t + e^{-2w_t}(1 - \epsilon_t)}$$

Intuition: AdaBoost forces WL to focus on problematic examples

- Claim: The error of h_t w.r.t. $\mathbf{D}^{(t+1)}$ is exactly $1/2$
- Proof:

$$\sum_{i=1}^{m} D_i^{(t+1)} \mathbb{1}_{[y_i \neq h_t(\mathbf{x}_i)]} = \frac{\sum_{i=1}^{m} D_i^{(t)} e^{-w_t y_i h_t(\mathbf{x}_i)} \mathbb{1}_{[y_i \neq h_t(\mathbf{x}_i)]}}{\sum_{j=1}^{m} D_j^{(t)} e^{-w_t y_j h_t(\mathbf{x}_j)}}$$

$$= \frac{e^{w_t} \epsilon_t}{e^{w_t} \epsilon_t + e^{-w_t}(1 - \epsilon_t)} = \frac{\epsilon_t}{\epsilon_t + e^{-2w_t}(1 - \epsilon_t)}$$

$$= \frac{\epsilon_t}{\epsilon_t + \frac{\epsilon_t}{1-\epsilon_t}(1 - \epsilon_t)} = \frac{1}{2} .$$

Theorem

If WL is $(1/2-\gamma,\delta)$ weak learner then, with probability at least $1-\delta T$,

$$L_S(h_s) \;\leq\; \exp(-2\,\gamma^2\,T) \quad .$$

Theorem

If WL is $(1/2 - \gamma, \delta)$ weak learner then, with probability at least $1 - \delta T$,

$$L_S(h_s) \;\leq\; \exp(-2\,\gamma^2\,T) \quad .$$

Remarks:

- For any $\epsilon > 0$ and $\gamma \in (0, 1/2)$, if $T \geq \frac{\log(1/\epsilon)}{2\gamma^2}$, then AdaBoost will output a hypothesis h_s with $L_S(h_s) \leq \epsilon$.

Theorem

If WL is $(1/2-\gamma,\delta)$ *weak learner then, with probability at least* $1-\delta T$*,*

$$L_S(h_s) \;\le\; \exp(-2\,\gamma^2\,T) \quad .$$

Remarks:

- For any $\epsilon > 0$ and $\gamma \in (0,1/2)$, if $T \ge \frac{\log(1/\epsilon)}{2\gamma^2}$, then AdaBoost will output a hypothesis h_s with $L_S(h_s) \le \epsilon$.
- Setting $\epsilon = 1/(2m)$ the hypothesis h_s must have a zero training error

Theorem

If WL is $(1/2 - \gamma, \delta)$ *weak learner then, with probability at least* $1 - \delta T$,

$$L_S(h_s) \;\le\; \exp(-2\,\gamma^2\,T) \quad .$$

Remarks:

- For any $\epsilon > 0$ and $\gamma \in (0, 1/2)$, if $T \ge \frac{\log(1/\epsilon)}{2\gamma^2}$, then AdaBoost will output a hypothesis h_s with $L_S(h_s) \le \epsilon$.
- Setting $\epsilon = 1/(2m)$ the hypothesis h_s must have a zero training error
- Since the weak learner is invoked on a distribution over S, in many cases δ can be 0. In any case, by "boosting the confidence", we can assume w.l.o.g. that δ is very small.

Outline

1. Weak learnability
2. Boosting the confidence
3. Boosting the accuracy using AdaBoost
4. AdaBoost as a learner for Halfspaces++
5. AdaBoost and the Bias-Complexity Tradeoff
6. Weak Learnability and Separability with Margin
7. AdaBoost for Face Detection

AdaBoost as a Learner for Halfspaces++

- Let B be the set of all hypotheses the WL may return

AdaBoost as a Learner for Halfspaces++

- Let B be the set of all hypotheses the WL may return
- Observe that AdaBoost outputs a hypothesis from the class

$$L(B,T) = \left\{ x \mapsto \text{sign}\left(\sum_{t=1}^{T} w_t h_t(x) \right) : \mathbf{w} \in \mathbb{R}^T,\ \forall t, \quad h_t \in B \right\} .$$

AdaBoost as a Learner for Halfspaces++

- Let B be the set of all hypotheses the WL may return
- Observe that AdaBoost outputs a hypothesis from the class

$$L(B,T) = \left\{ x \mapsto \text{sign}\left(\sum_{t=1}^{T} w_t h_t(x) \right) : \mathbf{w} \in \mathbb{R}^T,\ \forall t, \quad h_t \in B \right\} .$$

- Since WL is invoked only on distributions over S we can assume w.l.o.g. that $B = \{g_1, \ldots, g_d\}$ for some $d \leq 2^m$.

