Automatic vs Bootstrapped Uncertainty Quantification

Tyler Schmidt, Dylan Day, Camden Foster, Ashwin Dervesh

University of Iowa

12/10/2025

Introduction to Uncertainty Quantification

- We want to evaluate how accurate a prediction $(\hat{y})$ is, given a new set of covariates X that are not part of the training data.
- Real world examples include: Stroke risk, industrial quality control, crop yield, etc.
- Provides a measure or reliability through either a predictive interval or a posterior distribution.

Models

Random Forest

- Ensemble of many trees trained independently (bagging)
- Reduces variance by averaging predictions
- Strong baseline, low tuning burden

XGBoost

- Sequential tree boosting that corrects prior errors
- Gradient boosting with regularization for high accuracy
- Excels on structured/tabular data

SoftBART

- Uses soft probabilistic splits instead of hard splits.
- Provides full Bayesian uncertainty quantification via posterior draws.
- Produces smoother, more stable functions than standard BART.

Random Forests (ranger)

Random Forests (James et al., 2023) estimate the regression function using an ensemble of T decision trees grown on bootstrapped samples:

$$\hat{f}(X_i) = \frac{1}{T}\sum_{t=1}^{T} g_t(X_i),$$

where each g_t is a tree fit on a bootstrap sample with random feature subsampling.

Key elements:

- **Bootstrap sampling**: each tree is trained on a resampled dataset, inducing variability.
- **Feature subsampling**: at each split, only m_{try} features are considered, decorrelating trees.
- **Prediction**: the forest average reduces variance relative to individual trees.

Random Forests (ranger)

Uncertainty Quantification: Since trees are trained on perturbed datasets, variability across tree predictions,

$$\widehat{\mathrm{Var}}(X_i) = \frac{1}{T}\sum_{t=1}^{T}\left(g_t(X_i) - \hat{f}(X_i)\right)^2,$$

provides a natural measure of model-induced uncertainty.
Note: If you are measuring uncertainty in the regression function, this inherent variance is a good estimator, however, if you want a measure of prediction intervals, bootstrapping is required.

XGBoost

XGBoost (Chen and Guestrin, 2018) represents the regression function as an additive expansion of trees built sequentially to minimize a regularized loss:

$$\hat{f}(X_i) = \sum_{t=1}^{T} g_t(X_i), \qquad g_t = \arg\min_{g \in \mathcal{G}} \sum_{i=1}^{n} \ell\big(y_i, f_{t-1}(X_i) + g(X_i)\big) + \Omega(g),$$

where $\Omega(g)$ penalizes tree complexity (depth, number of leaves, leaf weights). Key elements:

- **Boosting**: each tree corrects residual errors from previous ones.
- **Second-order optimization**: uses gradients and Hessians of the loss.
- **Shrinkage and subsampling**: improves generalization and reduce variance.

XGBoost

Uncertainty Quantification: XGBoost is inherently deterministic; UQ typically requires *external* methods such as

- ensembles (bootstrapped XGBoost models),
- dropout-like randomization (XGBoost-DART),
- Bayesian approximations (e.g., Laplace, SWAG),
- quantile regression objectives.

These provide interval estimates but do not yield a Bayesian posterior like BART/SoftBART.

BART Model 1/3

Bayesian Additive Regression Trees (BART) (Chipman et al., 2010) represent the regression function as a sum of many small decision trees:

$$Y_i = f(X_i) + \varepsilon_i, \qquad f(X_i) = \sum_{t=1}^{T} g(X_i; \mathcal{T}_t, \mathcal{M}_t),$$

$$\varepsilon_i \sim \mathcal{N}(0, \sigma^2).$$

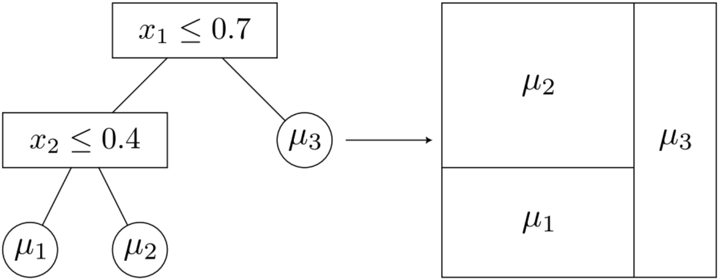
- Each $g(\cdot)$ is a shallow regression tree with structure $\mathcal{T}_t$ and leaf parameters $\mathcal{M}_t$.
- A strong shrinkage prior forces each tree to contribute only a small effect.

