MACHINE INTELLIGENCE

UNIT - 3

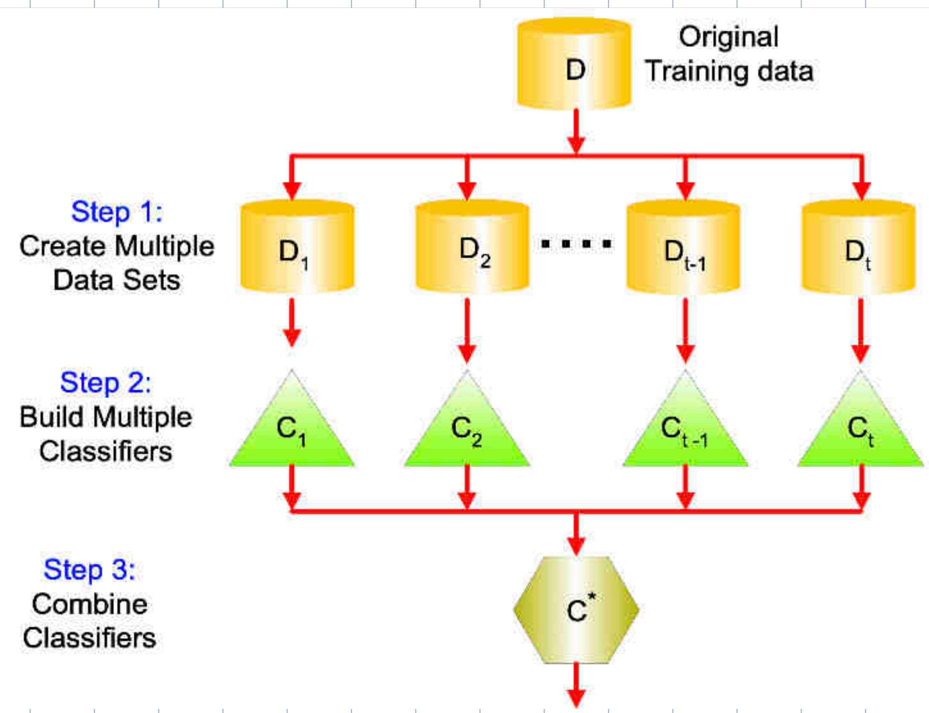
Ensemble Models

feedback/corrections: vibha@pesu.pes.edu

VIBHA MASTI

ENSEMBLE MODELS

- Combines predictions from multiple ML algorithms together (independent)
- Weak learners used
- Weak learners with accuracy better than chance (50%)
- Multiple models using
  - Diff algos (decision stump, perceptron)
  - Diff hyperparameters
  - Diff subsets of train (resampling)
  - Diff features of train (stumps)
- WLs must be independent, errors must be random
- Training data adapted (reweighted or resampled)



Combining Predictions

- Stats – mean, mode (majority voting)
- Weighted sum (learners assigned weights)
- Combined: low bias, low variance
- Problem of overfitting not there with WLs

Bias & Variance

- Given set of independent experiments $Z_1, \ldots, Z_n$ each with a variance of σ^2
- Variance of mean of observations = $\dfrac{\sigma^2}{n}$
- Averaging reduces variance
- Low variance: similar instances in same class (reliable)
- Low bias: instances get labelled to their true class (valid)

Variance of EL Models

- Decision trees: high variance
- Ensemble learning: low variance

Hyperparameters

- No. of WLs
- Sampling method

Multiple Weak Learners

- Assume n weak learners in binary classification problem

P(correct)

- Assume all WLs have uniform accuracy A and predict same class for given instance

P(wrong)

- Error rate of single predictor $E = 1 - A$

- Error rate of n such predictors (chance of all n getting it wrong)

$$E = (1-A)^n$$

- Chance of getting at least one prediction wrong

$$\binom{n}{0}(1-A)^n + \binom{n}{1} A^1 (1-A)^{n-1} + \ldots + \binom{n}{n-1} A^{n-1}(1-A)^1$$

$$= \sum_{k=0}^{n-1} \binom{n}{k} A^k (1-A)^{n-k}$$

- chance of at least one getting it right

$$C = 1 - (1-A)^n$$

- Realistically, assume n_1 learners predict class 1 and n_2 predict class 2 $(n_1 > n_2)$

