The Computational Curse of Big Data for Bayesian Additive Regression Trees: A Hitting Time Analysis

Yan Shuo Tan
NUS Department of Statistics and Data Science

BIRS-IASM Workshop 2023

The Computational Curse of Big Data for Bayesian Additive Regression Trees: A Hitting Time Analysis

Omer Ronen

Theo Saarinen

James Duncan

Bin Yu

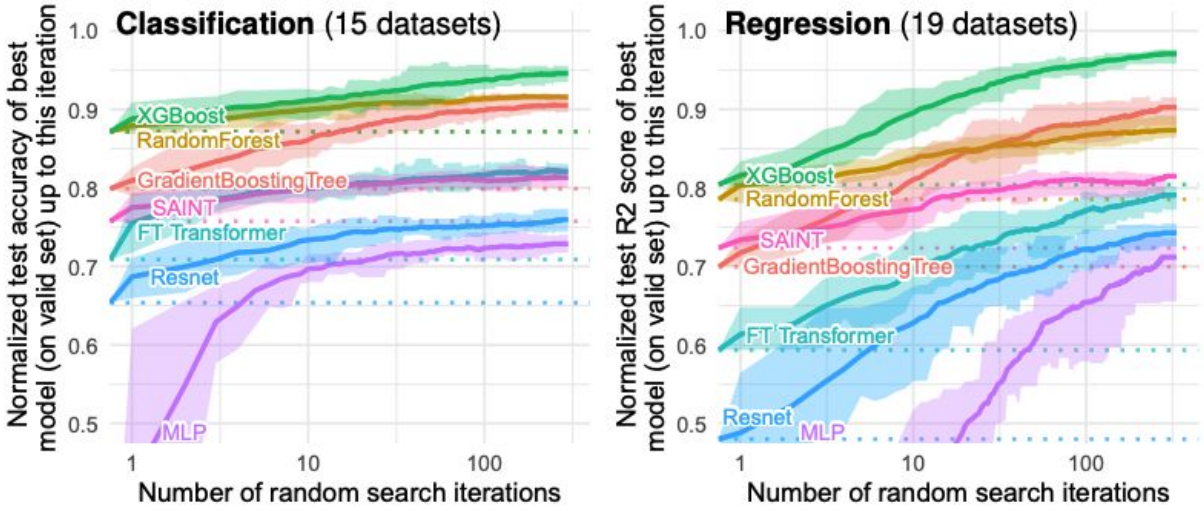
Why do tree-based models still outperform deep learning on typical tabular data?

Léo Grinsztajn
Soda, Inria Saclay
leo.grinsztajn@inria.fr

Edouard Oyallon
ISIR, CNRS, Sorbonne University

Gaël Varoquaux
Soda, Inria Saclay

Only numerical features



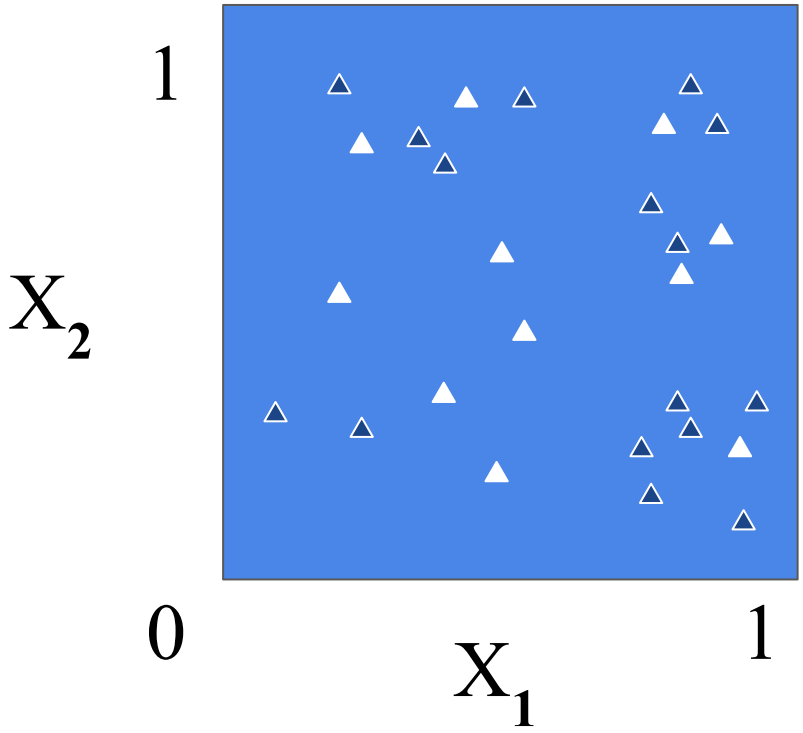
NeurIPS 2022

> … the method that performs consistently well across all dimensions is **random forests**, followed by neural nets, boosted trees, and SVMs. [11 datasets]
>
> - Caruana, Karampatziakis, Yessenalina (2008)

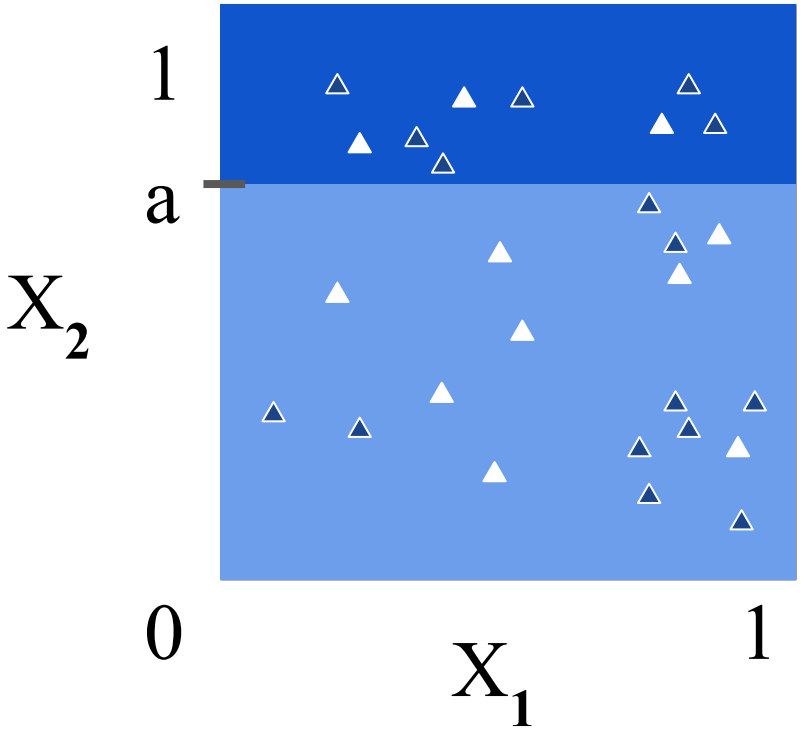
> The classifiers most likely to be the best are the **random forest** versions. [121 data sets, 179 models]
>
> - Fernandez-Delgado, Cernadas, Barro, Amorim (2014)

