EE6550 Machine Learning

Lecture Five – Boosting

Chung-Chin Lu

Department of Electrical Engineering

National Tsing Hua University

April 10, 2017

The Contents of This Lecture

- Introduction.
- AdaBoost.
- Theoretical results.
- Discussion.

Motivations

- It is often difficult, for a non-trivial learning task, to directly devise an accurate algorithm satisfying the strong PAC-learning requirements of Lecture 1 - Part 2.
- It is highly probable to find simple predictors guaranteed only to perform slightly better than random.
- Ensemble methods are general techniques in machine learning for combining several weak predictors to build a more accurate one.
- This lecture studies an important family of ensemble methods known as boosting, and more specifically the AdaBoost algorithm.

Weak Learning

Definition 6.1: A concept class C is said to be weakly PAC-learnable if there exist an algorithm $\mathbb{A}$, a constant $\gamma > 0$, and a polynomial function $\text{poly}(\cdot, \cdot, \cdot)$ such that for any $\delta > 0$, for all distributions P on the input space $\mathscr{I}$ and for any target concept $c \in C$, the following holds for any random sample S of size $m \geq \text{poly}(1/\delta, n, \text{size}(c))$:

$$P_m\left(R(h_S) \leq \frac{1}{2} - \gamma\right) \geq 1 - \delta,$$

where

- $h_S = \mathbb{A}(S; c, \mathcal{H})$ is the learned hypothesis by the algorithm $\mathbb{A}$ with a hypothesis set $\mathcal{H}$.
- $O(n)$: the cost of computational representation of an item ω.
- $O(\text{size}(c))$: cost of computational representation of a concept c.

Remarks

- When such an algorithm $\mathbb{A}$ exists, it is called a weak PAC-learning algorithm for $\mathcal{C}$ or a weak learner.
- The hypotheses set $\mathcal{H}$ associated with a weak learning algorithm is called a family of base classifiers.

Boosting Techniques

- Key Idea : to combine different base classifiers returned by a weak learner to create a more accurate predictor.
- Questions :
  - Which base classifiers should be used ?
  - How should they be combined?

The Contents of This Lecture

- Introduction.
- AdaBoost.
- Theoretical results.
- Discussion.

The AdaBoost Algorithm for $\mathcal{H} \subset \{-1, +1\}^{\mathscr{I}}$

AdaBoost$(S = ((\omega_1, c(\omega_1)), \ldots, (\omega_m, c(\omega_m))))$

1. **for** $i \leftarrow 1$ **to** m **do**
2. $\quad D_1(i) \leftarrow \frac{1}{m}$
3. **for** $t \leftarrow 1$ **to** T **do**
4. $\quad h_{S,t} \leftarrow$ base classifier in $\mathcal{H}$ with small error

$$\epsilon_t = \Pr_{i \sim D_t}(h_{S,t}(\omega_i) \neq c(\omega_i)) = \sum_{i=1}^{m} D_t(i) 1_{h_{S,t}(\omega_i) \neq c(\omega_i)}$$

5. $\quad \alpha_t \leftarrow \frac{1}{2} \log \frac{1-\epsilon_t}{\epsilon_t}$
6. $\quad Z_t \leftarrow 2[\epsilon_t(1-\epsilon_t)]^{1/2}$ $\quad \triangleright$ normalization factor

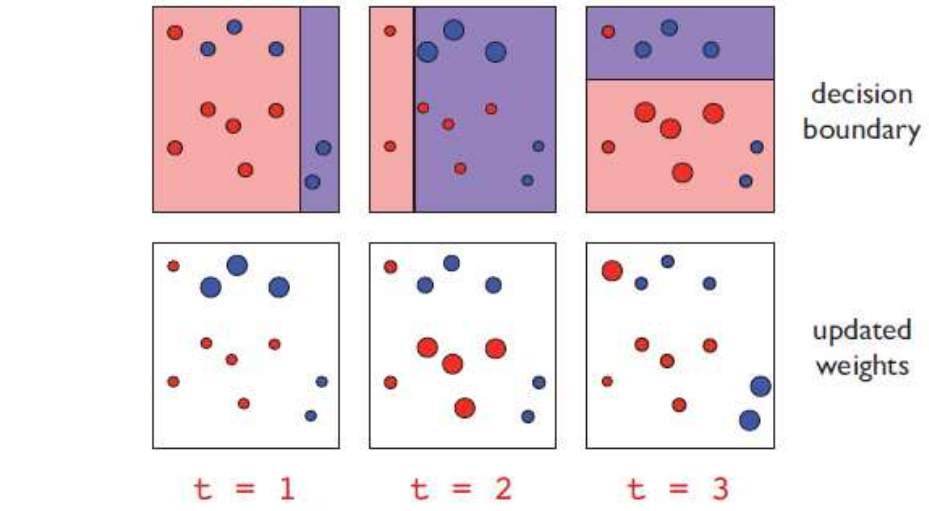
7. **for** $i \leftarrow 1$ **to** m **do**

8. $D_{t+1}(i) \leftarrow \frac{D_t(i)\exp(-\alpha_t h_{S,t}(\omega_i)c(\omega_i))}{Z_t}$

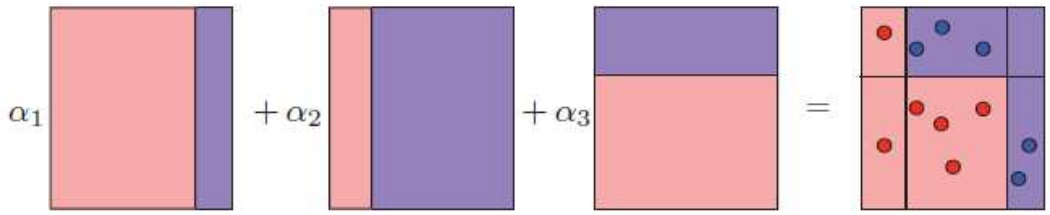
9. $g_S \leftarrow \sum_{t=1}^{T} \alpha_t h_{S,t}$

10. **return** $h_S = \mathrm{sgn}(g_S)$

AdaBoost with Axis-Aligned Hyperplanes as Base Classifiers (from Figure 6.2 of the *Foundation* Textbook)



(a)



(b)

