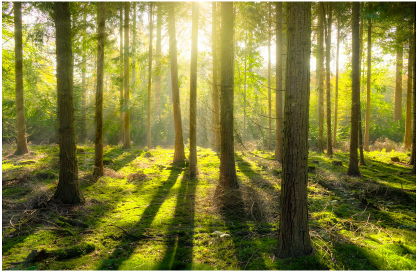
Born-Again Tree Ensembles

Thibaut Vidal[1], Maximilian Schiffer[2]
with the support of Toni Pacheco[1]

[1] Computer Science Department, Pontifical Catholic University of Rio de Janeiro
[2] TUM School of Management, Technical University of Munich



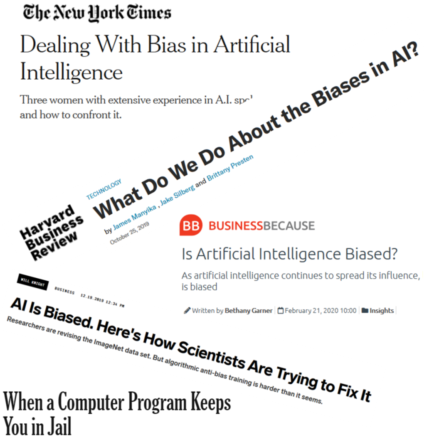
Our Concept

- We propose the first exact algorithm that transforms a tree ensemble into a born-again decision tree (BA tree) that is:
  - **Optimal** in size (number of leaves or depth), and
  - **Faithful** to the tree ensemble **in its entire feature space**.
- The BA tree is effectively **a different representation of the same decision function**.

We seek a single —minimal-size— decision tree that faithfully reproduces the decision function of the random forest.

Why interpretability is critical

- Machine learning is becoming widespread, even for high stakes decisions:
  - Recurrence predictions in medicine
  - Custody decisions in criminal justice
  - Credit risk evaluations...
- Some studies suggest that there is a *trade-off* between algorithm accuracy and interpretability
  - This is not always the case [1]



We need interpretable and accurate algorithms to leverage the best of both worlds

Related Research

Thinning tree ensembles

Pruning some weak learners [18, 21, 22, 25]

Replacing the tree ensemble by a simpler classifier [2, 7, 19, 23]

Rule extraction via bayesian model selection [14]

Extracting a single tree from a tree ensemble by actively sampling training points [3, 4]

Thinning neural networks

Model compression and knowledge distillation [8, 15]: Using a "teacher" to train a compact "student' with similar knowledge.

Creating soft decision trees from a neural network [11], or decomposing the gradient in knowledge distillation [12].

Simplifying neural networks [9, 10] or synthetizing them as an interpretable simulation model [17].

Optimal decision trees

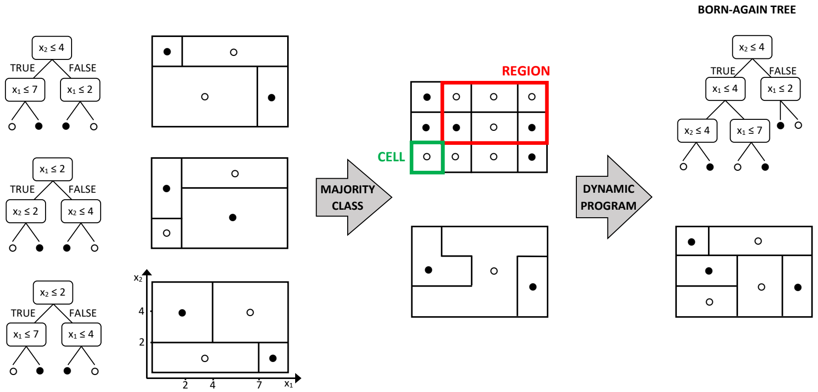
Linear programming algorithms have been exploited to find linear combination splits [5].

