ADABOOST

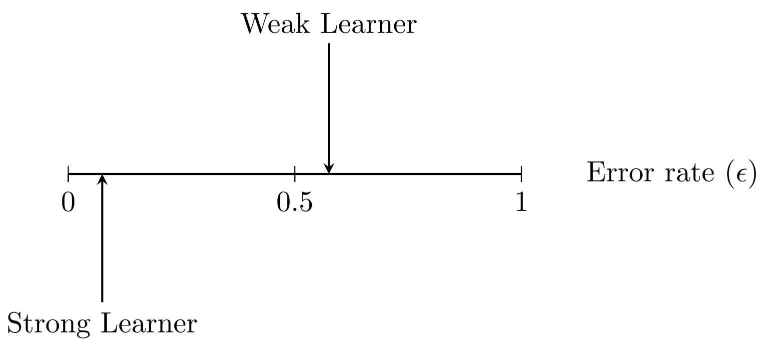
Boosting is a powerful technique to combine several 'weak' classifiers into a 'strong' classifier. *AdaBoost*, short for 'Adaptive Boosting' is one of the most frequently used boosting algorithms.

1 Weak and Strong Learners

A weak learner is a classification algorithm which is only slightly better than random guessing. On the other hand, a strong learner is one which almost always provides the true classification. Given training data of the form $(x_1, y_1),\ (x_2, y_2),\ \ldots, (x_n, y_n)$ where $y_i \in \{+1, -1\}\ \forall\ x_i \in \mathbf{X}$ [1] and a learner h the error ϵ is defined as,

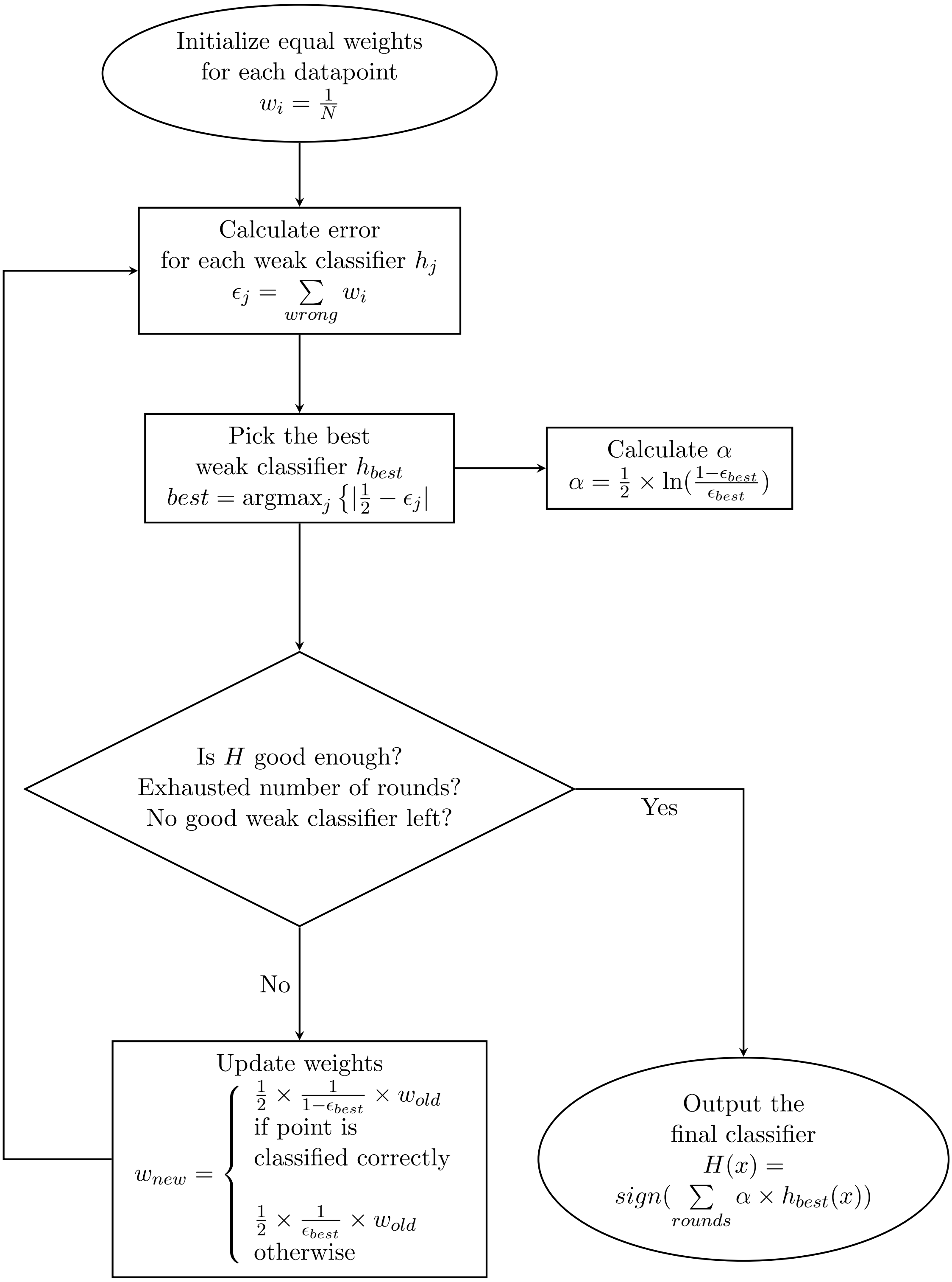
$$\epsilon = \frac{1}{N} \times \begin{cases} 0\ if\ y_i = h(x_i) \\ 1\ if\ y_i \neq h(x_i) \end{cases}$$

