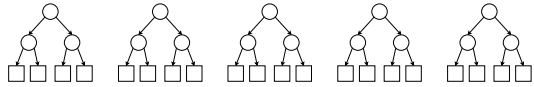
Towards Better Decision Forests: Forest Alternating Optimization

Miguel Á. Carreira-Perpiñán Magzhan Gabidolla Arman Zharmagambetov[1]

EECS, University of California, Merced
[1]currently at Meta AI (FAIR)

Tag: TUE-PM-331

Overview



$$\boldsymbol{\tau}_1(\mathbf{x}) \quad \boldsymbol{\tau}_2(\mathbf{x}) \quad \boldsymbol{\tau}_3(\mathbf{x}) \quad \boldsymbol{\tau}_4(\mathbf{x}) \quad \boldsymbol{\tau}_5(\mathbf{x})$$

$$\mathbf{F}(\mathbf{x}) = \boldsymbol{\tau}_1(\mathbf{x}) + \boldsymbol{\tau}_2(\mathbf{x}) + \boldsymbol{\tau}_3(\mathbf{x}) + \boldsymbol{\tau}_4(\mathbf{x}) + \boldsymbol{\tau}_5(\mathbf{x})$$

- Decision forests are among the most accurate models in machine learning.
- Unlike bagging or boosting, we show, for the first time, we can learn a forest by optimizing a desirable loss and regularization jointly over all its parameters.
- Our algorithm, Forest Alternating Optimization, is based on defining a forest as a parametric model with a fixed number of trees and structure (rather than adding trees indefinitely as in bagging or boosting).
- It iteratively updates each tree in alternation so that the objective function decreases monotonically. The algorithm is so effective at optimizing that it easily overfits, but this can be corrected by averaging.
- The result is a forest that consistently exceeds the accuracy of the state-of-the-art while using fewer, smaller trees.

Introduction

- **Decision forests (ensembles of decision trees)** are widely recognized as among the most accurate ML models for many tasks.
- However, neither the individual trees nor the forest are constructed to optimize a specific loss function.
- Here we propose an algorithm that optimize an objective function over all the forest parameters jointly, in all cases consistently improving over the state-of-the-art (such as XGBoost or LightGBM).

Tree Alternating Optimization (TAO)

- A scalable algorithm that can take a tree of arbitrary but parametric form and monotonically decrease an objective function of the form loss + regularization:

$$\min_{\boldsymbol{\tau}} \sum_n L(\mathbf{y}_n, \boldsymbol{\tau}(\mathbf{x}_n; \{\mathbf{w}_i\})) + \lambda \sum_{i \in \text{ nodes of } \boldsymbol{\tau}} \phi(\mathbf{w}_i) \tag{1}$$

- We focus on oblique trees (which are more powerful than axis-aligned ones):
  - Decision nodes: (sparse) hyperplane
  - Leaf nodes: constant label or value

Tree Alternating Optimization (TAO) cont.

No gradient descent (the tree defines a piecewise constant function) but alternating optimization over the nodes. Based on two theorems:

- Separability condition: the objective function separates over nodes which are not descendant of each other.
- Reduced problem over a node: optimizing over a node's parameters takes a special form that can be solved exactly or approximately:
  - decision node: weighted 0/1 loss binary classification
  - leaf node: majority vote or average

Forest: TAO + bagging/boosting

Using TAO as base learner with any ensembling mechanism results in better forests (higher accuracy, fewer trees):

- Bagging
  - *Smaller, more accurate regression forests using tree alternating optimization.* ICML 2020
  - *Ensembles of bagged TAO trees consistently improve over random forests, AdaBoost and gradient boosting.* FODS 2020.
- AdaBoost
  - *Improved multiclass AdaBoost for image classification: The role of tree optimization.* ICIP 2021
  - *Improved multiclass AdaBoost using sparse oblique decision trees.* IJCNN 22.
- Gradient Boosting (GB)
  - *Pushing the envelope of gradient boosting forests via globally-optimized oblique trees.* CVPR 22.

Forest Alternating Optimization (FAO)

- Bagging or boosting do not optimize the forest jointly. Trees are added independently (bagging) or greedily (boosting).
- In this work, we take this one step further and optimize globally over all the parameters (decision & leaf nodes) of a forest having a fixed number of trees of given structure, monotonically decreasing an objective function of the form loss + regularization:

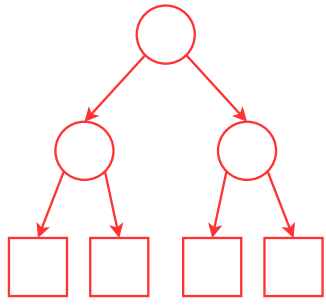
$$\min_{\boldsymbol{\tau}_1,\dots,\boldsymbol{\tau}_T} \sum_n L(\mathbf{y}_n, \mathbf{F}(\mathbf{x}_n)) + \lambda \sum_{t=1}^{T} \sum_{i \in \text{ nodes of } \boldsymbol{\tau}_t} \phi(\mathbf{w}_{ti}) \tag{2}$$

where $\mathbf{F}(\mathbf{x}) = \sum_{t=1}^{T} \boldsymbol{\tau}_t(\mathbf{x})$ is a forest of T trees.