BART Model 2/3

- The function $g(x; \mathcal{T}_t, \mathcal{M}_t)$ returns

$$g(x; \mathcal{T}_t, \mathcal{M}_t) = \sum_{\ell=1}^{L_t} \mu_{t,\ell}\, \phi_\ell(x; \mathcal{T}_t),$$

where $\phi_\ell(x; \mathcal{T}_t)$ is the indicator that x falls into leaf ℓ of tree $\mathcal{T}_t$ and $\mu_{t,\ell}$ is the leaf parameter.
- Given the tree structure, each branch node b is given by the decision rule $X_j \leq C_b$



BART Model 3/3

- Bayesian inference combines a prior distribution with the likelihood to form a posterior:

$$p(\theta \mid \text{data}) = \frac{p(\text{data} \mid \theta)\, p(\theta)}{\int p(\text{data} \mid \vartheta)\, p(\vartheta)\, d\vartheta}.$$

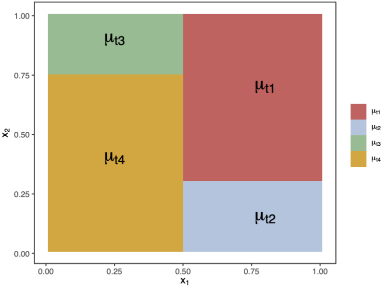
- For complex models, the posterior cannot be computed analytically.
- Markov Chain Monte Carlo (MCMC) approximates the posterior by constructing a Markov chain whose stationary distribution is $p(\theta \mid \text{data})$.
- Sampling from this chain allows intuitive inference and automatic uncertainty quantification.
- BART uses a specialized Bayesian backfitting MCMC: each tree is updated in turn, conditioning on the residual created by all other trees.

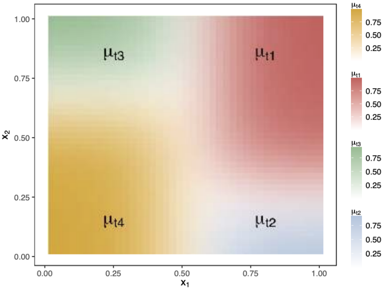
SoftBART Model 1/2

- (Linero and Yang, 2018) generalize the BART model by replacing the "hard" decision rules $I(x_j \leq C)$ with *soft* decision rules based on a smooth function ψ:

$$\psi\left(\frac{x_j - C}{\tau}\right),$$

- $\psi(x)$ is the cumulative distribution function (CDF) of a symmetric random variable (e.g., the logistic CDF $\psi(x) = (1 + e^{-x})^{-1}$), and $\tau > 0$ controls the softness of the split.





SoftBART Model 2/2

- Under SoftBART, each tree contributes

$$g(x; \mathcal{T}, \mathcal{M}) = \sum_{\ell=1}^{L} \mu_\ell \, w_\ell(x; \mathcal{T}, \tau),$$

  where $w_\ell(x; \mathcal{T}, \tau)$ is a probabilistic *soft* leaf-weight function.
- As $\tau \to 0$, SoftBART reduces to standard BART with hard indicator splits.
- (Linero and Yang, 2018) derived the following posterior contraction rate of SoftBART $n^{-\alpha/(2\alpha+p)}$ up to logarithmic terms for a smoothness-level α and number of covariates p.

Conformal Prediction and Bootstrapping

Conformal Prediction

- Quantifies uncertainty by using the distribution of calibration residuals to form prediction intervals.
- Procedure:
  - Fit model on training data
  - Compute residuals on the calibration set (data not used to train the model)
  - Choose a quantile of these residuals to ensure $(1-\alpha)$ coverage
- Gives valid prediction intervals regardless of underlying data distribution and works with any model.
- Computationally efficient

Bootstrap

- Repeatedly resample (with replacement) from the training data, refit the model, and record predicted values.
- Computationally expensive

Hypothesis

Based on prior knowledge and intial simulations we hypothesize that:

- SoftBART will achieve superior uncertainty quantification (interval length and coverage), especially for complex data-generating processes.
- This improvement comes at the cost of higher computational complexity compared to Random Forest and XGBoost.

