Improved classification accuracy of random forests via post-hoc regularization

Bastian Pfeifer[1]*
[1]Institute for Medical Informatics, Statistics and Documentation
Medical University Graz, Austria

Abstract

-

Keywords

random forest, feature importance, explainable AI, regularization

I. Introduction

–

II. Hierarchical Shrinkage

The approaches summarized in section regulate the tree models solely based on their structural aspects, while assuming that the prediction made by each leaf should be the average response of the training samples contained in it. Recently, [1] proposed a *post-hoc* regularization referred to as *Hierarchical Shrinkage* (HS), which does not modify the tree structure. Instead, it shrinks the predictions of the trees or the sample weights used during training. The authors demonstrate that this additional regularization can further improve generalization performance. By reducing the complexity of the model through shrinkage, one can effectively prevent overfitting and allow for the use of smaller ensembles for many datasets without sacrificing accuracy. Furthermore, they find that HS also enhances the quality of post-hoc interpretations. Specifically, by reducing the noise in the feature importance measures, HSregularization seems to lead to more reliable and robust interpretations. HS replaces the average prediction or response of a leaf node with a weighted average of the mean responses of the leaf and its ancestors. These weights are determined based on the number of samples in each leaf and are controlled by a single regularization parameter λ, as summarized in Eq. (2).

The following is a brief summary of the ideas proposed in [1]. Assume that we are given a training set $\mathcal{D}_n = (X; y)$. Our goal is to learn a tree model $\hat{f}$ that accurately represents the regression function based on this training data. Given a query point $\mathbf{x}$, let $t_L \subset t_{L-1} \subset \cdots \subset t_0$ denote its leaf-to-root path, with t_L and t_0 representing its leaf node and the root node respectively. For any node t, let $N(t)$ denote the number of samples it contains, and $\hat{\mathbb{E}}_t\{y\}$ the average response. The tree model prediction can be written as the telescoping sum

$$\hat{f}(\mathbf{x}) = \hat{\mathbb{E}}_{t_0}\{y\} + \sum_{l=1}^{L} \left(\hat{\mathbb{E}}_{t_l}\{y\} - \hat{\mathbb{E}}_{t_{l-1}}\{y\} \right) \tag{1}$$

HS transforms $\hat{f}$ into a shrunk model $\hat{f}_\lambda$ via the formula:

$$\hat{f}_\lambda(\mathbf{x}) := \hat{\mathbb{E}}_{t_0}\{y\} + \sum_{l=1}^{L} \frac{\hat{\mathbb{E}}_{t_l}\{y\} - \hat{\mathbb{E}}_{t_{l-1}}\{y\}}{1 + \lambda/N(t_{l-1})}, \tag{2}$$

*Corresponding author: Bastian Pfeifer (bastian.pfeifer@medunigraz.at).

where λ is a hyperparameter chosen by the user, for example by cross validation. HS maintains the tree structure, and only modifies the prediction over each leaf node.

III. Proposed method

A. Intuition

...

B. Mathematical formulation

$$\mathbf{B}_{posterior}(\alpha, \beta) = \mathbf{B}_{prior}(\alpha, \beta) + \sum_{l=0}^{L} \mathbf{B}_{t_l}(\alpha + \#y(0), \beta + \#y(1)) \tag{3}$$

$$\hat{f}(\mathbf{x}) = \frac{\alpha}{\alpha + \beta} \tag{4}$$

or

$$\hat{f}(\mathbf{x}) = \mathbf{PPF}(\frac{\alpha}{\alpha + \beta} | \mathbf{B}_{posterior}(\alpha, \beta)) \tag{5}$$