AdaBoost as a Learner for Halfspaces++

- Let B be the set of all hypotheses the WL may return
- Observe that AdaBoost outputs a hypothesis from the class

$$L(B,T) = \left\{ x \mapsto \text{sign}\left(\sum_{t=1}^{T} w_t h_t(x) \right) : \mathbf{w} \in \mathbb{R}^T,\ \forall t, \quad h_t \in B \right\} .$$

- Since WL is invoked only on distributions over S we can assume w.l.o.g. that $B = \{g_1, \ldots, g_d\}$ for some $d \leq 2^m$.
- Denote $\psi(x) = (g_1(x), \ldots, g_d(x))$. Therefore:

$$L(B,T) = \left\{ x \mapsto \text{sign}\left(\langle w, \psi(x) \rangle \right) : \mathbf{w} \in \mathbb{R}^d,\ \|\mathbf{w}\|_0 \leq T \right\} ,$$

where $\|\mathbf{w}\|_0 = |\{i : w_i \neq 0\}|$.

AdaBoost as a Learner for Halfspaces++

- Let B be the set of all hypotheses the WL may return
- Observe that AdaBoost outputs a hypothesis from the class

$$L(B,T) = \left\{ x \mapsto \operatorname{sign}\left(\sum_{t=1}^{T} w_t h_t(x) \right) : \mathbf{w} \in \mathbb{R}^T, \ \forall t, \quad h_t \in B \right\} .$$

- Since WL is invoked only on distributions over S we can assume w.l.o.g. that $B = \{g_1, \ldots, g_d\}$ for some $d \le 2^m$.
- Denote $\psi(x) = (g_1(x), \ldots, g_d(x))$. Therefore:

$$L(B,T) = \left\{ x \mapsto \operatorname{sign}\left(\langle w, \psi(x) \rangle \right) : \mathbf{w} \in \mathbb{R}^d, \ \|\mathbf{w}\|_0 \le T \right\} ,$$

where $\|\mathbf{w}\|_0 = |\{i : w_i \neq 0\}|$.

- That is, AdaBoost learns a composition of the class of halfspaces with sparse coefficients over the mapping $x \mapsto \psi(x)$

Expressiveness of $L(B,T)$

- Suppose $\mathcal{X} = \mathbb{R}$ and B is Decision Stumps,

$$B = \{x \mapsto \text{sign}(x - \theta) \cdot b : \quad \theta \in \mathbb{R}, b \in \{\pm 1\}\} \ .$$

Expressiveness of $L(B,T)$

- Suppose $\mathcal{X} = \mathbb{R}$ and B is Decision Stumps,

$$B = \{x \mapsto \text{sign}(x - \theta) \cdot b : \quad \theta \in \mathbb{R}, b \in \{\pm 1\}\} \ .$$

- Let $\mathcal{G}_T$ be the class of piece-wise constant functions with T pieces

Expressiveness of $L(B,T)$

- Suppose $\mathcal{X} = \mathbb{R}$ and B is Decision Stumps,

$$B = \{x \mapsto \text{sign}(x-\theta)\cdot b : \quad \theta \in \mathbb{R}, b \in \{\pm 1\}\} \ .$$

- Let $\mathcal{G}_T$ be the class of piece-wise constant functions with T pieces
- Claim: $\mathcal{G}_T \subseteq L(B,T)$

Expressiveness of $L(B,T)$

- Suppose $\mathcal{X} = \mathbb{R}$ and B is Decision Stumps,

$$B = \{x \mapsto \mathrm{sign}(x - \theta) \cdot b : \quad \theta \in \mathbb{R}, b \in \{\pm 1\}\} \ .$$

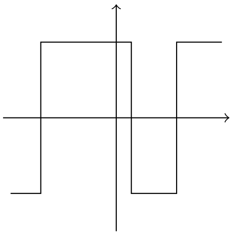
- Let $\mathcal{G}_T$ be the class of piece-wise constant functions with T pieces
- Claim: $\mathcal{G}_T \subseteq L(B,T)$

Expressiveness of $L(B,T)$

- Suppose $\mathcal{X} = \mathbb{R}$ and B is Decision Stumps,

$$B = \{x \mapsto \mathrm{sign}(x - \theta) \cdot b : \quad \theta \in \mathbb{R}, b \in \{\pm 1\}\} \ .$$

- Let $\mathcal{G}_T$ be the class of piece-wise constant functions with T pieces
- Claim: $\mathcal{G}_T \subseteq L(B,T)$



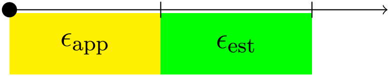
Composing halfspaces on top of simple classes can be very expressive !

