TOWARDS BETTER DECISION FORESTS: FOREST ALTERNATING OPTIMIZATION

Miguel Á. Carreira-Perpiñán and **Magzhan Gabidolla** and **Arman Zharmagambetov**, EECS, UC Merced

1 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.

In a series of papers, we have given algorithms that optimize very general types of losses (in the sense of monotonically decreasing the loss over iterations) over a single tree (axis-aligned or oblique), in combination with popular ensembling mechanisms (bagging, AdaBoost, gradient boosting), and here over all the forest parameters jointly, in all cases consistently improving over the state-of-the-art (such as XGBoost or LightGBM).

Work partially supported by NSF award IIS–2007147.

2 SINGLE TREE: 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:

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

We focus on oblique trees (which are far more powerful than axis-aligned ones):

- Decision nodes: (sparse) hyperplane
- Leaf nodes: constant label or value

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

TAO operates on an initial tree structure. The final structure is usually a subset of this because pruning occurs automatically via a ℓ_1 penalty on the decision nodes' weights: $\phi(\mathbf{w}_i) = \|\mathbf{w}_i\|_1$. This also sparsifies the decision hyperplanes.

The table below shows how a single TAO tree improves upon the traditional CART on test accuracy for several classification benchmarks:

Algorithm	MNIST	Letter	SensIT	Pendigits	Spambase
CART	88.05	86.07	81.00	91.62	89.62
TAO	94.74	90.41	85.44	96.80	93.31

Examples of tree-based models trainable with TAO:

- [1]: sparse oblique tree
- [4]: tree of neural nets
- [7]: softmax tree
- [8]: nonlinear embeddings with trees
- [12]: semi-supervised learning with trees
- [11]: clustering with trees

3 FOREST: TAO + BAGGING/BOOSTING

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

- bagging [2], [3]
- AdaBoost [5], [6], [10]
- gradient boosting [9]

We use sparse oblique trees, which are a much stronger learner than the traditional axis-aligned trees:

- more powerful model: oblique rather than axis-aligned tree
- better optimization: TAO rather than CART/C5.0/etc.

This results in more accurate forests; the ensemble is diverse enough.

The table below shows how simple TAO trees in bagging outperform other established tree ensembles for regression problems.

cpuact (N=8k,D=21)

Forest	E_{test}	T	Δ
CART	3.63±0.32	1	25
Bagging-TAO	2.71±0.04	1	6
Random Forest	2.62±0.04	100	36
AdaBoost	2.61±0.16	100	10
Random Forest	2.60±0.01	1k	37
XGBoost	2.60±0.00	100	10
XGBoost	2.57±0.00	1k	10
AdaBoost	2.56±0.11	1k	10
Bagging-TAO	2.39±0.05	30	7

CT slice (N=54k,D=384)

Forest	E_{test}	T	Δ
CART	2.71±0.06	1	51
Bagging-TAO	1.54±0.05	1	7
AdaBoost	1.48±0.03	100	10
XGBoost	1.45±0.00	100	10
AdaBoost	1.31±0.01	1k	10
XGBoost	1.18±0.00	1k	10
Random Forest	1.03±0.01	100	71
Random Forest	0.97±0.01	1k	78
Bagging-TAO	0.89±0.02	30	7

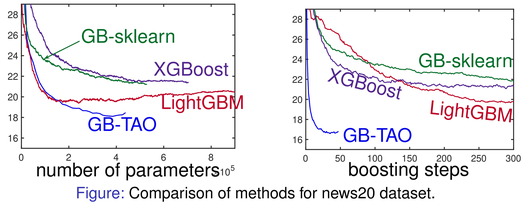


Figure: Comparison of methods for news20 dataset.

However, while each individual tree is well optimized, the forest is not. Trees are added independently (bagging) or greedily (boosting).

References:

[1] Alternating optimization of decision trees, with application to learning sparse oblique trees, *NEURIPS 2018*.
[2] Smaller, more accurate regression forests using tree alternating optimization. *ICML 2020*.
[3] Ensembles of bagged TAO trees consistently improve over random forests, AdaBoost and gradient boosting. *FODS 2020*.
[4] Learning a tree of neural nets. *ICASSP 21*.
[5] Improved boosted regression forests through non-greedy tree optimization. *IJCNN 21*.
[6] Improved multiclass AdaBoost for image classification: The role of tree optimization. *ICIP 2021*.
[7] Softmax tree: An accurate, fast classifier when the number of classes is large. *EMNLP 2021*.
[8] Learning interpretable, tree-based projection mappings for nonlinear embeddings. *AISTATS 2022*.
[9] Pushing the envelope of gradient boosting forests via globally-optimized oblique trees. *CVPR 22*.
[10] Improved multiclass AdaBoost using sparse oblique decision trees. *IJCNN 22*.
[11] Optimal interpretable clustering using oblique decision trees. *KDD 2022*.
[12] Semi-supervised learning with decision trees: Graph Laplacian tree alternating optimization. *NEURIPS 2022*.

