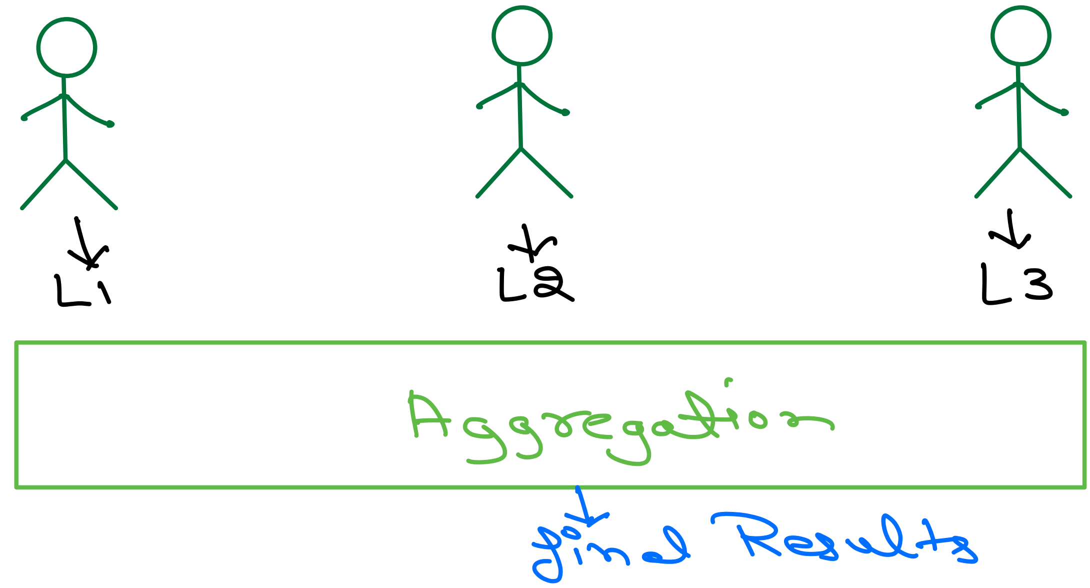
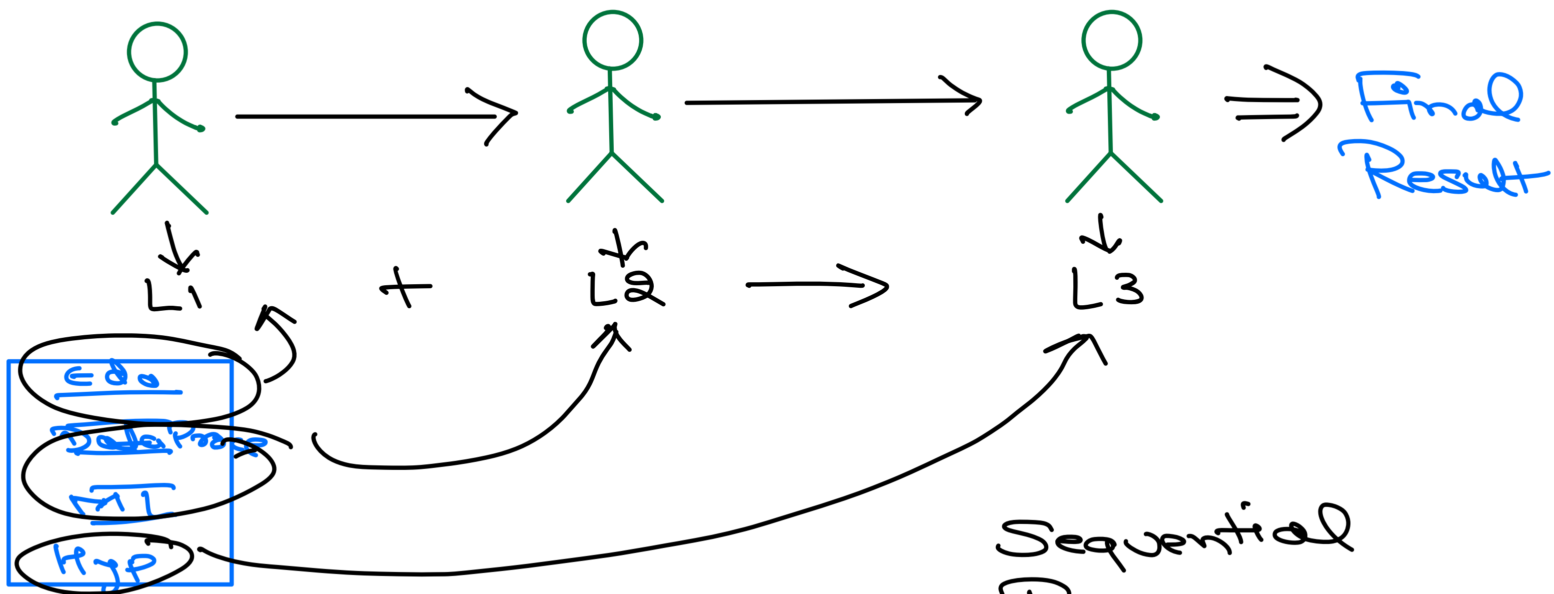
Recap

