LeafInfluence vs. BoostIn

- Consider a two-tree ensemble.
- Assume fixed structure.
- Analysis for a single train example i.

Tree 1

LeafInfluence (Single Point):

$$\frac{\partial f_{H;l}^t}{\partial w_i} = \frac{I_l^t(i)(h_i^t f_{H;l}^t + g_i^t) + \sum_{j \in I_l^t} w_j (k_j^t f_{H;l}^t + h_j^t) J(A^{t-1})_{ij}}{H_{H;l}^t} \quad (1)$$

$$\begin{aligned} f_{H;l}^t &= \alpha_l^t = \text{leaf value for leaf } l \text{ in tree } t. \\ w_j &= 1, \forall j. \\ J(A^{t-1})_{ij} &= 0 \ \forall \text{ i, j; initialization.} \\ \text{Also, } J(A^{t-1})_{ij} &= 0 \text{ when } i \neq j, \forall t. \end{aligned}$$

Thus, eq. (1) becomes:

$$\frac{\partial \alpha_l^t}{\partial w_i} = \frac{h_i^t \alpha_l^t + g_i^t}{H_{H;l}^t} \quad (2)$$

$$\frac{\partial \alpha_l^t}{\partial w_i} = 0, \forall l \text{ in which } I_l^t(i) = 0.$$

Update prediction derivative for train example i:

$$J(A^t)_{ii} = J(A^{t-1})_{ii} + eq.(2)$$

Note: In the code, eq. (2) is:

$$\frac{\partial \alpha_l^t}{\partial w_i} = \frac{h_i^t \alpha_l^t / \eta + g_i^t}{H_{H;l}^t / \eta} * \eta, \ \ \eta = \text{learning rate.}$$

BoostIn:

$$\frac{g_i^t}{n_l^t} * \eta, \ \ n_l^t = \text{no. examples at leaf } l \text{ for tree } t.$$

Tree 2

LeafInfluence (Single Point):

$$\frac{\partial \alpha_l^t}{\partial w_i} = \frac{(h_i^t \alpha_l^t + g_i^t) + (k_j^t \alpha_l^t + h_j^t) J(A^{t-1})_{ii}}{H_{H;l}^t} \tag{3}$$

Update prediction derivative for train example i:

$$J(A^t)_{ii} = J(A^{t-1})_{ii} + eq.(3)$$

Note: In the code, eq. (3) is:

$$\frac{\partial \alpha_l^t}{\partial w_i} = \frac{(h_i^t \alpha_l^t/\eta + g_i^t) + (k_j^t \alpha_l^t/\eta + h_j^t) J(A^{t-1})_{ii}}{H_{H;l}^t/\eta} * \eta$$

BoostIn:

$$\frac{g_i^t}{n_l^t} * \eta$$

Approximating Test Loss

LeafInfluence (Single Point):

$$Inf(x_i, x_\tau) = -\nabla \left(\sum_t \alpha_{P(x_\tau)_t}^t \right) \left(\sum_t \frac{\partial \alpha_{P(x_\tau)_t}^t}{\partial w_i} \right)$$

$$\begin{aligned} x_\tau = & \text{ test example.} \\ P(x_\tau)_t = & \text{ leaf that } x_\tau \text{ ends at for tree } t. \end{aligned}$$

BoostIn:

$$Inf(x_i, x_\tau) = \sum_t I[P(x_i)_t = P(x_\tau)_t] \frac{g_i^t g_\tau^t}{n_l^t} * \eta$$

Complexity Analysis

LeafInfluence (Single Point):
- Precompute 1st, 2nd, and 3rd derivatives.
- Needs to keep track of prediction derivatives, one for each train example.
- Can compute leaf derivatives for all train examples in one pass.
- Complexity: O(Tn).

BoostIn:
- Precompute 1st derivatives.
- Complexity: O(Tn).

Additional Notation

$$
\begin{aligned}
g_i^t =\ & \text{1st derivative for example } i \text{ in tree } t. \\
h_i^t =\ & \text{2nd derivative for example } i \text{ in tree } t. \\
k_i^t =\ & \text{3rd derivative for example } i \text{ in tree } t. \\
H_{H;l}^t =\ & \text{Sum of 2nd derivatives for leaf } l \text{ in tree } t. \\
J(A^t) =\ & \text{Jacobian prediction interaction matrix for tree } t. \\
I_l^t =\ & \text{Set of train examples at leaf } l \text{ for tree } t. \\
I_l^t(i) =\ & \text{1 if example } i \text{ is at leaf } l \text{ for tree } t\text{, 0 otherwise.}
\end{aligned}
$$