The training error theorem for boosting

Here is pseudocode for the AdaBoost boosting algorithm presented in class:

Given: \((x_1, y_1), \ldots, (x_N, y_N)\) where \(x_i \in X, y_i \in \{-1, +1\}\)

Initialize \(D_1(i) = 1/N\).

For \(t = 1, \ldots, T\):

- Train weak learner using training data weighted according to distribution \(D_t\).
- Get weak hypothesis \(h_t : X \rightarrow \{-1, +1\}\).
- Measure “goodness” of \(h_t\) by its weighted error with respect to \(D_t\):
  \[\epsilon_t = \Pr_{i \sim D_t} [h_t(x_i) \neq y_i] = \sum_{i : h_t(x_i) \neq y_i} D_t(i)\]
- Let \(\alpha_t = \frac{1}{2} \ln \left( \frac{1 - \epsilon_t}{\epsilon_t} \right)\).
- Update:
  \[D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } y_i = h_t(x_i) \\ e^{\alpha_t} & \text{if } y_i \neq h_t(x_i) \end{cases}\] (1)

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

Output the final classifier:

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

Although the notation is different, this algorithm is the same as in Fig. 18.10 of R&N.

In this note, we prove the training error theorem, which states that the training error of \(H\) is at most

\[\exp \left(-2 \sum_{t=1}^{T} \gamma_t^2\right)\]

where \(\epsilon_t = \frac{1}{2} - \gamma_t\).

We prove this in three steps.

**Step 1:** The first step is to show that

\[D_{T+1}(i) = \frac{1}{N} \cdot \frac{\exp(-y_i f(x_i))}{\prod_t Z_t}\]

where

\[f(x) = \sum_t \alpha_t h_t(x)\]

Proof: Note that Eq. (1) can be rewritten as

\[D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}\]

...
since $y_i$ and $h_t(x_i)$ are both in $\{-1,+1\}$. Unwrapping this recurrence, we get that

$$D_{T+1}(i) = D_1(i) \cdot \frac{\exp(-\alpha_1 y_i h_1(x_i))}{Z_1} \cdot \ldots \cdot \frac{\exp(-\alpha_T y_T h_T(x_i))}{Z_T}$$

$$= \frac{1}{N} \cdot \frac{\exp(-y_i \sum_t \alpha_t h_t(x_i))}{\prod_t Z_t}$$

$$= \frac{1}{N} \cdot \frac{\exp(-y_i f(x_i))}{\prod_t Z_t}.$$

**Step 2:** Next, we show that the training error of the final classifier $H$ is at most $\prod_{t=1}^T Z_t$.

Proof:

$$\text{training error}(H) = \frac{1}{N} \sum_i \left\{ \begin{array}{ll}
1 & \text{if } y_i \neq H(x_i) \\
0 & \text{else}
\end{array} \right. \quad \text{by definition of the training error}
$$

$$= \frac{1}{N} \sum_i \left\{ \begin{array}{ll}
1 & \text{if } y_i f(x_i) \leq 0 \\
0 & \text{else}
\end{array} \right. \quad \text{since } H(x) = \text{sign}(f(x)) \text{ and } y_i \in \{-1,+1\}
$$

$$\leq \frac{1}{N} \sum_i \exp(-y_i f(x_i)) \quad \text{since } e^{-z} \geq 1 \text{ if } z \leq 0
$$

$$= \sum_i D_{T+1}(i) \prod_t Z_t \quad \text{by Step 1 above}
$$

$$= \prod_t Z_t \quad \text{since } D_{T+1} \text{ is a distribution}
$$

**Step 3:** The last step is to compute $Z_t$.

We can compute this normalization constant as follows:

$$Z_t = \sum_i D_t(i) \times \begin{cases} 
eq y_i & \text{if } h_t(x_i) = y_i \\
\alpha & \text{if } h_t(x_i) \neq y_i \end{cases}
$$

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

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

$$= e^{-\alpha t} (1 - \epsilon_t) + e^{\alpha t} \epsilon_t \quad \text{by definition of } \epsilon_t
$$

$$= 2 \sqrt{\epsilon_t (1 - \epsilon_t)} \quad \text{by our choice of } \alpha (\text{which was chosen to minimize this expression})
$$

$$= \sqrt{1 - 4 \gamma_t^2} \quad \text{plugging in } \epsilon_t = \frac{1}{2} - \gamma_t
$$

$$\leq e^{-2 \gamma_t^2} \quad \text{using } 1 + x \leq e^x \text{ for all real } x
$$

Combining with Step 2 gives the claimed upper bound on the training error of $H$. 