The AdaBoost Algorithm for $\mathcal{H} \subset \{-1, +1\}^{\mathscr{I}}$

- Input : a labeled sample $S = ((\omega_1, c(\omega_1)), \ldots, (\omega_m, c(\omega_m)))$ of size m;
- Output : the sign function $h_S = \mathrm{sgn}(g_S)$ of a linear combination $g_S = \sum_{t=1}^{T} \alpha_t h_{S,t}$ of base classifiers $h_{S,t}$ in the hypothesis set $\mathcal{H}$ obtained from T rounds of boosting.
- AdaBoost maintains a distribution on the index set $[1, m]$ which will be updated at each round of boosting. The initial distribution D_1 is set to be the uniform distribution, i.e., $D_1(i) = 1/m$ for all $i \in [1, m]$. Let D_t be the distribution on $[1, m]$ at the t-th round of boosting.
- At the t-th round of boosting, the base classifier $h_{S,t}$ is selected that minimizes the error on the training sample weighted by

the distribution D_t:

$$h_{S,t} \in \arg\min_{h \in \mathcal{H}} \Pr_{i \sim D_t}(h(\omega_i) \neq c(\omega_i)) = \arg\min_{h \in \mathcal{H}} \sum_{i=1}^{m} D_t(i) 1_{h(\omega_i) \neq c(\omega_i)}.$$

  - Instead of a hypothesis with minimal weighted error, $h_{S,t}$ can be more generally the base classifier returned by a weak learning algorithm trained on the distribution D_t;

- Let ϵ_t be the error of the base classifier $h_{S,t}$ weighted by the distribution D_t:

$$\epsilon_t = \Pr_{i \sim D_t}(h_{S,t}(\omega_i) \neq c(\omega_i)) = \sum_{i=1}^{m} D_t(i) 1_{h_{S,t}(\omega_i) \neq c(\omega_i)},$$

  which is better than random. In particular, $\epsilon_t < 1/2$.

- Now AdaBoost updates the distribution D_i by substantially increasing the weight on i if item ω_i is incorrectly classified (i.e., $c(\omega_i) h_{S,t}(\omega_i) \leq 0$), and, on the contrary, decreasing it if ω_i

is correctly classified:

$$D_{t+1}(i) = \frac{D_t(i)e^{-\alpha c(\omega_i)h_{S,t}(\omega_i)}}{Z_t},$$

where Z_t is the normalization factor and $\alpha > 0$ is to be determined.

- This has the effect of focusing more on the items incorrectly classified at the next round of boosting, less on those correctly classified by $h_{S,t}$.

- The normalization factor Z_t is

$$\begin{aligned} Z_t &= \sum_{i=1}^{m} D_t(i)e^{-\alpha c(\omega_i)h_{S,t}(\omega_i)} \\ &= \sum_{i:c(\omega_i)h_{S,t}(\omega_i)=+1} D_t(i)e^{-\alpha} + \sum_{i:c(\omega_i)h_{S,t}(\omega_i)=-1} D_t(i)e^{\alpha} \\ &= (1-\epsilon_t)e^{-\alpha} + \epsilon_t e^{\alpha}. \end{aligned}$$

– We will want Z_t to be minimized so that a good upper bound of the empirical error of the AdaBoost classifier h_S can be obtained later.

- Z_t is minimized if and only if

$$(1-\epsilon_t)e^{-\alpha} = \epsilon_t e^{\alpha}, \tag{1}$$

whose solution is denoted as

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right). \tag{2}$$

And the minimum value is assigned to

$$Z_t = 2\sqrt{(1-\epsilon_t)\epsilon_t}. \tag{3}$$

Also the update rule of the distribution on the index set $[1, m]$ becomes

$$D_{t+1}(i) = \frac{D_t(i)e^{-\alpha_t c(\omega_i)h_{S,t}(\omega_i)}}{Z_t}, \tag{4}$$

- When the base classifier $h_{S,t}$ is more accurate, i.e., ϵ_t is smaller, the parameter α_t is bigger.
- After T rounds of boosting, the classifier returned by AdaBoost is based on the sign of a linear combination

$$g_S = \sum_{t=1}^{T} \alpha_t h_{S,t}$$

of the base classifiers $h_{S,t}$ in a way that more accurate base classifiers are assigned a larger weight in that sum.

A Usefull Identity

- $g_{S,t} \triangleq \sum_{j=1}^{t} \alpha_j h_{S,j}$, $t \in [1,T]$.
  - $g_S = g_{S,T}$.

$$\forall\ i \in [1,m],\quad D_{t+1}(i) = \frac{e^{-c(\omega_i) g_{S,t}(\omega_i)}}{m \prod_{j=1}^{t} Z_j}.$$

Proof. By repeatedly using (4), we have

$$\begin{aligned} D_{t+1}(i) &= \frac{D_t(i) e^{-\alpha_t c(\omega_i) h_{S,t}(\omega_i)}}{Z_t} \\ &= \frac{D_{t-1}(i) e^{-\alpha_{t-1} c(\omega_i) h_{S,t-1}(\omega_i)} e^{-\alpha_t c(\omega_i) h_{S,t}(\omega_i)}}{Z_{t-1} Z_t} \\ &= \frac{e^{-c(\omega_i) \sum_{j=1}^{t} \alpha_j h_{S,j}(\omega_i)}}{m \prod_{j=1}^{t} Z_j}. \end{aligned}$$

□

Empirical Error Bound for AdaBoost Classifiers

Theorem 6.1: Let

- $S = ((\omega_1, c(\omega_1)), \ldots, (\omega_m, c(\omega_m)))$: a labeled sample of size m;
- $h_S = \mathrm{sgn}(g_S)$: the binary classifier returned by AdaBoost, where $g_S = \sum_{t=1}^{T} \alpha_t h_{S,t}$ with $h_{S,t} \in \mathcal{H}$, $t \in [1, T]$ the base classifiers obtained from T boosting rounds.