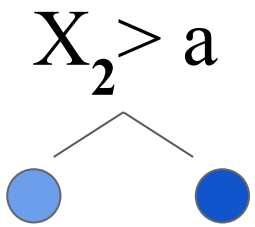
> The post-hoc test underlines the impressive performance of **Gradient Tree Boosting**, which significantly outperforms every algorithm except **Random Forest** at the $p < 0.01$ level. [165 data sets, 13 models]
>
> - Olson, Randal S., et al. (2018)

A decision tree is a piecewise constant model obtained from recursive partitioning of the covariate space

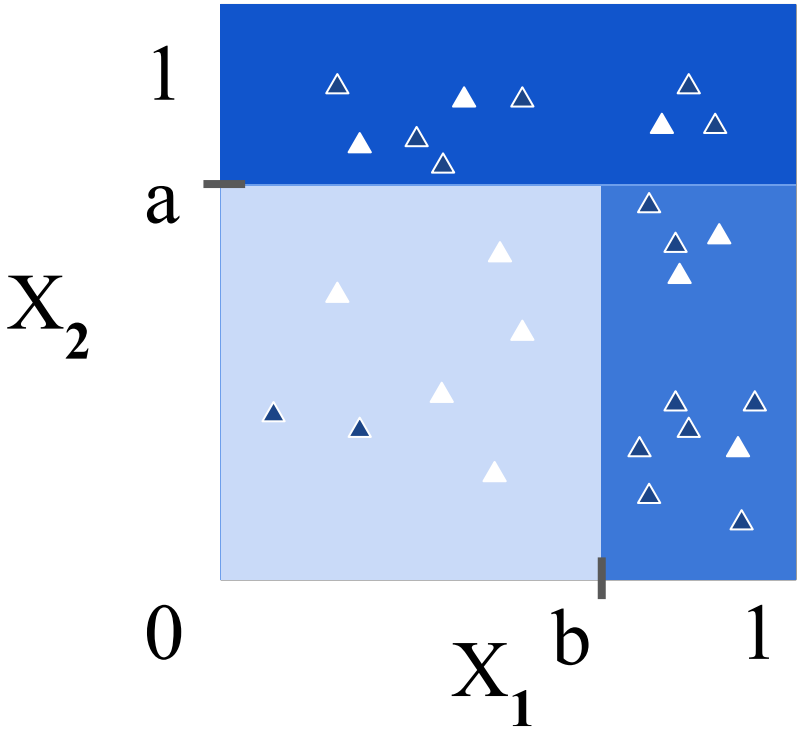


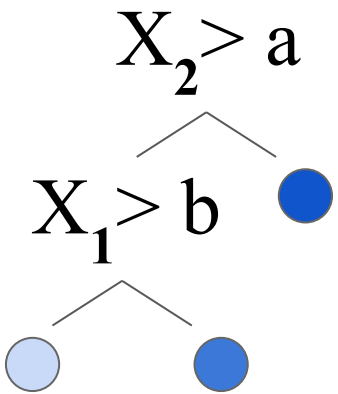
A decision tree is a piecewise constant model obtained from recursive partitioning of the covariate space



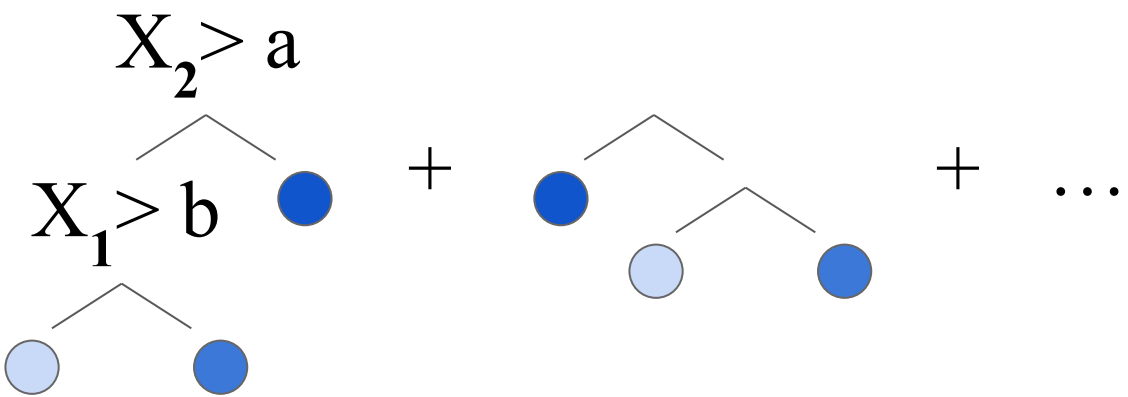


A decision tree is a piecewise constant model obtained from recursive partitioning of the covariate space





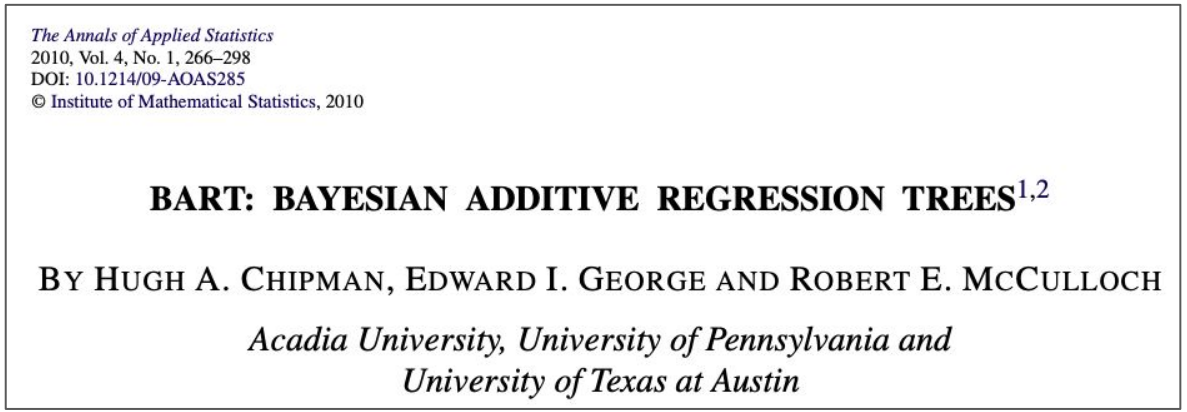
Random forests (RFs) and Gradient Boosted Trees combine decision trees in an ensemble



Drawbacks of RFs / gradient boosting

- Unclear how to perform uncertainty quantification
- Greedy splitting criterion may lead to ensemble models being stuck in local optima
- Inefficiency with additive structure *[Tan, Agarwal, Yu (2021)]*

Overcome these drawbacks using a *Bayesian* formulation of tree ensembles



The Annals of Applied Statistics
2010, Vol. 4, No. 1, 266–298
DOI: 10.1214/09-AOAS285

BART: BAYESIAN ADDITIVE REGRESSION TREES[1,2]

BY HUGH A. CHIPMAN, EDWARD I. GEORGE AND ROBERT E. MCCULLOCH

Acadia University, University of Pennsylvania and University of Texas at Austin

How does BART work?

