Generalized AdaBoost Algorithm

Given: \((x_1, y_1), \ldots, (x_m, y_m)\); \(x_i \in \mathcal{X}, y_i \in \{-1, +1\}\)
Initialize \(D_1(i) = 1/m\).
For \(t = 1, \ldots, T\):

- Train weak learner using distribution \(D_t\).
- Get weak hypothesis \(h_t : \mathcal{X} \rightarrow \mathbb{R}\).
- Choose \(\alpha_t \in \mathbb{R}\).
- Update:

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

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

Output the final hypothesis:

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

Given: \((x_1, y_1), \ldots, (x_m, y_m)\); \(x_i \in \mathcal{X}, y_i \in \{-1, +1\}\)
Initialize \(D_1(i) = 1/m\).
For \(t = 1, \ldots, T\):

- Train weak learner using distribution \(D_t\).
- Get weak hypothesis \(h_t : \mathcal{X} \to \mathbb{R}\).
- Choose \(\alpha_t \in \mathbb{R}\).
- Update:

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

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

Output the final hypothesis:

\[
H(x) = \text{sign} \left( \sum_{t=1}^{T} \alpha_t h_t(x) \right).
\]
AdaBoost and AdaBoost.MH on Train (left) and Test (right) data from Irvine repository. [Schapire and Singer, ML 1999]
Boosting Minimizes Exponential Loss Function
[Collins et al., 2002]

One reasonable loss function to minimize during learning is the sum of training errors weighted by classifier confidence

$$\sum_{i=1}^{m} [y_i f_{\lambda}(x_i) \leq 0]$$

where

$$f(x) = \sum_{t} \alpha_t h_t(x)$$

AdaBoost has been proven [Collins et al., MLJournal, 2002] to minimize the exponential lost function

$$\sum_{i=1}^{m} \exp (-y_i f_{\lambda}(x_i)).$$
AdaBoost minimizes exponential loss:

\[ \sum_{i=1}^{m} \exp (-y_i f_\lambda (x_i)). \]

Which is similar to the loss function minimized by logistic regression, which learns \( \Pr[y = +1 | x] = \frac{1}{1 + e^{-f_\lambda (x)}}. \)

The likelihood of the labels occurring in the sample then is

\[ \prod_{i=1}^{m} \frac{1}{1 + \exp (-y_i f_\lambda (x_i))}. \]

Maximizing this likelihood then is equivalent to minimizing the log loss of this model

\[ \sum_{i=1}^{m} \ln (1 + \exp (-y_i f_\lambda (x_i))). \]