4 FOREST ALTERNATING OPTIMIZATION (FAO)

JUNE 18-22, 2023 CVPR VANCOUVER, CANADA

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*:

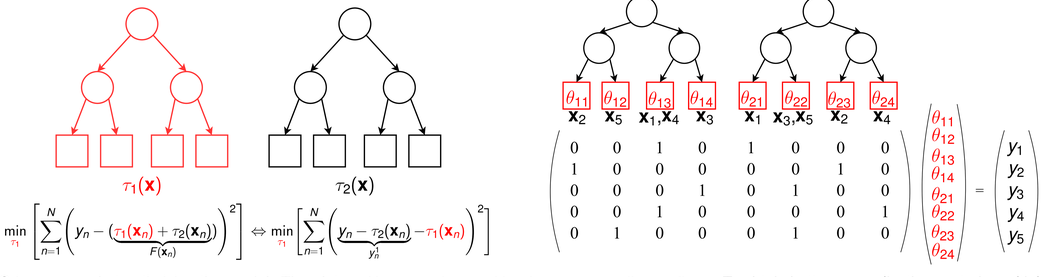
$$\min_{\tau_1,\dots,\tau_T} \sum_{n} L(\mathbf{y}_n, \mathbf{F}(\mathbf{x}_n)) + \lambda \sum_{t=1}^{T} \sum_{i \in \text{ nodes of } \tau_t} \phi(\mathbf{w}_{ti}) \qquad (2)$$

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

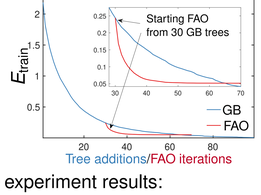
Alternating optimization over trees:

- 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.

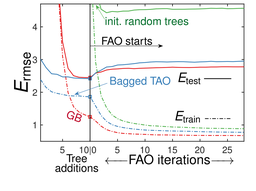
As with a single tree, each individual tree's structure is still pruned automatically via an ℓ_1 penalty on the decision nodes' weights.



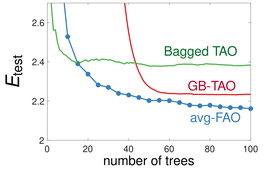
FAO is very good at optimizing the model. The plot shows how 30 trees trained with FAO can exceed the training accuracy of 60 greedily added GB trees.



However, the resulting forest can easily overfit: the plot shows the increasing test error of FAO trained on 10 trees of various initialization.



To obtain better generalization we train multiple small FAO forests on different random initializations, and just average their predictions.



Some experiment results:

MNIST ($N = 60k, D = 784, K = 10$)

Forest	E_{test} (%)	#pars.	T	Δ
SPORF	2.89±0.04	(143M)	1k	50
XGBoost	2.20±0.00	107k	1k	6
LightGBM	2.02±0.00	121k	1k	10
XGBoost	1.91±0.00	505k	10k	6
GB-TAO	1.65±0.02	3M	500	7
LightGBM	1.62±0.00	642k	10k	21
GB-TAO	1.55±0.02	7.2M	1.4k	7
avg-FAO	1.48±0.06	658k	60	6
avg-FAO	1.39±0.04	968k	90	6
avg-FAO	1.33±0.04	4.9M	300	8

SUSY ($N = 4.5M, D = 18, K = 2$)

Forest	E_{test} (%)	#pars.	T	Δ
SPORF	19.91	(271M)	100	102
SPORF	19.73	(2.7B)	1k	109
XGBoost	19.63	151k	300	8
XGBoost	19.62	196k	100	10
LightGBM	19.62	153k	100	23
LightGBM	19.60	230k	300	21
XGBoost	19.59	2.0M	1k	10
LightGBM	19.57	1.5M	1k	23
avg-FAO	19.51	233k	50	8
avg-FAO	19.50	459k	100	8

casp ($N = 45k, D = 9$)

Forest	$E_{\text{test.rmse}}$	#pars.	T	Δ
XGBoost	3.66±0.00	119k	100	10
XGBoost	3.58±0.00	793k	1k	10
LightGBM	3.54±0.00	153k	100	114
GB-TAO	3.49±0.01	256k	50	12
LightGBM	3.48±0.00	766k	1k	109
avg-FAO	3.45±0.02	359k	50	12
GB-TAO	3.43±0.00	481k	100	12
avg-FAO	3.40±0.01	711k	100	12
GB-TAO	3.39±0.01	887k	200	12
avg-FAO	3.37±0.01	1.4M	200	12