Bayesian nonparametric regression

Step 1: Define prior on space of regression functions

Step 2: Combine prior and data likelihood to get posterior

Step 3: “Sample” from posterior using MCMC

Randomized tree ensemble method

Trees in ensemble are grown using probabilistic moves

BART has become widely used in the applied statistics community

- Social sciences *[Green and Kern (2010), Yeager et al. (2018), Dorie et al. (2019), ...]*
- Biostatistics *[Wendling et al. (2018), Starling et al. (2020), ...]*
- Several popular software implementations: `dbarts`, `BART`, `bartCause`, `bartMachine` (15K combined monthly downloads)

Theoretical analysis of BART

BART posterior has good predictive and inferential properties

- Posterior concentration around true regression function at minimax rate
  - Sobolev/Holder smoothness *[Rockova and Saha, 2019], [Linero and Yang, 2018], [Rockova and van der Pas, 2020], …*
  - Anisotropic and heterogeneous smoothness *[Jeong and Rockova, 2020], [Rockova and Rousseau, 2023], …*
- Variable selection consistency *[Linero, 2018], [Liu et al., 2021]*

However: Can only access the posterior *approximately* via MCMC.

Theoretical analysis of BART

BART posterior has good predictive and inferential properties

However: Can only access the posterior *approximately* via MCMC.

We would like to know:

- How close is the approximate posterior to the true posterior?
- How long must we run the chain to achieve convergence?

Or in technical terms, **what is the mixing time of the BART MCMC?**

Problem: MCMC chain does not mix well

"... while this algorithm is often effective, it **does not always mix well**, and recent work suggests that it can be important to *run many chains* (as many as 10 or 12) to encourage proper mixing (Carnegie 2019)..."

Bayesian additive regression trees: A review and look forward
J Hill, A Linero, J Murray - Annual Review of Statistics and Its ..., 2020 - annualreviews.org

"... *warm-start initialization* yields considerable improvement in the estimation, which may indicate inadequate chain length of BART (that is, **poor mixing**)..."

Stochastic tree ensembles for regularized nonlinear regression
J He, PR Hahn - Journal of the American Statistical Association, 2021 - Taylor & Francis

Seems to be "folklore" in the literature, but no rigorous study

Rest of this talk:

1. How does BART work?
2. How to frame computational lower bounds for BART?
3. Hitting time lower bounds for BART and practical takeaways

Part 1: How does BART work? (more details)

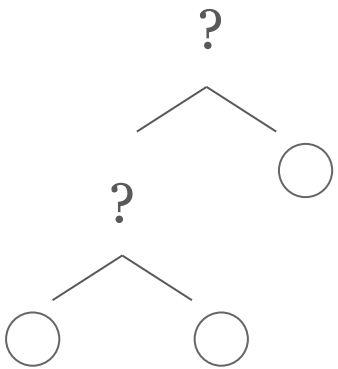
A. Parameterization of space of regression trees
B. Priors and likelihoods
C. MCMC algorithm

Parameterization of space of regression trees

Parameterization of space of regression trees

Binary tree structure $\mathcal{T}$

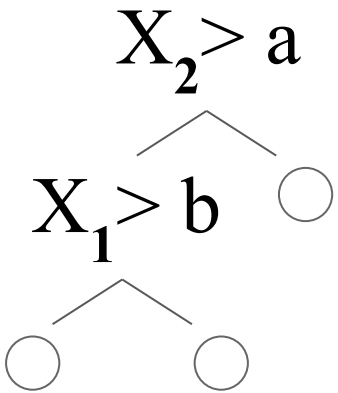
- Tree topology



Parameterization of space of regression trees

Binary tree structure $\mathcal{T}$

- Tree topology
- Splitting rule for each internal node

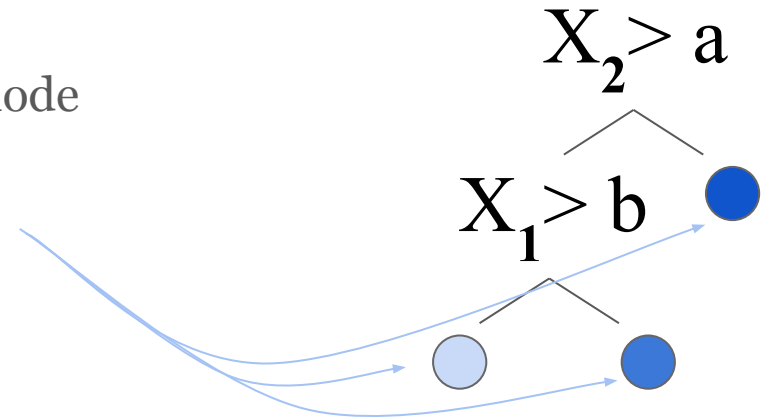


Parameterization of space of regression trees

Binary tree structure $\mathcal{T}$

- Tree topology
- Splitting rule for each internal node

Values on each leaf in partition Θ

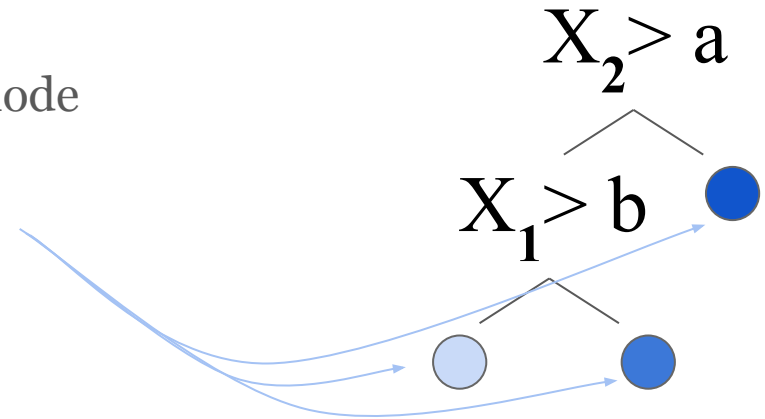


Parameterization of space of regression trees

Binary tree structure $\mathcal{T}$

- Tree topology
- Splitting rule for each internal node

Values on each leaf in partition Θ



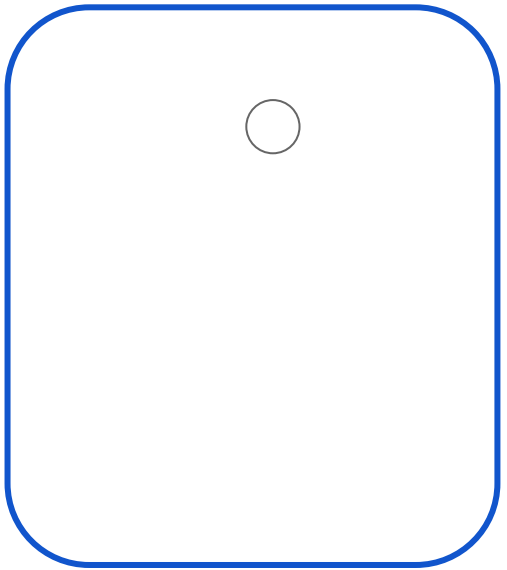
Assume covariate space is $\mathcal{X} = \{1, 2, \ldots, m\}^d$ *

