INDIAN INSTITUTE OF TECHNOLOGY, KHARAGPUR

CS60020 : Foundations of Algorithm Design and Machine Learning

- Prof. Aritra Hazra

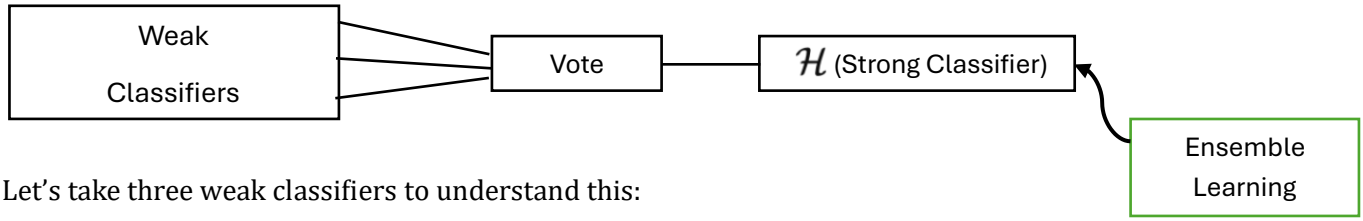
TOPIC: ADABOOST

Scribe: 8th April, 2025 (11:00 AM – 11:55 AM)

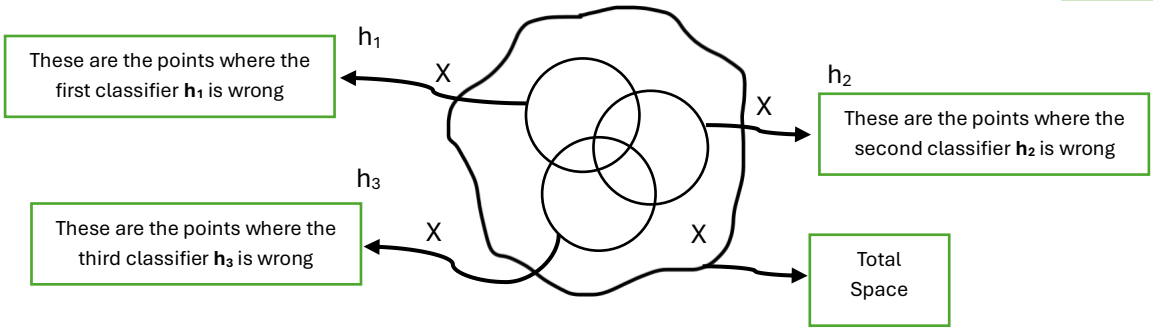
Submitted by: Utkarsh Attela (24BM6JP59)

ADABOOST

AdaBoost, which stands for "**Ada**ptive **Boost**ing", is an ensemble learning algorithm that uses the boosting paradigm. The core idea is to combine multiple weak learners into a single strong learner by training each new learner to focus more on the examples that previous learners got wrong. It was formulated by Yoav Freund and Robert Schapire in 1995. They also won the 2003 Gödel Prize for their work.



Let's take three weak classifiers to understand this:



We assume that we are given a training set **D** and a pool of hypothesis functions **H** from which we are to pick T hypotheses in order to form an $\mathcal{H}$ ensemble.

$\mathcal{H}$ = sign ($\alpha_1 h_1 + \alpha_2 h_2 + \alpha_3 h_3$)

Where h_t are weak classifiers and α_t are the corresponding weights. ($\sum \alpha_t = 1$)

More formally, $\mathcal{H}$ = sign($\alpha_1 h_1 + \alpha_2 h_2 + \alpha_3 h_3$ + ---------- + $\alpha_T h_T$) = sign($\sum_{t=1}^{T} \alpha_t h_t$)

That is, $\mathcal{H}$ uses a linear combination of the decisions of each of the h_i hypotheses in the ensemble. The AdaBoost algorithm sequentially chooses h_i from $\mathcal{H}$ and assigns this hypothesis a weight α_i. We let H_t be the classifier formed by the first t hypotheses. That is,

$\mathcal{H}$ = sign ($\mathcal{H}_{t-1}$ + $\alpha_t h_t$)