The error rate ϵ is related to the strength of a learner as shown below:

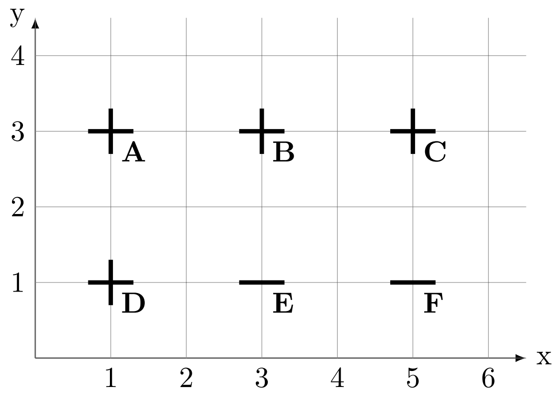


2 Algorithm

Given training data of the form [1] in the preceding section and weak learners $h_1,\ h_2,\ \ldots\ h_m$, the goal is to iteratively come up with a strong learner H. AdaBoost's algorithmic flowchart is as follows:


Initialize equal weights
for each datapoint
$w_i = \frac{1}{N}$
Calculate error
for each weak classifier h_j
$\epsilon_j = \sum_{wrong} w_i$
Pick the best
weak classifier h_{best}
$best = \text{argmax}_j \left\{ |\frac{1}{2} - \epsilon_j| \right.$
Calculate α
$\alpha = \frac{1}{2} \times \ln(\frac{1-\epsilon_{best}}{\epsilon_{best}})$
Is H good enough?
Exhausted number of rounds?
No good weak classifier left?
Yes
No
Update weights
$w_{new} = \begin{cases} \frac{1}{2} \times \frac{1}{1-\epsilon_{best}} \times w_{old} & \text{if point is classified correctly} \\ \frac{1}{2} \times \frac{1}{\epsilon_{best}} \times w_{old} & \text{otherwise} \end{cases}$
Output the
final classifier
$H(x) =$
$sign(\sum_{rounds} \alpha \times h_{best}(x))$

3 Example



Following weak classifiers are considered. Their error points are shown below:

Weak Classifier	Error Points
$x >= 2$	**A, D, E, F**
$x < 2$	**B, C**
$x >= 4$	**A, B, D, F**
$x < 4$	**C, E**
$x >= 6$	**A, B, C, D**
$x < 6$	**E, F**
$y >= 2$	**D**
$y < 2$	**A, B, C, E, F**
$y >= 4$	**A, B, C, D**
$y < 4$	**E, F**

3.1 Iteration 1

At the start, each point has an equal weight of 1/6 since there are 6 points.

Point	**A**	**B**	**C**	**D**	**E**	**F**
Error	1/6	1/6	1/6	1/6	1/6	1/6

The error points and their corresponding weights determine the error rate of a particular weak classifier. The error rate for each classifier is shown below.