The empirical error of the AdaBoost classifier h_S w.r.t. the concept c on the labeled sample S satisfies:

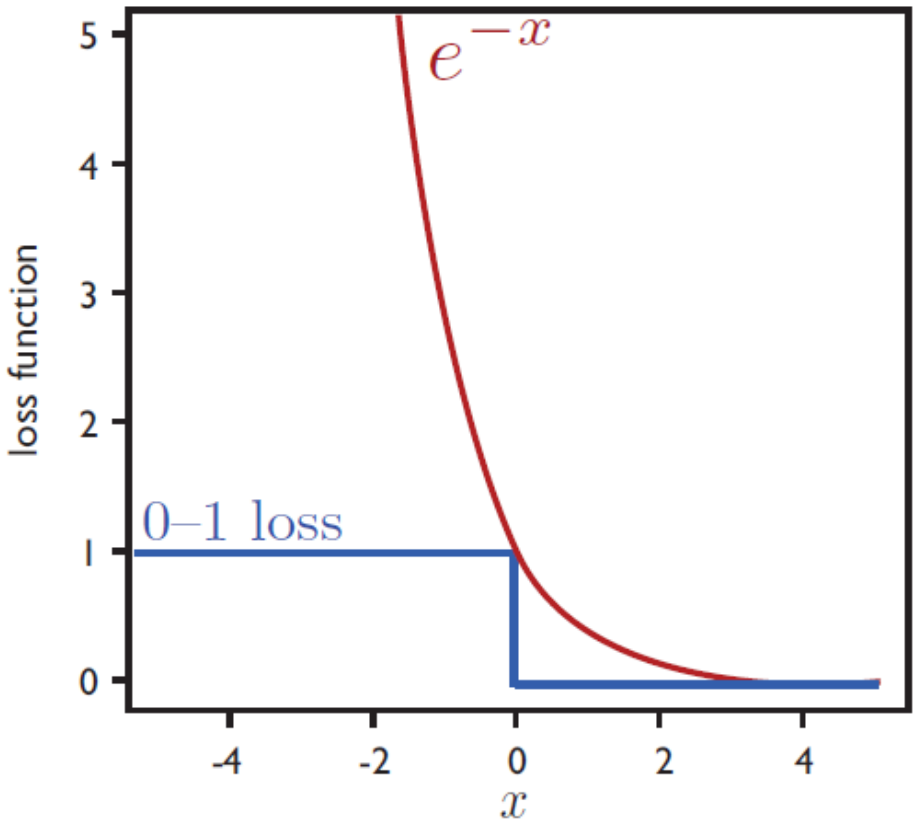
$$\hat{R}_S(h_S) \leq \exp\left\{-2\sum_{t=1}^{T}\left(\frac{1}{2} - \epsilon_t\right)^2\right\}.$$

Furthermore, if $\epsilon_t \leq 1/2 - \gamma$ for all $t \in [1, T]$, then

$$\hat{R}_S(h_S) \leq \exp\left\{-2\gamma^2 T\right\}.$$

Proof. Note that

- $1_{u \leq 0} \leq e^{-u}$ for all $u \in \mathbb{R}$.



- $D_{T+1}(i) = \frac{\exp\{-c(\omega_i) g_S(\omega_i)\}}{m \prod_{t=1}^{T} Z_t}$ for all $i \in [1, m]$.

Now the empirical error of the AdaBoost classifier h_S w.r.t. the concept c on the labeled sample S is

$$\begin{aligned}
\hat{R}_S(h_S) &= \frac{1}{m}\sum_{i=1}^{m} 1_{c(\omega_i)g_S(\omega_i)\leq 0} \\
&\leq \frac{1}{m}\sum_{i=1}^{m} e^{-c(\omega_i)g_S(\omega_i)} \\
&= \frac{1}{m}\sum_{i=1}^{m} \left(m\prod_{t=1}^{T} Z_t\right) D_{T+1}(i) \\
&= \prod_{t=1}^{T} Z_t.
\end{aligned}$$

Since $Z_t = 2\sqrt{\epsilon_t(1-\epsilon_t)}$, we have

$$\begin{aligned}
\prod_{t=1}^{T} Z_t &= \prod_{t=1}^{T} 2\sqrt{\epsilon_t(1-\epsilon_t)} = \prod_{t=1}^{T} \sqrt{1-4\left(\frac{1}{2}-\epsilon_t\right)^2} \\
&\leq \prod_{t=1}^{T} \sqrt{\exp\left\{-4\left(\frac{1}{2}-\epsilon_t\right)^2\right\}} \text{ since } 1-x \leq e^{-x} \ \forall \ x \in \mathbb{R} \\
&= \exp\left\{-2\sum_{t=1}^{T}\left(\frac{1}{2}-\epsilon_t\right)^2\right\}
\end{aligned}$$

and then

$$\hat{R}_S(h_S) \leq \exp\left\{-2\sum_{t=1}^{T}\left(\frac{1}{2}-\epsilon_t\right)^2\right\}.$$

□

Remarks

- Note that the value of γ, which is known as the edge, and the accuracy ϵ_t of the base classifiers $h_{S,t}$ do not need to be known to the algorithm. The algorithm adapts to their accuracy and defines a solution based on these values. This is the source of the extended name of AdaBoost: adaptive boosting.
- Since the empirical error $\hat{R}_S(h_S)$ of the classifier $h_S = \text{sgn}(g_S)$ returned by the AdaBoost is upper bounded by the product $\prod_{t=1}^{T} Z_t$, the chosen α_t in (2) which minimizes Z_t will also minimize this upper bound.
- In fact, for base classifiers whose range is $[-1, +1]$ or $\mathbb{R}$, α_t can be chosen in a similar fashion to minimize Z_t, and this is the way AdaBoost is extended to these more general cases.

