Supplemental Material
A Unified Framework for Knowledge Intensive Gradient Boosting: Leveraging Human Experts for Noisy Sparse Domains

Harsha Kokel[1], **Phillip Odom**[2], **Shuo Yang**[3], **Sriraam Natarajan**[1]
[1]The University of Texas at Dallas, [2]Georgia Tech Research Institute, [3]Indiana University Bloomington
[1]{hkokel,sriraam.natarajan}@utdallas.edu, [2]phodom@gatech.edu, [3]Shuoyang@iu.edu

1 Appendix

1.1 Derivation

Assume tree (ψ_t) splits at node n on variable a. As described in the paper, for monotonic influence $a \overset{Q+}{\prec} y$, we expect $\mathbb{E}_{\boldsymbol{\psi_t}}[\mathbf{n_L}] \leq \mathbb{E}_{\boldsymbol{\psi_t}}[\mathbf{n_R}]$, where $\mathbf{n_L}$ (resp. $\mathbf{n_R}$) is the set of all examples following the left (resp. right) sub-tree at $\mathbf{n}$. This expectation can be characterized using current tree ψ_t as:

$$\frac{1}{|\mathbf{n_L}|}\sum_{x_i \in \mathbf{n_L}} \psi_t(x_i) \leq \frac{1}{|\mathbf{n_R}|}\sum_{x_i \in \mathbf{n_R}} \psi_t(x_i) \tag{1}$$

Each tree, in turn, is represented as sum of its leaves ($\boldsymbol{\ell}$):

$$\psi_t(x) = \sum_{\boldsymbol{\ell} \in \psi_t} \psi_t^{\boldsymbol{\ell}} \cdot \mathbb{I}(x \in \boldsymbol{\ell}) \tag{2}$$

where, $\mathbb{I}(x \in \boldsymbol{\ell})$ represents whether example x is captured in leaf $\boldsymbol{\ell}$

In KiGB, we use expectation as a constraint and introduce ζ_n to measure the violation of this constraints at each node. $\zeta_{\mathbf{n}} = (\mathbb{E}_{\boldsymbol{\psi_t}}[\mathbf{n_L}] - \mathbb{E}_{\boldsymbol{\psi_t}}[\mathbf{n_R}] - \varepsilon)$. We modify the standard objective of squared-error loss by adding a squared-penalty for violation of this constraint.

$$\underset{\boldsymbol{\psi_t}}{argmin} \underbrace{\sum_{i=1}^{N} \left(\tilde{y}_i - \psi_t(x_i)\right)^2}_{\text{loss function w.r.t data}} + \underbrace{\frac{\lambda}{2} \sum_{\mathbf{n} \in \mathcal{N}(\mathbf{x}_c)} \max\left(\zeta_{\mathbf{n}} \cdot |\zeta_{\mathbf{n}}|, 0\right)}_{\text{loss function w.r.t. advice}} \tag{3}$$

With this objective, value of each parameter/leaf node can be derived taking partial derivation w.r.t that parameter.

$$\frac{\partial}{\partial \psi_t^j}\left(\sum_{i=1}^{N}\left(\tilde{y}_i - \sum_{\boldsymbol{\ell} \in \psi_t} \psi_t^{\boldsymbol{\ell}} \cdot \mathbb{I}(x_i \in \boldsymbol{\ell})\right)^2 + \frac{\lambda}{2} \sum_{\mathbf{n} \in \mathcal{N}(\mathbf{x}_c)} \max\left(\zeta_{\mathbf{n}} \cdot |\zeta_{\mathbf{n}}|, 0\right)\right) \tag{4}$$

Over the next few equations, we take the derivative and then equate it to zero.

$$\sum_{i=1}^{N} \frac{\partial}{\partial \psi_t^j}\left(\tilde{y}_i - \sum_{\boldsymbol{\ell} \in \psi_t} \psi_t^{\boldsymbol{\ell}} \cdot \mathbb{I}(x_i \in \boldsymbol{\ell})\right)^2 + \frac{\lambda}{2} \sum_{\mathbf{n} \in \mathcal{N}(\mathbf{x}_c)} \frac{\partial}{\partial \psi_t^j}\left(\max\left(\zeta_{\mathbf{n}} \cdot |\zeta_{\mathbf{n}}|, 0\right)\right) \tag{5}$$

Here, we use the derivative of ReLU. For, $f(x) = max(x, 0)$, the derivative is defined as:

$$f'(x) = \mathbb{I}(x > 0) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases}$$

Also,

$$\max\left(\zeta_{\mathbf{n}} \cdot |\zeta_{\mathbf{n}}|, 0\right) = \begin{cases} (\zeta_{\mathbf{n}})^2, & \text{if } \mathbb{I}(\zeta_{\mathbf{n}} > 0) \\ 0, & \text{otherwise} \end{cases}$$