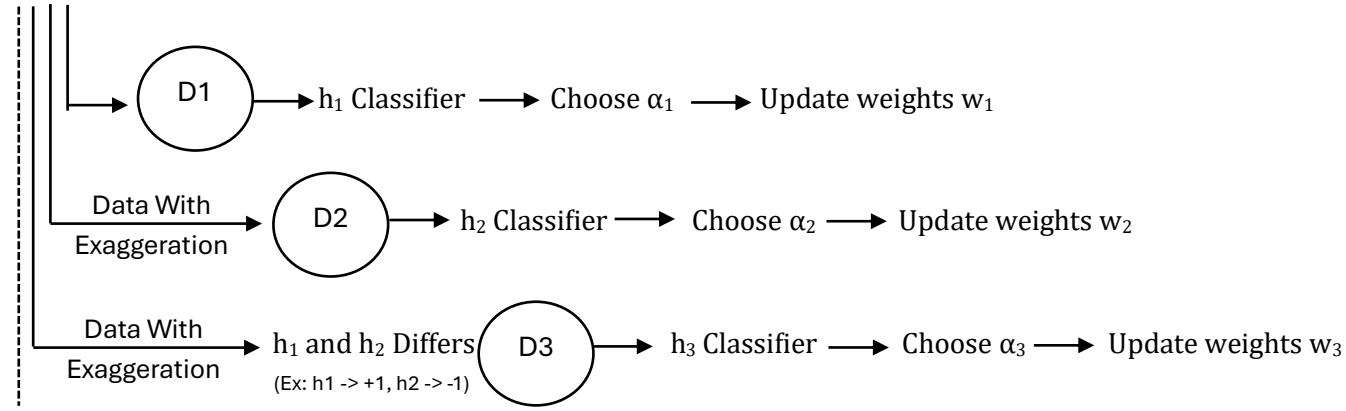
Let's say we have N data points. Since it is a supervised learning algorithm, we have labelled data **D**: <x1, y1>, <x2, y2>-------, <x_N, y_N>. Initially the weights of all these data points are same $\left(w_i^{(1)} = \frac{1}{N}\right)$. Now, we create a first sample of dataset **D1** from **D** and with respect to that we get our first classifier $\mathbf{h_1}$. The error of the learner is: $\varepsilon_t = \sum_{misclassified} w_i^{(t)}$

Now, since $\mathbf{h_1}$ is a weak classifier, some of the data points will not be classified correctly. So, we take the same Data **D** and do data exaggeration such that we try to increase the probability of the wrongly classified points and decrease the probability of correct points. With this new probabilities for the data points, we will create another sample of dataset which is **D2** and this dataset will give us another classifier $\mathbf{h_2}$.

Now, again we will do data exaggeration. Here the exaggeration will mean that $\mathbf{h_1}$ and $\mathbf{h_2}$ differs i.e. one of the classify a data point correctly and another classify is wrongly. Because, if $\mathbf{h_1}$ and $\mathbf{h_2}$ differs, the choice of $\mathbf{h_3}$ will be important to classify the data point. Again, we will increase or decrease the probability of those data points where $\mathbf{h_1}$ and $\mathbf{h_2}$ differs. Again, we will create a new sample dataset **D3** and we get another classifier $\mathbf{h_3}$. And we will keep on doing so on till **M** sample datasets or till we achieve required results.

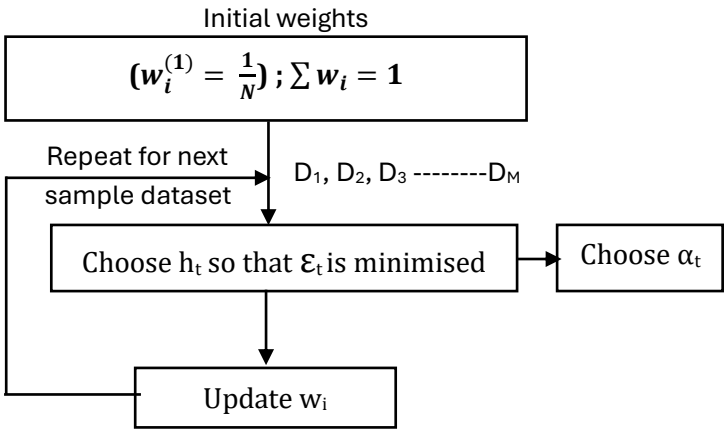
The flow of this process is:

Data **D**: <x1, y1>, <x2, y2>-------, <x_N, y_N>.