- Observe also that the balanced equation of weight

$$(1 - \epsilon_t)e^{-\alpha_t} = \epsilon_t e^{\alpha_t}$$

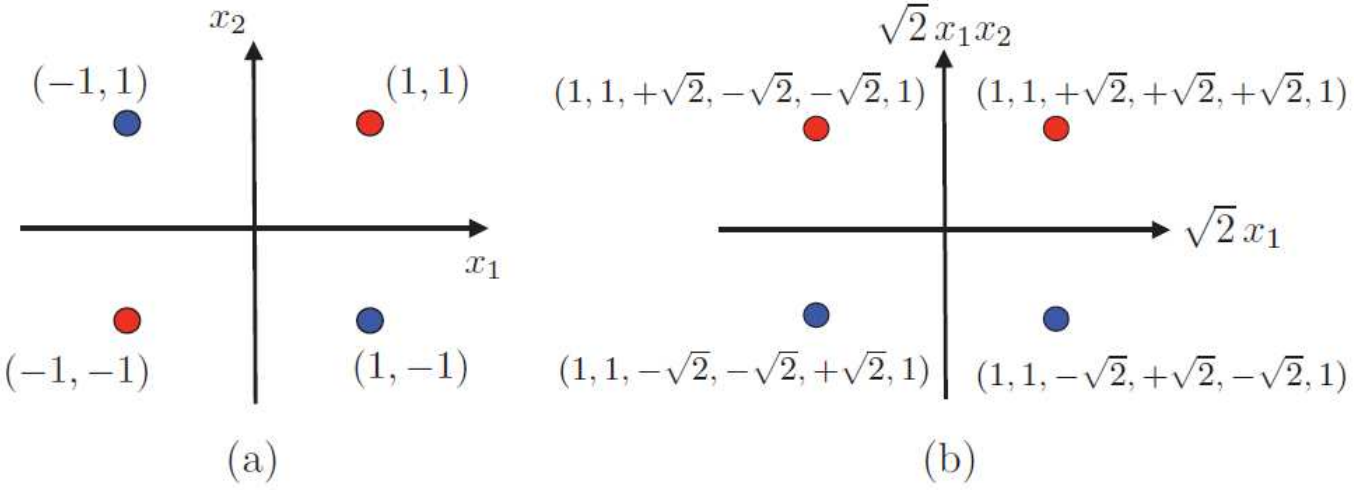
  in (1) implies that at each iteration, AdaBoost assigns equal distribution mass to correctly and incorrectly classified instances.

- This may seem to contradict the fact that AdaBoost increases the weights of incorrectly classified points and decreases those of others, but there is in fact no inconsistency: the reason is that there are always fewer incorrectly classified points, since the base classifier's accuracy is better than random.

Standard Use of AdaBoost in Practice

- Decision trees of depth one, also known as stumps or boosting stumps are by far the most frequently used base classifiers for AdaBoost.
- Boosting stumps are threshold functions associated to a single feature. Thus, a stump corresponds to a single axis-aligned partition of the space, as illustrated in Figure 6.2 of the *Foundation* textbook.
- If the data is in $\mathbb{R}^N$, we can associate a stump to each of the N components. Thus, to determine the stump with the minimal weighted error at each of round of boosting, the best component and the best threshold for each component must be computed.

- To do so, we can first presort each component in $O(m \log m)$ time with a total computational cost of $O(mN \log m)$.
- For a given component, there are only $m + 1$ possible distinct thresholds, since two thresholds between the same consecutive component values are equivalent. To find the best threshold at each round of boosting, all of these possible $m + 1$ values can be compared, which can be done in $O(m)$ time.
- Thus, the total computational complexity of the algorithm for T rounds of boosting is $O(mN \log m + mNT)$.
- The learning algorithm in the Adaboost algorithm which returns the stump with the minimal weighted empirical error may not be a weak learner in general. A counterexample is shown in Figure 5.2a of the *Foundation* textbook, where, for the concept c of XOR, no decision stump can achieve an accuracy better than 1/2:



- To develop a weak learner for a concept class, the family $\mathcal{H}$ of base classifiers may be chosen to be other than the family of stumps such as a family of decision trees up to a certain depth or a family of neural networks. But such a family should not be too sophisticated as implied by the generalization bounds derived in the next section.

The Contents of This Lecture

- Introduction.
- AdaBoost.
- Theoretical results.
- Discussion.

VC-Dimension of AdaBoost Hypothesis Set

- $\mathcal{H} \subseteq \{-1, +1\}^{\mathscr{I}}$: a family of base classifiers.
- d : the VC-dimension of $\mathcal{H}$.
- $\mathcal{H}_T$: the AdaBoost hypothesis set with T rounds of boosting from $\mathcal{H}$,

$$\mathcal{H}_T \triangleq \left\{ \operatorname{sgn}\left(\sum_{t=1}^{T} \alpha_t h_t \right) \mid \alpha_t \in \mathbb{R}, h_t \in \mathcal{H}, t \in [1, T] \right\}.$$

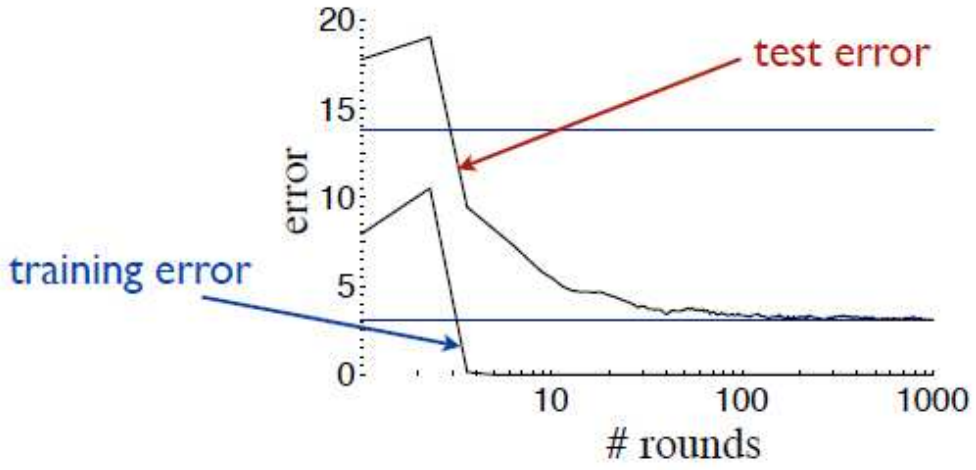
Then we have

$$\operatorname{VCdim}(\mathcal{H}_T) \leq 2(d+1)(T+1)\log_2((T+1)e)^{\text{a}}.$$

[a] Y. Freund and R.E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Science* **55**, 1997, pp. 119–139, Theorem 8.

Remarks

- The upper bound of $\text{VCdim}(\mathcal{H}_T)$ grows as $O(dT \log T)$ which suggests that AdaBoost could overfit for large values of T, and indeed this can occur.
- However, in many practical cases, it has been observed empirically that the generalization error of AdaBoost decreases as a function of the number of rounds of boosting T.



C4.5 decision trees (Schapire et al., 1998).