Bagging vs Boosting

Bagging ⇒ Bootstrap Aggregation



Boosting ⇒ Additive Combining



Sequential Process

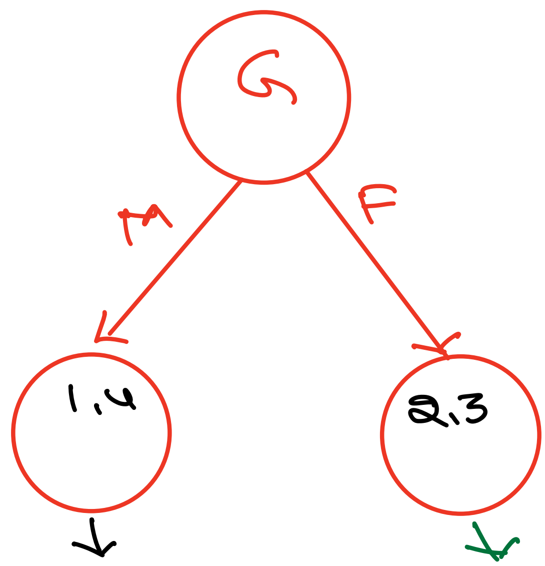
Steps in Boosting

Step 0: Mean Model and Error residual

	Height	Gender	Weight	$h_0(x)$	Error
1	1.6	M	82	67	15
2	1.5	F	55	67	−12
3	1.4	F	66	67	−1
4	1.4	M	65	67	−2

$$\frac{82+55+66+65}{4} \Rightarrow 67$$

Step 1: Model 1 ⇒ $D \Rightarrow \{x_i, \text{Error}\}$



Preds ⇒ $\frac{13}{2} \Rightarrow 6.5$

Preds: $\frac{-12+-1}{2} \Rightarrow -6.5$

h_1x

	$h_0(x)$	Error	$f_1(x) \Rightarrow h_0x + h_1x$	Err_1
1	67	15	73.5	
2	67	−12	60.5	
3	67	−1	60.5	
4	67	−2	73.5	

and Combine Stage 1 with stage 0 Predictions

$f_1(x) \Rightarrow h_0 x + \gamma_1 (h_1 x)$

Height	Gender	Weight	Err.	$h_1(x)$	$h_0(x)$	$f_1 x = h_0(x) + h_1(x)$
1.6	M	82	15	6.5	67	73.5
1.5	F	55	-12	-6.5	67	60.5
1.4	F	66	-1	-6.5	67	60.5
1.4	M	65	-2	6.5	67	73.5

$f_2(x) \Rightarrow h_0(x) + h_1(x) + h_2(x)$

(x_i, Err_2)