Extensive study of global optimization methods, based on mixed-integer programming or dynamic programming, for the construction of optimal decision trees [6, 13, 16, 20, 24]

Thinning algorithms do not guarantee faithfulness

Methodology

Construction Process



Methodology

Problem 1: Born-Again Tree Ensemble

Given a tree ensemble $\mathcal{T}$, we search for a decision tree T of **minimal size** such that $F_T(\mathbf{x}) = F_{\mathcal{T}}(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^p$.

Theorem 1

Problem 1 is NP-hard when optimizing depth, number of leaves, or any hierarchy of these two objectives.

Verifying that a given solution is feasible (faithful) is NP-hard.

Methodology

Dynamic Program 1

Let $\Phi(\mathbf{z}^{\mathrm{L}}, \mathbf{z}^{\mathrm{R}})$ be the depth of an optimal born-again decision tree for a region $(\mathbf{z}^{\mathrm{L}}, \mathbf{z}^{\mathrm{R}})$. Then:

$$\Phi(\mathbf{z}^{\mathrm{L}}, \mathbf{z}^{\mathrm{R}}) = \begin{cases} 0 \quad \text{if} \quad \textsc{id}(\mathbf{z}^{\mathrm{L}}, \mathbf{z}^{\mathrm{R}}) \\ \min\limits_{1 \leq j \leq p} \left\{ \min\limits_{z_j^{\mathrm{L}} \leq l < z_j^{\mathrm{R}}} \left\{ 1 + \max\{\Phi(\mathbf{z}^{\mathrm{L}}, \mathbf{z}_{jl}^{\mathrm{R}}), \Phi(\mathbf{z}_{jl}^{\mathrm{L}}, \mathbf{z}^{\mathrm{R}})\} \right\} \right\}, \end{cases}$$

in which $\textsc{id}(\mathbf{z}^{\mathrm{L}}, \mathbf{z}^{\mathrm{R}})$ takes value TRUE iff all cells $\mathbf{z}$ such that $\mathbf{z}^{\mathrm{L}} \leq \mathbf{z} \leq \mathbf{z}^{\mathrm{R}}$ are from the same class (i.e. base case).

Issue 1

Detecting base cases

Issue 2

Numerous recursive calls

Circumventing Issue 1

We tried several alternatives to efficiently check base cases. The best approach we found consisted in including the base case evaluation within the DP:

Dynamic Program 2

Let $\Phi(\mathbf{z}^{\mathrm{L}}, \mathbf{z}^{\mathrm{R}})$ be the depth of an optimal born-again decision tree for a region $(\mathbf{z}^{\mathrm{L}}, \mathbf{z}^{\mathrm{R}})$. Then:

$$\Phi(\mathbf{z}^{\mathrm{L}}, \mathbf{z}^{\mathrm{R}}) = \min_{1 \le j \le p} \left\{ \min_{z_j^{\mathrm{L}} \le l < z_j^{\mathrm{R}}} \left\{ \mathbb{1}_{jl}(\mathbf{z}^{\mathrm{L}}, \mathbf{z}^{\mathrm{R}}) + \max\{\Phi(\mathbf{z}^{\mathrm{L}}, \mathbf{z}_{jl}^{\mathrm{R}}), \Phi(\mathbf{z}_{jl}^{\mathrm{L}}, \mathbf{z}^{\mathrm{R}})\} \right\} \right\}$$

where
$$\mathbb{1}_{jl}(\mathbf{z}^{\mathrm{L}}, \mathbf{z}^{\mathrm{R}}) = \begin{cases} 0 & \text{if} \quad \Phi(\mathbf{z}^{\mathrm{L}}, \mathbf{z}_{jl}^{\mathrm{R}}) = \Phi(\mathbf{z}_{jl}^{\mathrm{L}}, \mathbf{z}^{\mathrm{R}}) = 0 \\ & \text{and} \;\; F_{\mathcal{T}}(\mathbf{z}^{\mathrm{L}}) = F_{\mathcal{T}}(\mathbf{z}^{\mathrm{R}}); \\ 1 & \text{otherwise.} \end{cases}$$

Circumventing Issue 2

We exploit two simple properties to reduce the number of recursive calls:

Property 2

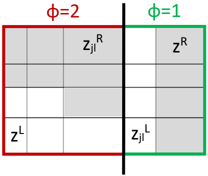
If $\Phi(\mathbf{z}^{\mathrm{L}}, \mathbf{z}^{\mathrm{R}}_{jl}) \geq \Phi(\mathbf{z}^{\mathrm{L}}_{jl}, \mathbf{z}^{\mathrm{R}})$ then for all $l' > l$:

$$\begin{aligned}\mathbb{1}_{jl}(\mathbf{z}^{\mathrm{L}}, \mathbf{z}^{\mathrm{R}}) &+ \max\{\Phi(\mathbf{z}^{\mathrm{L}}, \mathbf{z}^{\mathrm{R}}_{jl}), \Phi(\mathbf{z}^{\mathrm{L}}_{jl}, \mathbf{z}^{\mathrm{R}})\} \\ &\leq \mathbb{1}_{jl'}(\mathbf{z}^{\mathrm{L}}, \mathbf{z}^{\mathrm{R}}) + \max\{\Phi(\mathbf{z}^{\mathrm{L}}, \mathbf{z}^{\mathrm{R}}_{jl'}), \Phi(\mathbf{z}^{\mathrm{L}}_{jl'}, \mathbf{z}^{\mathrm{R}})\}\end{aligned}$$

Property 3

If $\Phi(\mathbf{z}^{\mathrm{L}}, \mathbf{z}^{\mathrm{R}}_{jl}) \leq \Phi(\mathbf{z}^{\mathrm{L}}_{jl}, \mathbf{z}^{\mathrm{R}})$ then for all $l' < l$:

$$\begin{aligned}\mathbb{1}_{jl}(\mathbf{z}^{\mathrm{L}}, \mathbf{z}^{\mathrm{R}}) &+ \max\{\Phi(\mathbf{z}^{\mathrm{L}}, \mathbf{z}^{\mathrm{R}}_{jl}), \Phi(\mathbf{z}^{\mathrm{L}}_{jl}, \mathbf{z}^{\mathrm{R}})\} \\ &\leq \mathbb{1}_{jl'}(\mathbf{z}^{\mathrm{L}}, \mathbf{z}^{\mathrm{R}}) + \max\{\Phi(\mathbf{z}^{\mathrm{L}}, \mathbf{z}^{\mathrm{R}}_{jl'}), \Phi(\mathbf{z}^{\mathrm{L}}_{jl'}, \mathbf{z}^{\mathrm{R}})\}\end{aligned}$$



Allowing us to search for the best hyperplane level for each feature with a binary search.

Experimental Analyses

Datasets

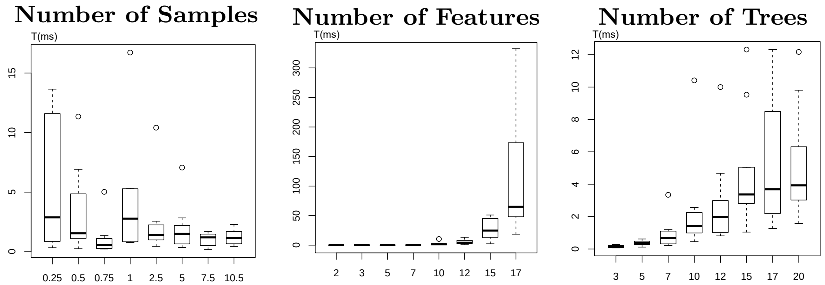
We used datasets from diverse applications, including medicine (BC, PD), criminal justice (COMPAS), and credit scoring (FICO).

Data set	n	p	K	CD	Src.
BC – Breast-Cancer	683	9	2	65-35	UCI
CP – COMPAS	6907	12	2	54-46	HuEtAl
FI – FICO	10459	17	2	52-48	HuEtAl
HT – HTRU2	17898	8	2	91-9	UCI
PD – Pima-Diabetes	768	8	2	65-35	SmithEtAl
SE – Seeds	210	7	3	33-33-33	UCI

Data Preparation

One-hot encoding for categorical variables.
Continuous variables binned into ten ordinal scales.
Generate training and test samples for all data sets by ten-fold cross validation.
For each fold and each dataset, generate a random forest composed of 10 trees with a depth of 3.