How can these empirical results be explained?

- We will explain this with a margin-based analysis.
- Note that AdaBoost classifiers are linear combinations of base classifiers with non-negative coefficients.

The Convex Hull co($\mathcal{H}$) of a Hypothesis Set $\mathcal{H}$

- $\mathcal{H}$: a hypothesis set of real-valued measurable functions over the input space $\mathscr{I}$.

The convex hull of $\mathcal{H}$ is defined as

$$\mathrm{co}(\mathcal{H}) \triangleq \left\{ \sum_{k=1}^{p} \mu_k h_k \mid p \geq 1, \mu_k \geq 0, h_k \in \mathcal{H} \; \forall \; k \in [1,p], \sum_{k=1}^{p} \mu_k = 1 \right\}.$$

The Empirical Rademacher Complexity of $\mathbf{co}(\mathcal{H})$

Theorem 6.2 : Let

- $\mathscr{I}$: the input space of all possible items ω, associated with a probability space $(\mathscr{I}, \mathcal{F}, P)$, where P is unknown;
- $\mathscr{Y}' \subseteq \mathbb{R}$: the output space;
- $\mathcal{H}$: a hypothesis set of $\mathscr{Y}'$-valued measurable functions over the input space $\mathscr{I}$;
- $S = (\omega_1, \ldots, \omega_m)$: a sample of m items drawn i.i.d. from $\mathscr{I}$ according to the distribution P;
- $\sup_{g \in \mathrm{co}(\mathcal{H})} \sum_{i=1}^{m} \sigma_i g(\omega_i) < \infty$ for all $\sigma_i \in \{-1, +1\}$, $i \in [1, m]$.
  - This always holds when $\mathscr{Y}'$ is a bounded subset of $\mathbb{R}$.

The empirical Rademacher complexity of the convex hull $\mathrm{co}(\mathcal{H})$ w.r.t. the sample S is equal to that of $\mathcal{H}$ w.r.t. the same sample S,

$$\hat{\mathfrak{R}}_S(\mathrm{co}(\mathcal{H})) = \hat{\mathfrak{R}}_S(\mathcal{H}).$$

Proof.

$$
\begin{aligned}
& \hat{\mathfrak{R}}_S(\mathrm{co}(\mathcal{H})) \\
= \quad & \frac{1}{2^m} \sum_{\sigma_1,\sigma_2,\ldots,\sigma_m \in \{-1,+1\}} \sup_{g \in \mathrm{co}(\mathcal{H})} \frac{1}{m} \sum_{i=1}^{m} \sigma_i g(\omega_i) \\
= \quad & \frac{1}{2^m} \sum_{\sigma_1,\sigma_2,\ldots,\sigma_m \in \{-1,+1\}} \sup_{p \geq 1, \mathbf{h} \in \mathcal{H}^p} \sup_{\mu \geq \mathbf{0}, \|\mu\|_1 = 1} \frac{1}{m} \sum_{i=1}^{m} \sigma_i \sum_{k=1}^{p} \mu_k h_k(\omega_i) \\
= \quad & \frac{1}{2^m} \sum_{\sigma_1,\sigma_2,\ldots,\sigma_m \in \{-1,+1\}} \sup_{p \geq 1, \mathbf{h} \in \mathcal{H}^p} \sup_{\mu \geq \mathbf{0}, \|\mu\|_1 = 1} \sum_{k=1}^{p} \mu_k \frac{1}{m} \sum_{i=1}^{m} \sigma_i h_k(\omega_i) \\
= \quad & \frac{1}{2^m} \sum_{\sigma_1,\sigma_2,\ldots,\sigma_m \in \{-1,+1\}} \sup_{p \geq 1, \mathbf{h} \in \mathcal{H}^p} \max_{k \in [1,p]} \frac{1}{m} \sum_{i=1}^{m} \sigma_i h_k(\omega_i) \\
= \quad & \frac{1}{2^m} \sum_{\sigma_1,\sigma_2,\ldots,\sigma_m \in \{-1,+1\}} \sup_{h \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^{m} \sigma_i h(\omega_i) = \hat{\mathfrak{R}}_S(\mathcal{H}).
\end{aligned}
$$

□

Ensemble Rademacher Margin Bound

Corollary 6.1 : Let

- $\mathscr{I}$: the input space of all possible items ω, associated with a probability space $(\mathscr{I}, \mathcal{F}, P)$, where P is unknown;
- $c : \mathscr{I} \to \{-1, +1\}$: a fixed but unknown target concept in the concept class $\mathcal{C}$;
- $\mathscr{Y}' = [a, b]$: the output space with $-\infty < a < b < \infty$;
- $\mathcal{H}$: a hypothesis set of $\mathscr{Y}'$-valued measurable functions on the input space $\mathscr{I}$;
  - In Adaboost, $\mathcal{H}$ is the family of base classifiers.
- co($\mathcal{H}$): the convex hull of $\mathcal{H}$, which is also a hypothesis set of $\mathscr{Y}'$-valued measurable functions on the input space $\mathscr{I}$;

- $S = (\omega_1, \omega_2, \ldots, \omega_m)$: a sample of m items drawn i.i.d. from $\mathscr{I}$ according to P with labels $(c(\omega_1), c(\omega_2), \ldots, c(\omega_m))$;
- $\rho > 0$: a given margin;
- $L_\rho(y', y) = \Phi_\rho(y'y) : \mathbb{R} \times \mathbb{R} \to [0, 1]$: the ρ-margin loss function;
- $g_h : \mathscr{I} \times \{-1, +1\} \to [0, 1]$: the loss function associated with h in $\mathrm{co}(\mathcal{H})$ under the ρ-margin loss function L_ρ, defined as $g_h(\omega, y) \triangleq L_\rho(h(\omega), y) = \Phi_\rho(h(\omega)y)$.
- $\mathcal{G} = \{g_h \mid h \in \mathrm{co}(\mathcal{H})\}$: the family of loss functions associated with hypotheses in $\mathrm{co}(\mathcal{H})$ under the ρ-margin loss function L_ρ;
- $\mathscr{Z} = \mathscr{I} \times \{-1, +1\}$: the input set of loss functions g_h, associated with a probability space $(\mathscr{Z}, \tilde{\mathcal{F}}, \tilde{P})$ where $\tilde{P}$ is an extension of P from on $\mathcal{F}$ to on $\tilde{\mathcal{F}} = \mathcal{F} \times 2^{\{-1,+1\}}$;
- $\tilde{S} = ((\omega_1, c(\omega_1)), \ldots, (\omega_m, c(\omega_m))$: the labeled sample corresponding to S;