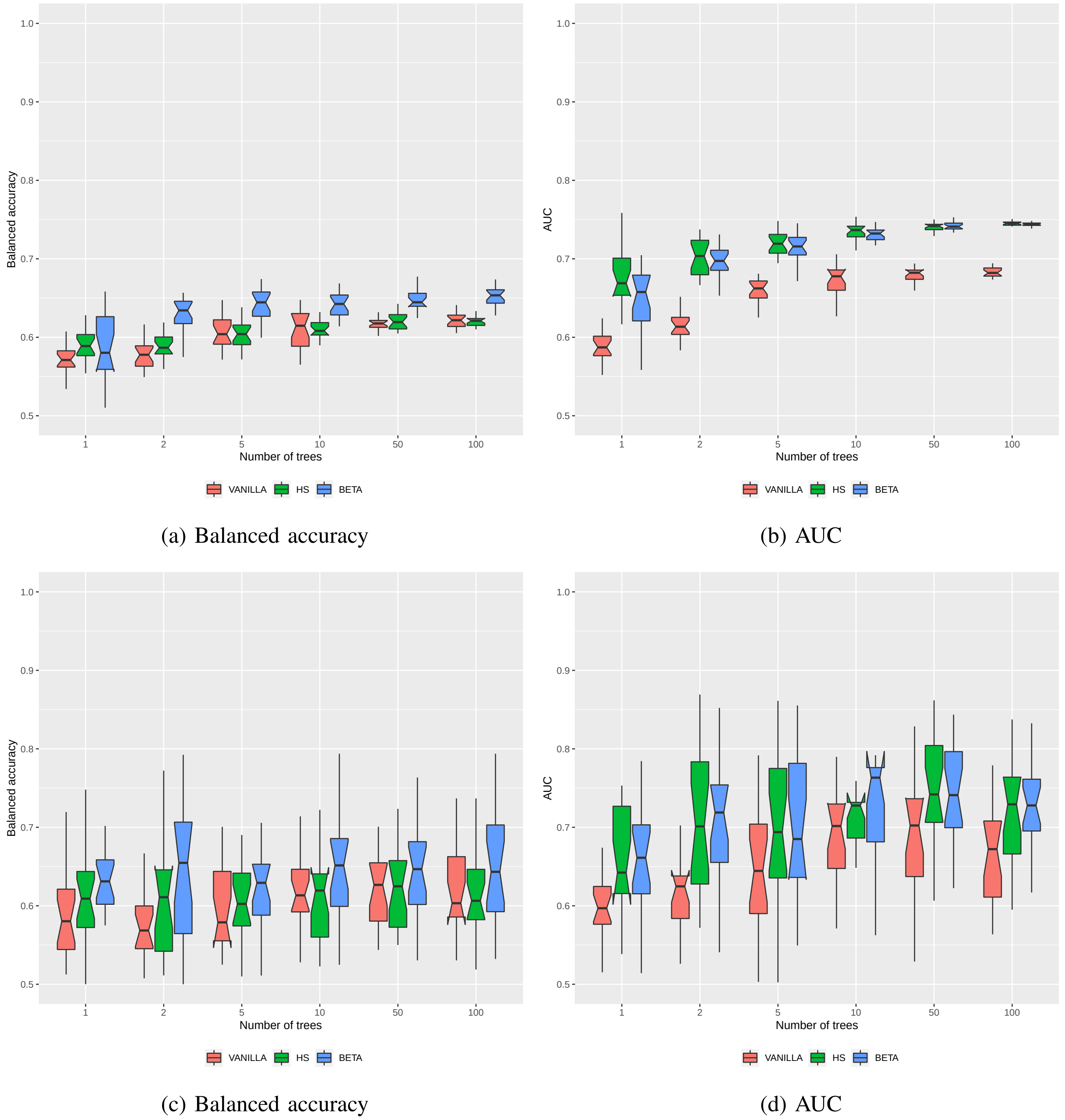
IV. Evaluation

V. Results and discussion

-

References

[1] A. Agarwal, Y. S. Tan, O. Ronen, C. Singh, and B. Yu, “Hierarchical shrinkage: Improving the accuracy and interpretability of tree-based models.” in International Conference on Machine Learning. PMLR, 2022, pp. 111–135.