Scalability



Computational time(ms) of the DP as a function of the number of samples, features and trees.

Experimental Analyses

Simplicity

Depth and number of leaves of the born-again trees:

	D		L		DL	
Data set	Depth	# Leaves	Depth	# Leaves	Depth	# Leaves
BC	12.5	2279.4	18.0	890.1	12.5	1042.3
CP	8.9	119.9	8.9	37.1	8.9	37.1
FI	8.6	71.3	8.6	39.2	8.6	39.2
HT	6.0	20.2	6.3	11.9	6.0	12.0
PD	9.6	460.1	15.0	169.7	9.6	206.7
SE	10.2	450.9	13.8	214.6	10.2	261.0
Avg.	9.3	567.0	11.8	227.1	9.3	266.4

Analysis

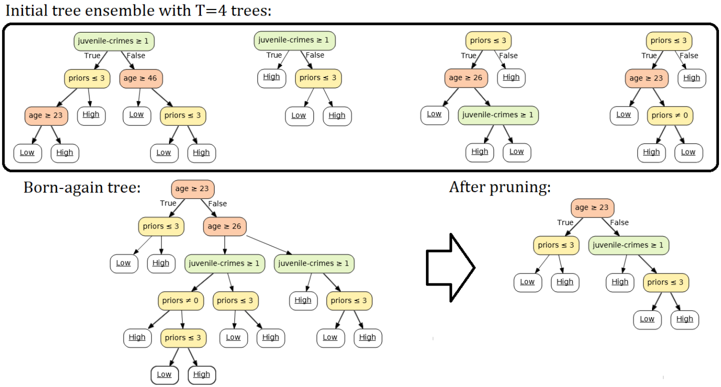
The decision function of a random forest is visibly complex
One main reason: *Incompatible feature combinations* are being represented, and the decision function of the RF is not necessarily uniform on these regions due to the other features.

Experimental Analyses

Post-Pruning

Eliminate inexpressive tree sub-regions. From bottom to top:

- Verify whether both sides of a split contain at least one sample
- Eliminate every such *empty* split



Experimental Analyses

Analysis

With post-pruning, faithfulness is no longer guaranteed per definition. We need to experimentally evaluate:

- Impact on simplicity
- Impact on accuracy

Depth and number of leaves:

	RF	BA-Tree		BA+P	
	Leaves	Depth	Leaves	Depth	Leaves
BC	61.1	12.5	2279.4	9.1	35.9
CP	46.7	8.9	119.9	7.0	31.2
FI	47.3	8.6	71.3	6.5	15.8
HT	42.6	6.0	20.2	5.1	13.2
PD	53.7	9.6	460.1	9.4	79.0
SE	55.7	10.2	450.9	7.5	21.5
Avg.	51.2	9.3	567.0	7.4	32.8

Accuracy and F1 score comparison:

	RF		BA-Tree		BA+P	
	Acc	F1	Acc	F1	Acc	F1
BC	0.953	0.949	0.953	0.949	0.946	0.941
CP	0.660	0.650	0.660	0.650	0.660	0.650
FI	0.697	0.690	0.697	0.690	0.697	0.690
HT	0.977	0.909	0.977	0.909	0.977	0.909
PD	0.746	0.692	0.746	0.692	0.750	0.700
SE	0.790	0.479	0.790	0.479	0.790	0.481
Avg.	0.804	0.728	0.804	0.728	0.803	0.729

Conclusions

- Compact representations of the decision functions of random forests, as a single —minimal size— decision tree.
- Sheds a new light on **random forests visualization and interpretability**.
- Progressing towards interpretable models is an important step towards addressing bias and data mistakes in learning algorithms.
- Optimal classifiers can be fairly complex. Indeed, BA-trees reproduce the complete decision function for *all regions of the feature space*.
  - Pruning can solve this issue
  - Heuristics can be used for datasets which are too large to be solved to optimality

Bibliography I

[1] Angelino, E., N. Larus-Stone, D. Alabi, M. Seltzer, C. Rudin. 2018. Learning certifiably optimal rule lists for categorical data. *Journal of Machine Learning Research* **18** 1–78.