And so on.... Till **M** sample datasets
or till achieved required results

FLOW CHART



Derivation of α_t

Our objective is to minimise the exponential loss function. We start with defining the exponential loss function that is to be minimised:

$$E_t = \sum_i w_i^{(t)} e^{-\alpha_t y_i h_t(x_i)} \longrightarrow \text{Equation 1}$$

Note that: $y_i h_t(x_i) = +1$ if the point is classified correctly and $y_i h_t(x_i) = -1$ if the point is classified incorrectly. Error is boosted if values of $h_t(x_i)$ and y_i differs. (i.e. if $h_t(x_i) = +1$ and $y_i = -1$ or vice-versa)

Now, split the Sum into correct and incorrect classifications.

$$E_t = \sum_{i \in \text{wrong}} w_i^{(t)} e^{\alpha_t} + \sum_{i \in \text{correct}} w_i^{(t)} e^{-\alpha_t}$$

Let $\varepsilon_t = \sum_{wrong} w_i^{(t)}$ and $1 - \varepsilon_t = \sum_{correct} w_i^{(t)}$, Thus

$$E_t = \varepsilon_t e^{\alpha_t} + (1 - \varepsilon_t) e^{-\alpha_t}$$

Now, to minimise the loss, we need to find the optimal α_t. So, differentiate E_t with respect to α_t and set to 0.

$$\frac{dE_t}{d\alpha_t} = \varepsilon_t e^{\alpha_t} - (1 - \varepsilon_t) e^{-\alpha_t} = 0$$

$$\Rightarrow \varepsilon_t e^{\alpha_t} = (1 - \varepsilon_t) e^{-\alpha_t}$$

Multiply both sides by $e^{\alpha t}$:

$$\varepsilon_t e^{2\alpha_t} = 1 - \varepsilon_t$$

$$e^{2\alpha_t} = \frac{1 - \varepsilon_t}{\varepsilon_t}$$

Taking log on both sides:

$$2\alpha_t = \ln\left(\frac{1 - \varepsilon_t}{\varepsilon_t}\right)$$

$$\boxed{\alpha_t = \frac{1}{2} \ln\left(\frac{1 - \varepsilon_t}{\varepsilon_t}\right)} \longrightarrow \text{Equation 2}$$

Weight Update Derivation

From Adaboost weight update rule, we know that,

$$w_i^{(t+1)} \longleftarrow w_i^{(t)} \cdot e^{-\alpha_t h_t(x_i) y_i}$$

To normalize, we divide by a constant Z:

$$w_i^{(t+1)} \longleftarrow \frac{w_i^{(t)} \cdot e^{-\alpha_t h_t(x_i) y_i}}{Z}$$

We will do case-wise update when a class is wrongly classified or correctly classified.

When a point is wrongly classified: then and $y_i h_t(x_i) = -1$

$$w_i^{(t+1)} = \frac{w_i^{(t)} \cdot e^{\alpha_t}}{Z}$$

When a point is correctly classified: then and $y_i h_t(x_i) = +1$

$$w_i^{(t+1)} = \frac{w_i^{(t)} \cdot e^{-\alpha_t}}{Z}$$

Substitute the α_t value that we derived in equation 2:

For wrong points:

$$e^{\alpha_t} = \sqrt{\frac{1 - \varepsilon_t}{\varepsilon_t}} \Rightarrow w_i^{(t+1)} = \frac{w_i^{(t)}}{Z} \cdot \sqrt{\frac{1 - \varepsilon_t}{\varepsilon_t}} \longrightarrow \text{Equation 3}$$

For correct points:

$$e^{-\alpha_t} = \sqrt{\frac{\varepsilon_t}{1-\varepsilon_t}} \Rightarrow w_i^{(t+1)} = \frac{w_i^{(t)}}{Z} \cdot \sqrt{\frac{\varepsilon_t}{1-\varepsilon_t}} \longrightarrow \text{Equation 4}$$

Since total weight must be 1 (sum over wrong points + sum over correct points) i.e.,

$$\sum_i w_i^{(t+1)} = 1$$

Breaking into sum over wrong (WR) and correct (COR) points:

$$\sum_{\text{WR}} \frac{w_i^{(t)}}{Z} \cdot \sqrt{\frac{1-\varepsilon_t}{\varepsilon_t}} + \sum_{\text{COR}} \frac{w_i^{(t)}}{Z} \cdot \sqrt{\frac{\varepsilon_t}{1-\varepsilon_t}} = 1$$

ε_t $\qquad$ $1-\varepsilon_t$

This simplifies to:

$$\boxed{Z = 2\sqrt{\varepsilon_t(1-\varepsilon_t)}}$$

Substituting Z in equation 3 and 4:

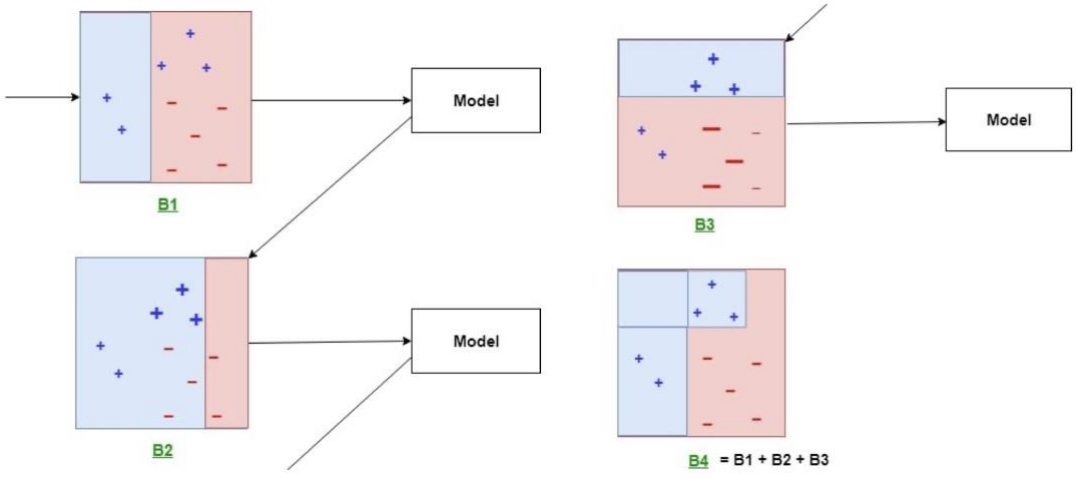
$$w_i^{(t+1)} = \begin{cases} \frac{w_i^{(t)}}{2\varepsilon_t}, & \text{if wrong point} \\ \frac{w_i^{(t)}}{2(1-\varepsilon_t)}, & \text{if correct point} \end{cases}$$

Example for AdaBoost:

(Reference: Madha Engineering College. (2024). *MCA 2nd year notes*. https://www.madhaengineeringcollege.com/wp-content/uploads/2024/01/MCA-2nd-year-notes.pdf)

Algorithm:

1. Initialise the dataset and assign equal weight to each of the data point.
2. Provide this as input to the model and identify the wrongly classified data points.
3. Increase the weight of the wrongly classified data points.
4. if (got required results)
   Goto step 5
   else
   Goto step 2
5. End



Explanation:

The above diagram explains the AdaBoost algorithm in a very simple way. Let's try to understand it in a stepwise process:

- **B1** consists of 10 data points which consist of two types namely plus(+) and minus(-) and 5 of which are plus(+) and the other 5 are minus(-) and each one has been assigned equal weight initially. The first model tries to classify the data points and generates a vertical separator line but it wrongly classifies 3 plus(+) as minus(-).

- **B2** consists of the 10 data points from the previous model in which the 3 wrongly classified plus(+) are weighted more so that the current model tries more to classify these pluses(+) correctly. This model generates a vertical separator line that correctly classifies the previously wrongly classified pluses(+) but in this attempt, it wrongly classifies three minuses(-).

- **B3** consists of the 10 data points from the previous model in which the 3 wrongly classified minus(-) are weighted more so that the current model tries more to classify these minuses(-) correctly. This model generates a horizontal separator line that correctly classifies the previously wrongly classified minuses(-).

- **B4** combines together B1, B2, and B3 in order to build a strong prediction model which is much better than any individual model used.

Making Predictions with AdaBoost:

Predictions are made by calculating the weighted average of the weak classifiers. For a new input instance, each weak learner calculates a predicted value as either +1.0 or -1.0. The predicted values are weighted by each weak learners stage value. The prediction for the ensemble model is taken as a the sum of the weighted predictions. If the sum is positive, then the first class is predicted, if negative the second class is predicted.

For example, 5 weak classifiers may predict the values 1.0, 1.0, -1.0, 1.0, -1.0. From a majority vote, it looks like the model will predict a value of 1.0 or the first class. These same 5 weak classifiers may have the stage values 0.2, 0.5, 0.8, 0.2 and 0.9 respectively. Calculating the weighted sum of these predictions results in an output of -0.8, which would be an ensemble prediction of -1.0 or the second class.