Boosting I

Gradient Boosting

Gradient Boosting

Starting point: A decision tree...

$$T(x;\Theta) = \sum_{j=1}^{J} \gamma_j I(x \in R_j)$$

with tree parameters $\Theta = \{R_j, \gamma_j\}$.

Goal of tree construction:

$$\Theta = \arg\min_{\Theta} \sum_{j=1}^{J} \sum_{x_i \in R_j} L(y_i, \gamma_i)$$

$\rightarrow$ Boosting: Minimizing loss over a *sequence* of trees

Gradient Boosting

Boosting: Reduce the loss given $f_{m-1}(x_i)$

$$\Theta = \arg\min_{\Theta_m} \sum_{i=1}^{N} L(y_i, f_{m-1}(x_i) + T(x_i; \Theta_m))$$

$\rightarrow$ Focus on pseudo-residuals for the ith obs on iteration m

$$r_{im} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f_{m-1}}$$

$\rightarrow$ Fit regression tree to pseudo-residuals/ negative gradients

$$\tilde{\Theta}_m = \arg\min_{\Theta} \sum_{i=1}^{N} (-g_{im} - T(x_i; \Theta))^2$$

Gradient Boosting

Boosting: Reduce the loss given $f_{m-1}(x_i)$

$$\Theta = \arg\min_{\Theta_m} \sum_{i=1}^{N} L(y_i, f_{m-1}(x_i) + T(x_i; \Theta_m))$$

$\rightarrow$ Focus on pseudo-residuals for the ith obs on iteration m

$$r_{im} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f_{m-1}}$$

$\rightarrow$ Fit regression tree to pseudo-residuals/ negative gradients

$$\tilde{\Theta}_m = \arg\min_{\Theta} \sum_{i=1}^{N} (-g_{im} - T(x_i; \Theta))^2$$

Gradient Boosting

Boosting: Reduce the loss given $f_{m-1}(x_i)$

$$\Theta = \arg\min_{\Theta_m} \sum_{i=1}^{N} L(y_i, f_{m-1}(x_i) + T(x_i; \Theta_m))$$

$\rightarrow$ Focus on pseudo-residuals for the ith obs on iteration m

$$r_{im} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f_{m-1}}$$

$\rightarrow$ Fit regression tree to pseudo-residuals/ negative gradients

$$\tilde{\Theta}_m = \arg\min_{\Theta} \sum_{i=1}^{N} (-g_{im} - T(x_i; \Theta))^2$$

Gradient Boosting Machines

- General approach to sequential learning
- Applicable with various loss functions
- **Boosting trees**
  1. Initialize model (with a constant $f_0(x)$)
  2. Compute pseudo-residuals based on current model

$$r_{im} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f_{m-1}}$$

  3. Fit a regression tree to the pseudo-residuals
  4. Compute $\gamma_{jm} = \arg\min_\gamma \sum_{x_i \in R_{jm}} L(y_i, f_{m-1}(x_i) + \gamma)$
  5. Update the current model: $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$
- Output $\hat{f}(x) = f_M(x)$

$\rightarrow$ Analogue to steepest descent

Gradient Boosting Machines

Table: GBM components for different loss functions

Setting	Loss function	r_i	$f_0(x)$
Regression	$\frac{1}{2}(y_i - f(x_i))^2$	$y_i - f(x_i)$	mean(y_i)
Regression	$\lvert y_i - f(x_i)\rvert$	sign$(y_i - f(x_i))$	median(y_i)
Classification	Deviance	$I(y_i = G_k) - p_k(x_i)$	prior p's

Boosting for Regression

Algorithm 1: Gradient Boosting for regression

Set number of trees M;
Set interaction depth D;
Set shrinkage parameter λ;
Use $\bar{y}$ as initial prediction;
for $m = 1$ *to* M **do**
    compute residuals based on current predictions;
    assign data to root node, using the residuals as the outcome;
    while *current tree depth* $< D$ **do**
        tree growing process;
    end
    compute the predicted values of the current tree;
    add the shrinked new predictions to the previous predicted values;
end

Boosting for Regression

Table: Boosting Example with 5 obs and 2 x's

ID	x_1	x_2	y	$f_0(x)$
1	0	0	1	1.2
2	0	2	3	1.2
3	1	2	2	1.2
4	2	3	0	1.2
5	0	1	0	1.2

Boosting for Regression

Table: Step 1: Split $x_2 > 2.5$

ID	x_1	x_2	y	$f_0(x)$	r_{i1}	γ_{j1}	$f_1(x)$
1	0	0	1	1.2	–0.2	0.3	1.5
2	0	2	3	1.2	1.8	0.3	1.5
3	1	2	2	1.2	0.8	0.3	1.5
4	2	3	0	1.2	–1.2	–1.2	0
5	0	1	0	1.2	–1.2	0.3	1.5

Boosting for Regression

Table: Step 2: Split $x_2 < 1.5$

ID	x_1	x_2	y	$f_0(x)$	$f_1(x)$	r_{i2}	γ_{j2}	$f_2(x)$
1	0	0	1	1.2	1.5	−0.5	−1	0.5
2	0	2	3	1.2	1.5	1.5	0.66	2.166
3	1	2	2	1.2	1.5	0.5	0.66	2.166
4	2	3	0	1.2	0	0	0.66	0.66
5	0	1	0	1.2	1.5	−1.5	−1	0.5