1 AdaBoost

Algorithm 1 AdaBoost

\[
\forall i : D_1(i) = \frac{1}{m} \\
\text{for } t = 1..T \text{ do} \\
\quad h_t \leftarrow \text{Run A on } D_t \\
\quad \epsilon_t = err_{D_t}(h_t) = \frac{1}{2} - \gamma_t \\
\quad \alpha_t = \frac{1}{2} \ln \left( \frac{1-\epsilon_t}{\epsilon_t} \right) \\
\quad \forall i : D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \left\{ \begin{array}{ll}
    e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \\
    e^{-\alpha_t} & \text{if } h_t(x_i) = y_i
\end{array} \right. \\
\text{return } H(x) = \text{sign} \left( \sum_{t=1}^{T} \alpha_t h_t(x) \right)
\]

In this algorithm, \( Z_t \) represents a normalizing factor since \( D_{t+1} \) is a probability distribution.

1.1 Bounding the training error.

In the previous class, we gave the basic intuition behind the AdaBoost algorithm. Now, having defined the value for \( \alpha_t \), we tracked the three rounds of the algorithm in a toy example (see slides on the course website).

Theorem 1.1. The training error is bounded by the following expression:

\[
e\hat{err}(H) \leq \prod_{t=1}^{T} 2\sqrt{\epsilon_t (1 - \epsilon_t)} = \exp \left( -\sum_t RE \left( \frac{1}{2} \| \epsilon_t \right) \right) \tag{By definition of RE}
\]

\[
= \prod_t \sqrt{1 - 4\gamma_t^2} \tag{\epsilon_t = \frac{1}{2} - \gamma_t}
\]

\[
\leq \exp \left( -2 \sum_t \gamma_t^2 \right) \tag{1 + x \leq e^x}
\]

Considering the weak learning assumption: \( \gamma_t \geq \gamma > 0 \)

\[
\leq e^{-2\gamma^2 T}
\]

Step 1: \( D_{T+1}(i) = \frac{\exp[-y_i F(x_i)]}{m^T Z_t}, F(x) = \sum_t \alpha_t h_t(x) \)
Proof.

$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times e^{-\alpha_t h_t(x_i)} = D_t(i) \frac{e^{-y_i h_t(x_i)}}{Z_t}$

Then, we can find this expression for $t = T$, and solve recursively:

$D_{T+1} = D_1(i) \frac{e^{-y_i a_1 h_1(x_i)}}{Z_1} \times \cdots \times \frac{e^{-y_i a_T h_T(x_i)}}{Z_T}$

$= \frac{1}{m} \exp \left( -y_i \sum_{t} \alpha_t h_t(x_i) \right) \prod_t Z_t$

$= \exp \left[ -y_i F(x_i) \right] \frac{1}{m \prod_t Z_t}$

Step 2: $\hat{e}^r(H) \leq \prod_t Z_t$

Proof.

$\hat{e}^r(H) = \frac{1}{m} \sum_{i=1}^{m} \{ y_i \neq H(x_i) \}$

$= \frac{1}{m} \sum_i \{ y_i F(x_i) \leq 0 \}$

$\leq \frac{1}{m} \sum_i e^{-y_i F(x_i)}$

$= \frac{1}{m} \sum_i D_{T+1}(i) m \prod_t Z_t$

$= \prod_t Z_t \sum_i D_{T+1}(i)$

$= \prod_t Z_t$

(3) follows since $e^{-y_i F(x_i)} > 0$ if $-y_i F(x_i) > 0$ and $e^{-y_i F(x_i)} \geq 1$ if $-y_i F(x_i) \leq 0$. (4) follows from Step 1. (6) follows from the fact that we are adding all values over distribution $D_{T+1}$ so we are getting 1.

Step 3: $Z_t = 2\sqrt{\epsilon_t(1-\epsilon_t)}$

Proof.