*In practice, BART “discretizes” features

Prior for tree structure $\mathcal{T}$

Defined in terms of stochastic process

- Start with trivial tree



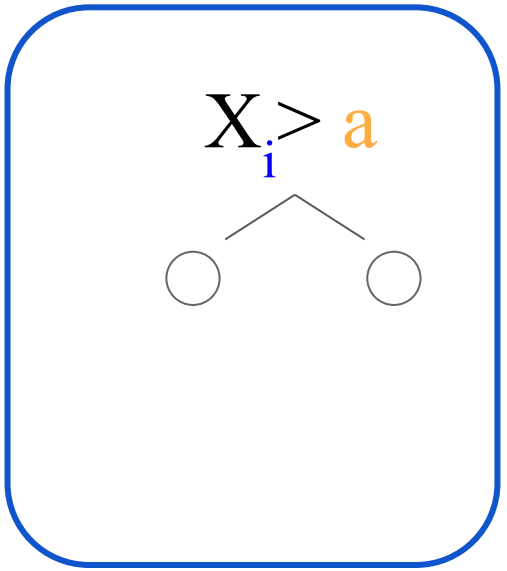
Prior for tree structure $\mathcal{T}$

Defined in terms of stochastic process

- Start with trivial tree
- With probability p, split root node (else stop)
- If node is split, draw split feature and threshold uniformly at random, i.e.

  Features: {1, 2, …. i d}

  Thresholds: {1, 2, …. a m-1}



Prior for tree structure $\mathcal{T}$

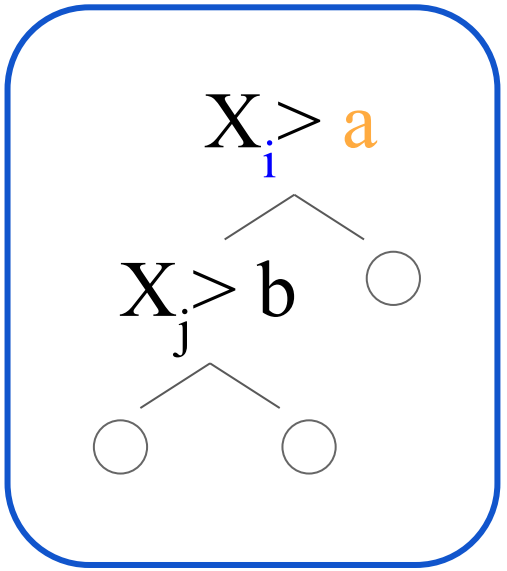
Defined in terms of stochastic process

- Start with trivial tree
- With probability *p*, split root node (else stop)
- If node is split, draw split feature and threshold uniformly at random, i.e.

  Features: {1, 2, …. i d}

  Thresholds: {1, 2, …. a m-1}

- Repeat with each newly created node



Prior for leaf values Θ

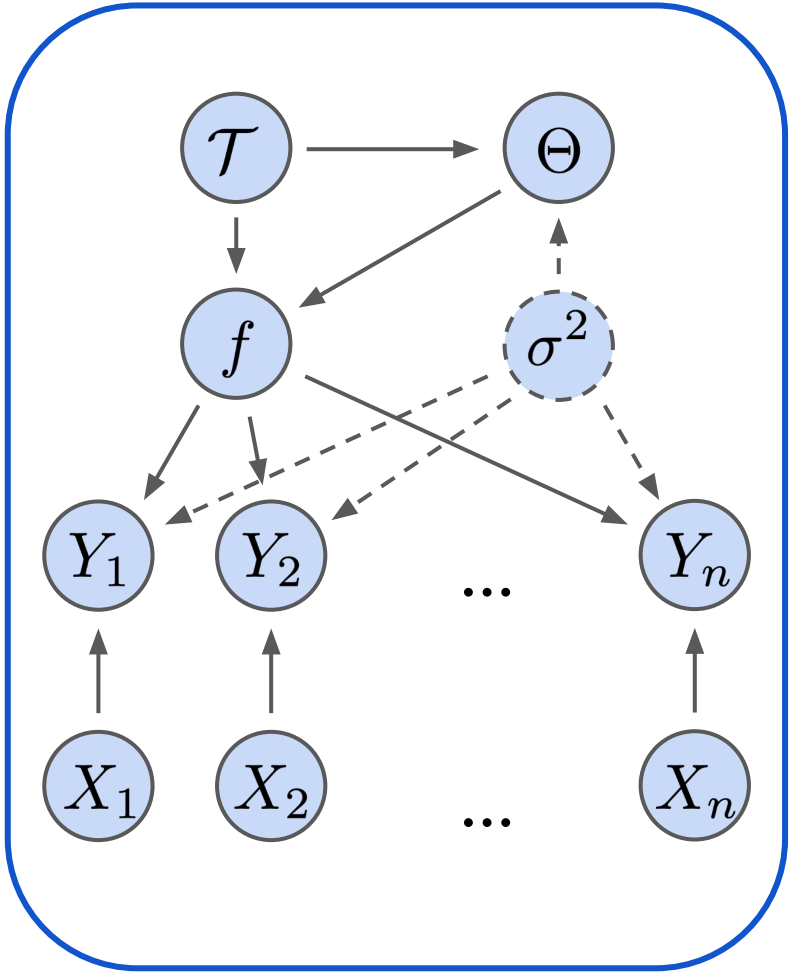
Independent Gaussian priors for the leaf values

$$\Theta|\mathcal{T} \sim \mathcal{N}(\bar{\mu}\mathbf{1}, \sigma^2\mathbf{I}_b)$$

Data likelihood

Independent Gaussian likelihood for errors in responses

$$\mathbf{y}|\mathbf{X}, \Theta, \mathcal{T} \sim \mathcal{N}\left((f(\mathbf{x}_i))_{i=1}^n, a\sigma^2\mathbf{I}_n\right)$$