- Probability of class actually being class 1 (n_1 correct and n_2 wrong)

$$\binom{n}{n_2} A^{n-n_2} (1-A)^{n_2} = \binom{n}{n_1} A^{n_1} (1-A)^{n-n_1}$$

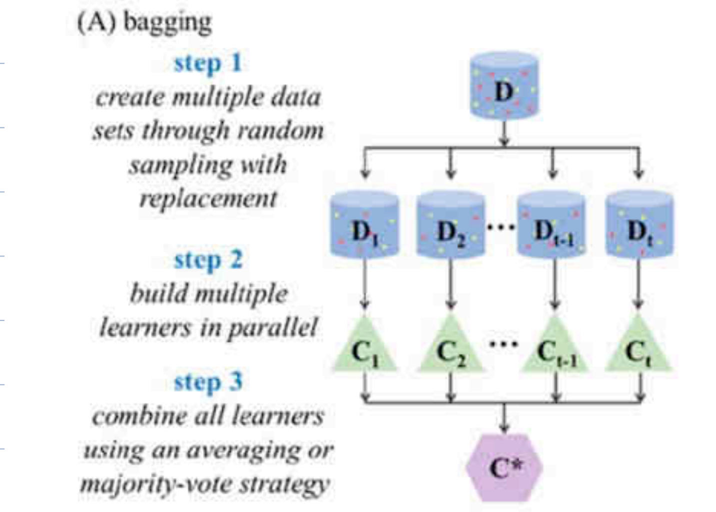
- As number of WLs increase, accuracy increases but algorithmic complexity increases

Types of Ensemble Models

1. Manipulate data distribution } syllabus
   - bagging, boosting
2. Manipulate input features
   - random forests
3. Manipulate class labels
   - error-correcting output coding

(A) BAGGING

- Bootstrap Aggregating



Bootstrap sample: subset of dataset obtained by sampling with replacement

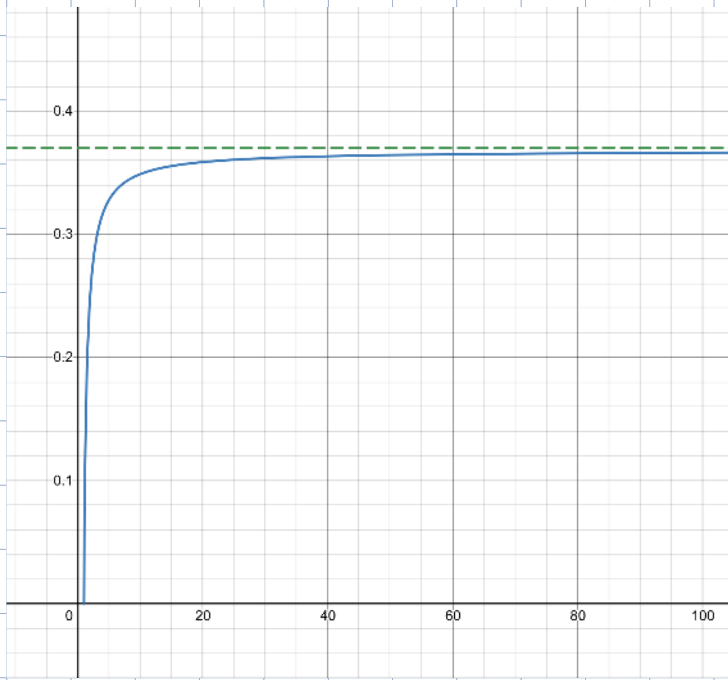
(duplicates are introduced)

