Globally Induced Forest: A Prepruning Compression Scheme Supplementary material

Anonymous Authors[1]

1. Optimization problem

We are building an additive model by inserting progressively nodes in the forest. At time t, we are trying to find the best node $j*$ from the candidate list C_t and its associated optimal weight w_j^*:

$$j^{(t)}, w_j^{(t)} = \underset{j \in C_t, w \in \mathbb{R}^K}{\arg\min} \sum_{i=1}^{N} L\left(y_i, \hat{y}^{(t-1)}(x_i) + w z_j(x_i)\right) \tag{1}$$

where $(x_i, y_i)_i^N$ is the learning sample, $\hat{y}^{(t-1)}()$ is the model at time $t-1$, $z_j()$ is the node indicator functions, meaning that it is 1 if its argument reaches node j and 0 otherwise.

This problem is solved in two steps. First a node j is selected from C_t and the corresponding optimal weight, alongside the error reduction, are computed. This is repeated for all nodes and the one achieving the best improvement is selected.

Regression For regression, we used the L2-norm:

$$w_j^{(t)} = \underset{w \in \mathbb{R}}{\arg\min} \sum_{i=1}^{N} L\left(y_i, \hat{y}^{(t-1)}(x_i) + w z_j(x_i)\right)^2 \tag{2}$$

and the solution is given by

$$w_j^{(t)} = \frac{1}{|Z_j|} \sum_{i \in Z_j} r_i^{(t-1)} \tag{3}$$

where $r_i^{(t-1)} = y_i - \hat{y}^{(t-1)}(x_i)$ is the residual at time $t-1$ for the ith training instance and $Z_j = \{1 \leq i \leq N | z_j(x_i) = 1\}$ is the subset of instances reaching node j.

Classification For classification we used the multi-exponential loss. First, we need to encode the labels so that

$$y_i^{(k)} = \begin{cases} 1, & \text{if the class of } y_i \text{ is } k \\ -\frac{1}{K-1}, & \text{otherwise} \end{cases} \tag{4}$$

where K is the number of classes. Notice that $\sum_{k=1}^{K} y_i^{(k)} = 0$. The optimization then becomes

$$w_j^{(t)} = \underset{w \in \mathbb{R}^K}{\arg\min} \sum_{i=1}^{N} \exp\left(\frac{-1}{K} y_i^T \left(\hat{y}^{(t-1)}(x_i) + w z_j(x_i)\right)\right) \tag{5}$$

$$= \underset{w \in \mathbb{R}^K}{\arg\min} F_j^{(t-1)}(w) \tag{6}$$

Solving for $\nabla F_j^{(t-1)}(w) = 0$ yields

$$\alpha_j^{(t-1,k)} \phi^{(k)}(w) = \frac{1}{K} \sum_{l=1}^{K} \alpha_j^{(t-1,l)} \phi^{(l)}(w) \quad 1 \leq k \leq K \tag{7}$$

for , where

$$\alpha_j^{(t-1,k)} \triangleq \sum_{i \in Z_j^{(k)}} \exp\left(-\mu_i^{(t-1)}\right) \tag{8}$$

$$\mu_i^{(t-1)} \triangleq \frac{1}{K} \sum_{k=1}^{K} y_i \hat{y}^{(t-1,k)}(x_i) \tag{9}$$

$$\phi^{(k)}(w) \triangleq \exp\left(-\frac{1}{K} \psi^{(k)}(w)\right) \tag{10}$$

$$\psi^{(k)}(w) \triangleq -w^{(k)} + \frac{1}{K-1} \sum_{l=1, l \neq k}^{K} w^{(l)} \tag{11}$$

where $Z_j^{(k)} = \{1 \leq i \leq N | z_{i,j} = 1 \wedge y_i^{(k)} = 1\}$ is the subset of learning instances of class k reaching node j. In words, $\mu_i^{(t-1)}$ is the hyper-margin of instance i at time $t-1$ and $\alpha_j^{(t-1,k)}$ is the class error of label k for node j at time $t-1$.