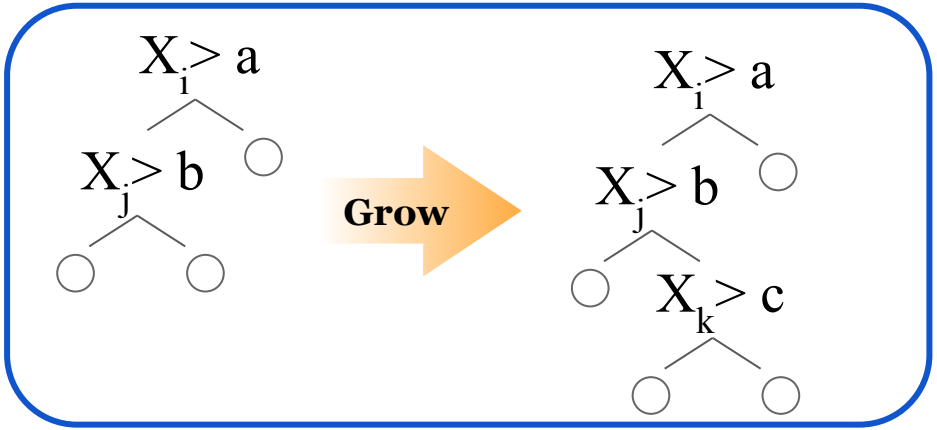
MCMC algorithm

Decompose posterior $p(\mathcal{T}, \Theta|\mathbf{X}, \mathbf{y}) = \underbrace{p(\Theta|\mathcal{T}, \mathbf{X}, \mathbf{y})}_{\text{Have closed form expression}} p(\mathcal{T}|\mathbf{X}, \mathbf{y})$

Hence just need to perform MCMC for space of trees $\mathcal{T}$ to sample from $p(\mathcal{T}|\mathbf{X}, \mathbf{y})$

Will use Metropolis-Hastings. 4 proposal moves:

1. Grow



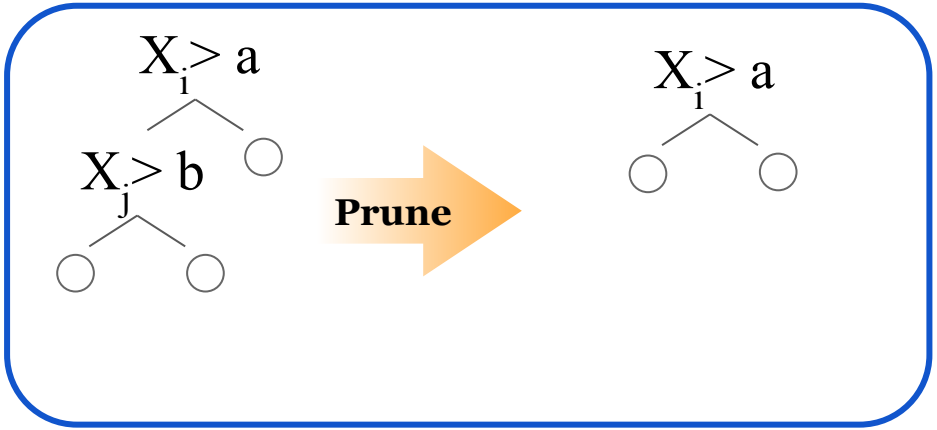
MCMC algorithm

Decompose posterior $p(\mathcal{T}, \Theta|\mathbf{X}, \mathbf{y}) = \underbrace{p(\Theta|\mathcal{T}, \mathbf{X}, \mathbf{y})}_{\text{Have closed form expression}} p(\mathcal{T}|\mathbf{X}, \mathbf{y})$

Hence just need to perform MCMC for space of trees $\mathcal{T}$ to sample from $p(\mathcal{T}|\mathbf{X}, \mathbf{y})$

Will use Metropolis-Hastings. 4 proposal moves:

1. Grow
2. Prune



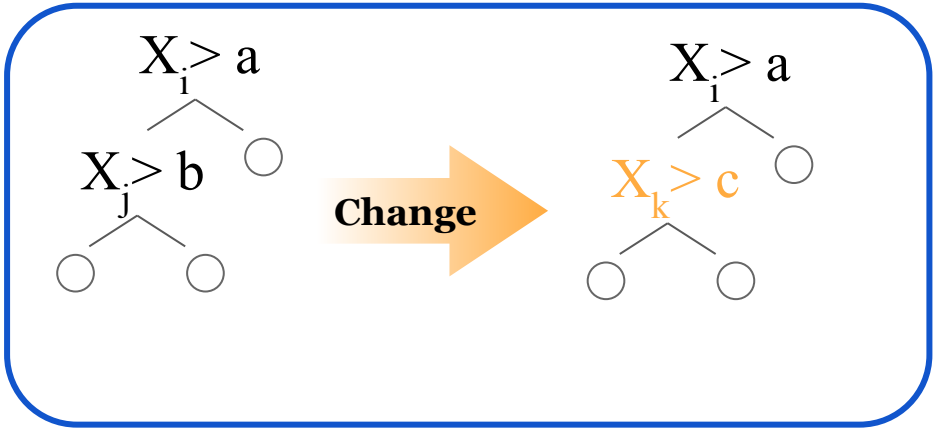
MCMC algorithm

Decompose posterior $p(\mathcal{T}, \Theta|\mathbf{X}, \mathbf{y}) = \underbrace{p(\Theta|\mathcal{T}, \mathbf{X}, \mathbf{y})}_{\text{Have closed form expression}} p(\mathcal{T}|\mathbf{X}, \mathbf{y})$

Hence just need to perform MCMC for space of trees $\mathcal{T}$ to sample from $p(\mathcal{T}|\mathbf{X}, \mathbf{y})$

Will use Metropolis-Hastings. 4 proposal moves:

1. Grow
2. Prune
3. Change



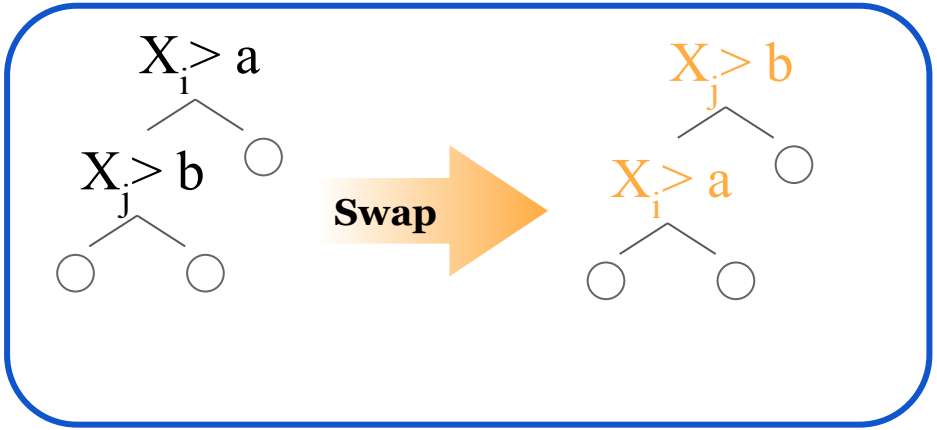
MCMC algorithm

Decompose posterior $p(\mathcal{T}, \Theta|\mathbf{X}, \mathbf{y}) = \underbrace{p(\Theta|\mathcal{T}, \mathbf{X}, \mathbf{y})}_{\text{Have closed form expression}} p(\mathcal{T}|\mathbf{X}, \mathbf{y})$