Outline

1. Weak learnability
2. Boosting the confidence
3. Boosting the accuracy using AdaBoost
4. AdaBoost as a learner for Halfspaces++
5. AdaBoost and the Bias-Complexity Tradeoff
6. Weak Learnability and Separability with Margin
7. AdaBoost for Face Detection

Bias-complexity

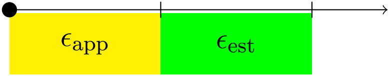
Recall:



- We have argued that the expressiveness of $L(B, T)$ grows with T

Bias-complexity

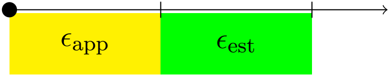
Recall:



- We have argued that the expressiveness of $L(B, T)$ grows with T
- In other words, the approximation error decreases with T

Bias-complexity

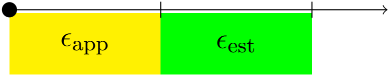
Recall:



- We have argued that the expressiveness of $L(B,T)$ grows with T
- In other words, the approximation error decreases with T
- We'll show that the estimation error increases with T

Bias-complexity

Recall:



- We have argued that the expressiveness of $L(B, T)$ grows with T
- In other words, the approximation error decreases with T
- We'll show that the estimation error increases with T
- Therefore, the parameter T of AdaBoost enables us to control the bias-complexity tradeoff

The Estimation Error of $L(B,T)$

- Claim:

$$\mathrm{VCdim}(L(B,T)) \;\leq\; \tilde{O}(T \cdot \mathrm{VCdim}(B))$$

The Estimation Error of $L(B, T)$

- Claim:

$$\text{VCdim}(L(B,T)) \ \leq \ \tilde{O}(T \cdot \text{VCdim}(B))$$

- Corollary: if $m \geq \tilde{\Omega}\left(\frac{\log(1/\delta)}{\gamma^2 \epsilon}\right)$ and $T = \log(m)/(2\gamma^2)$, then w.p. of at least $1 - \delta$,

$$L_{(\mathcal{D},f)}(h_s) \leq \epsilon \ .$$

Outline

1. Weak learnability
2. Boosting the confidence
3. Boosting the accuracy using AdaBoost
4. AdaBoost as a learner for Halfspaces++
5. AdaBoost and the Bias-Complexity Tradeoff
6. **Weak Learnability and Separability with Margin**
7. AdaBoost for Face Detection

Weak Learnability and Separability with Margin

- We have essentially shown: if $\mathcal{H}$ is weak learnable, then $\mathcal{H} \subseteq L(B, \infty)$

Weak Learnability and Separability with Margin

- We have essentially shown: if $\mathcal{H}$ is weak learnable, then $\mathcal{H} \subseteq L(B, \infty)$
- What about the other direction ?

Weak Learnability and Separability with Margin

- We have essentially shown: if $\mathcal{H}$ is weak learnable, then $\mathcal{H} \subseteq L(B, \infty)$
- What about the other direction ?
- Using von Neumanns minimax theorem, it can be shown that if $L(B, \infty)$ separates a training set with ℓ_1 margin γ then ERM_B is a γ weak learner for $\mathcal{H}$

Weak Learnability and Separability with Margin

- We have essentially shown: if $\mathcal{H}$ is weak learnable, then $\mathcal{H} \subseteq L(B, \infty)$
- What about the other direction ?
- Using von Neumanns minimax theorem, it can be shown that if $L(B, \infty)$ separates a training set with ℓ_1 margin γ then ERM_B is a γ weak learner for $\mathcal{H}$
- This is beyond the scope of the course

Outline

1. Weak learnability
2. Boosting the confidence
3. Boosting the accuracy using AdaBoost
4. AdaBoost as a learner for Halfspaces++
5. AdaBoost and the Bias-Complexity Tradeoff
6. Weak Learnability and Separability with Margin
7. AdaBoost for Face Detection

Face Detection

- Classify rectangles in an image as face or non-face

Weak Learner for Face Detection

Rules of thumb:

- "eye region is often darker than the cheeks"
- "bridge of the noise is brighter than the eyes"

Weak Learner for Face Detection

Rules of thumb:

- "eye region is often darker than the cheeks"
- "bridge of the noise is brighter than the eyes"

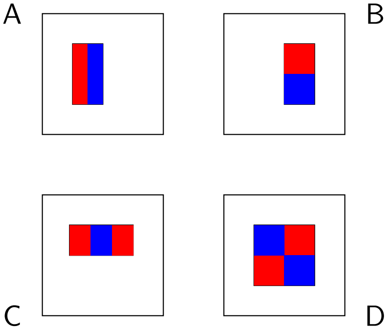
Goal:

- We want to combine few rules of thumb to obtain a face detector
- "Sparsity" reflects both small estimation error but also speed !

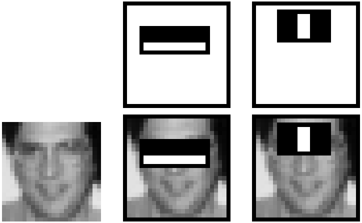
Weak Learner for Face Detection

Each hypothesis in the base class is of the form $h(x) = f(g(x))$, where f is a decision stump and $g : \mathbb{R}^{24,24} \to \mathbb{R}$ is parameterized by:

- An axis-aligned rectangle R. Since each image is of size 24×24, there are at most 24^4 axis-aligned rectangles.
- A type, $t \in \{A, B, C, D\}$. Each type corresponds to a mask:



The first and second features selected by AdaBoost, as implemented by Viola and Jones.



Summary

- Boosting the confidence using validation
- Boosting the accuracy using AdaBoost
- The power of composing halfspaces over simple classes
- The bias-complexity tradeoff
- AdaBoost works in many practical problems !