L_0

L_1

L_3

$f_n(x) \Rightarrow h_0 x + \gamma_1 h_1 x + \gamma_2 h_2(x) \ldots \gamma_n h_n(x)$

⇒ γ_n Can't be calculated at Same Time

LR ⇒ $w_0 x + w_1 x_1 + w_2 x_2 \ldots w_n x_n$

⇒ all of w_i's can be calculated at Same time

⇒ Boosting is Slow due to Sequential Nature

Gradient Boosting

(Regression

$L(y_i, \hat{y}_{(i)}) \Rightarrow$ m.s.e $\Rightarrow \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$

$\hat{y} \Rightarrow f_k(x)$

(Prediction at Stage k)

$$\frac{\partial L}{\partial \hat{y}} \Rightarrow \frac{(y_i - \hat{y})^2}{\partial \hat{y}}$$

$$\frac{\partial L}{\partial \hat{y}} \Rightarrow -2(y - \hat{y})$$

$$-\frac{\partial L}{\partial \hat{y}} \Rightarrow 2(y - \hat{y})$$

negative gradient of Loss w.r.t output

ignore

residual

Pseudo-Residual $\Rightarrow -\frac{\partial L}{\partial \hat{y}}$

Residual is proportional to -ve gradient of Loss w.r.t prediction

How do we use Pseudo-Residual?

⇒ for some model m_j

$$M_j \longrightarrow \{x_j, err^i_{j-1}\}$$

→ $y_i - F_{j-1}(x)$

→ residual

→ Replace err_{j-1} with it's pseudo Residual

Optimizing for pseudo Residual will also optimize for the Loss function ← $-\dfrac{\partial L}{\partial F_{j-1}(x_i)}$

* L ⇒ MSE or RMSE for Regression

Log-Loss for classification

⇒ At any stage k, to optimize we need to calculate gradient of Loss function w.r.t output at Stage k-1

GBDT

Pseudo-Residual using Gradient
(optimize for this)

$\{x_i, y_i\}_{i=1}^{n}$

$L(y, f(x))$ — MSE (Regression)
— LogLoss (Classification)

M

Input: training set $\{(x_i, y_i)\}_{i=1}^{n}$, a differentiable loss function $L(y, F(x))$, number of iterations M.

Algorithm:

1. Initialize model with a constant value:

$$F_0(x) = \underset{\gamma}{\arg\min} \sum_{i=1}^{n} L(y_i, \gamma).$$ ← G

2. For m = 1 to M:
   1. Compute so-called *pseudo-residuals*:

   $$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} \quad \text{for } i = 1, \ldots, n.$$

   2. Fit a base learner (or weak learner, e.g. tree) closed under scaling $h_m(x)$ to pseudo-residuals, i.e. train it using the training set $\{(x_i, r_{im})\}_{i=1}^{n}$.
   3. Compute multiplier γ_m by solving the following one-dimensional optimization problem:

   $$\gamma_m = \underset{\gamma}{\arg\min} \sum_{i=1}^{n} L\left(y_i, F_{m-1}(x_i) + \gamma h_m(x_i)\right).$$ ←

   optimal value of γ

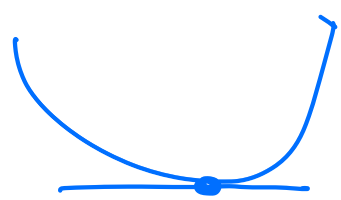
   4. Update the model:

   $$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x).$$

3. Output $F_M(x)$.

$$\arg\min \sum_{i=1}^{n} L(y_i, \gamma)$$

$$\arg\min \sum_{i=1}^{n} (y_i - \gamma)^2$$



To find minima Take diff and set it equal to Zero

$$\frac{\partial L}{\partial \gamma} = \frac{\sum_{i=1}^{n} (y-\gamma)^2}{\partial y} \Rightarrow 0$$

