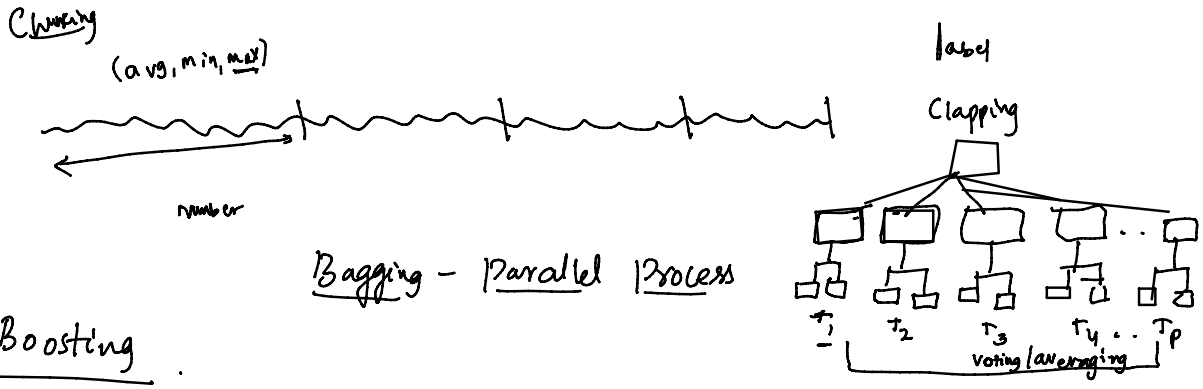
Boosting

21 February 2022 20:45

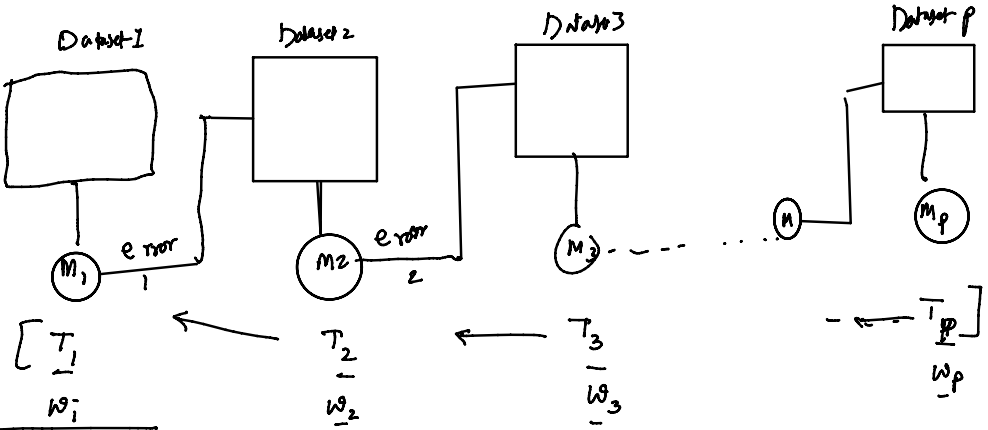
16



Bagging - Parallel Process

Boosting

→ Sequential Process (My triple person marathon)