- $\hat{A}_{\tilde{S}}(g_h) = \frac{1}{m}\sum_{i=1}^{m} g_h(\omega_i, c(\omega_i)) = \frac{1}{m}\sum_{i=1}^{m} L_\rho(h(\omega_i), c(\omega_i)) = \hat{R}_{S,\rho}(h)$, the empirical ρ-margin loss of h w.r.t. c on sample S;
- $\underset{z\sim\tilde{P}}{E}[g_h(z)] = \underset{\tilde{S}\sim\tilde{P}_m}{E}[\hat{A}_{\tilde{S}}(g_h)] = \underset{S\sim P_m}{E}[\hat{R}_{S,\rho}(h)] \geq \underset{S\sim P_m}{E}[\hat{R}_S(h)] = R(h)$.

For any $\delta > 0$, with probability at least $1-\delta$, each of the following holds for all h in $\mathrm{co}(\mathcal{H})$:

$$R(h) \leq \hat{R}_{S,\rho}(h) + \frac{2}{\rho}\mathfrak{R}_m(\mathcal{H}) + \sqrt{\frac{\ln\frac{1}{\delta}}{2m}},$$

$$R(h) \leq \hat{R}_{S,\rho}(h) + \frac{2}{\rho}\hat{\mathfrak{R}}_S(\mathcal{H}) + 3\sqrt{\frac{\ln\frac{2}{\delta}}{2m}}.$$

Proof. By the Rademacher complexity bound for the family $\mathcal{G}$ in Theorem 3.1, for any $\delta > 0$, with probability at least $1 - \delta$, each of the following holds for all g_h in $\mathcal{G}$:

$$\begin{aligned} \underset{z\sim\tilde{P}}{E}[g_h(z)] &\leq \frac{1}{m}\sum_{i=1}^{m} g_h(\omega_i, c(\omega_i)) + 2\mathfrak{R}_m(\mathcal{G}) + \sqrt{\frac{\ln\frac{1}{\delta}}{2m}} \\ \underset{z\sim\tilde{P}}{E}[g_h(z)] &\leq \frac{1}{m}\sum_{i=1}^{m} g_h(\omega_i, c(\omega_i)) + 2\hat{\mathfrak{R}}_{\tilde{S}}(\mathcal{G}) + 3\sqrt{\frac{\ln\frac{2}{\delta}}{2m}}. \end{aligned}$$

Let

$$\tilde{\mathcal{H}} \triangleq \{z = (\omega, y) \mapsto h(\omega)y \mid h \in \mathrm{co}(\mathcal{H})\},$$

which is a family of $(-\mathscr{Y}' \cup \mathscr{Y}')$-valued functions on the input set $\mathscr{Z} = \mathscr{I} \times \{-1, +1\}$, where $(-\mathscr{Y}' \cup \mathscr{Y}')$ is a bounded subset of $\mathbb{R}$. It is clear that $\mathcal{G} = \Phi_\rho \circ \tilde{\mathcal{H}}$. Since Φ_ρ is a $1/\rho$-Lipschitz function, by

Talagrand's lemma,

$$\hat{\mathfrak{R}}_{\tilde{S}}(\mathcal{G}) \leq \frac{1}{\rho}\hat{\mathfrak{R}}_{\tilde{S}}(\tilde{\mathcal{H}}) \text{ and then } \mathfrak{R}_m(\mathcal{G}) \leq \frac{1}{\rho}\mathfrak{R}_m(\tilde{\mathcal{H}}).$$

The empirical Rademacher complexity of $\tilde{\mathcal{H}}$ is

$$\begin{aligned}
\hat{\mathfrak{R}}_{\tilde{S}}(\tilde{\mathcal{H}}) &= \frac{1}{2^m} \sum_{\sigma_1,\sigma_2,\ldots,\sigma_m \in \{-1,+1\}} \sup_{h \in \mathrm{co}(\mathcal{H})} \frac{1}{m} \sum_{i=1}^{m} \sigma_i c(\omega_i) h(\omega_i) \\
&= \frac{1}{2^m} \sum_{\sigma_1,\sigma_2,\ldots,\sigma_m \in \{-1,+1\}} \sup_{h \in \mathrm{co}(\mathcal{H})} \frac{1}{m} \sum_{i=1}^{m} \sigma_i h(\omega_i) \\
&= \hat{\mathfrak{R}}_S(\mathrm{co}(\mathcal{H}))
\end{aligned}$$

and then $\mathfrak{R}_m(\tilde{\mathcal{H}}) = \mathfrak{R}_m(\mathrm{co}(\mathcal{H}))$. Now by Theorem 6.2, $\hat{\mathfrak{R}}_S(\mathrm{co}(\mathcal{H})) = \hat{\mathfrak{R}}_S(\mathcal{H})$, and with $\frac{1}{m}\sum_{i=1}^{m} g_h(\omega_i, c(\omega_i)) = \hat{R}_{S,\rho}(h)$ and $R(h) \leq \underset{z\sim\tilde{P}}{E}[g_h(z)]$, the corollary is proved. □

Ensemble VC-Dimension Margin Bound

Corollary 6.2 : Let

- $\mathscr{I}$: the input space of all possible items ω, associated with a probability space $(\mathscr{I}, \mathcal{F}, P)$, where P is unknown;
- $c : \mathscr{I} \to \{-1, +1\}$: a fixed but unknown target concept in the concept class $\mathcal{C}$;
- $\mathscr{Y}' = [a, b]$: the output space with $-\infty < a < b < \infty$;
- $\mathcal{H}$: a hypothesis set of $\mathscr{Y}'$-valued measurable functions on the input space $\mathscr{I}$;
  - In Adaboost, $\mathcal{H}$ is the family of base classifiers.
- $S = (\omega_1, \omega_2, \ldots, \omega_m)$: a sample of m items drawn i.i.d. from $\mathscr{I}$ according to P with labels $(c(\omega_1), c(\omega_2), \ldots, c(\omega_m))$;

- $\rho > 0$: a given margin;
- $d < \infty$: the VC-dimension of $\mathcal{H}$.

