AdaBoost

S. Sumitra
Department of Mathematics
Indian Institute of Space Science and Technology

1 Introduction

In this chapter, we are considering AdaBoost algorithm for the two class classification problem.

AdaBoost (Adaptive Boosting) generates a sequence of hypothesis and combines them with weights. That is

$$H(x) = sgn\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

where $h_t : \mathcal{X} \rightarrow \{1, -1\}, t = 1, 2, \ldots T$ are called base learners or weak learners and α_t is the weight associated with h_t. Hence two questions are there: how to generate the hypothesis $h_t's$? and how to determine the proper weights $\alpha_t's$?

Let $D = \{(x_1, y_1), ..., (x_N, y_N)\},\ x_i \in \mathcal{X} \subseteq R^n, y_i \in \{-1, 1\}$ be the given data. For generating T classifiers, there would be T iterations and in each iteration training data is chosen from N points with replacement. Each data point is associated with a weight and it decides the probabity of each point getting selected as a training point.

Initially all the data points have equal probability of getting selected, that is each data point has a weight equal to $1/N$. In each iteration the weight of a data point gets changed in such a way, that it gets decreased, if it is correctly classified by the model generated in that iteration and increased otherwise.

Given the training data, choose an appropriate classification algorithm to find h_t. To find the weight corresponding to each classifier we need to formulate an objective function and find α to minimize it. The objective function used is: to minimize

$$\sum_{i=1}^{N} 1_{y_i \neq sgn\left(\sum_{k=1}^{t} \alpha_k h_k(x_i)\right)} \tag{1}$$

That is at each step the weight of base classifier is chosen in such a way that the error of $H(x)$ is minimized.

(1) is difficult to minimize and therefore for finding the optimal weight of each classifier the following function which is an upper bound of (1) is used:

$$\sum_{i=1}^{N} e^{-y_i\left(\sum_{k=1}^{t}\alpha_k h_k(x_i)\right)} \tag{2}$$

This is because if $y_i \neq sgn\left(\sum_{k=1}^{t}\alpha_k h_k(x_i)\right)$, then $e^{-y_i\left(\sum_{k=1}^{t}\alpha_k h_k(x_i)\right)} \geq 1$ and if $y_i = sgn\left(\sum_{k=1}^{t}\alpha_k h_k(x_i)\right)$, then $0 \leq e^{-y_i\left(\sum_{k=1}^{t}\alpha_k h_k(x_i)\right)} \leq 1$. Therefore

$$\sum_{i=1}^{N} 1_{y_i \neq sgn\left(\sum_{k=1}^{t}\alpha_k h_k(x_i)\right)} \quad \leq \quad \sum_{i=1}^{N} e^{-y_i\left(\sum_{k=1}^{t}\alpha_k h_k(x_i)\right)} \tag{3}$$

Also, $e^{-y_i\left(\sum_{k=1}^{t}\alpha_k h_k(x_i)\right)}$ is smooth and differentiable in all places.

2 Updating the weight of the classifier

Consider the t^{th} iteration. To find α_t, the objective is to minimize

$$\sum_{i=1}^{N} e^{-y_i\sum_{k=1}^{t}\alpha_k h_k(x_i)}$$

For the next iteration, that is $t = (t+1)$ the objective is to minimize,

$$\sum_{i=1}^{N} e^{-y_i\left(\sum_{k=1}^{t}\alpha_k h_k(x_i)+\alpha_{t+1}h_{(t+1)}(x_i)\right)}$$

Let $obj_t = \sum_{i=1}^{N} e^{-y_i\sum_{k=1}^{t}\alpha_k h_k(x_i)}$ and $obj_{t+1} = e^{-y_i\left(\sum_{k=1}^{t}\alpha_k h_k(x_i)+\alpha_{t+1}h_{(t+1)}(x_i)\right)}$

$$\begin{aligned}
\frac{obj_{(t+1)}}{obj_t} &= \frac{\sum_{i=1}^{N} e^{-y_i\left(\sum_{k=1}^{t}\alpha_k h_k(x_i)+\alpha_{t+1}h_{t+1}(x_i)\right)}}{\sum_{i=1}^{N} e^{-y_i\sum_{k=1}^{t}\alpha_k h_k(x_i)}} \\
&= \sum_{i=1}^{N} \frac{e^{-y_i\sum_{k=1}^{t}\alpha_k h_k(x_i)}}{\sum_{i=1}^{N} e^{-y_i\sum_{k=1}^{t}\alpha_k h_k(x_i)}} e^{-y_i\alpha_{t+1}h_{t+1}(x_i)} \\
&= \sum_{i=1}^{N} D_{t+1}(i) e^{-y_i\alpha_{(t+1)}h_{t+1}(x_i)} \qquad (4)
\end{aligned}$$

where

$$D_{t+1}(i) = \frac{e^{-y_i \sum_{k=1}^{t} \alpha_k h_k(x_i)}}{\sum_{i=1}^{N} e^{-y_i \sum_{k=1}^{t} \alpha_k h_k(x_i)}} \tag{5}$$

$D_{t+1}(i)$ is the weight that is assigned to i^{th}sample during the $(t+1)^{th}$ iteration. Hence in $(t+1)^{th}$ iteration, the weight of all the data points which is classified correctly by the t^{th} ensemble model is less than those which it misclassified. That is, if for (x_l, y_l) and (x_m, y_m), $y_l = sgn\left(\sum_{k=1}^{t} \alpha_k h_k(x_l)\right)$ and $y_m \neq sgn\left(\sum_{k=1}^{t} \alpha_k h_k(x_m)\right)$, then $D_{t+1}(l) < D_{t+1}(m)$.