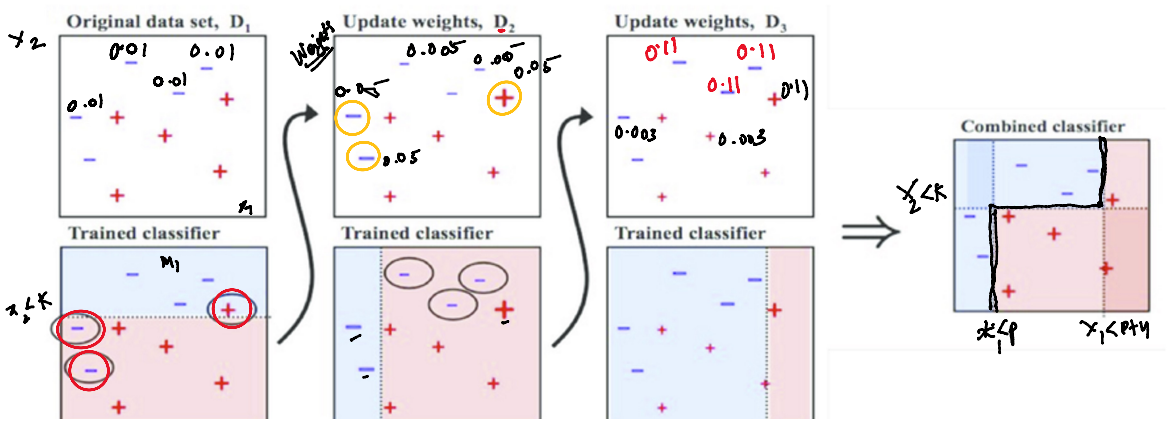


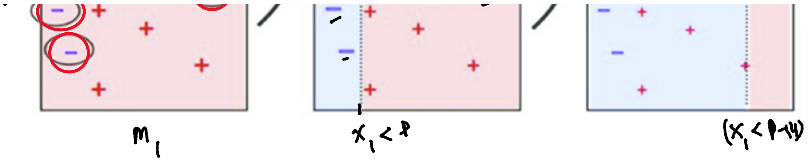
$$\gamma = \sum_{i=1}^{p} (w_i \wedge T_i)$$

→ Adaptive Boosting (Adaboost)

→ Gradient Boosting (GBM), xgboost extreme Gradient Boosting

Adaboost (decision stumps) — decision tree with only one split)
↖ (weak learners)



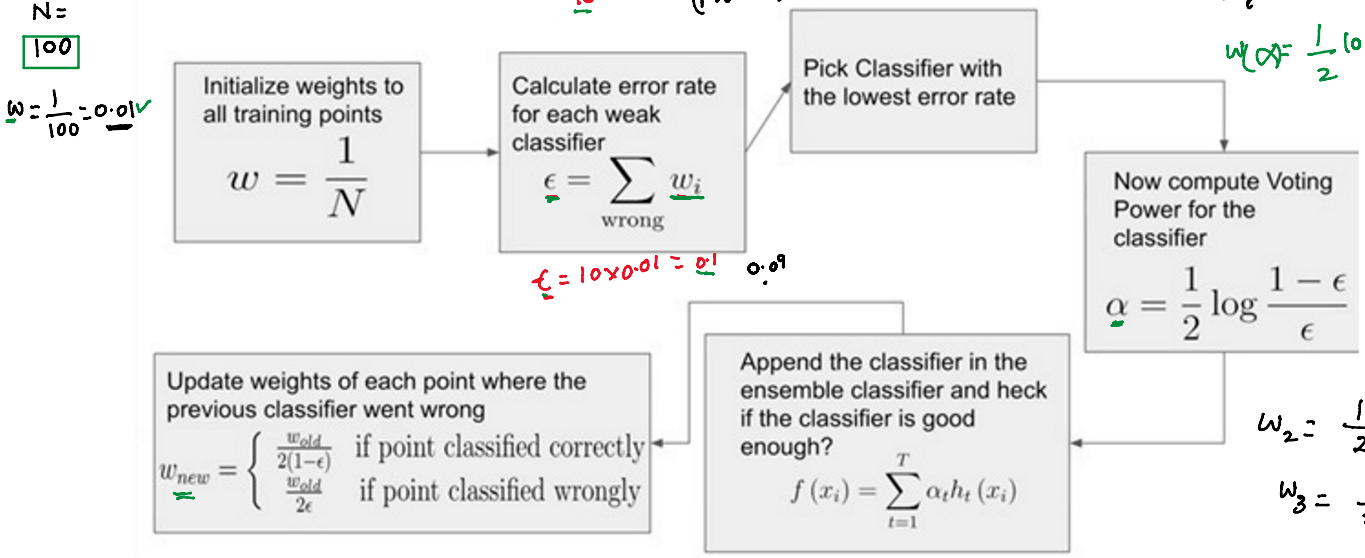


'amount of say'

Iteration 1 (Model-1) $M_1 = 0.9$ $e_{error} = 0.1$ 90 ✓ 10 ✗ (Model 2)

$N = 100$

$w = \frac{1}{100} = 0.01$ ✓



$\epsilon = 10 \times 0.01 = 0.1$ 0.09

w_1

$\alpha = \frac{1}{2} \log \frac{1-0.1}{0.1}$

$w_2 = \frac{1}{2} \log \frac{1-0.09}{0.1}$

$w_3 = \frac{1}{2} \log \frac{1-\epsilon_3}{\epsilon_3}$

$w_{new} = \frac{w_{old}}{2(1-\epsilon)}$ if Correctly Classified

$w_{new} = \frac{w_{old}}{2\epsilon}$ if Wrongly Classified.

