Gradient Boost

→ Gradient Boost - builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by adding the feature of optimizing an arbitrary differentiable loss function.

REGRESSION

→ Input: $\{(x_i, y_i)\}_{i=1}^{n}$ → Data

$L(y_i, F(x))$ → Loss function.

Most common loss function = $\frac{1}{2}(\text{Observed} - \text{Predicted})^2$

→ Step 1: Initialize model with a const value

$$F_0(x) = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^{n} L(y_i, \gamma)$$

for, γ = avg (all elements), Loss function is least.

Thus, Prediction = average of all values for γ each

→ Step 2:

for m = 1 to M [M: No of trees (8-32) or more]

(A) Gradient

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

$F(x) = F_{m-1}(x)$

$$= -\frac{d}{d\,\text{Pred.}} \frac{1}{2}(\text{Observed} - \text{Pred}^2)^2 = \text{Observed} - \text{Predicted}$$

Thus this calculates residuals.

but, r_{im} is called **pseudo-residuals**.

(B) Create tree to fit residuals **NOT outputs**

(C) For $j = 1 \cdots J_m$, $\gamma_{jm} = \underset{\gamma}{\text{argmin}} \sum_{x_i \in R_{ij}} L(y_i, F_{m-1}(x_i))$

↳ Argmin of loss function takes average of values

↳ This means convert all leaves of tree to average of all values in that leaf.

(D) $F_m(x) = F_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$

New pred = Old pred + learning rate × value of residual from tree traversal

NOTE: Step 2 is done M times

→ Step 3: Output $F_m(x)$

NOTE: Gradient boost have full sized trees not just stumps.

CLASSIFICATION

Input: Data: $\{(x_i, y_i)\}_{i=1}^{n}$

Loss fn: $-[\text{Observed} \cdot \log(p) + (1-\text{Observed}) \cdot \log(1-p)]$

$= -[(-\text{observed} \times \log\text{odds}) + \log(1 + e^{\log\text{odds}})]$

$$\frac{dl}{d(\log \text{odds})}\left(\text{loss function}\right) = -\text{Observed} + \frac{e^{\log(\text{odds})}}{1+e^{\log(\text{odds})}}$$

$$= -\text{Observed} + p$$

Step 1:

Initialize model with constant value

$$F_0(x) = \underset{\gamma}{\text{argmin}} \sum_{i=1}^{n} L(y_i, \gamma) \qquad [\gamma = \log(\text{odds})]$$

p = This is again average of all values, 1 → yes, 0 → no

$$\log(\text{odds}) = \log\left(\frac{p}{1-p}\right)$$

$$F_0(x) = \log(\text{odds})$$

Step 2:

for m = 1 to M.

(A) $$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]$$ $i = 1 \dots n$

$F(x) = f_{m-1}(x)$

pseudo residual = Observed $- \frac{e^{\log(\text{odds})}}{1+e^{\log\text{odds}}}$

= Observed - p [Any one]

Calculate for entire dataset.

(B)

→ Build regression tree to residuals not output

(C) for $j = 1 \dots J_m$ compute,

$$\gamma_{jm} = \underset{\gamma}{\text{argmin}} \sum_{x_i \in R_{ij}} L(y_i, F_{m-1}(x_i) + \gamma)$$

This reduces to,

$$\frac{\sum_{j=1}^{J_m} \text{Residuals}}{\sum_{j=1}^{J_m} p_j(1-p_j)}$$

[In a particular leaf]

[p → last prediction]

(D)

$$F_m(x) = F_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_{jm} \mathbb{I}(x \in R_{jm})$$

New pred = Last pred + Learning Rate × Output from tree

This is in log(odds)

Do everything until $m = M$

→ Step 3: Output $F_m(x)$. If $p > 0.5 \rightarrow 1$ else $\rightarrow 0$