- Weak model created on each subset and final prediction — voting or averaging
- 63% of data from original dataset selected with bagging (observed)
- Sample size = train dataset size = n
- Consider a certain data instance in the train dataset
- Probability of picking that instance = $\frac{1}{n}$
- Probability of not picking it = $1 - \frac{1}{n}$
- If we have n randomly-bagged samples (WLs), probability of never picking a particular instance

$$p = \left(1 - \frac{1}{n}\right)^n$$

- We have assumed no. of instances = no. of WLs

$\left(1 - \frac{1}{n}\right)^n$

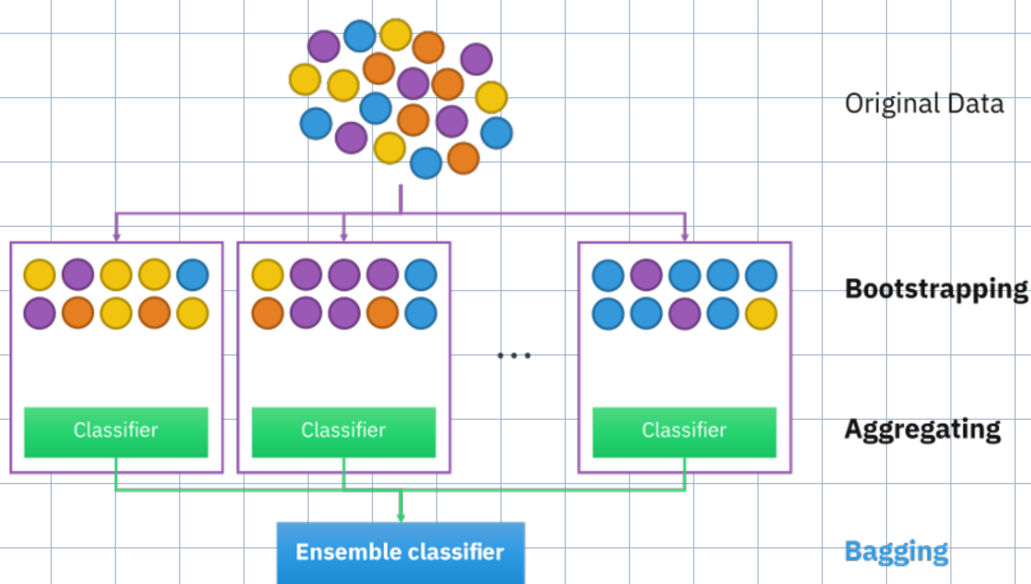


n

Advantages

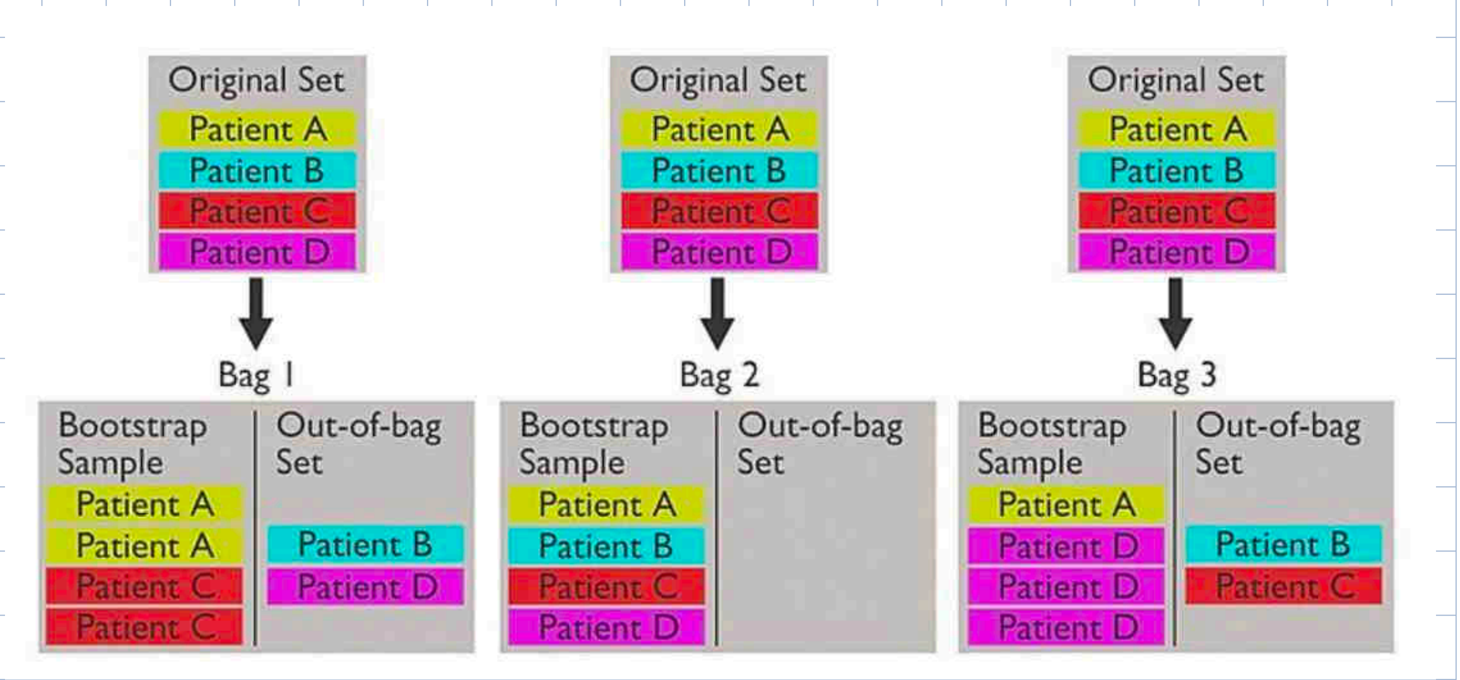
- parallel computation
- less variance than DT
- no preprocessing

