# AdaBoost in a Nutshell

Ilker Birbil

September, 2019

In this short note, I will try to give a few additional details about the infamous AdaBoost algorithm. Consider a binary classification problem. The data is $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ where $x_i \in \mathbb{R}^p$ and $y_i \in \{+1, -1\}$.

Assume that we know how to train a series of *weak* binary classifiers $y_\ell(x) \in \{+1, -1\}$, $\ell = 1, 2, \ldots$. We shall boost the compounded performance of these classifiers in an iterative manner by giving different weights to the data points. In particular, the misclassified points shall get higher weights so that the classifiers will pay more attention to those points to reduce the error. First, the algorithm.

**Algorithm 1:** AdaBoost

1 Initialize weights $w_i^1 = \frac{1}{n}$ for $i = 1, \ldots, n$

2 **for** $k = 1, \ldots, K$ **do**

3 Fit a classifier $y_k(x)$ that minimizes the weighted classification error

$$\sum_{i=1}^{n} w_i^k I(y_k(x_i) \neq y_i)$$

4 Evaluate the relative weight of the misclassified data points

$$\varepsilon^k = \frac{\sum_{i=1}^{n} w_i^k I(y_k(x_i) \neq y_i)}{\sum_{i=1}^{n} w_i^k}$$

5 Set the update parameter $\alpha_k = \ln \frac{1-\varepsilon_k}{\varepsilon_k}$

6 Update the weights $w_i^{k+1} = w_i^k e^{\alpha_k I(y_k(x_i) \neq y_i)}$ for $i = 1, \ldots, n$

7 **Output:** $\operatorname{sign}\left(\sum_{k=1}^{K} \alpha_k y_k(x)\right)$

There are a couple of important points in Algorithm 1. Note that in line 5, $\alpha_k$ becomes large whenever the relative weights of the misclassified points are small. Thus, the equation in line 6 makes sure that the weights of the misclassified points go up at the next iteration. Moreover, a large $\alpha_k$ value promotes the corresponding classifier $k$ in line 7 .

Next, we shall try to give an interpretation to AdaBoost. Consider a classifier of the form $f_k(x) = \frac{1}{2}\sum_{\ell=1}^{k} \alpha_\ell y_\ell(x)$. That is, $f_k(x)$ is a linear combination of $k$ classifiers and its

response is simply $\text{sign}(f_k(x))$. We want to train $f_k(x)$ by minimizing the following error function

$$\sum_{i=1}^{n} e^{-y_i f_k(x_i)} = \sum_{i=1}^{n} e^{-y_i \frac{1}{2} \sum_{\ell=1}^{k} \alpha_\ell y_\ell(x_i)}.$$

We need to choose $\alpha_1, \ldots, \alpha_k$ and the parameters of the classifiers $y_\ell(x)$, $\ell = 1, \ldots, k$.

How about minimizing this error function incrementally rather than minimizing the whole function at once? In particular, we shall assume that the classifiers $y_1(x), \ldots, y_{k-1}(x)$ and their multipliers $\alpha_1, \ldots, \alpha_{k-1}$ have already been fixed. Then, we want to minimize the error function by choosing $\alpha_k$ and $y_k(x)$.

The error function is given by

$$\begin{aligned}
\textstyle\sum_{i=1}^{n} e^{-y_i \frac{1}{2} \sum_{\ell=1}^{k} \alpha_\ell y_\ell(x_i)} &= \textstyle\sum_{i=1}^{n} e^{-y_i f_{k-1}(x_i) - \frac{1}{2} y_i \alpha_k y_k(x_i)} \qquad \boxed{w_i^k \triangleq e^{-y_i f_{k-1}(x_i)}} \\
&= \textstyle\sum_{i=1}^{n} w_i^k e^{-\frac{1}{2} y_i \alpha_k y_k(x_i)} \\
&= \textstyle\sum_{i=1}^{n} w_i^k \left( I(y_k(x_i) = y_i) e^{-\frac{1}{2}\alpha_k} + I(y_k(x_i) \neq y_i) e^{\frac{1}{2}\alpha_k} \right) \\
&= \textstyle\sum_{i=1}^{n} w_i^k e^{-\frac{\alpha_k}{2}} + (e^{\frac{\alpha_k}{2}} - e^{-\frac{\alpha_k}{2}}) \sum_{i=1}^{n} w_i^k I(y_k(x_i) \neq y_i).
\end{aligned}$$

Minimizing this error function with respect to the parameters of the classifier $y_k(x)$ is equivalent to minimizing

$$\sum_{i=1}^{n} w_i^k I(y_k(x_i) \neq y_i)$$

as done in line 3 of Algorithm 1. On the other hand, minimizing the error function with respect to $\alpha_k$ boils down to taking the derivative and finding the root of the resulting equation[1]. This step leads to the $\alpha_k$ evaluation in line 5.

By definition, we also have

$$\begin{aligned}
w_i^{k+1} &= e^{-y_i f_k(x_i)} \\
&= e^{-y_i f_{k-1}(x_i) - \frac{1}{2} y_i \alpha_k y_k(x_i)} \\
&= w_i^k e^{-\frac{1}{2} y_i \alpha_k y_k(x_i)} \\
&= w_i^k e^{-\frac{1}{2}\alpha_k (1 - 2I(y_k(x_i) \neq y_i))} \\
&= w_i^k \underbrace{e^{-\frac{1}{2}\alpha_k}}_{c_k} e^{\alpha_k I(y_k(x_i) \neq y_i)}.
\end{aligned}$$

Note that the term marked as $c_k$ just normalizes each weight with the same amount. Thus, we can safely drop it and obtain line 6 of Algorithm 1 as

$$w_i^{k+1} = w_i^k e^{\alpha_k I(y_k(x_i) \neq y_i)}.$$

Finally, we output classification response according to

$$\text{sign}(f_k(x)) = \text{sign}\left( \frac{1}{2} \sum_{l=1}^{k} \alpha_l y_l(x) \right) = \text{sign}\left( \sum_{l=1}^{k} \alpha_l y_l(x) \right)$$

giving line 7 of the AdaBoost algorithm.

[1] Please try this step on your own.