# Discrete AdaBoost

Cary Goltermann

July 13, 2017

## Introduction

Discrete AdaBoost, referred to hereafter as adaboost, is an application of forward stagewise additive modeling, the goal of which is to minimize, at each stage, $m$:

$$\min_{\phi} \sum_{i=1}^{N} L(y_i, f_{m-1}(x_i) + \phi(x_i)),$$

where $L(y, \hat{y})$ is come loss function and $f$ is a sum of adaptive basis functions, $\phi$, often referring to as a weak learning, frequently chosen to be a decision tree. The

additive part of the model can be seen in the equation that at each stage we will train a model $\phi(x)$ that minimizes the loss when it's opinion as added to the previous $f$, $f_{m-1}$.

## AdaBoost

In the situation of a binary classification problem we can use exponential loss as our $L$, $L(y, f) = e^{-yf}$.

Here we will label $y \in \{-1, 1\}$, different than the usual $y \in \{0, 1\}$, will make the math work out more simply. Therefore, at step $m$ we have to minimize:

$$\begin{aligned} L_m(\phi) &= \sum_{i=1}^{N} e^{-y_i(f_{m-1}(x_i) + \beta\phi(x_i))} \\ &= \sum_{i=1}^{N} w_{i,m} e^{-\beta y \phi(x_i)} \end{aligned}$$

where $w_{i,m} = e^{-y f_{m-1}(x_i)}$ is a weight applied to observation $i$. This objective can be rewritten as:

$$
\begin{aligned}
L_m &= e^{-\beta}\sum_{y_i=\phi x_i} w_{i,m} = e^{\beta}\sum y_i \neq \phi x_i w_{i,m} \\
&= (e^{\beta} - e^{-}\beta)\sum_{i=1}^{N} w_{i,m}\mathbb{I}(y \neq \phi(x_i)) + e^{-\beta}\sum_{i=1}^{N} w_{i,m}
\end{aligned}
$$

Consequently the optimal function to add is:

$$
\phi_m = \operatorname*{argmin}_{\phi}\, w_{i,m}\mathbb{I}(y \neq \phi(x_i))
$$

This can be found by fitting $\phi$ to a weighted version of the dataset, with weights $w_{i,m}$. Substituting $\phi_m$ into $L(m)$ and solving for $\beta$ we find:

$$
\beta_m = \frac{1}{2} log \frac{1 - err_m}{err_m}
$$

where

$$
err_m = \frac{\sum_{i=1}^{N} w_i \mathbb{I}(y \neq \phi(x_i))}{\sum_{i=1}^{N} w_{i,m}}
$$

The overall update becomes:

$$
f_m(x) = f_{m-1}(x) + \beta_m \phi_m(x)
$$

With this, the weights at the next iteration, $w + 1$, become:

$$
\begin{aligned}
w_{i,m+1} &= w_{i,m} e^{-\beta_m y_i \phi_m(x_i)} \\
&= w_{i,m} e^{-\beta_m (2\mathbb{I}(y_i \neq \phi(x_i) - 1)} \\
&= w_{i,m} e^{-\beta_m (2\mathbb{I}(y_i \neq \phi(x_i))} e^{-\beta_m}
\end{aligned}
$$

Notice, we were exploiting the fact that $-y_i\phi_m(x_i) = -1$ if $y_i = \phi_m(x_i)$ and $-y_i\phi_m(x_i) = 1$ otherwise.

# AdaBoost: Algorithm

This leads us to the full AdaBoost algorithm:

1. $w_i = \frac{1}{N}$;
2. for $m = 1$ to $M$:
   (a) Fit a classifier $\phi_m(x)$ to the training set using weights $w$;
   (b) Compute $err_m = \frac{\sum_{i=1}^{N} w_i \mathbb{I}(y \neq \phi(x_i))}{\sum_{i=1}^{N} w_{i,m}}$;
   (c) Compute $\alpha_m = log\frac{1-err_m}{err_m}$;
   (d) Set $w_i \leftarrow w_i e^{\alpha_m \mathbb{I}(y \neq \phi(x_i))}$;
3. Return $f(x) = sgn\left[\sum_{m=1}^{M} \alpha_m \phi_m(x)\right]$