Scalable Smoothing in High-Dimensions with BART

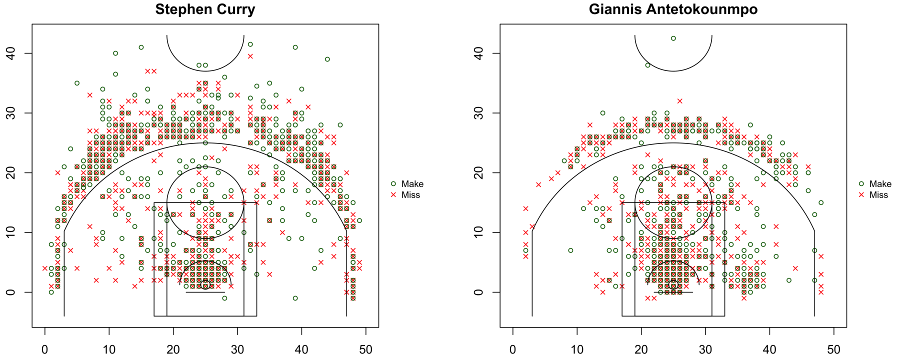
Ryan Yee

University of Wisconsin–Madison

JSM 2024

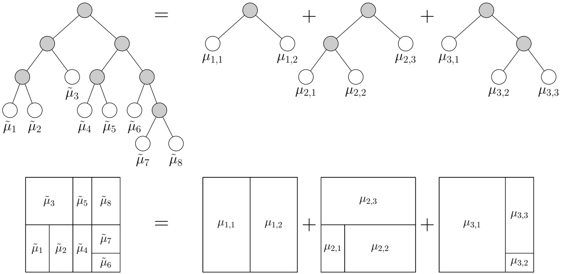
Motivating Example

- **Goal:** estimate $P(\text{make})$ for an NBA player given:
  - Player, position, size (height and weight)
  - Location



Review of BART

- **Problem:** non-parametric regression: $y_n \sim \mathcal{N}(f(\mathbf{x}_n), \sigma)$
- **Main Idea:** approximate $f(\mathbf{x})$ with step-function (i.e., tree)

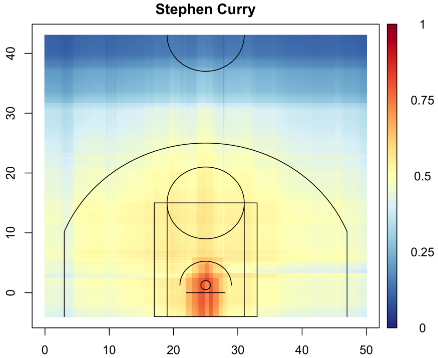


☺ Ideal for modeling nonlinear data with complex interactions

☺ Do not need to specify the functional form of f

☺ Bayesian approach facilitates uncertainty quantification

BART on Motivating Example



Our Contribution

- **Goal:** given covariates $(\mathbf{x}, \mathbf{z})$, output a smooth function in $\mathbf{z}$

Our Contribution

- **Goal:** given covariates $(\mathbf{x}, \mathbf{z})$, output a smooth function in $\mathbf{z}$
  - Note: if $\mathbf{z} \not\subset \mathbf{x}$: *targeted smoothing*, if $\mathbf{z} \subset \mathbf{x}$: *piecewise continuous*

Our Contribution

- **Goal:** given covariates $(\mathbf{x}, \mathbf{z})$, output a smooth function in $\mathbf{z}$
  - Note: if $\mathbf{z} \not\subset \mathbf{x}$: *targeted smoothing*, if $\mathbf{z} \subset \mathbf{x}$: *piecewise continuous*
- **Our contribution:** a scalable smoothing BART
  - Split on $\mathbf{x}$, output function in $\mathbf{z}$:

$$\sum_{d=1}^{D} \beta_d \cdot h(\omega_d^\top \mathbf{z} + b_d)$$

  - If $h(\cdot) = \sqrt{2}\cos(\cdot)$: *random Fourier feature GP approximation*
  - If $\omega_j \sim \mathcal{N}(0, 1/\rho)$, $\rho \sim \pi_\rho$, $\pi_\rho \sim \mathrm{DP}(\alpha, F_0)$: *infinite mixture of GPs*
  - If $h(\cdot) = \tanh(\cdot); \mathrm{ReLU}(\cdot); \sigma(\cdot)$: *single-layer random neural network*