Let obj_t is fixed. We want to find α_{t+1} such that with a fixed h_{t+1}, the objective function is minimized.

Now,

$$\frac{obj_{t+1}}{obj_t} = \sum_{i:y_i = h_{t+1(xi)}} D_{t+1}(i) e^{-\alpha_{t+1}} + \sum_{i:y_i \neq h_{t+1(xi)}} D_{t+1}(i) e^{\alpha_{t+1}}$$

$$\frac{obj_{t+1}}{obj_t} = (1 - \epsilon_{t+1}) e^{-\alpha_{t+1}} + \epsilon_{t+1} e^{\alpha_{t+1}} \tag{6}$$

where $\epsilon_{t+1} = \sum_{y_i \neq h_{t+1}(x_i)} D_{(t+1)}(i)$ is the error rate of h_{t+1} on the weighted samples.

Taking the derivative of (6) and equating to zero (for finding the optimal α_{t+1}),

$$(1 - \epsilon_{t+1}) e^{-\alpha_{t+1}} = \epsilon_{t+1} e^{\alpha_{t+1}}$$

Therefore,

$$\alpha_{t+1} = \frac{1}{2} \log\left(\frac{1 - \epsilon_{t+1}}{\epsilon_{t+1}}\right) \tag{7}$$

Sub: (7) into (6) ,

$$\frac{obj_{t+1}}{obj_t} = 2\sqrt{(1 - \epsilon_{t+1})\epsilon_{t+1}} \leq 1$$

[The maximum value of $\sqrt{(1 - \epsilon_{t+1})\epsilon_{t+1}} = \sqrt{.25}$]

Therefore $obj_{t+1} \leq obj_t$. Thus at each step α_t is chosen in such a way that the error rate of $H(x)$ is minimized.

3 Updating the weight of data points

Using (5),

$$\frac{D_{t+1}(i)}{D_t(i)} = \frac{e^{-y_i \sum_{i=1}^{t} \alpha_t h_t(x_i)} \left(\sum_{i=1}^{N} e^{-y_i \sum_{k=1}^{t-1} \alpha_k h_k(x_i)} \right)}{\left(\sum_{i=1}^{N} e^{-y_i \sum_{k=1}^{t} \alpha_k h_k(x_i)} \right) e^{-y_i \sum_{i=1}^{t-1} \alpha_t h_t(x_i)}} \tag{8}$$

Hence,

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

Thus,

$$D_{t+1}(i) = \frac{D_t(i) e^{-y_i \alpha_t h_t(x_i)}}{Z_t} \tag{9}$$

where $Z_t = \sum_{i=1}^{N} D_t(i) e^{-y_i \alpha_t h_t(x_i)}$, is a normalization factor such that D_{t+1} will be a distribution.

From (4) it is clear that $Z_t = \dfrac{obj_t}{obj_{t-1}}$ and thus error is minimized by minimizing Z_t.

3.1 AdaBoost Algorithm

The weak learner h_t is modeled using a sample D_t, which is created in the following way:

- Repeat the following steps N times:
  - Choose a number p from (0,1). Select all the data points from D whose weight is greater than p and randomly choose a data point from that subset. The chosen point becomes a member of D_t.

The AdaBoost algorithm's pseudocode is given below:

Algorithm 1 AdaBoost algorithm

Input N examples $D = \{(x_1, y_1), ..., (x_N, y_N)\},\ x_i \in \mathcal{X} \subseteq R^n, y_i \in \{-1, 1\}$

T: number of hypotheses in the ensemble

Initialize $D_1(i) = 1/N, i = 1, 2, \ldots N$

1: **for** $t = 1$ to T **do**

2: Create a sample D_t by sampling D with replacement by taking into consideration the data points weights (as given in subsection 3.1)

3: Train a Weak Learner using D_t and obtain the hypothesis $h_t : \mathcal{X} \rightarrow \{1, -1\}$

4: Computed weighted error $\epsilon_t = \sum_{i=1}^{N} D_t(i)_{\{h_t(x_i) \neq y_i\}}$

5: If $\epsilon_t \leq 0.5$ continue else go to step (2)

6: Compute hypothesis weight $\alpha_t = \frac{1}{2} \log \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$

7: If $t < T$, update the data points weights:

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

8: **end for**

9: Final vote $H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$ is the weighted sum.