Equation 7 is equivalent to

$$\alpha_j^{(t-1,k)} \phi^{(k)}(w) = \alpha_j^{(t-1,l)} \phi^{(l)}(w) \quad 1 \leq k, l \leq K \tag{12}$$

In keeping with the output representation (Equation 4), we can impose a zero-sum constraint on the prediction to get

a unique solution for the kth component of $w_j^{(t)}$. If it is imposed at each stage, it means that

$$\sum_{k=1}^{K} \hat{y}^{(t-1,k)} = \sum_{k=1}^{K} \hat{y}^{(t,k)} = 0 = \sum_{k=1}^{K} w^{(k)} \tag{13}$$

and this is not impacted by the learning rate. The corresponding solution is

$$\phi^{(k)}(w) = \exp\left(-\frac{1}{K-1} w^{(k)}\right) \tag{14}$$

$$\alpha_j^{(t-1,k)} = \sum_{i \in Z_j^{(k)}} \exp\left(-\frac{1}{K-1} \hat{y}^{(t-1,k)}(x_i)\right) \tag{15}$$

$$w_j^{(t,k)} = \frac{K-1}{K} \sum_{l=1}^{K} \log \frac{\alpha_j^{(t-1,k)}}{\alpha_j^{(t-1,l)}} \tag{16}$$

2. Equivalence of GIF and the underlying tree

In the case of a single tree and a unit learning rate, both the square loss in regression and the multiexponential loss in classification produce the same prediction as the underlying tree. This is due to the fact that, when examining the weight to give to node j at time t, the prediction of time $t-1$ relates to the parent π_j of j. It is thus independent of t and is also the same for all instance reaching that node. Consequently, we will adopt the following slight change in notation:

$$\hat{y}_j = \hat{y}_{(\pi_j)} + w_j \tag{17}$$

Meaning that the prediction associated to any object reaching node j is the weight of j plus the prediction associated to its parent π_j. With $\hat{y}_{(\pi_1)} = 0$, the prediction of the root's pseudo-parent.

2.1. Regression

In regression, the tree prediction Tr_j of any leaf j is the average of the learning set's outputs reaching that node: $Tr_j = \frac{1}{|Z_j|} \sum_{i \in Z_j} y_i$. We need to show that the GIF prediction is:

$$\hat{y}_j = \frac{1}{|Z_j|} \sum_{i \in Z_j} y_i \tag{18}$$

The prediction of node j is

$$\hat{y}_j = \hat{y}_{\pi_j} + w_j \tag{19}$$

$$= \hat{y}_{\pi_j} + \frac{1}{|Z_j|} \sum_{i \in Z_j} \left(y_i - \hat{y}_{\pi_j}\right) \tag{20}$$

$$= \hat{y}_{\pi_j} + \frac{1}{|Z_j|} \sum_{i \in Z_j} (y_i) - \hat{y}_{\pi_j} \tag{21}$$

$$= \frac{1}{|Z_j|} \sum_{i \in Z_j} y_i \tag{22}$$

The first step is how the additive model is built. The second is the optimal weight value of node j derived in Equation 3, the third step is due to the fact that the prediction at π_j is constant since there is only one tree.

2.2. Classification

In order to have the same prediction as the underlying tree, we must demonstrate that the probability of being in class l associated to node j will be $\frac{Z_j^{(l)}}{|Z_j|}$. Under the zero-sum constraint, we have

$$\exp\left(\frac{1}{K-1} w_j^{(l)}\right) = \frac{1}{c_j} \alpha_{\pi_i}^{(l)} \tag{23}$$

$$= \frac{1}{c_j} \sum_{i \in Z_j^{(l)}} \exp\left(-\frac{1}{K-1} \hat{y}_{\pi_i}^{(l)}\right) \tag{24}$$

$$= |Z_j^{(l)}| \exp\left(-\frac{1}{K-1} \hat{y}_j^{(l)}\right) \tag{25}$$

$$\exp\left(\frac{1}{K-1} \hat{y}_j^{(l)}\right) = \exp\left(\frac{1}{K-1} \hat{y}_{\pi_j}^{(l)}\right) \exp\left(\frac{1}{K-1} w_j^{(l)}\right) \tag{26}$$

$$= \frac{1}{c_j} |Z_j^{(l)}| \tag{27}$$

$$P_j(l) = \frac{\exp\left(\frac{1}{K-1} \hat{y}_j^{(l)}\right)}{\sum_{k=1}^{K} \exp\left(\frac{1}{K-1} \hat{y}_j^{(k)}\right)} = \frac{|Z_j^{(l)}|}{|Z_j|} \tag{28}$$

where $c_j = \left(\prod_{k=1}^{K} \alpha_j^{(k)}\right)^{\frac{1}{K}}$ is a constant. The first equality is a consequence of the value of $w_j^{(l)}$ (Equation 16). The second is a due to the definition of $\alpha_j^{(l)}$ (Equation 15). The third is a consequence of having a single tree: the prediction of the parent is the same for all instances.

Notice that, in both regression and classification, the equivalence also holds for an internal node: the prediction is the one the tree would have yielded if that node had been a leaf.