Plugging this in the previous equation:

$$-2\sum_{i=1}^{N}\left(\tilde{y}_i - \sum_{\boldsymbol{\ell} \in \psi_t} \psi_t^{\boldsymbol{\ell}} \cdot \mathbb{I}(x_i \in \boldsymbol{\ell})\right) \cdot \mathbb{I}(x_i \in j) + \frac{\lambda}{2} \sum_{\mathbf{n} \in \mathcal{N}(\mathbf{x}_c)} \mathbb{I}(\zeta_n > 0) \frac{\partial}{\partial \psi_t^j}\left(\zeta_{\mathbf{n}}\right)^2 \tag{6}$$

substituting ζ_n,

$$-2\sum_{i=1}^{N}\left(\tilde{y}_i-\psi_t^j\right)\cdot\mathbb{I}(x_i\in j)+$$
$$\lambda\sum_{\mathbf{n}\in\mathcal{N}(\mathbf{x}_c)}\mathbb{I}(\zeta_n>0)\cdot\zeta_n\frac{\partial}{\partial\psi_t^j}\left(\mathbb{E}_{\boldsymbol{\psi_t}}[\mathbf{n_L}]\ -\ \mathbb{E}_{\boldsymbol{\psi_t}}[\mathbf{n_R}]-\varepsilon\right) \tag{7}$$

Substituting $\mathbb{E}_{\boldsymbol{\psi_t}}[\mathbf{n}]$

$$-2\sum_{i=1}^{N}\left(\tilde{y}_i\cdot\mathbb{I}(x_i\in j)\right)+2\psi_t^j\sum_{i=1}^{N}\left(\mathbb{I}(x_i\in j)\right)+$$
$$\lambda\sum_{\mathbf{n}\in\mathcal{N}(\mathbf{x}_c)}\mathbb{I}(\zeta_n>0)\cdot\zeta_n\left(\frac{\partial}{\partial\psi_t^j}\left(\frac{1}{|\mathbf{n_L}|}\sum_{x_i\in\mathbf{n_L}}\psi_t(x_i)\right)-\frac{\partial}{\partial\psi_t^j}\left(\frac{1}{|\mathbf{n_R}|}\sum_{x_i\in\mathbf{n_R}}\psi_t(x_i)\right)\right) \tag{8}$$

Here, $\sum_{i=1}^{N}\mathbb{I}(x_i\in j)$ is the number of samples at leaf node j, we use the notation $|j|$ for it.

$$-2\sum_{i=1}^{N}\left(\tilde{y}_i\cdot\mathbb{I}(x_i\in j)\right)+2\psi_t^j\cdot|j|+$$
$$\lambda\sum_{\mathbf{n}\in\mathcal{N}(\mathbf{x}_c)}\mathbb{I}(\zeta_n>0)\cdot\zeta_n\left(\frac{1}{|\mathbf{n_L}|}\sum_{x_i\in\mathbf{n_L}}\mathbb{I}(x_i\in j)-\frac{1}{|\mathbf{n_R}|}\sum_{x_i\in\mathbf{n_R}}\mathbb{I}(x_i\in j)\right) \tag{9}$$

$\sum_{x_i\in\mathbf{n_L}}^{N}\mathbb{I}(x_i\in j)$ is true only if the $j\in\mathbf{n_L}$ and when it is true, it will be equivalent to $|j|$.

$$-2\sum_{i=1}^{N}\left(\tilde{y}_i\cdot\mathbb{I}(x_i\in j)\right)+2\psi_t^j\cdot|j|+$$
$$\lambda\sum_{\mathbf{n}\in\mathcal{N}(\mathbf{x}_c)}\mathbb{I}(\zeta_n>0)\cdot\zeta_n\left(\frac{\mathbb{I}(j\in\mathbf{n_L})\cdot|j|}{|\mathbf{n_L}|}-\frac{\mathbb{I}(j\in\mathbf{n_R})\cdot|j|}{|\mathbf{n_R}|}\right) \tag{10}$$

Now, equating the derivation with zero will give us the following equation for leaf values:

$$\psi_t^j=\frac{1}{|j|}\sum_{i=1}^{N}\left(\tilde{y}_i\cdot\mathbb{I}(x_i\in j)\right)+$$
$$\frac{\lambda}{2}\sum_{\mathbf{n}\in\mathcal{N}(\mathbf{x}_c)}\mathbb{I}(\zeta_n>0)\cdot\zeta_n\left(\frac{\mathbb{I}(j\in\mathbf{n_R})}{|\mathbf{n_R}|}-\frac{\mathbb{I}(j\in\mathbf{n_L})}{|\mathbf{n_L}|}\right) \tag{11}$$