Visualisation of Bagging



Out of Bag Set

- Bootstrap sample — in bag



- OOB sets can be used as train datasets for performance of model

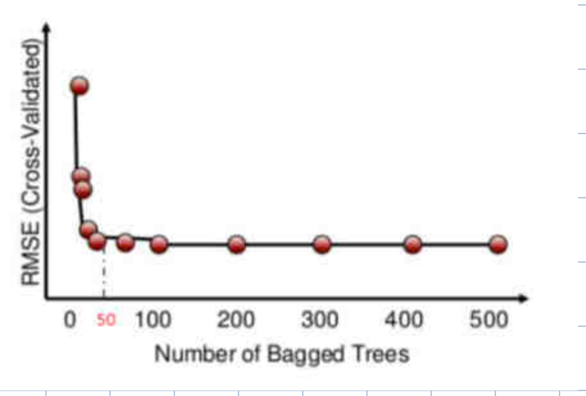
- OOB error
  1. Find all models not trained by an OOB instance
  2. Take majority vote of the models' predictions for the OOB instance
  3. Average of errors for an instance gives OOB error for that instance
  4. Compile error of all instances in OOB dataset

Error Calculation

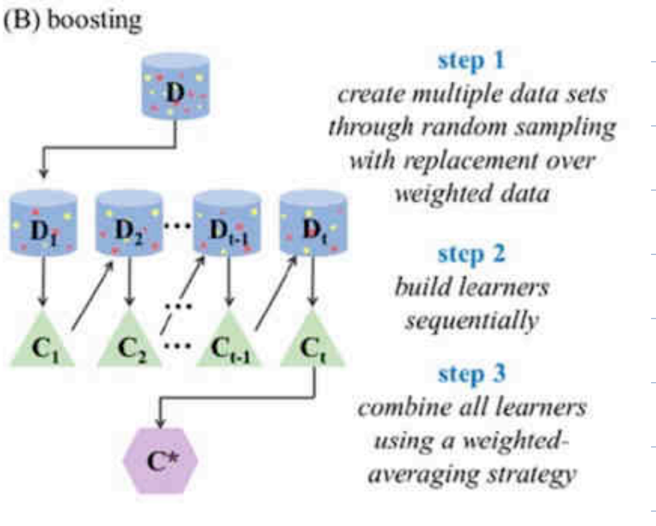
- k-fold cross-validation
- OOB error

Number of Bagged Trees

- Max improvement at ~50 stumps
- ~100 WLs good enough



(B) BOOSTING

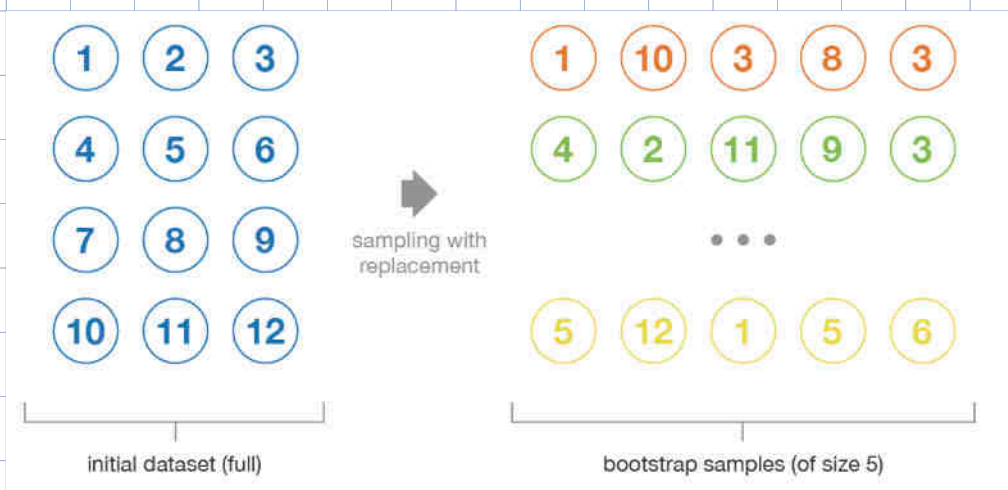


- Sequentially build models such that instances that were misclassified by the prev model are given more weight in next model

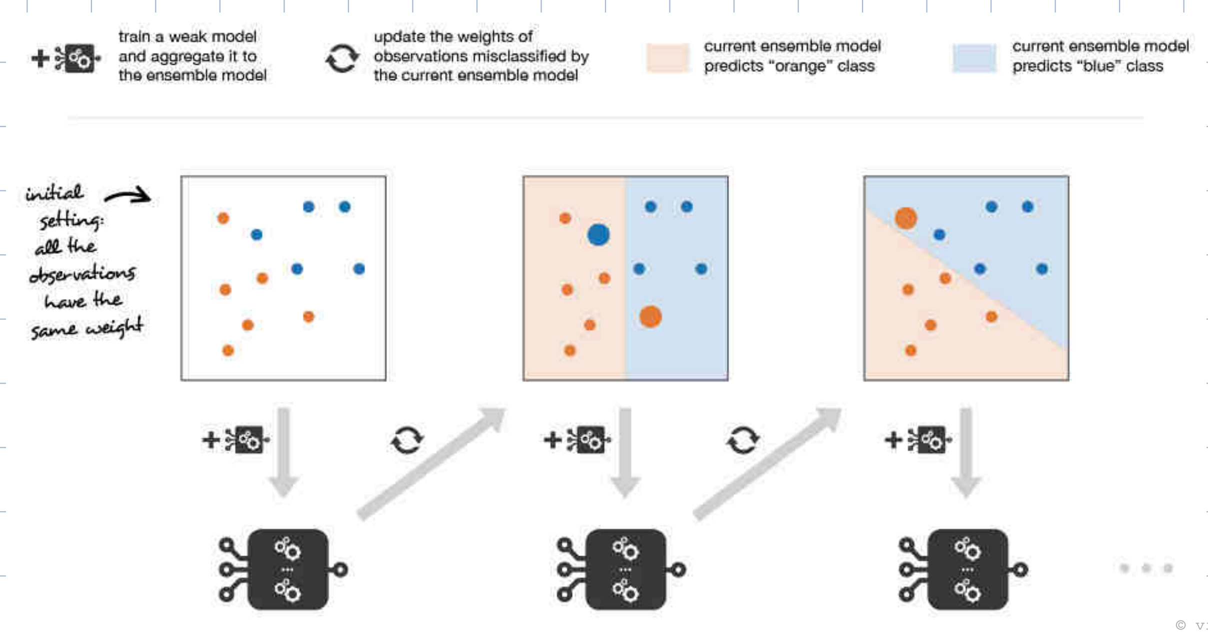
- Weighted observations, dependent on prev learners

