~~GB~~ XG ⇒ large complicated data set

Gradient ~~boost~~ does regression

Regulation

XGBoost

(A unique Regression tree) ∈ part I

part II) XGBoost Classification

part III) Math

part I Effectiveness starts with 0.5

⇒ Residual

⇒ unique Regression Tree = XGBoost

↳ Many Ways to build XGBoost Tree

but for now focus on $\boxed{\text{Common way}}$

⇒ Starts with single leaf

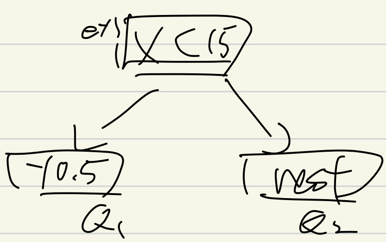
$$\boxed{\text{Residuals}} \Rightarrow Q = \frac{(\Sigma \text{ Resi})^2}{\text{Num(Resi)} + \lambda}$$

(similarity score)

λ → Regularation Parameter

ex) thresold = 15.

let $\lambda = 0$



$$Q_1 = \frac{(-10.5)^2}{1 + \lambda} = 110.25 \quad (\lambda = 0)$$

$$Q_2 = 14.08$$

⇒ 제곱전에 더하기 때문에 부호가 같으면 (수가 비슷) similarity ↑

⇒ root Q 와 비교 : Gain = $Q_1 + Q_2 - Q$

= 120

small Bam

ex thre : 22.5

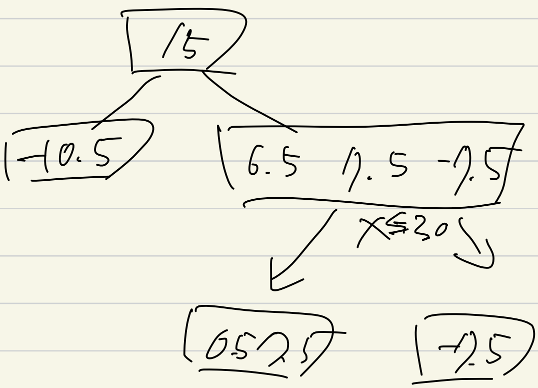
$\boxed{x < 22.5}$ ⇒ Gain = 8 + 0 − 4 = 4

⇒ x < 15 is better at splitting the residuals into clusters of similar value

ex threshold : 30

$\boxed{x < 30}$ ⇒ Gain = 4.08 + 56.25 − 4 = 56.33

then use x < 15 ∵ Gain is biggest.



likely calculate.

and limit depth=2 (default=6)

(Prune) : 가지치기. If set $\gamma = 130$

x≤30 의 Gain = 140.17

and if Gain − γ > 0 not remove branch

" < 0 remove.

but branch 가 remove 되지 않으면 상위 branch는 Gain − γ < 0 이더라도 pruning 하지않는다.

($\lambda = 0$: default)

If $\lambda = 1$: Regularization Parameter reduce the prediction's sensitivity to individual observation.

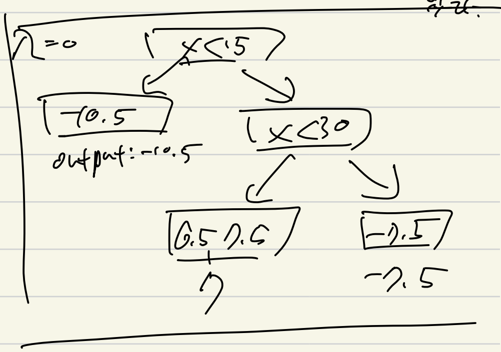
$\mathcal{Q} = 3.2$

$\mathcal{Q}_1 = 55.12$　　$\mathcal{Q}_2 = 10.56$

$\therefore$ λ ↑ then easier to prune

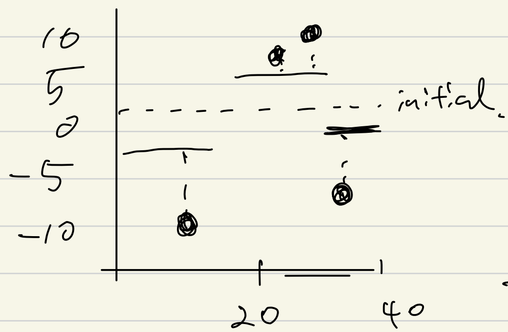
— $\lambda > 0$ 이라도 Gain < 0 일 수 있어 $\gamma = 0$ 이라도 prune 될 수 있다.

↳ prevent overfitting



— Output Value = $\dfrac{\sum \text{Resi}}{\text{Num(Resi)} + \lambda}$!

Prediction : initial pred + learning rate × Output Value
(0.5)　($\varepsilon = 0.3$ default)



⇒ initial prediction 에 비해 각각 더 작은 Resi가 나온다.