Hence just need to perform MCMC for space of trees $\mathcal{T}$ to sample from $p(\mathcal{T}|\mathbf{X}, \mathbf{y})$

Will use Metropolis-Hastings. 4 proposal moves:

1. Grow
2. Prune
3. Change
4. Swap



MCMC algorithm

Decompose posterior $p(\mathcal{T}, \Theta|\mathbf{X}, \mathbf{y}) = \underbrace{p(\Theta|\mathcal{T}, \mathbf{X}, \mathbf{y})}_{\text{Have closed form expression}} p(\mathcal{T}|\mathbf{X}, \mathbf{y})$

Hence just need to perform MCMC for space of trees $\mathcal{T}$ to sample from $p(\mathcal{T}|\mathbf{X}, \mathbf{y})$

Will use Metropolis-Hastings. 4 proposal moves:

1. Grow
2. Prune
3. Change
4. Swap

Pick a move at random

Apply accept-reject filter

BART with multiple trees

Convention: Use m trees

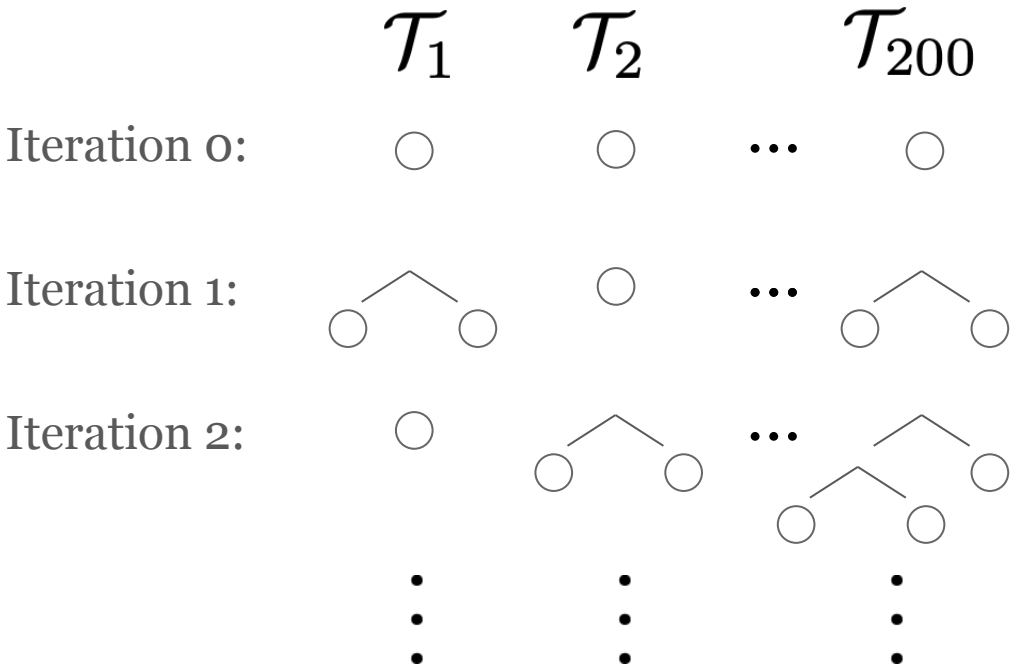
Parameterization and priors:

- Each tree is parameterized in the same way as before
- Regression function f is defined as the sum of the functions for each tree

MCMC

- Combine Gibbs sampling with Metropolis-Hastings

BART with multiple trees



Convention: Use 100 burn-in iterations, then 1000 iterations for computing “posterior”

Part 2: How to frame computational lower bounds?

A. Prior work on mixing time lower bounds for BART
B. What is wrong with this definition?
C. How to fix it?

A frequentist analysis of computational lower bounds

Assume we observe training dataset comprising n i.i.d. samples

$$\mathcal{D}_n \coloneqq \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$$

$$\mathbf{x}_i \sim \nu, \qquad y_i = f^*(\mathbf{x}_i) + \epsilon_i,$$

$$\epsilon_i \text{ sub-Gaussian}$$

Warning: Generative distribution can and will be different from Bayesian parameterization!

Defining the mixing time t_{mix}

t - time step of the Markov chain

π - Stationary distribution

$$Q^t(-|\mathcal{T}) - \pi$$

Q - Transition kernel

T - initialization

Defining the mixing time t_{mix}

t - time step of the Markov chain

π - Stationary distribution

$$\max_{\mathcal{T} \in \Omega} \| Q^t(-|\mathcal{T}) - \pi \|_{\mathrm{TV}}$$

Q - Transition kernel

T - initialization

Defining the mixing time t_{mix}

t - time step of the Markov chain

π - Stationary distribution

$$t_{mix} := \min\{t : \max_{\mathcal{T} \in \Omega} \|Q^t(-|\mathcal{T}) - \pi\|_{\text{TV}} \leq 0.25\}$$

Q - Transition kernel

T - initialization

Previous result with t_{mix}

Theorem (informal): Mixing time for BART with one tree grows **exponentially** in the sample size.

[Ronen, Saarinen, Tan, Duncan, Yu (2022)]

> This paper has the potential to be a significant contribution to the BART literature. However, I believe that the paper is missing a crucial discussion about *whether mixing in tree space is practically relevant or even necessary*.
>
> - Reviewer #3

What is wrong with

$$t_{mix} := \min\{t : \max_{\mathcal{T} \in \Omega} \|Q^t(-|\mathcal{T}) - \pi\|_{\text{TV}} \leq 0.25\} \text{ ?}$$

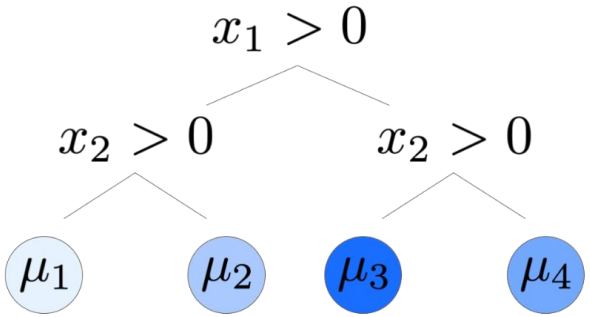
1. *Worst case* over all initializations

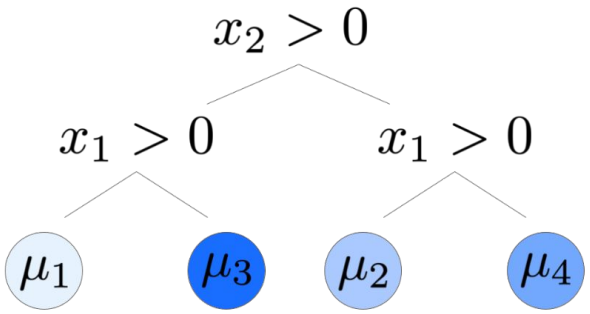
Whereas BART MCMC initializes from empty tree ensemble