BAGGING VS BOOSTING

- Bagging - bootstrapping



- Boosting



ADABOOST (with stumps)

- Weights associated with each training example (importance of getting that instance right)
- Initial stump used as first learner (best attribute chosen
- Second learner gives importance to misclassified instances due to first learner
- Each model given amount of say α (voting rights) — quality of model
- Sensitive to outliers if large no. of classifiers used

Algorithm

Algorithm 11.3: Boosting($D, T, \mathscr{A}$) – train an ensemble of binary classifiers from reweighted training sets.

WL

Input : data set D; ensemble size T; learning algorithm $\mathscr{A}$.
Output : weighted ensemble of models.

$w_{1i} \leftarrow 1/|D|$ for all $x_i \in D$; // start with uniform weights
for $t = 1$ to T **do**
  run $\mathscr{A}$ on D with weights w_{ti} to produce a model M_t;
  calculate weighted error ϵ_t;
  if $\epsilon_t \geq 1/2$ **then**
    set $T \leftarrow t-1$ and break
  end
  $\alpha_t \leftarrow \frac{1}{2}\ln\frac{1-\epsilon_t}{\epsilon_t}$; // confidence for this model
  $w_{(t+1)i} \leftarrow \frac{w_{ti}}{2\epsilon_t}$ for misclassified instances $x_i \in D$; // increase weight
  $w_{(t+1)j} \leftarrow \frac{w_{tj}}{2(1-\epsilon_t)}$ for correctly classified instances $x_j \in D$; // decrease weight
end
return $M(x) = \sum_{t=1}^{T} \alpha_t M_t(x)$

Peter Flach

Algorithm Simplified

1. Initialise sample weights of all N instances to be equal
$$w_i = \frac{1}{N} \quad \forall \quad i \in [1, N]$$

2. Choose learner $h_1(x)$ with highest accuracy as start learner (from set of possible/chosen weak learners)

3. Run prediction and calculate error rate ε as sum of weights of misclassified instances

4. Repeat to find $h_2(x)$, ensuring that sample weights of instances misclassified by $h_1(x)$ are increased

5. Take weighted vote of all the hypotheses and make final prediction

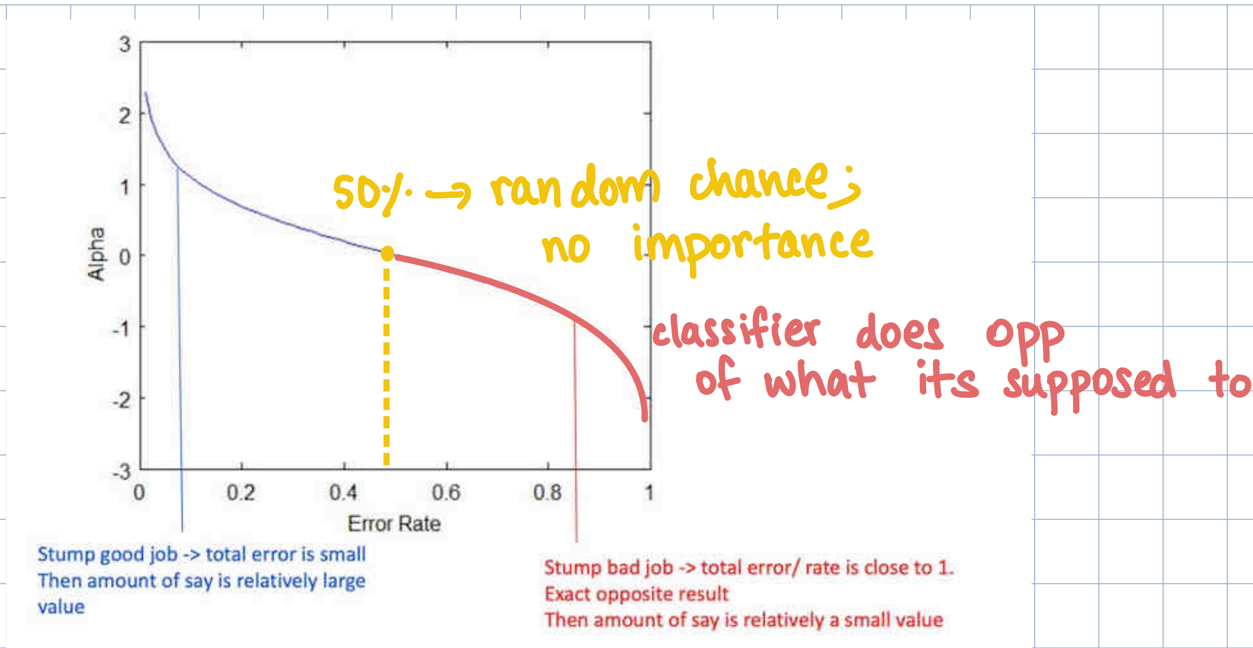
6. 2 sets of weights - sample & hypothesis

