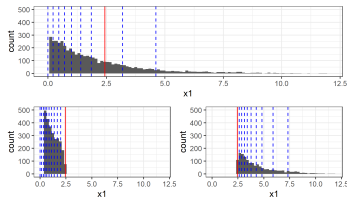
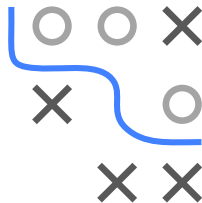


Introduction to Machine Learning

Boosting

Gradient Boosting: Deep Dive XGBoost

Optimization



Learning goals

- Understand details of the regularized risk in XGBoost
- Understand approximation of loss used in optimization
- Understand split finding algorithm

RISK MINIMIZATION

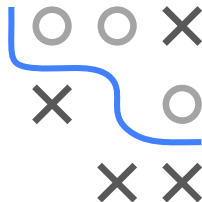
XGBoost uses a risk function with 3 regularization terms:

$$\mathcal{R}_{\text{reg}}^{[m]} = \sum_{i=1}^n L\left(y^{(i)}, f^{[m-1]}(\mathbf{x}^{(i)}) + b^{[m]}(\mathbf{x}^{(i)})\right) \\ + \lambda_1 J_1(b^{[m]}) + \lambda_2 J_2(b^{[m]}) + \lambda_3 J_3(b^{[m]}),$$

with $J_1(b^{[m]}) = T^{[m]}$ the number of leaves in the tree to penalize tree depth.

$J_2(b^{[m]}) = \|\mathbf{c}^{[m]}\|_2^2$ and $J_3(b^{[m]}) = \|\mathbf{c}^{[m]}\|_1$ are $L2$ and $L1$ penalties of the terminal region values $c_t^{[m]}$, $t = 1, \dots, T^{[m]}$.

We define $J(b^{[m]}) := \lambda_1 J_1(b^{[m]}) + \lambda_2 J_2(b^{[m]}) + \lambda_3 J_3(b^{[m]})$.



RISK MINIMIZATION / 2

To approximate the loss in iteration m , a second-order Taylor expansion around $f^{[m-1]}(\mathbf{x})$ is computed:

$$L(y, f^{[m-1]}(\mathbf{x}) + b^{[m]}(\mathbf{x})) \approx L(y, f^{[m-1]}(\mathbf{x})) + g^{[m]}(\mathbf{x})b^{[m]}(\mathbf{x}) + \frac{1}{2}h^{[m]}(\mathbf{x})b^{[m]}(\mathbf{x})^2,$$

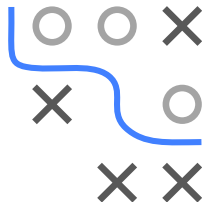
with gradient

$$g^{[m]}(\mathbf{x}) = \frac{\partial L(y, f^{[m-1]}(\mathbf{x}))}{\partial f^{[m-1]}(\mathbf{x})}$$

and Hessian

$$h^{[m]}(\mathbf{x}) = \frac{\partial^2 L(y, f^{[m-1]}(\mathbf{x}))}{\partial f^{[m-1]}(\mathbf{x})^2}.$$

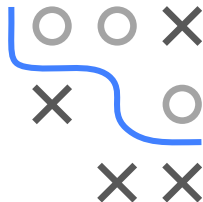
Note: $g^{[m]}(\mathbf{x})$ are the negative pseudo-residuals $-\tilde{r}^{[m]}$ we use in standard gradient boosting to determine the direction of the update.



RISK MINIMIZATION / 4

Expanding $J(b^{[m]})$:

$$\begin{aligned}\mathcal{R}_{\text{reg}}^{[m]} &= \sum_{t=1}^{T^{[m]}} \left(G_t^{[m]} c_t^{[m]} + \frac{1}{2} H_t^{[m]} (c_t^{[m]})^2 + \frac{1}{2} \lambda_2 (c_t^{[m]})^2 + \lambda_3 |c_t^{[m]}| \right) + \lambda_1 T^{[m]} \\ &= \sum_{t=1}^{T^{[m]}} \left(G_t^{[m]} c_t^{[m]} + \frac{1}{2} (H_t^{[m]} + \lambda_2) (c_t^{[m]})^2 + \lambda_3 |c_t^{[m]}| \right) + \lambda_1 T^{[m]}.\end{aligned}$$



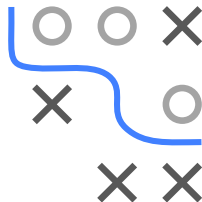
Note: The factor $\frac{1}{2}$ is added to the $L2$ regularization to simplify the notation as shown in the second step. This does not impact estimation since we can just define $\lambda_2 = 2\tilde{\lambda}_2$.

LOSS MINIMIZATION - SPLIT FINDING

To evaluate the performance of a candidate split that divides the instances in region $R_t^{[m]}$ into a left and right node we use the **risk reduction** achieved by that split:

$$\tilde{S}_{LR} = \frac{1}{2} \left[\frac{t_{\lambda_3} \left(G_{tL}^{[m]} \right)^2}{H_{tL}^{[m]} + \lambda_2} + \frac{t_{\lambda_3} \left(G_{tR}^{[m]} \right)^2}{H_{tR}^{[m]} + \lambda_2} - \frac{t_{\lambda_3} \left(G_t^{[m]} \right)^2}{H_t^{[m]} + \lambda_2} \right] - \lambda_1,$$

where the subscripts L and R denote the left and right leaves after the split.



LOSS MINIMIZATION - SPLIT FINDING / 2

```

1: Input  $I$ : instance set of current node
2: Input  $p$ : dimension of feature space
3:  $gain \leftarrow 0$ 
4:  $G \leftarrow \sum_{i \in I} g(\mathbf{x}^{(i)}), H \leftarrow \sum_{i \in I} h(\mathbf{x}^{(i)})$ 
5: for  $j = 1 \rightarrow p$  do
6:    $G_L \leftarrow 0, H_L \leftarrow 0$ 
7:   for  $i$  in sorted( $I$ , by  $x_j$ ) do
8:      $G_L \leftarrow G_L + g(\mathbf{x}^{(i)}), H_L \leftarrow H_L + h(\mathbf{x}^{(i)})$ 
9:      $G_R \leftarrow G - G_L, H_R \leftarrow H - H_L$ 
10:    compute  $\tilde{S}_{LR}$ 
11:   end for
12: end for
13: Output Split with maximal  $\tilde{S}_{LR}$ 

```