Weak Classifier	Error Rate
$x >= 2$	2/3
$x < 2$	1/3
$x >= 4$	2/3
$x < 4$	1/3
$x >= 6$	2/3
$x < 6$	1/3
$y >= 2$	1/6
$y < 2$	5/6
$y >= 4$	2/3
$y < 4$	1/3

$y >= 2$ is chosen as the best weak classifier since it has the minimum error rate. α_1 is calculated as follows:

$$\begin{aligned} \alpha_1 &= \frac{1}{2} \times ln(\frac{1 - 1/6}{1/6}) \\ &= \frac{1}{2} \times ln(5) \end{aligned}$$

3.2 Iteration 2

The weights are now updated as per flow chart above,

Point	**A**	**B**	**C**	**D**	**E**	**F**
Error	1/10	1/10	1/10	1/2	1/10	1/10

An interesting fact is that the sum of updated weights of the points misclassified by the chosen classifier $y >= 2$ (**D**) is equal to 1/2 and so the sum of updated weights of the points correctly classified (**A, B, C, E, F**) is also 1/2.

$$\begin{aligned} Error(\mathbf{D}) &= 1/2 \\ \sum_{P \in \{\mathbf{A},\ \mathbf{B},\ \mathbf{C},\ \mathbf{E},\ \mathbf{F}\}} Error(P) &= 1/10 + 1/10 + 1/10 + 1/10 + 1/10 \\ &= 1/2 \end{aligned}$$

Error rates are updated as follows:

Weak Classifier	Error Rate
$x >= 2$	4/5
$x < 2$	1/5
$x >= 4$	4/5
$x < 4$	1/5
$x >= 6$	4/5
$x < 6$	1/5
$y >= 2$	1/2
$y < 2$	1/2
$y >= 4$	4/5
$y < 4$	1/5

$x < 2$ is chosen as the best weak classifier since it has the minimum error rate (for breaking tie, classifier occuring first in the above list is chosen). α_2 is calculated as follows,

$$\begin{aligned}\alpha_2 &= \frac{1}{2} \times ln(\frac{1 - 1/5}{1/5}) \\ &= \frac{1}{2} \times ln(4)\end{aligned}$$

3.3 Iteration 3

Weights are again updated as follows:

Point	**A**	**B**	**C**	**D**	**E**	**F**
Error	1/16	1/4	1/4	5/16	1/16	1/16

Given that (**B, C**) are misclassified by $x < 2$ and points (**A, D, E, F**) are correctly classified; notice again that,

$$\begin{aligned}\sum_{P \in \{\mathbf{B},\ \mathbf{C}\}} Error(P) &= 1/4 + 1/4 \\ &= 1/2 \\ \sum_{P \in \{\mathbf{A},\ \mathbf{D},\ \mathbf{E},\ \mathbf{F}\}} Error(P) &= 1/16 + 5/16 + 1/16 + 1/16 \\ &= 1/2\end{aligned}$$

Updated error rates look like this,

Weak Classifier	Error Rate
$x >= 2$	1/2
$x < 2$	1/2
$x >= 4$	11/16
$x < 4$	5/16
$x >= 6$	7/8
$x < 6$	1/8
$y >= 2$	5/16
$y < 2$	11/16
$y >= 4$	7/8
$y < 4$	1/8

$x < 6$ is chosen as the best weak classifier since it has the minimum error rate. α_3 is calculated as follows,

$$\begin{aligned} \alpha_3 &= \frac{1}{2} \times ln(\frac{1 - 1/8}{1/8}) \\ &= \frac{1}{2} \times ln(7) \end{aligned}$$

After three iterations, the resultant strong classifier is

$$\begin{aligned} H =& sign(\frac{1}{2} \times ln(5) \times (y >= 2) \\ &+ \frac{1}{2} \times ln(4) \times (x < 2) \\ &+ \frac{1}{2} \times ln(7) \times (x < 6)) \end{aligned}$$

The classification of H for each point is as follows,

Point	Calculation	Classification
A	$sign(1/2 \times ln(5 \times 4 \times 7))$	$+$
B	$sign(1/2 \times ln((5 \times 7)/4))$	$+$
C	$sign(1/2 \times ln((5 \times 7)/4))$	$+$
D	$sign(1/2 \times ln((4 \times 7)/5))$	$+$
E	$sign(1/2 \times ln(7/(5 \times 4)))$	$-$
F	$sign(1/2 \times ln(7/(5 \times 4)))$	$-$

It is evident the H classifies each point correctly after 3 iterations. Further iterations are hence terminated.