$$\Rightarrow -2\left(\sum_{i=1}^{n} (y_i - \gamma)\right) = 0$$

$$\Rightarrow \sum_{i=1}^{n} (y_i) - n\gamma = 0 \times -2$$

$$\Rightarrow n\gamma = \sum_{i=1}^{n} (y_i)$$

$$\Rightarrow \gamma \Rightarrow \frac{1}{n}\sum_{i=1}^{n} (y_i)$$

Step 1 is Nothing but Creation of Mean Model

Step 1 $\Rightarrow$ initialize with mean Model

$\arg\min \sum_{i=1}^{n} L(y_i, \gamma)$

avg

For $\Rightarrow$ Avg Model

Step 2

2.1 $\Rightarrow$ Calculate pseudo Residual

$$-\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}$$

2.2) fit another Low Depth Tree on pseudo-residual

2.3 : Using gradient descent to calculate the Best Value of γ_m

(the one which gives min Loss)

2.4

$m = 1$

$$F_m = F_{m-1} \cdot x + \gamma_m h_m(x)$$

$$F_1 = F_0 x + \gamma_m h_0(x)$$

Repeat Step 2 for M times

We are optimizing

→ model $h_1 x, h_2 x \ldots h_m x$

→ $\gamma_1, \gamma_2 \ldots \gamma_m$

weighted addition

Base Learner → High Bias
→ Low Variance

M → Number of Base-learner

M → ↑ Bias Reduce
and
Variance Variance

if m is a very high number, variance of Final Boosted Model will also be very high
(Overfit)

Base learners → Depth
↓
High Depth will lead to high Variance and Overfitting

To Find right Balance we will have to Tune
M
D

Q) Is there a regularization Term in GBDT

$$F_m(x) \Rightarrow h_0 x + \sum_{i=1}^{m} \gamma_m h_m(x)$$

+

Regularization (Shrinkage)

⇓

$$F_m(x) \Rightarrow h_0 x + \eta \sum_{i=1}^{m} \gamma_m h_m(x)$$

Constant Value (0.1)

(learning Rate)

$\eta \Rightarrow 0 \Rightarrow$ Underfitted Model

$\eta = \infty \Rightarrow$ Overfitting

Issues with GBDT:

1) No parallelization due to Sequential learning

2) Prone to Overfitting

Stochastic GBDT $\Rightarrow$ GBDT + Randomization (Column Sampling)

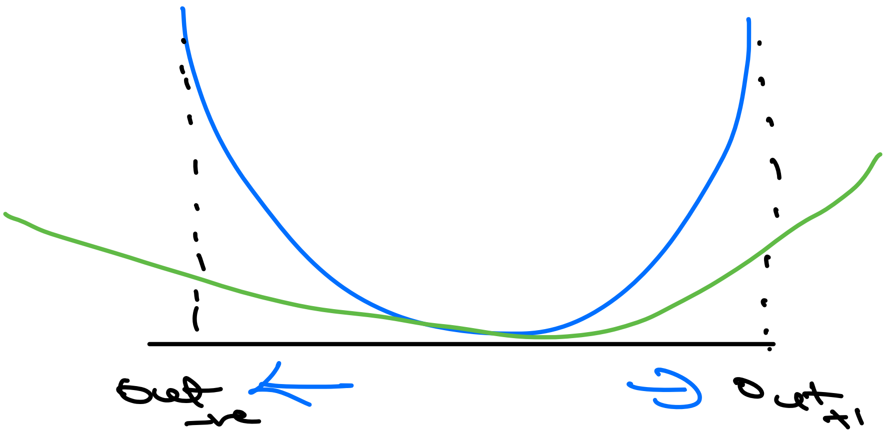
Row Samplin

$d = 3$

	m_1	m_2	m_3
x_1	e_{11}	e_{12}	e_{13}
x_2	e_{21}	e_{22}	e_{23}
x_3	e_{31}	e_{32}	e_{33}

$e_{23} \Rightarrow$ Very High or Very Low

RMSE or Huber loss



* Replacing Loss function with Something that doesn't Explode ie gives High Value for Outliers

* GBDT Implemention
* Variations of GBDT