w_{new} $w_{new} = \frac{0.01}{2(1-0.1)} = 0.0055$

$w_i = 1.09$

$w_2 = 0.81$

$w_3 = 0.76$

$M = \left[\frac{1.09 \times M_1 + 0.81 \times M_2}{1.09 + 0.81} \right.$

$\underline{M} = \left[\frac{1.09 \times M_1 + 0.81 \times M_2 + 0.76 M_3}{1.09 + 0.81 + 0.76} \right.$

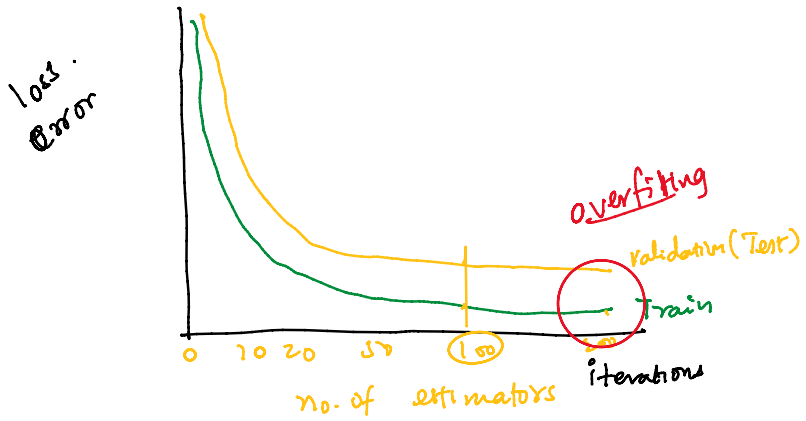
$\frac{w_{old}}{2\epsilon} = \frac{0.01}{2 \times 0.1} = 0.05$

w_{new} $w_{new} =$

$w_{new} =$

Validation accuracy = 0.91 - - -

? n_estimators :- No. of models



Classification: step by step

1. Initialize the weight 1/m for all i=1..m examples
2. We create a decision stump for each variable and see how well each stump classifies samples to their target classes. We'd look at how many samples are correctly or incorrectly classified as Fit or Unfit for each individual stump.
3. We calculate the Gini index and build the tree
4. Total error for a stump is the sum of the weights associated with the incorrectly classified samples
5. We use the total error to determine the Amount of Say.
   Amount of say =1/2 log (1-total error/total error)
6. We need to modify the weights by using the below formula
   New Sample Weight = Sample Weight * e^amount of say
7. More weight is assigned to the incorrectly classified samples so that they're classified correctly in the next decision stump. Weight is also assigned to each classifier based on the accuracy of the classifier, which means high accuracy = high weight!
8. Reiterate from Step 2 until all the data points have been correctly classified, or the maximum iteration level has been reached