$Z_t = \sum_i D_t(i) \times \left\{ \begin{array}{ll} e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \\ e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \end{array} \right.$

$= \sum_{i : y_i \neq h_t(x_i)} D_t(i) e^{\alpha_t} + \sum_{i : y_i = h_t(x_i)} D_t(i) e^{-\alpha_t}$

$= \epsilon_t e^{\alpha_t} + (1-\epsilon_t) e^{-\alpha_t}$
(2) follows from just decomposing the sum for the two cases. (3) follows from the fact that $e^{\alpha t}$ or $e^{-\alpha t}$ can be taken outside of the sum, and $\sum_{i:y_i \neq h_{2t}(x_i)} D_t(i) = \epsilon_t$ and $\sum_{i:y_i = h_{1t}(x_i)} D_t(i) = 1 - \epsilon_t$.

We choose $\alpha_t$ to minimize the empirical error, so we get:

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

*This is how we choose $\alpha_t$ in the algorithm.

1.2 Bounding the generalization error.

Of the many tools we have used over the past classes, we choose the growth function to bound the generalization error.

$$H(x) = \text{sign} \left( \sum_t \alpha_t h_t(x) \right)$$

$$= g(h_1(x), \ldots, h_T(x))$$

We defined $g(z_1, z_2, \ldots, z_t) = \text{sign}(\sum_t \alpha_t z_t) = \text{sign}(w \cdot z)$, with $w = (\alpha_1, \alpha_2, \ldots, \alpha_T)$, which represents linear threshold functions in $\mathbb{R}^T$. Let us define now the following spaces:

$$\mathcal{J} = \{\text{LTFs in } \mathbb{R}^T\}$$

$$\mathcal{H} = \text{weak hypothesis space}$$

$$\mathcal{F} = \text{all functions } f \text{ (as above), where } g \in \mathcal{J}, h_1, h_2, \ldots, h_T \in \mathcal{H}$$

As proved in problem 2 of Homework 2, we can set the following bound:

$$\Pi_{\mathcal{F}}(m) \leq \Pi_{\mathcal{J}}(m) \prod_{t=1}^T \Pi_{\mathcal{H}}(m)$$

$$= \Pi_{\mathcal{J}}(m) [\Pi_{\mathcal{H}}(m)]^T$$

We have that $\text{VC-dim}(\mathcal{J}) = T$ since we are considering linear threshold functions going through the origin in $\mathbb{R}^T$, and we define $\text{VC-dim}(\mathcal{H}) = d$. Then, using Sauer’s Lemma:

$$\Pi_{\mathcal{J}}(m) \leq \left(\frac{em}{T}\right)^T$$

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

Plugging the above inequalities in equation (4):

$$\Pi_{\mathcal{F}}(m) \leq \left(\frac{em}{T}\right)^T \left(\frac{em}{d}\right)^{dT}$$

Using “soft-oh” notation (not only hides constant but also log factors), given $m$ examples, with probability at least $1 - \delta, \forall H \in \mathcal{F}$:

$$\text{err}(H) \leq \hat{\text{err}}(H) + \tilde{O} \left( \sqrt{\frac{Td + \ln 1/\delta}{m}} \right)$$
1.3 Margin

Contrary to what we would expect based on the previous equation, as we increase $T$ (the complexity) we do not always get a worse generalization error even when the training error is already 0. The following image is the one in the slides from class that represents this behavior:

![Graph I: Error versus # of rounds of boosting](image)

The reason behind this behavior is that, as we keep increasing the number of rounds, the classifier becomes more “confident”. This confidence translates into a lower generalization error. We have:

$$H(x) = \text{sign} \left( \sum_{t=1}^{T} a_t h_t(x) \right), \quad \text{where} \quad a_t = \frac{\alpha_t}{\sum_{t'=1}^{T} \alpha_{t'}}$$

In this way, we are normalizing the weights for each hypothesis, having $a_t \geq 0, \sum a_t = 1$. We define the margin as the difference between the weighted fraction of $h_t$’s voting correctly and the fraction corresponding to those voting incorrectly. Then for an example $x$ with correct label $y$, the margin is:

$$\text{margin} = \sum_{t: h_t(x) = y} a_t - \sum_{t: h_t(x) \neq y} a_t = \sum_{t} a_t y h_t(x) = y \sum_{t} a_t h_t(x) = y f(x) \quad \text{where} \quad f(x) = \sum_{t} a_t h_t(x)$$