7. Hypothesis weights - amount of say α

Amount of Say - α

- α for a classifier where ε = error

$$\alpha = \frac{1}{2} \ln \left(\frac{1 - \varepsilon}{\varepsilon} \right)$$



Indicator Function

$$I(a,b) = \begin{cases} 1, a \neq b \\ 0, a = b \end{cases}$$

Error ε_m of m^{th} classifier

$$\varepsilon_m = \frac{\sum_{n=1}^{N} w_n^{(m)} I\left(y_m\left(x_n\right), t_n\right)}{\sum_{n=1}^{N} w_n^{(m)}}$$

w_n = weight of n^{th} data instance in m^{th} classifier

not sure of m

Updation of Weights

(1) Correctly classified

$$W_i^{s+1} = \frac{W_i^s}{N^s} e^{-\alpha}$$

(2) Wrongly classified

$$W_i^{s+1} = \frac{W_i^s}{N^s} e^{\alpha}$$

Replacing α

$$W_i^{s+1} = \frac{W_i^s}{N^s} e^{-\frac{1}{2}\ln\left(\frac{1-\varepsilon}{\varepsilon}\right)} \longrightarrow \text{correct}$$

$$W_i^{s+1} = \frac{W_i^s}{N^s} \sqrt{\frac{\varepsilon}{1-\varepsilon}}$$

$$W_i^{s+1} = \frac{W_i^s}{N^s} e^{\frac{1}{2}\ln\left(\frac{1-\varepsilon}{\varepsilon}\right)} \longrightarrow \text{wrong}$$

$$W_i^{s+1} = \frac{W_i^s}{N^s} \sqrt{\frac{1-\varepsilon}{\varepsilon}}$$

Normalisation factor

$$N^s = \sum_{i \in wrong} e^{\alpha} W_i^s + \sum_{i \in right} e^{-\alpha} W_i^s$$

$$N^s = \sqrt{\frac{1-\varepsilon}{\varepsilon}} \sum_{i \in wrong} W_i^s + \sqrt{\frac{\varepsilon}{1-\varepsilon}} \sum_{i \in right} W_i^s$$

$$N^s = \sqrt{\frac{1-\varepsilon}{\varepsilon}}\, \varepsilon + \sqrt{\frac{\varepsilon}{1-\varepsilon}}\, (1-\varepsilon)$$

$$N^s = 2\sqrt{\varepsilon\,(1-\varepsilon)}$$

Simplified Weight Updation

(1) Correct

$$W_i^{s+1} = \frac{W_i^s}{2\sqrt{\varepsilon(1-\varepsilon)}}\sqrt{\frac{\varepsilon}{1-\varepsilon}}$$

$$W_i^{s+1} = \frac{W_i^s}{2(1-\varepsilon)}$$

(2) Wrong

$$W_i^{s+1} = \frac{W_i^s}{2\sqrt{\varepsilon(1-\varepsilon)}}\sqrt{\frac{1-\varepsilon}{\varepsilon}}$$