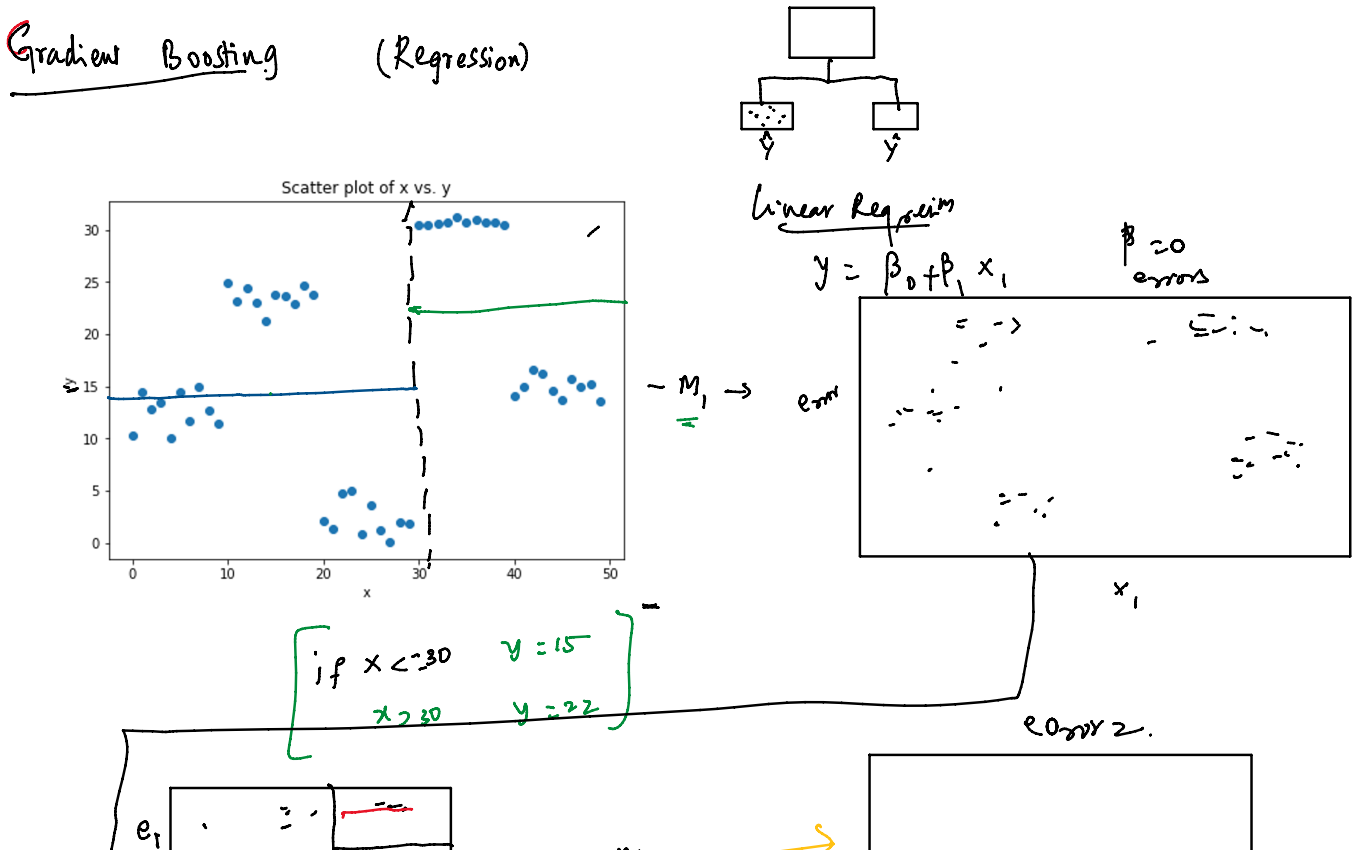
From <https://www.kaggle.com/questions-and-answers/181589>