For any $\delta > 0$, with probability at least $1 - \delta$, we have

$$\forall\ h \in \mathrm{co}(\mathcal{H}), \quad R(h) \leq \hat{R}_{S,\rho}(h) + \frac{2}{\rho}\sqrt{\frac{2d \ln \frac{em}{d}}{m}} + \sqrt{\frac{\ln \frac{1}{\delta}}{2m}}.$$

Proof. By Corollary 3.1, the Rademacher complexity $\mathfrak{R}_m(\mathcal{H})$ of the family $\mathcal{H}$ of base calssifiers is upper bounded by

$$\mathfrak{R}_m(\mathcal{H}) \leq \sqrt{\frac{2 \ln \Pi_{\mathcal{H}}(m)}{m}},$$

where $\Pi_{\mathcal{H}}(m)$ is the growth function for the family $\mathcal{H}$. And by Corollary 3.3, the growth function for the family $\mathcal{H}$ is upper bounded by

$$\Pi_{\mathcal{H}}(m) \leq \left(\frac{em}{d}\right)^d,$$

where d is the VC-dimension of the family $\mathcal{H}$. Thus we have

$$\mathfrak{R}_m(\mathcal{H}) \leq \sqrt{\frac{2d \ln \frac{em}{d}}{m}}$$

and by the first inequality in Corollary 6.1, the corollary is proved. □

Remarks

- Corollaries 6.1 and 6.2 cannot apply directly to AdaBoost since the returned hypothesis is not a convex combination of base classifiers.
- But they can be applied to the following normalized version of the returned hypothesis g:

$$\omega \to \frac{g(\omega)}{\|\alpha\|_1} = \frac{\sum_{t=1}^{T} \alpha_t h_t(\omega)}{\|\alpha\|_1} \in \mathrm{co}(\mathcal{H})$$

  - $\|\alpha\|_1 = \sum_{t=1}^{T} |\alpha_t| = \sum_{t=1}^{T} \alpha_t$ since $\alpha_t \geq 0$ for all t.
- Since $\mathrm{sgn}(g) = \mathrm{sgn}(g/\|\alpha\|_1)$, they have the same generalization error

$$R(g) = R(g/\|\alpha\|_1)$$

  so that for any $\delta > 0$, with probability at least $1 - \delta$, we have

for all $g \in \mathrm{co}(\mathcal{H})$,

$$
\begin{aligned}
R(g) &\leq \hat{R}_{S,\rho}(g/\|\alpha\|_1) + \frac{2}{\rho}\mathfrak{R}_m(\mathcal{H}) + \sqrt{\frac{\ln \frac{1}{\delta}}{2m}}, \\
R(g) &\leq \hat{R}_{S,\rho}(g/\|\alpha\|_1) + \frac{2}{\rho}\hat{\mathfrak{R}}_S(\mathcal{H}) + 3\sqrt{\frac{\ln \frac{2}{\delta}}{2m}}, \\
R(g) &\leq \hat{R}_{S,\rho}(g/\|\alpha\|_1) + \frac{2}{\rho}\sqrt{\frac{2d \ln \frac{em}{d}}{m}} + \sqrt{\frac{\ln \frac{1}{\delta}}{2m}}.
\end{aligned}
$$

- These bounds guarantee an effective generalization if the margin loss $\hat{R}_{S,\rho}(g/\|\alpha\|_1)$ is small for a relatively large ρ.

- Since

$$\begin{aligned}\hat{R}_{S,\rho}(g/\|\alpha\|_1) &= \frac{1}{m}\sum_{i=1}^{m}\Phi_\rho(c(\omega_i)g(\omega_i)/\|\alpha\|_1)\\ &\leq \frac{1}{m}\sum_{i=1}^{m}1_{c(\omega_i)g(\omega_i)/\|\alpha\|_1\leq\rho}\\ &= \frac{|\{i\in[1,m]\mid c(\omega_i)g(\omega_i)/\|\alpha\|_1\leq\rho\}|}{m},\end{aligned}$$

  the ρ-margin loss is upper bounded by the fraction of the items ω_i in the training sample with $c(\omega_i)g(\omega_i)/\|\alpha\|_1 \leq \rho$.

  – The quantity

$$\rho_1(\omega_i) \triangleq \frac{c(\omega_i)g(\omega_i)}{\|\alpha\|_1}$$

    is defined as the L_1-margin of the item ω_i with label $c(\omega_i)$ for an AdaBoost classifier $g = \sum_{t=1}^{T}\alpha_t h_t$.

  – $\{i\in[1,m]|c(\omega_i)g(\omega_i)/\|\alpha\|_1\leq\rho\} = \{i\in[1,m]|\rho_1(\omega_i)\leq\rho\}$.

Bound on the ρ-Margin Loss

Theorem 6.3: Let

- $g_S = \sum_{t=1}^{T} \alpha_t h_{S,t}$: the classifier returned by AdaBoost after T rounds of boosting;
- $\epsilon_t < 1/2$ for all $t \in [1, T]$ which implies that $\alpha_t = \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t} > 0$.

Then for any $\rho > 0$,

$$\hat{R}_{S,\rho}(g_S/\|\alpha\|_1) \leq 2^T \prod_{t=1}^{T} \sqrt{\epsilon_t^{1-\rho}(1-\epsilon_t)^{1+\rho}}.$$

Proof. Note that

- $1_{u \leq 0} \leq e^{-u}$ for all $u \in \mathbb{R}$.
- $D_{T+1}(i) = \frac{\exp\{-c(\omega_i) g_S(\omega_i)\}}{m \prod_{t=1}^{T} Z_t}$ for all $i \in [1, m]$.
- $Z_t = 2\sqrt{\epsilon_t(1-\epsilon_t)}$ for all $t \in [1, T]$.