[2] Bai, J., Y. Li, J. Li, Y. Jiang, S. Xia. 2019. Rectified decision trees: Towards interpretability, compression and empirical soundness. *arXiv preprint arXiv:1903.05965* .

[3] Bastani, O., C. Kim, H. Bastani. 2017. Interpretability via model extraction. *arXiv preprint arXiv:1706.09773* .

[4] Bastani, O., C. Kim, H. Bastani. 2017. Interpreting blackbox models via model extraction. *arXiv preprint arXiv:1705.08504* .

[5] Bennett, K. 1992. Decision tree construction via linear programming. *Proceedings of the 4th Midwest Artificial Intelligence and Cognitive Science Society Conference, Utica, Illinois*.

[6] Bertsimas, D., J. Dunn. 2017. Optimal classification trees. *Machine Learning* **106**(7) 1039–1082.

[7] Breiman, L., N. Shang. 1996. Born again trees. Tech. rep., University of California Berkeley.

[8] Buciluǎ, C., R. Caruana, A. Niculescu-Mizil. 2006. Model compression. *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[9] Clark, K., M.-T. Luong, U. Khandelwal, C. D. Manning, Q. V. Le. 2019. Bam! born-again multi-task networks for natural language understanding. *arXiv preprint arXiv:1907.04829* .

Bibliography II

[10] Frankle, J., M. Carbin. 2018. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635* .

[11] Frosst, N., G. Hinton. 2017. Distilling a neural network into a soft decision tree. *arXiv preprint arXiv:1711.09784* .

[12] Furlanello, Tommaso, Zachary C Lipton, Michael Tschannen, Laurent Itti, Anima Anandkumar. 2018. Born again neural networks. *arXiv preprint arXiv:1805.04770* .

[13] Günlük, O., J. Kalagnanam, M. Menickelly, K. Scheinberg. 2018. Optimal decision trees for categorical data via integer programming. *arXiv preprint arXiv:1612.03225* .

[14] Hara, S., K. Hayashi. 2016. Making tree ensembles interpretable: A bayesian model selection approach. *arXiv preprint arXiv:1606.09066* .

[15] Hinton, G., O. Vinyals, J. Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* .

[16] Hu, X., C. Rudin, M. Seltzer. 2019. Optimal sparse decision trees. *Advances in Neural Information Processing Systems*.

[17] Kisamori, K., K. Yamazaki. 2019. Model bridging: To interpretable simulation model from neural network. *arXiv preprint arXiv:1906.09391* .

[18] Margineantu, D., T. Dietterich. 1997. Pruning adaptive boosting. *Proceedings of the Fourteenth International Conference Machine Learning*.

[19] Meinshausen, N. 2010. Node harvest. *The Annals of Applied Statistics* 2049–2072.

[20] Nijssen, S., E. Fromont. 2007. Mining optimal decision trees from itemset lattices. *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Bibliography III

[21] Rokach, L. 2016. Decision forest: Twenty years of research. *Information Fusion* **27** 111–125.

[22] Tamon, C., J. Xiang. 2000. On the boosting pruning problem. *Proceedings of the 11th European Conference on Machine Learning*.

[23] Tan, H. F., G. Hooker, M. T. Wells. 2016. Tree space prototypes: Another look at making tree ensembles interpretable. *arXiv preprint arXiv:1611.07115* .

[24] Verwer, S., Y. Zhang. 2019. Learning optimal classification trees using a binary linear program formulation. *Proceedings of the AAAI Conference on Artificial Intelligence*.

[25] Zhang, Y., S. Burer, W. N. Street. 2006. Ensemble pruning via semi-definite programming. *Journal of Machine Learning Research* **7**(Jul) 1315–1338.