$$W_i^{s+1} = \frac{W_i^s}{2\varepsilon}$$

Q: Perform 2 iterations of Adaboost using the given classifiers

X1	X2	CLASS LABEL	Weights
2	3	+1	0.1
2.1	2	+1	0.1
4.5	6	+1	0.1
4	3.5	-1	0.1
3.5	1	-1	0.1
5	7	+1	0.1
5	3	-1	0.1
6	5.5	+1	0.1
8	6	-1	0.1
8	2	-1	0.1

mis classified

classifier (for +)	Wrong	Error
$X1 \leq 2.1$ $X2 > 3.5$	3/10	0.3

New weights for correct: $W^{s+1} = \dfrac{W^s}{2(1-\varepsilon)}$

New weights for wrong: $W^{s+1} = \dfrac{W^s}{2\varepsilon}$

$$\alpha_1 = \frac{1}{2} \ln\left(\frac{1-\varepsilon}{\varepsilon}\right) = 0.4237$$

$h_1(x) = 0.4237$ [$X1 \leq 2.1$ gives 1]

X1	X2	CLASS LABEL	Weights
2	3	+1	0.1/1.4
2.1	2	+1	0.1/1.4
4.5	6	+1	0.1/0.6
4	3.5	-1	0.1/1.4
3.5	1	-1	0.1/1.4
5	7	+1	0.1/0.6
5	3	-1	0.1/1.4
6	5.5	+1	0.1/0.6
8	6	-1	0.1/1.4
8	2	-1	0.1/1.4

X1	X2	CLASS LABEL	Weights
2	3	+1	1/14
2.1	2	+1	1/14
4.5	6	+1	1/6
4	3.5	-1	1/14
3.5	1	-1	1/14
5	7	+1	1/6
5	3	-1	1/14
6	5.5	+1	1/6
8	6	-1	1/14
8	2	-1	1/14

classifier (for +)	Wrong	Error	α
$X1 \leq 2.1$	3/10	0.3	0.4237
$X2 > 3.5$	3/10	3/14	

X1	X2	CLASS LABEL	Weights
2	3	+1	1/14
2.1	2	+1	1/14
4.5	6	+1	1/6
4	3.5	-1	1/14
3.5	1	-1	1/14
5	7	+1	1/6
5	3	-1	1/14
6	5.5	+1	1/6
8	6	-1	1/14
8	2	-1	1/14

$$\alpha = \frac{1}{2} \ln\left(\frac{1-3/14}{3/14}\right) = \frac{1}{2} \ln\left(\frac{11}{3}\right) = 0.6496$$

classifier (for +)	Wrong	Error	α
$X1 \leq 2.1$	3/10	0.3	0.4237
$X2 > 3.5$	3/10	3/14	0.6496

$h_2(x) = 0.6496$ [X2 > 3.5 gives 1]

Update weights

X1	X2	CLASS LABEL	Weights
2	3	+1	$1/14 \div (6/14)$
2.1	2	+1	$1/14 \div (6/14)$
4.5	6	+1	$1/6 \div (22/14)$
4	3.5	-1	$1/14 \div (22/14)$
3.5	1	-1	$1/14 \div (22/14)$
5	7	+1	$1/6 \div (22/14)$
5	3	-1	$1/14 \div (22/14)$
6	5.5	+1	$1/6 \div (22/14)$
8	6	-1	$1/14 \div (6/14)$
8	2	-1	$1/14 \div (22/14)$

X1	X2	CLASS LABEL	
2	3	+1	1/6
2.1	2	+1	1/6
4.5	6	+1	7/66
4	3.5	-1	1/22
3.5	1	-1	1/22
5	7	+1	7/66
5	3	-1	1/22
6	5.5	+1	7/66
8	6	-1	1/6
8	2	-1	1/22

Hypothesis $H(x) = \text{sign}\left(\sum_{s=1}^{m} \alpha_s h_s(x)\right)$