4 Mathematics

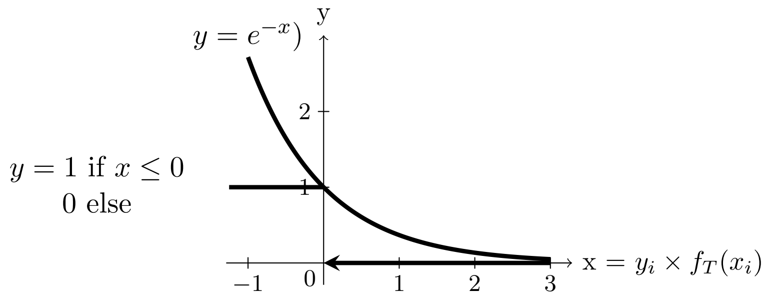
4.1 Upper bound on training error

4.1.1 Step 1

$$Error(H_T) = \frac{1}{N} \times \sum_i \begin{cases} 1 \text{ if } y_i \neq H_T(x_i) \\ 0 \text{ else} \end{cases}$$

Let $H_T(x_i) = sign(f_T(x_i))$ where $f_T(x_i) = \sum_t \alpha_t \times h_t(x_i)$

$$Error(H_T) = \frac{1}{N} \times \sum_i \begin{cases} 1 \text{ if } y_i \times f_T(x_i) <= 0 \\ 0 \text{ else} \end{cases}$$



$$Error(H_T) \leq \frac{1}{N} \times \sum_i e^{-y_i \times f_T(x_i)} \tag{1}$$

4.1.2 Step 2

$$w_{t+1}(i) = \frac{1}{2} \times w_t(i) \begin{cases} \frac{1}{1-\epsilon_t} & \text{if } y_i = h_t(x_i) \\ \frac{1}{\epsilon_t} & \text{else} \end{cases}$$

$$w_{t+1}(i) = \frac{w_t(i)}{2 \times \sqrt{\epsilon_t \times (1-\epsilon_t)}} \begin{cases} \sqrt{\frac{\epsilon_t}{1-\epsilon_t}} & \text{if } y_i = h_t(x_i) \\ \sqrt{\frac{1-\epsilon_t}{\epsilon_t}} & \text{else} \end{cases}$$

$$w_{t+1}(i) = \frac{w_t(i)}{2 \times \sqrt{\epsilon_t \times (1-\epsilon_t)}} \times e^{-\alpha_t \times y_i \times h_t(x_i)}$$

$$w_{t+1}(i) = \frac{1}{2 \times \prod_t \sqrt{\epsilon_t \times (1-\epsilon_t)}} \times e^{-y_i \times f_t(x_i)} \times \frac{1}{N}$$

Transposing and replacing t by T,

$$\frac{1}{N} \times e^{-y_i \times f_T(x_i)} = 2 \times w_{T+1}(i) \times \prod_t \sqrt{\epsilon_t \times (1 - \epsilon_t)}$$

Summing over i,

$$\sum_i \frac{1}{N} \times e^{-y_i \times f_T(x_i)} = 2 \times \prod_t \sqrt{\epsilon_t \times (1 - \epsilon_t)} \times \sum_i w_{T+1}(i)$$

Since sum of weights is one at every round,

$$\frac{1}{N} \times \sum_i e^{-y_i \times f_T(x_i)} = 2 \times \prod_t \sqrt{\epsilon_t \times (1 - \epsilon_t)} \tag{2}$$

4.1.3 Step 3

Using (1) and (2)

$$Error(H_T) \leq 2 \times \prod_t \sqrt{\epsilon_t \times (1 - \epsilon_t)}$$

Using $\gamma_t = (1/2 - \epsilon_t)$

$$Error(H_T) \leq 2 \times \prod_t \sqrt{1/4 - \gamma_t^2}$$

$$Error(H_T) \leq \prod_t \sqrt{1 - 4\gamma_t^2}$$

Since $e^x \geq (1 + x)$ for all real values of x,

$$Error(H_T) \leq \prod_t e^{-2\gamma_t^2}$$

$$Error(H_T) \leq e^{-2\sum_t \gamma_t^2}$$

Since $\sum_t \gamma_t^2$ monotonically increases with each iteration, $Error(H_t)$ monotonically decreases exponentially.