↳ 이 과정을 반복

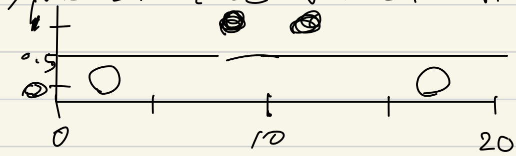
λ

Summary

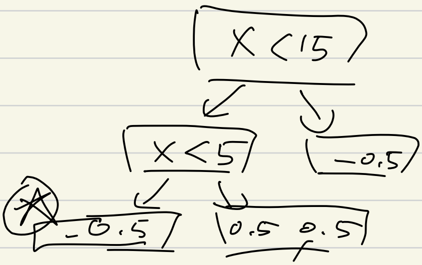
1) Similarity score
2) Gain : split data
3) prune
4) output value

Video 2

XGBoost Tree for Classification



initial 0.5



★ $\text{similarity} = \dfrac{(\sum \text{Resi})^2}{\sum[(\text{previous probabi}) \times (1-pp_i)] + \lambda}$

first leaf : -0.5, 0.5, 0.5, -0.5

Simi = 0

try X<15 : 15는 뒤 두개의 평균

Simi Left = 0.33

Simi Right = 1

Gain = 1.33

⇒ 나머지로 구한다 find biggest one.

이후 3개 남은 left side 에 대해 진행.

$\text{Cover} = \sum[pp_i \times (1-pp_i)]$

ex) ★ leaf : cover = $0.5 \times (1-0.5) = 0.25$

default cover = 1, ∴ ★ leaf is not allowed

∵ If cover=1, then 모든 leaf 는 제거

∴ maybe want to set cover as 0

which means "min_child_weight" = 0

prune. Gain − γ

→ λ 증 Simi 감 Gain 감 Gain − γ 감 prune 증

$$\text{Output} = \frac{\sum (\text{Resi})}{\sum [PP_i \times (1 - PP_i)] + \lambda}$$

Note : λ가 없으면 Output 식은 unextreme Gradient Boost와 같다.

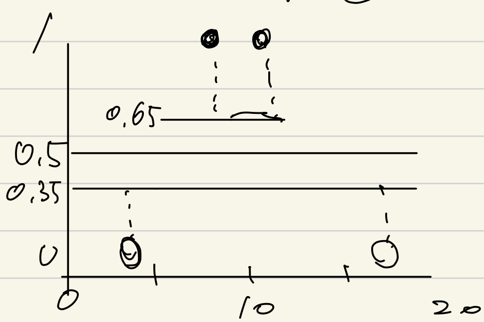
—— First Tree Done ——

$\log(\text{odds}) : \log\left(\frac{PP_i}{1 - PP_i}\right)$ ⇒ Youtube 보기

(p.prediction$_i$ output

$\log(\text{odds})\ \text{prediction} = \log(\text{odds}) + \text{learning rate} \times \text{output}$

$$\text{probability} = \frac{e^{\log(\text{odds})}}{1 + e^{\log(\text{odds})}}$$ (logistic function)



Summary

1) Similarity
2) Gain
3) prune
4) Output Value

consider
λ
cover

Second Tree 는 PP_i 가 모두 다르다.

↳ 반복

Video 3 Mathematical Detail

Only difference between Regression & Classification is "Loss Function"

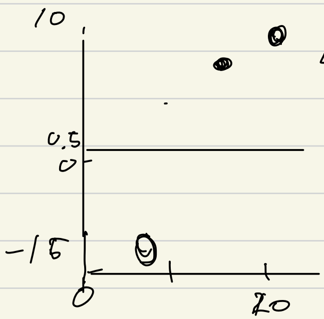
1) Regression

$$L(y_i, p_i) = \frac{1}{2}(y_i - p_i)^2$$

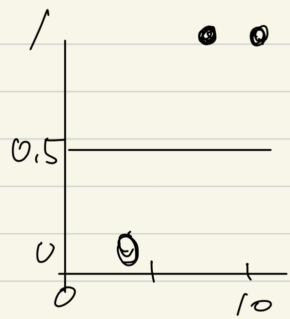
2) negative log-likelyhood Loss function : classification

$$L(y_i, p_i) = -[y_i \log(p_i) + (1-y_i)\log(1-p_i)]$$

$$\Leftarrow \sum_{i=1}^{3} L(y_i, p_i) = 104.4$$



Regression



Classification

set $\gamma = 0$ Regularization

$$\left[\sum L(y_i, p_i)\right] + \cancel{\gamma} T + \frac{1}{2}\lambda O_{value}^2$$

set $\lambda = 0$

For first tree $\left[\sum L(y_i, p_i^0 + O_{val})\right] + \cancel{\frac{1}{2}}\lambda O_{val}^2$

$$= \frac{1}{2}(y_i - p_i)^2 = 104.4$$