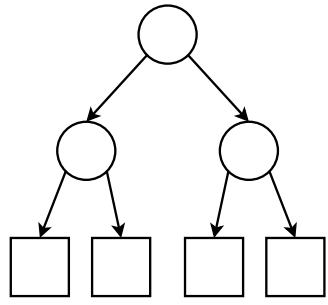
Forest Alternating Optimization (FAO) cont.

- Gradient descent is not applicable because the forest makes hard decisions.
- Therefore, we perform alternating optimization over trees:
  - If we fix all trees but one, the resulting problem over that tree can be optimized by TAO.
  - Also, if we fix all the decision nodes of all the trees, the resulting problem over all the trees' leaves can be optimized exactly.

Alternating optimization: first tree



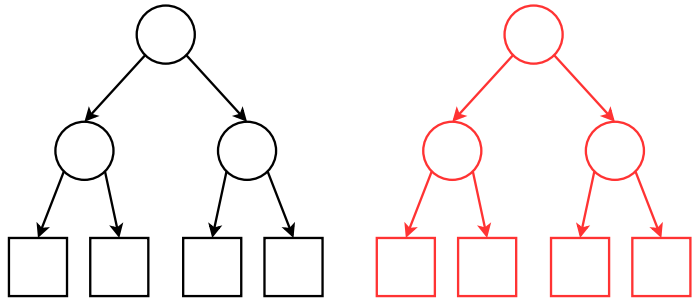
$\tau_1(\mathbf{x})$



$\tau_2(\mathbf{x})$

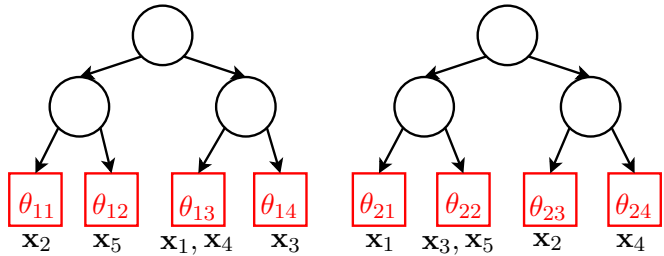
$$\min_{\tau_1} \left[\sum_{n=1}^{N} \left(y_n - \underbrace{(\tau_1(\mathbf{x}_n) + \tau_2(\mathbf{x}_n))}_{F(\mathbf{x}_n)} \right)^2 \right] \Leftrightarrow \min_{\tau_1} \left[\sum_{n=1}^{N} \left(\underbrace{y_n - \tau_2(\mathbf{x}_n)}_{y_n^1} - \tau_1(\mathbf{x}_n) \right)^2 \right]$$

Alternating optimization: second tree



$$\min_{\tau_2} \left[\sum_{n=1}^{N} \left(y_n - \underbrace{(\tau_1(\mathbf{x}_n) + \tau_2(\mathbf{x}_n))}_{F(\mathbf{x}_n)} \right)^2 \right] \Leftrightarrow \min_{\tau_2} \left[\sum_{n=1}^{N} \left(\underbrace{y_n - \tau_1(\mathbf{x}_n)}_{y_n^2} - \tau_2(\mathbf{x}_n) \right)^2 \right]$$

Alternating optimization: all leaves



$$\begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \theta_{11} \\ \theta_{12} \\ \theta_{13} \\ \theta_{14} \\ \theta_{21} \\ \theta_{22} \\ \theta_{23} \\ \theta_{24} \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{pmatrix}$$

FAO: optimization ability

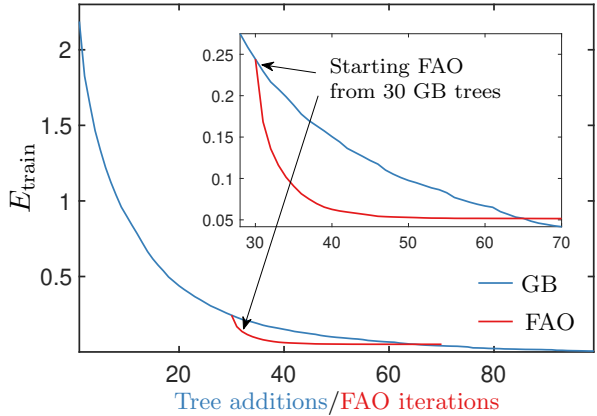


Figure: Optimization ability of FAO on the cpuact dataset. The errors are RMSE. All trees are oblique and complete of depth $\Delta = 6$. *Left:* optimizing 30 trees (initialized from GB) with FAO can exceed the performance of 60 GB trees with no shrinkage.

