Algorithmic description

The Mathematically the classification of Adaboosting is defined as:

$$\boldsymbol{H(x) = sign\left(\sum_{t=1}^{T} a_t h_t(x)\right)}$$

Breaking down this formula we fine that h_t is the weak classification of attributes which separating them into two difference groups $X \to \{-1, +1\}$. The goal of the selected weak hypothesis is to produce a split that give the lowest error between groups. This hypothesis can be calculated in a range of ways however is typically done using decision trees.

Where total error is the sum of weights within a stun that incorrectly classifies a sample meaning it can either be between one and zero, where a total error of zero would indicate a perfect stump. For example, if the adabooster had produced the following stump from a sample size of 10 with all the same weights. For $D_1$the first value of distribution is given as 1/m where m is the number of attributes y_m within the selected data set, meaning within this case would be 1/10.

Active Smoker	Correct	Incorrect
Cancer	4	1
No Cancer	3	2

As we can observe the total error calculates to 3/10 and therefor this stump would have a say of would be 0.42. The amount of say is also logarithmically adjusted meaning that the closer you get to zero or one the more drastic difference there is between each interval. Gini Index is calculated for feature stump and the stump with the lowest index is selected to be added to the forest.

Next $\boldsymbol{a_t}$ is classifies as the "amount of say" at Interval t= 1,…,T. The Amount of say defines how well a stump defines a classifying sample and provide a larger voice to stumps that more accurately separate data compared to other stumps. This is calculated by the formula

$$a_t \;=\; \frac{1}{2} ln\left(\frac{1 - Total\ Error}{Total\ Error}\right)$$

The sample weights are then updated after each iteration as the adabooster trains weak learners using the new distribution D_{t+1}. This is calculated by multiplying the current sample weight D_t with the e to the power of the inverted amount of say, normalised by the normalization factor Z_t . This calculation takes the form of for the new sample weight that were correctly classified.

$$D_{t+1} = \frac{D_t e^{(-a_t)}}{z_t}$$

Understanding of method

Adaboosting also knowing as adaptive boosting is a supervised machine learning algorithm that iteratively creates and classifies weak learners In addition to iteratively corrects mistakes and improves accuracy of weak classifiers by combining weak learners in order to create a strong classifier.

As adaboosters typically only use very weak learners such as 1-level stump decision trees, means that the algorithm is less likely to be affected by overfitting as stumps are very flexible and are more resilient to noise. However, if within the adabooster is the decision tree size was increase then overfitting would more become more likely to occur.

However, on the flip side adabooster can become highly effected by outliers as they can be given massive weighting when miss classified causing it to have more effect on the end result then it would on other types of learners.

In addition, there is also two ways the sample weights can be re-calculated after a stump is selected the first being weighted gini indexes which is a variable that is factored into making the following stumps. However, the method that will be used within this implementation is creating a new collection of samples after each iteration by normalising the new sample weights and then randomly selecting values which correspond the normalise weighted values. meaning that the classes with larger weight have more likely to be included multiple times in the new collection which ultimately makes there collective weight larger.