2. MCMC is over *space of tree parameters*

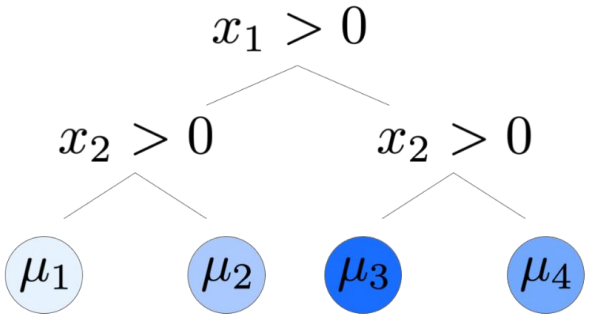
Whereas what we care about is the *realized regression function*

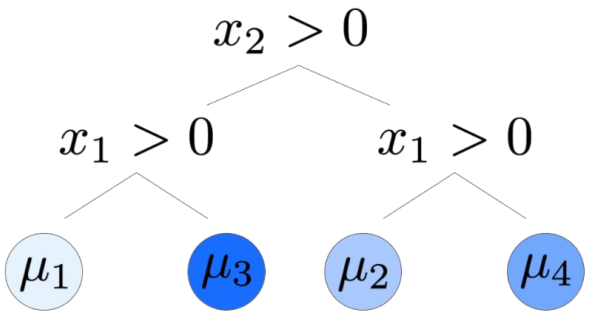
BART tree parameters are not identifiable

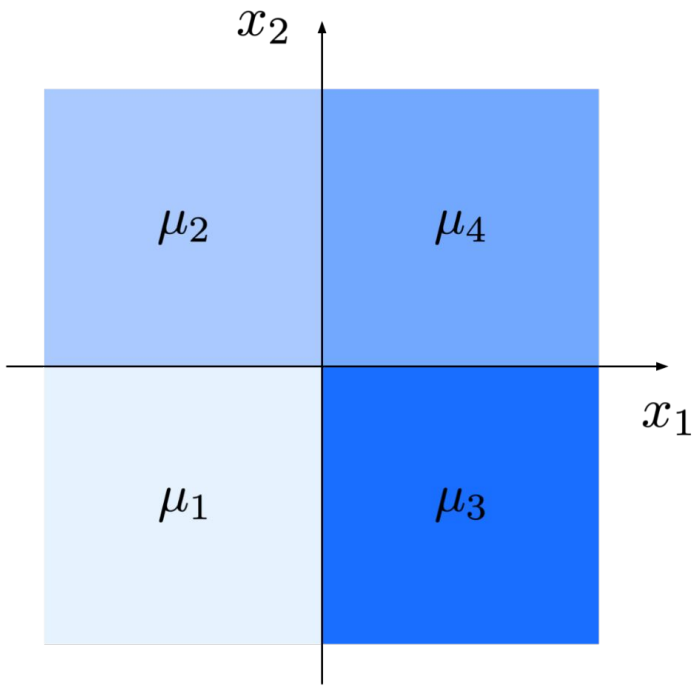




BART tree parameters are not identifiable







Fix this using *hitting times* for *highest posterior density regions (HPDR)*

$$\mathrm{OPT} := \left\{ \begin{matrix} \text{Trees } \mathcal{T} \text{ with zero bias and} \\ \text{minimal degrees of freedom} \end{matrix} \right\}$$

Proposition: OPT is a HPDR and BART posterior concentrates on OPT as $n \to \infty$.

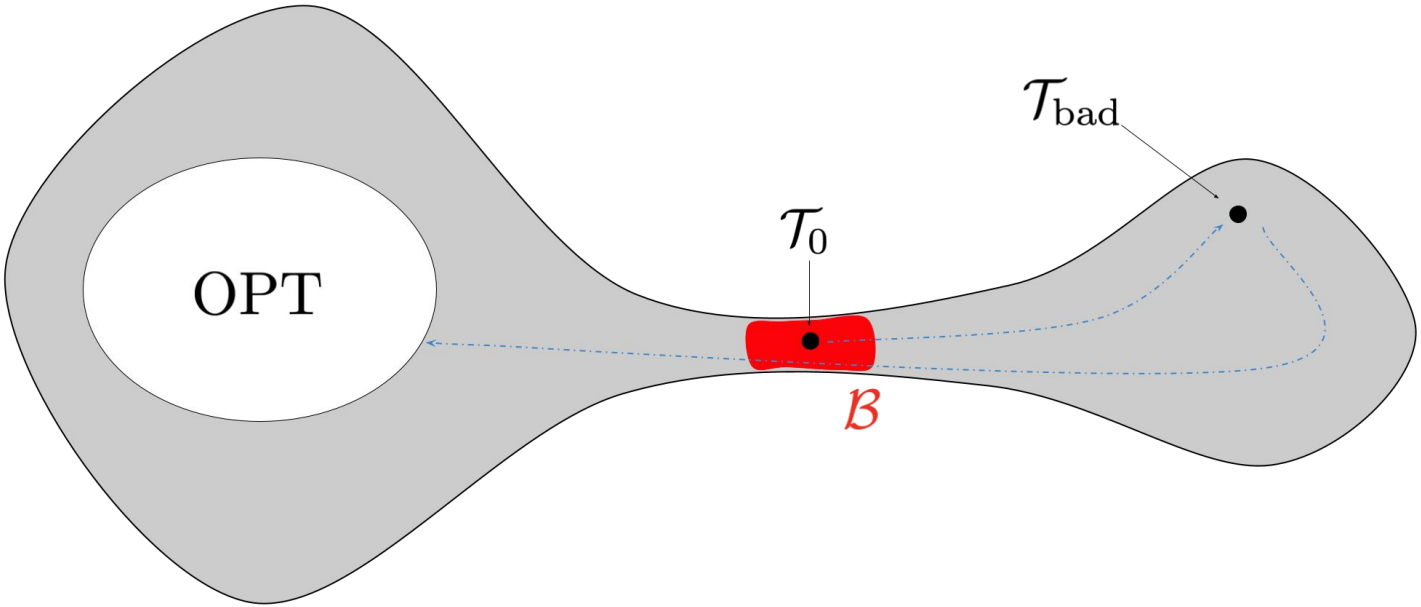
Tan, Ronen, Saarinen, Duncan, Yu (2023, in preparation)

$$\tau_{\mathrm{OPT}} := \min \{ t \geq 0 \colon \mathcal{T}_t \in \mathrm{OPT} \}$$