Our Contribution

- **Goal:** given covariates $(\mathbf{x}, \mathbf{z})$, output a smooth function in $\mathbf{z}$
  - Note: if $\mathbf{z} \not\subset \mathbf{x}$: *targeted smoothing*, if $\mathbf{z} \subset \mathbf{x}$: *piecewise continuous*
- **Our contribution:** a scalable smoothing BART
  - Split on $\mathbf{x}$, output function in $\mathbf{z}$:

$$\sum_{d=1}^{D} \beta_d \cdot h(\omega_d^\top \mathbf{z} + b_d)$$

  - If $h(\cdot) = \sqrt{2}\cos(\cdot)$: *random Fourier feature GP approximation*
  - If $\omega_j \sim \mathcal{N}(0, 1/\rho)$, $\rho \sim \pi_\rho$, $\pi_\rho \sim \mathrm{DP}(\alpha, F_0)$: *infinite mixture of GPs*
  - If $h(\cdot) = \tanh(\cdot); \mathrm{ReLU}(\cdot); \sigma(\cdot)$: *single-layer random neural network*
- Previous authors have taken similar approaches:
  - BART with B-splines (Low-Kam et. al., 2015)
  - Treed Gaussian processes (Gramacy and Lee, 2007)
  - BART with targeted smoothing (Starling et. al., 2020)
  - GP-BART (Maia et. al., 2024)

BART Metropolis-within-Gibbs Sampler

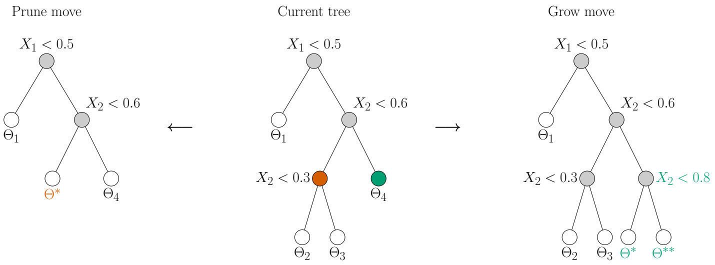
- BART updates tree in two steps:
  1. Tree structure via Metropolis-Hastings
  2. Leaf parameters via conjugate-normal update

BART Metropolis-within-Gibbs Sampler

- BART updates tree in two steps:
  1. Tree structure via Metropolis-Hastings
  2. Leaf parameters via conjugate-normal update
- New output function: $\sum_{d=1}^{D} \beta_d \cdot h(\omega_d^\top \mathbf{z} + b_d)$
- **Key Idea:** separate *input weights* $\Theta = (\boldsymbol{\omega}, \mathbf{b})$ and *output weights* $\boldsymbol{\beta}$

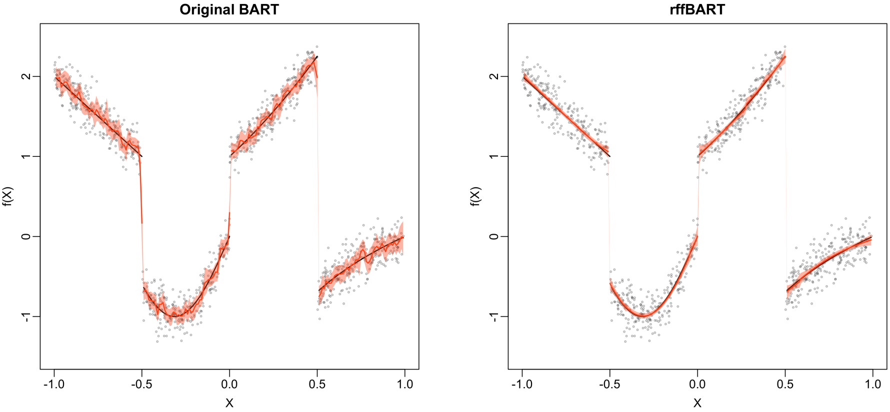
BART Metropolis-within-Gibbs Sampler

- BART updates tree in two steps:
  1. Tree structure via Metropolis-Hastings
  2. Leaf parameters via conjugate-normal update
- New output function: $\sum_{d=1}^{D} \beta_d \cdot h(\omega_d^\top \mathbf{z} + b_d)$
- **Key Idea:** separate *input weights* $\Theta = (\boldsymbol{\omega}, \mathbf{b})$ and *output weights* $\boldsymbol{\beta}$