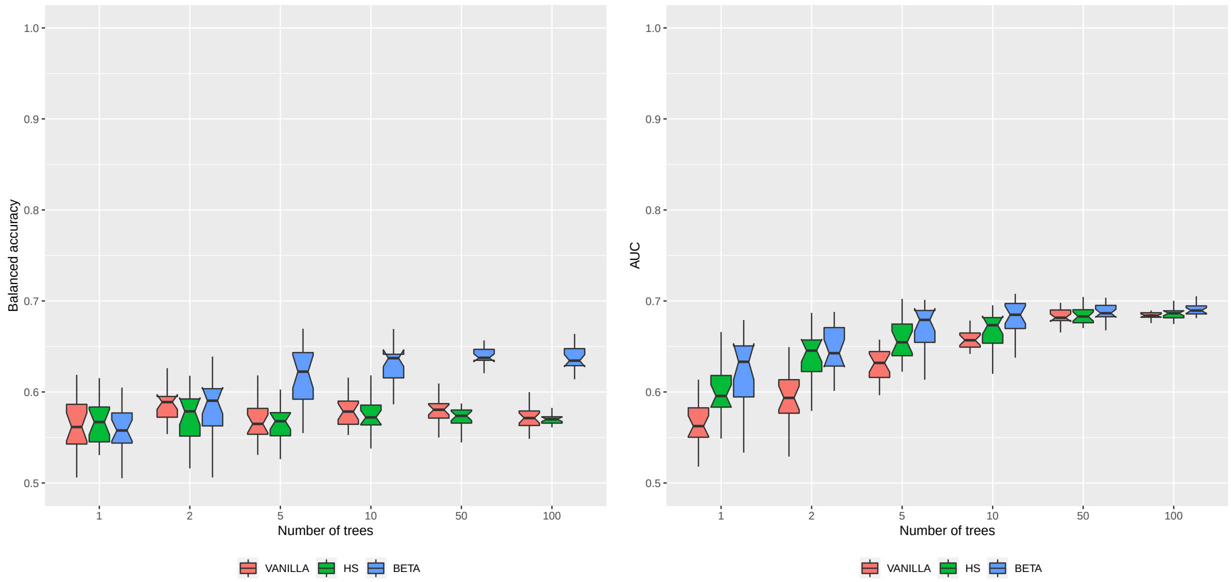
(a) Balanced accuracy

(b) AUC

(c) Balanced accuracy

(d) AUC

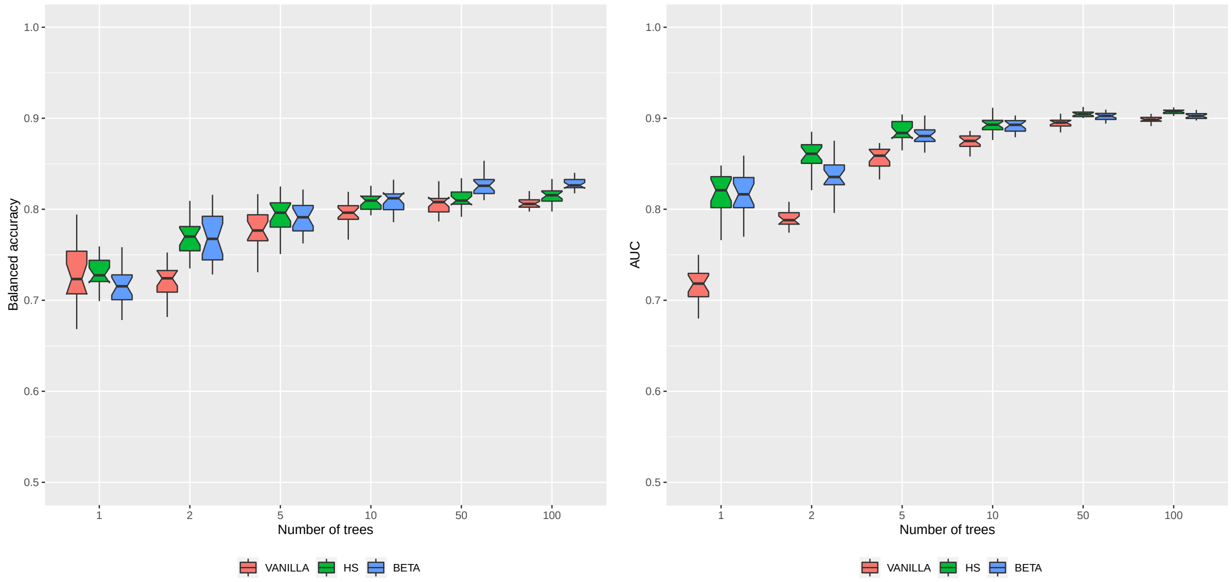
Fig. 1: Breast cancer dataset. (a) and (b) 20 times 5-fold crossvaliation on the whole dataset. (c) and (d) 20 times evaluation on independent test dataset.



(a) Balanced accuracy

(b) AUC

Fig. 2: Habermann dataset. (a) and (b) 20 times 5-fold crossvaliation on the whole dataset.



(a) Balanced accuracy

(b) AUC

Fig. 3: Heart cancer dataset. 20 times 5-fold crossvaliation on the whole dataset.