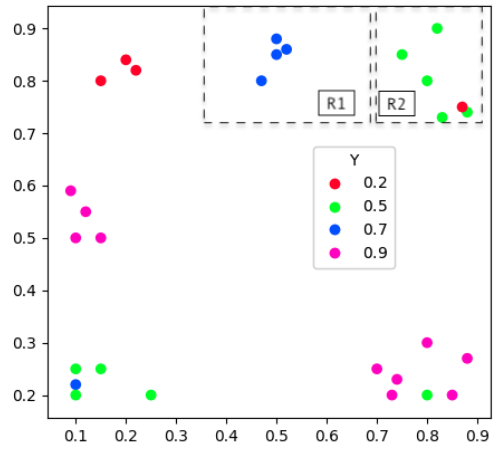
Below is the same equation mentioned in the paper, reproduced here for convenience:

$$\psi_t^{\boldsymbol{\ell}}(\mathbf{x})=\underbrace{\frac{1}{|\boldsymbol{\ell}|}\sum_{i=1}^{N}\tilde{y}_i\cdot\mathbb{I}(x_i\in\boldsymbol{\ell})}_{\text{mean}}+$$
$$\underbrace{\frac{\lambda}{2}\sum_{\mathbf{n}\in\mathcal{N}(\mathbf{x}_c)}\mathbb{I}(\zeta_\mathbf{n}>0)\zeta_\mathbf{n}\cdot\left(\frac{\mathbb{I}(\boldsymbol{\ell}\in\mathbf{n_R})}{|\mathbf{n_R}|}-\frac{\mathbb{I}(\boldsymbol{\ell}\in\mathbf{n_L})}{|\mathbf{n_L}|}\right)}_{\text{penalty for advice violation}} \tag{12}$$

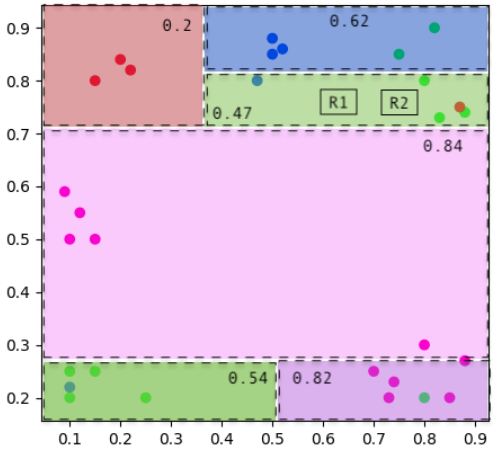
1.2 Overfitting by strict monotonicity

When a model is fitted w.r.t data under strict monotonicity constraints, as done by LMC, it may overfit the data sometime. We illustrate this with an example in this section. Consider a noisy data as shown in figure 1a, with feature a on the horizontal axis, b on the vertical axis, and different colors represent different regression values of the target y. Assume that some expert provided the *monotonic influence advice* – $a \overset{Q+}{\prec} y$ for this data.

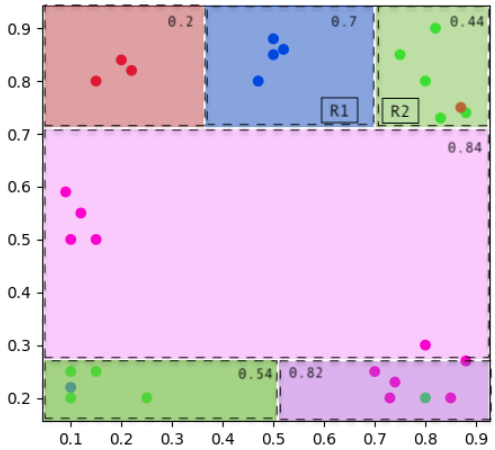
The noisy data clearly violates this constraint in region R1 & R2. In the scenario where the advice is significantly more important than the noisy data, it might seem reasonable to use the strict monotonic boosting provided by LightGBM (LMC). However, as seen in figure 1b the LMC overfits the training data by splitting horizontally in the region $R1$ & $R2$. Standard boosting method (LGBM) (figure 1c), on the other hand use natural splits but has no way of correcting the noise. Our LKiGB approach (figure 1d) uses the monotonic influence information from the expert to provide correction and learn a monotonic function.



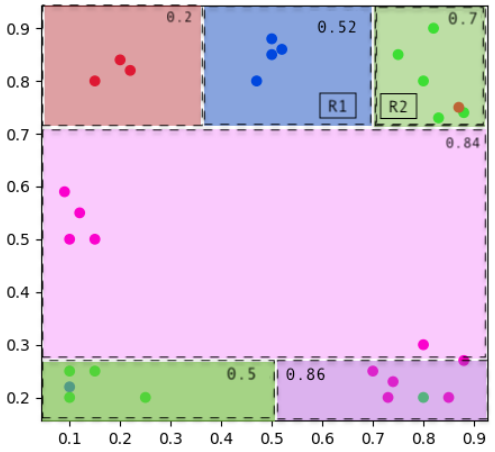
(a) data



(b) LMC



(c) LightGBM standard boosting (LGBM)



(d) KiGB

Figure 1: Illustration of the overfitting by LMC. As can be seen, LGBM, without any monotonic influence statements, learned an incorrect model due to the presence of noisy data. With LMC, the model learns a monotonic function but it overfits the training data. LKiGB provides a correction to the LGBM and generalizes to a better model.