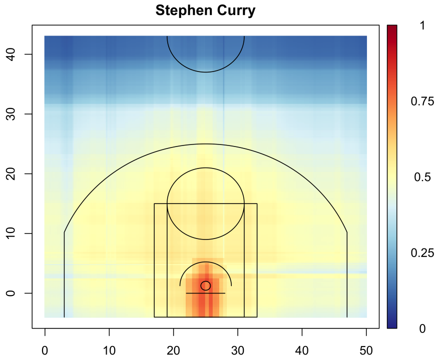
- Treat Θ as part of the tree structure and update via MCMC
- Place conjugate-normal prior on $\boldsymbol{\beta}$ and update accordingly

Illustrative Example

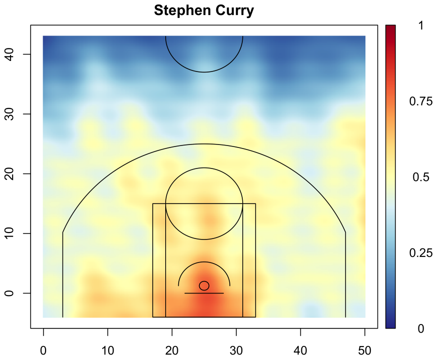


$$f(x) = \mathbb{1}(x \leq -0.5)(-2x) + \mathbb{1}(-0.5 < x \leq 10)\sin(5x) + \mathbb{1}(0 < x \leq 0.5)(x+1)^2 + \mathbb{1}(x > 0.5)\log x + \epsilon$$

Motivating Example

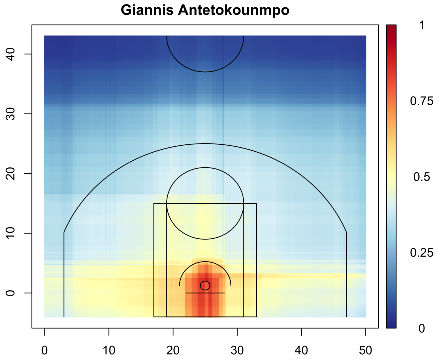


Original BART

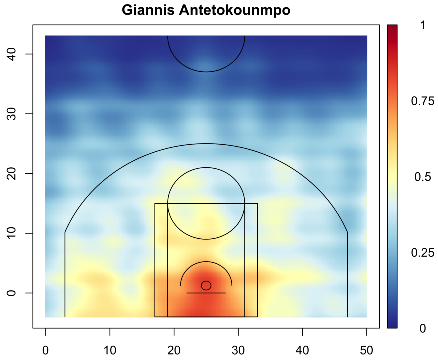


`rffBART`

Motivating Example



Original BART



`rffBART`

Key Takeaways

- Introduced extendable framework for computationally scalable and representationally flexible continuous-response BART model
  - Ridge functions facilitate scalability without sacrificing flexibility
  - Minimal changes to sampler offer extensibility
- Predictive performance competitive with BART
  - Improved function recovery and tighter pointwise credible intervals on piecewise-continuous test function
  - Comparable to BART on motivating example despite imposing restrictive continuity assumptions
- **Work in progress:** posterior contraction

Key Takeaways

- Introduced extendable framework for computationally scalable and representationally flexible continuous-response BART model
  - Ridge functions facilitate scalability without sacrificing flexibility
  - Minimal changes to sampler offer extensibility
- Predictive performance competitive with BART
  - Improved function recovery and tighter pointwise credible intervals on piecewise-continuous test function
  - Comparable to BART on motivating example despite imposing restrictive continuity assumptions
- **Work in progress:** posterior contraction

Thanks!

Email: ryee2@wisc.edu
Website: https://ryanyee3.github.io