Now the empirical ρ-margin loss of the normalized AdaBoost classifier $g_S/\|\alpha\|_1$ w.r.t. the concept c on the labeled sample S is

$$\begin{aligned}
\hat{R}_{S,\rho}(g_S/\|\alpha\|_1) &\leq \frac{1}{m}\sum_{i=1}^{m} 1_{c(\omega_i)g_S(\omega_i)-\rho\|\alpha\|_1\leq 0} \\
&\leq \frac{1}{m}\sum_{i=1}^{m} e^{-c(\omega_i)g_S(\omega_i)+\rho\|\alpha\|_1} \\
&= \frac{1}{m}\sum_{i=1}^{m} e^{\rho\|\alpha\|_1}\left(m\prod_{t=1}^{T} Z_t\right) D_{T+1}(i) \\
&= e^{\rho\|\alpha\|_1}\prod_{t=1}^{T} Z_t = e^{\rho\sum_{t=1}^{T}\alpha_t}\prod_{t=1}^{T} Z_t \\
&= 2^T\prod_{t=1}^{T}\left(\sqrt{\frac{1-\epsilon_t}{\epsilon_t}}\right)^{\rho}\sqrt{\epsilon_t(1-\epsilon_t)},
\end{aligned}$$

which completes the proof. □

A Function $f(\epsilon) = (1-\rho)\ln\epsilon + (1+\rho)\ln(1-\epsilon)$ on $\epsilon \in (0,1)$

- $0 < \rho < 1$.
- $f'(\epsilon) = \frac{1-\rho}{\epsilon} - \frac{1+\rho}{1-\epsilon} = 2\frac{\left(\frac{1}{2}-\frac{\rho}{2}\right)-\epsilon}{\epsilon(1-\epsilon)}$ for all $\epsilon \in (0,1)$.
- $f'(\epsilon) > 0$ for $\epsilon \in (0, \frac{1}{2} - \frac{\rho}{2})$; $f'(\epsilon) = 0$ for $\epsilon = \frac{1}{2} - \frac{\rho}{2}$; and $f'(\epsilon) < 0$ for $\epsilon \in (\frac{1}{2} - \frac{\rho}{2}, 1)$.
- If $\epsilon \leq \frac{1}{2} - \gamma$ and $\frac{\rho}{2} \leq \gamma$, then $f(\epsilon)$ is maximized at $\epsilon = \frac{1}{2} - \gamma$.

Remarks

- If $\epsilon_t \leq 1/2 - \gamma$ for all $t \in [1, T]$ and $\frac{\rho}{2} \leq \gamma < \frac{1}{2}$, then $4\epsilon_t^{1-\rho}(1-\epsilon_t)^{1+\rho}$ is maximized at $\epsilon_t = 1/2 - \gamma$ with maximum value $(1-2\gamma)^{1-\rho}(1+2\gamma)^{1+\rho}$ and the empirical ρ-margin loss of the AdaBoost classifier is upper bounded by

$$\hat{R}_{S,\rho}(g_S/\|\alpha\|_1) \leq \left((1-2\gamma)^{1-\rho}(1+2\gamma)^{1+\rho}\right)^{T/2}.$$

- Since

$$(1-2\gamma)^{1-\rho}(1+2\gamma)^{1+\rho} = (1-4\gamma^2)\left(\frac{1+2\gamma}{1-2\gamma}\right)^{\rho}$$

is an increasing function of ρ, if $\rho < \gamma$, then we have

$$(1-2\gamma)^{1-\rho}(1+2\gamma)^{1+\rho} < (1-2\gamma)^{1-\gamma}(1+2\gamma)^{1+\gamma}.$$

- It can be seen that

$$(1-2\gamma)^{1-\gamma}(1+2\gamma)^{1+\gamma} < 1 \; \forall \; \gamma \in (0, \frac{1}{2}).$$

- Thus if $\rho < \gamma < \frac{1}{2}$, then $(1-2\gamma)^{1-\rho}(1+2\gamma)^{1+\rho} < 1$ and then the empirical ρ-margin loss $\hat{R}_{S,\rho}(g_S/\|\alpha\|_1)$ decreases exponentially with T.
- To have informative margin bound, it is required that

$$\rho \gg O(1/\sqrt{m}) \Rightarrow \gamma \gg O(1/\sqrt{m}).$$

- In practice, the error ϵ_t of the base classifier at round t may increase as a function of t. Informally, this is because boosting presses the weak learner to concentrate on instances that are harder and harder to classify, for which even the best base classifier could not achieve an error significantly better than random. If ϵ_t becomes close to $1/2$ relatively fast as a function of t, then the bound of Theorem 6.3 becomes uninformative.

The Contents of This Lecture

- Introduction.
- AdaBoost.
- Theoretical results.
- Discussion.

Discussions

- AdaBoost offers several advantages: it is simple, its implementation is straightforward, and the time complexity of each round of boosting as a function of the sample size is rather favorable.
  - As already discussed, when using decision stumps, the time complexity of each round of boosting is in $O(mN)$. Of course, if the dimension of the feature space N is very large, then the algorithm could become in fact quite slow.
- AdaBoost additionally benefits from a rich theoretical analysis. Nevertheless, there are still many theoretical questions. For example, as we saw, the algorithm in fact does not maximize the margin, and yet algorithms that do maximize the margin do not always outperform it. This suggests that perhaps a finer analysis based on a notion different from that of margin could

shed more light on the properties of the algorithm.

- The main drawbacks of the algorithm are the need to select the parameter T and the base classifiers, and its poor performance in the presence of noise. The choice of the number of rounds of boosting T (stopping criterion) is crucial to the performance of the algorithm. As suggested by the VC-dimension analysis, larger values of T can lead to overfitting.
- In practice, T is typically determined via cross-validation.
- The choice of the base classifiers is also crucial. The complexity of the family $\mathcal{H}$ of base classifiers appeared in all the bounds presented and it is important to control it in order to guarantee generalization. On the other hand, insufficiently complex hypothesis sets could lead to low margins.

- Probably the most serious disadvantage of AdaBoost is its performance in the presence of noise; it has been shown empirically that noise severely damages its accuracy. The distribution weight assigned to examples that are harder to classify substantially increases with the number of rounds of boosting, by the nature of the algorithm. These examples end up dominating the selection of the base classifiers, which, with a large enough number of rounds, will play a detrimental role in the definition of the linear combination defined by AdaBoost.
- The behavior of AdaBoost in the presence of noise can be used, however, as a useful feature for detecting outliers, that is, examples that are incorrectly labeled or that are hard to classify. Examples with large weights after a certain number of rounds of boosting can be identified as outliers.