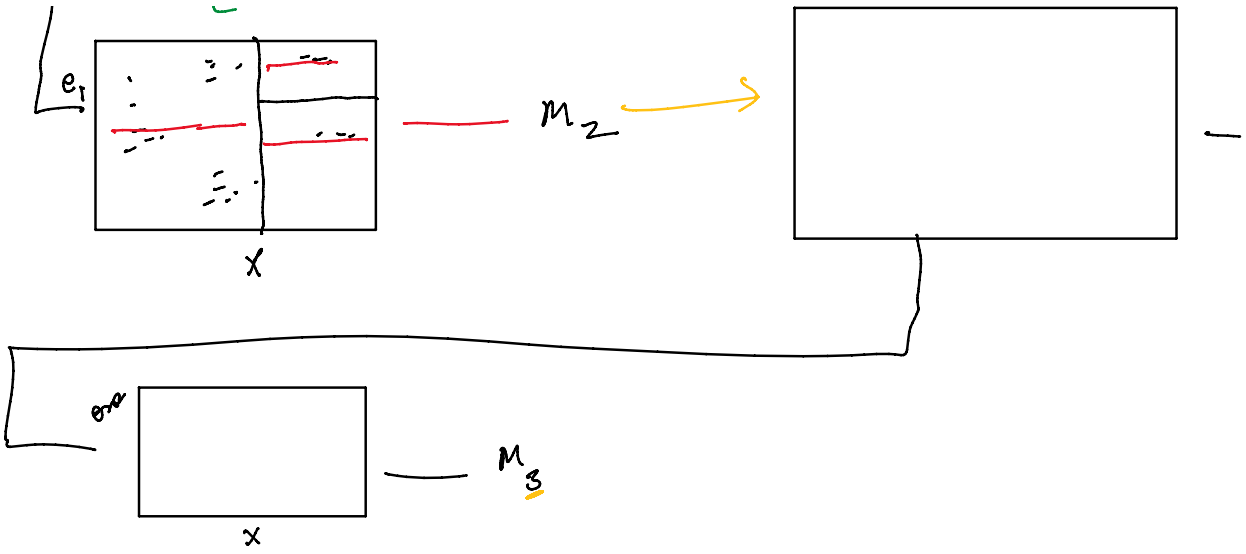
$\eta = 0.5$

$\text{amount of say} = \frac{1}{2} \log \frac{1-\varepsilon_i}{\varepsilon_i}$

$\text{amount of say} = \eta \log \frac{1-\varepsilon_i}{\varepsilon_i}$

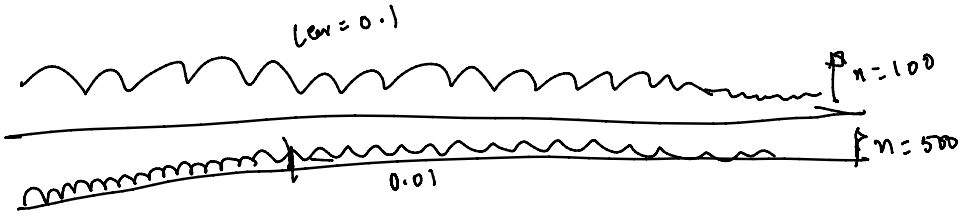
η - learning rate





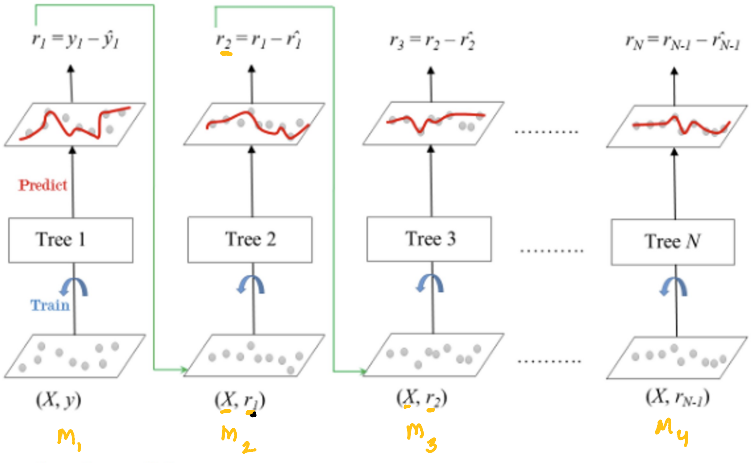
Layman

learning



- where $F_i(x)$ - previous base learner, $h_m(x)$ - current learner.

(log odds)



lets perform the predictions

$y = m_1 + m_2 + m_3 + m_4$

Gradient descent

y_i^p

$$\text{loss} = L(y_i, y_i^p)$$

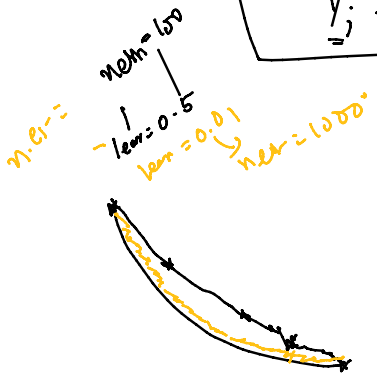
$$\partial L = \sum_{i=1}^{n} (y_i - y_i^p)^2$$

$$y_i^p = y_i^p + \underbrace{\alpha}_{\text{learning rate}} \cdot \frac{\partial L(\quad)}{\partial y}$$

$$= y_i^p + \alpha(-2)(y_i - y_i^p)$$

$$\boxed{= y_i^p - \alpha 2 \cdot (y_i - y_i^p)}$$

Gradient Boosting machine

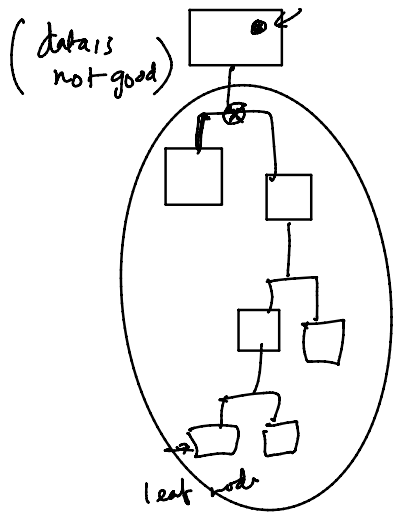


disadvantages

Too slow

Bias variance tradeoff

High variance



Xgboost — Whole bunch of cost functions

2. Parallelization (Tree - entropy | Gini) ✓
3. Subsampling
4. Regularization
5. 'Gpu' execution ✓

Overfits - (Variance)

Ensembly RiF, Boosting (Sub Sampling)