FAO: overfitting

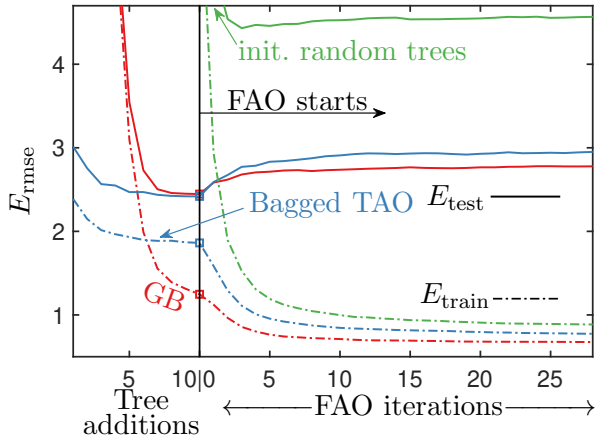
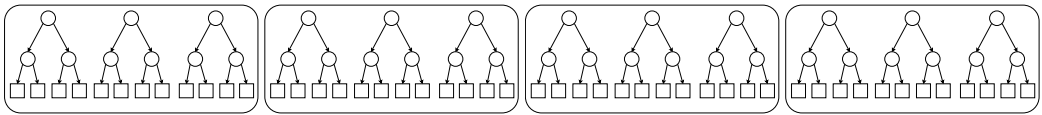


Figure: Illustration of overfitting: starting FAO with 10 trees initialized from Gradient Boosting, Bagging, averaged small FAO forests or random initialization results in a significant increase in test error.

Averaging multiple FAO forests

- We experimentally find that averaging multiple FAO forests solves the overfitting problem, and produces more accurate results than bagging and boosting.

$$\bar{\mathbf{F}}(\mathbf{x}) = \frac{1}{Q}\sum_{q=1}^{Q}\mathbf{F}_q(\mathbf{x}) \tag{3}$$



$\mathbf{F}_1(\mathbf{x}, \mathbf{\Theta}_1)$ $\mathbf{F}_2(\mathbf{x}, \mathbf{\Theta}_2)$ $\mathbf{F}_3(\mathbf{x}, \mathbf{\Theta}_3)$ $\mathbf{F}_4(\mathbf{x}, \mathbf{\Theta}_4)$

$$\bar{\mathbf{F}}(\mathbf{x}) = \frac{1}{4}\left(\mathbf{F}_1(\mathbf{x}, \mathbf{\Theta}_1) + \mathbf{F}_2(\mathbf{x}, \mathbf{\Theta}_2) + \mathbf{F}_3(\mathbf{x}, \mathbf{\Theta}_3) + \mathbf{F}_4(\mathbf{x}, \mathbf{\Theta}_4)\right)$$

Averaged FAO forests achieve best results

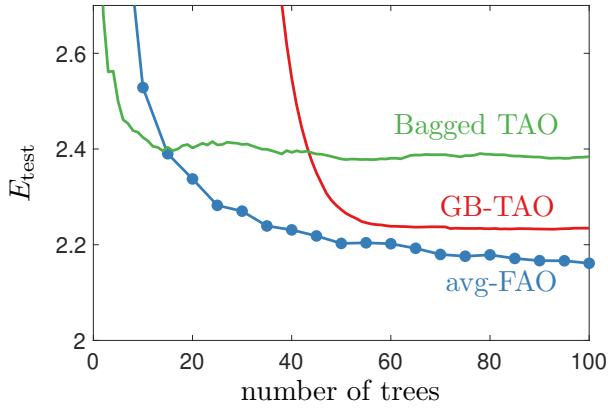


Figure: Test error for FAO, gradient boosting and bagging. Each FAO forest consists of 5 trees. Then, for FAO, 100 trees in **x**-axis means: 20 FAO forests each consisting 5 trees.

Conclusion

- We propose the first ever algorithm to jointly optimize a decision forest, Forest Alternating Optimization (FAO), and show empirically that it produces even better results in terms of overall accuracy and the number of parameters.
- **Acknowledgments.** Work supported in part by NSF award IIS–2007147.