On AdaBoost

AdaBoost is a committee machine, which consists of a set of weak classifiers $h_t(x_i) \in \{+, -\}$, $t = 1, \ldots, T$ given a training set $\{(x_i, y_i), i = 1, \ldots, n\}$. The final strong classifier is a perceptron based on the weak classifiers,

$$\hat{y}_i = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x_i)\right), \tag{1}$$

where α_t can be interpreted as the weight of the vote of weak classifier t. The loss is in the exponential form:

$$\ell(\alpha) = \sum_{i=1}^{n} \exp\left(-y_i \sum_{t=1}^{T} \alpha_t h_t(x_i)\right). \tag{2}$$

When training the strong classifier, we sequentially add members to the committee. Suppose the current committee has m classifiers, and we want to add a new member h_{new}. After adding a new member, the loss function becomes

$$\ell(\alpha_{new}, h_{new}) = \sum_{i=1}^{n} \exp\left(-y_i \sum_{t=1}^{m} \alpha_t h_t(x_i) + \alpha_{new} h_{new}(x_i)\right), \tag{3}$$

where we assume the current members and their weights of votes are fixed. Take derivative

$$\frac{\partial \ell}{\partial \alpha_{new}} = \sum_{i=1}^{n} \exp\left(-y_i (\sum_{t=1}^{m} \alpha_t h_t(x_i) + \alpha_{new} h_{new}(x_i))\right) \cdot (-y_i h_{new}(x_i)). \tag{4}$$

When choosing ***a new member***, consider the current committee without adding a new member, $\alpha_{new} = 0$, then the above gradient can be written as

$$\left.\frac{\partial \ell}{\partial \alpha_{new}}\right|_{\alpha_{new}=0} = \sum_{i=1}^{n} \exp\left(-y_i (\sum_{t=1}^{m} \alpha_t h_t(x_i))\right) \cdot (-y_i h_{new}(x_i)) \tag{5}$$

$$= -\sum_{i=1}^{n} w_i y_i h_{new}(x_i) \tag{6}$$

where

$$w_i = \exp\left(-y_i \sum_{t=1}^{m} \alpha_t h_t(x_i)\right). \tag{7}$$

Then normalize $w_i \leftarrow w_i / \sum_{i=1}^n w_i$ to make it a distribution. Observe, this distribution focuses on those examples that are not well classified by the current committee. Thus, we want to choose a weak classifier h_{new} by maximizing $\sum_{i=1}^n w_i y_i h_{new}(x_i)$

$$h_{new} = \arg\max_{h_{new}} \sum_{i=1}^{n} w_i y_i h_{new}(x_i) \tag{8}$$

for the steepest drop in loss. Note, this implies that we want to choose the new weak classifier based on current distribution of the data (x_i, y_i, w_i) where the weights w_i keep changing.

To ***determine the voting weight*** α_{new}, we set the derivative to 0,

$$\frac{\partial \ell}{\partial \alpha_{new}} = 0, \tag{9}$$

$$\sum_{i=1}^{n} w_i \exp(-y_i \alpha_{new} h_{new}(x_i)) \cdot y_i h_{new}(x_i) = 0, \tag{10}$$

$$\sum_{i \in correct} w_i \exp(-\alpha_{new}) = \sum_{i \in wrong} w_i \exp(\alpha_{new}), \tag{11}$$

$$\sum_{i \in correct} w_i = \sum_{i \in wrong} w_i \exp(2\alpha_{new}), \tag{12}$$

$$\alpha_{new} = \frac{1}{2} \log \left(\frac{\sum_{i \in correct} w_i}{\sum_{i \in wrong} w_i} \right). \tag{13}$$

Note, in (11) we used the fact that if $i \in correct$, then $y_i h_{new}(x_i) = 1$. If we define the error rate as

$$\epsilon = \frac{\sum_{i \in wrong} w_i}{\sum_i w_i}, \tag{14}$$

then α_{new} is

$$\alpha_{new} = \frac{1}{2} \log \frac{1-\epsilon}{\epsilon}. \tag{15}$$

Finally, we assemble the derived steps in Algorithm 1.

Algorithm 1 AdaBoost learning

Input:

- Samples $\{x_i : i = 1, \ldots, n\}$,
- Desired outputs $\{y_i \in \{-1, +1\} : i = 1, \ldots, n\}$,
- Weak learners $H = \{h : x \to \{-1, +1\}\}$,
- Steps T.

Output:

- Ensemble $\{h_t : t = 1, \ldots, T\}$,
- Voting weights $\{\alpha_t : t = 1, \ldots, T\}$.

Steps:

1: Let $\{w_{1,i} = \frac{1}{n} : i = 1, \ldots, n\}$.

2: **for** t in $1, \ldots, T$ **do**

3: Compute weighted error:

$$\epsilon_t(h) = \sum_{i=1}^{n} w_{t,i} 1(h(x_i) \neq y_i) \ \ \forall h \in H.$$

4: Choose weak learner:

$$h_t = \arg\min_{h \in H} \epsilon_t(h).$$

5: Assign voting weight:

$$\alpha_t = \frac{1}{2} \log \frac{1 - \epsilon_t(h_t)}{\epsilon_t(h_t)}.$$

6: Update weights:

$$w_{i,t+1} = w_{i,t} \exp\left(-y_i \alpha_t h_t(x_i)\right) \ \ \forall i \in 1, \ldots, n.$$

7: Renormalize weights:

$$w_{i,t+1} = \frac{w_{i,t+1}}{\sum_{i=1}^{n} w_{i,t+1}}.$$

8: **end for**