Part 3: Hitting time lower bounds and takeaways

A. Proof recipe
B. 3 hitting time lower bounds
C. Practical takeaways

Proof recipe



Result 1: Lower bounds for additive models

Def: (Additive model)

$$f^*(\mathbf{x}) = f_1(x_1) + f_2(x_2) + \cdots + f_{m'}(x_{m'})$$

Recall: BART model is

$$f(\mathbf{x}) = \mathcal{T}_1(\mathbf{x}) + \mathcal{T}_2(\mathbf{x}) + \cdots + \mathcal{T}_m(\mathbf{x})$$

Theorem 1 *(informal)*: If f^* is additive, m ≤ m', then

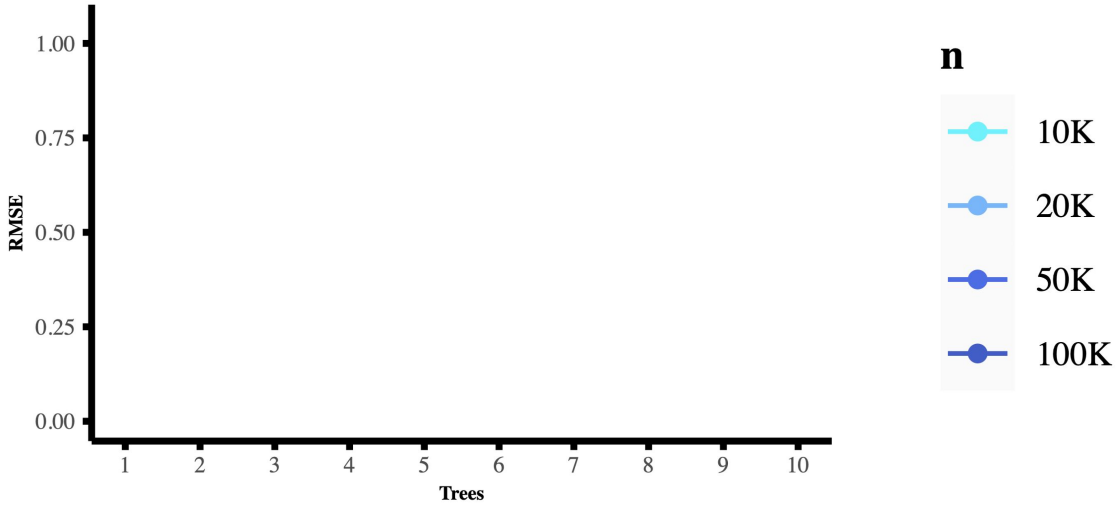
$$E\{\tau_{\mathrm{OPT}}\} = \Omega(n^{1/2})$$

If furthermore, m < m', and we allow only *grow* and *prune* moves, then

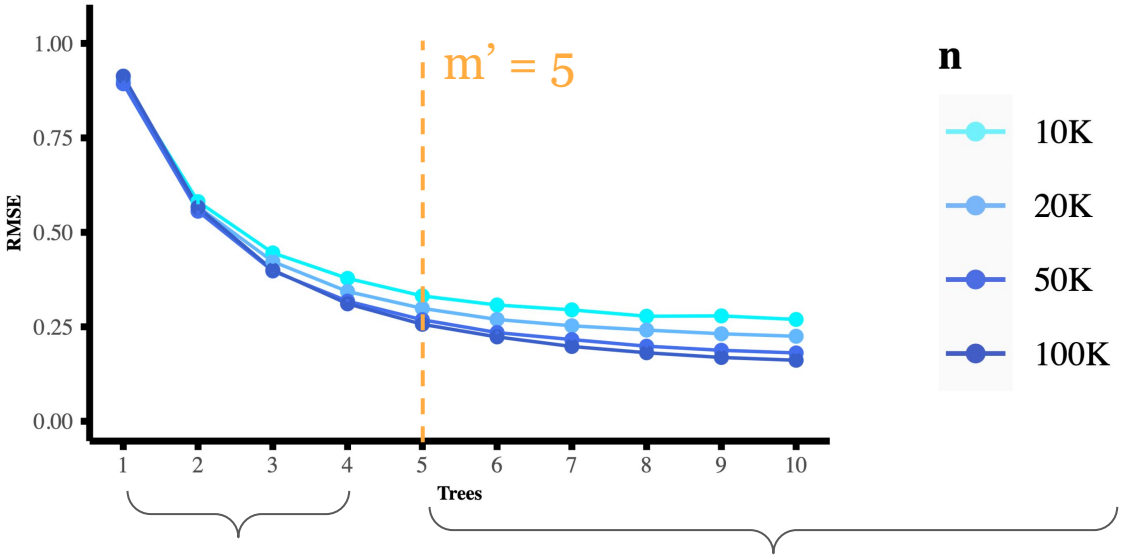
$$E\{\tau_{\mathrm{OPT}}\} = \Omega(n^{a/2})$$

$$a = \min_{1 \le j \le m'} \deg(f_j) - 2$$

Simulate f* linear, with m’ = 5.



Simulate f* linear, with m’ = 5.



(m < m’) Inefficient representation [Tan et al. (2021)]

(m ≤ m’) Theorem 1 shows computational bottleneck

(m ≤ ?) Simulation shows computational bottleneck persists

Other results

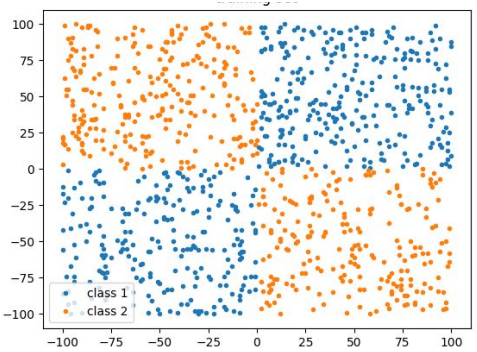
Assume only *grow* and *prune* moves are allowed.

Theorem 2 *(informal)*: If f^* contains a *pure interaction*, then

$$E\{\tau_{\text{OPT}^*}\} = \Omega(n^{1/2})$$

OPT*: All trees with zero bias

E.g. XOR function



Other results

Assume only *grow* and *prune* moves are allowed.

Theorem 2 *(informal)*: If f^* contains a *pure interaction*, then

$$E\{\tau_{\mathrm{OPT}^*}\} = \Omega(n^{1/2})$$

Theorem 3 *(informal)*: If we fit BART with only one tree (i.e. m=1), *then*

$$E\{\tau_{\mathrm{OPT}}\} = \exp\left(\Omega(n)\right)$$

What do our results mean for practice?

Short-term

- Should run multiple MCMC chains and average the results
- Should not take BART credible intervals at face value

Long-term

- BART sampler has large room for improvement
  - Temperature control
  - Using “informed” proposals instead of uniform proposals
  - More global proposal moves

Key Contributions

- Created framework for meaningful computational lower bounds for BART
- First analysis of BART with multiple trees
- Provide HPDR hitting time lower bounds for BART under three different settings, show that they grow with sample size
- Extensive simulation study (in the paper)
- Obtain insights on why BART sampler may experience computational issues and suggests how to overcome them.

Paper to appear on arxiv soon (a few weeks)!