Simulation Setup

- We generate a design matrix $X \in \mathbb{R}^{n \times 5}$ with rows

$$X \sim \text{MVN}(\mu, \sigma^2 I_5),$$

i.e. independent Gaussian features and no intercept, with $\mu = (5, 9, -3, 2, 1)^T$ and $\sigma^2 = 1$

- For the linear case, responses are generated as

$$y = 2X_1 - 3.1X_2 + 4.7X_3 + 0.5X_4 + 1.5X_5 + \varepsilon$$

and for the nonlinear case, responses are generated as

$$\begin{aligned} y = &2 * X_1^3 * X_4 - 3.1 * X_2^2 + 4.7 * \sin(X_3) + \\ &0.5 * X_4 * X_5 + \log(\text{abs}(X_5) + 1) + \varepsilon \end{aligned}$$

where $\varepsilon \sim N(0, 1)$ and X_i is the ith column of X for $i \in \{1, 2, \ldots, 5\}$

Simulation Algorithm

Algorithm 1 Uncertainty Quantification Across Different Models

Inputs: Sample size $n = \{100, 250, 500\}$, Number of Bootstrap Replicates $B = 1000$, Number of simulations $J = 20$, linear or nonlinear DGP
for $j = 1$ to J **do**
    Set seed, generate data with set sample size n and DGP,
    Split sample into 60/20/20 training/test/calibration sets
    Fit models on training set, predict on calibration set
    bootstrap and determine conformal prediction intervals
    Compute test bias, MSE, interval length, and coverage for each method
end for
Compute average test statistics across simulations $j = 1, \ldots, J$
Output: Test statistics for each model

HPC Optimization - Argon

High Performance Computing (HPC) opens the door for extreme optimization of complex processes or big data analysis. The University of Iowa has their own HPC system called Argon:

- Campus shared system with clusters of GPU/CPU nodes
- 23,000 CPU processors and 350 GPU accelerators
- Queues available to Iowa students/faculty to submit jobs

To implement in our workflow:

- Package R scripts and execute commands in bash shell files
- Submit simulations in parallel via an Array job to advisor's queue
- Save results to file

Linear Results

n	Method	Bias	RMSE	Bootstrap		Conformal	
				Int Length	Coverage	Int Length	Coverage
100	ranger	0.68	3.03	13.32	0.95	14.66	0.96
	xgboost	0.52	2.36	11.57	0.96	11.56	0.96
	SoftBART	0.36	1.57	6.16	0.94	7.08	0.96
250	ranger	0.42	2.59	10.75	0.94	11.42	0.95
	xgboost	0.26	1.90	8.98	0.97	8.63	0.97
	SoftBART	0.19	1.32	5.07	0.94	5.64	0.95
500	ranger	0.21	2.24	9.11	0.94	9.31	0.95
	xgboost	0.17	1.66	7.29	0.97	6.63	0.95
	SoftBART	0.11	1.16	4.61	0.96	4.63	0.95

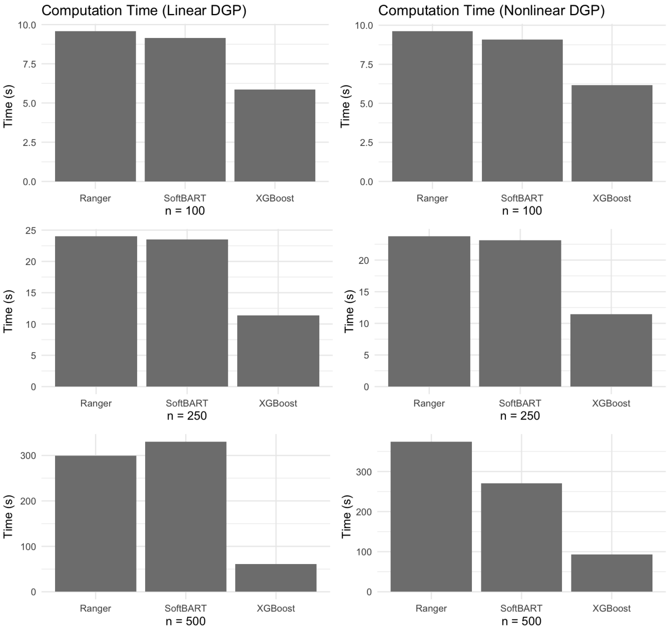
Table: Linear results for tree-based methods across different sample sizes, showing Bootstrap and Conformal intervals.

Nonlinear Results

n	Method	Bias	RMSE	Bootstrap		Conformal	
				Int Length	Coverage	Int Length	Coverage
	ranger	44.54	226.39	963.06	0.93	1137.52	0.94
100	xgboost	36.65	185.19	925.79	0.97	1140.42	0.94
	SoftBART	21.27	90.25	217.65	0.92	589.10	0.94
	ranger	31.24	184.49	755.14	0.95	786.89	0.95
250	xgboost	22.16	136.70	621.34	0.97	520.61	0.94
	SoftBART	7.44	55.22	95.96	0.95	216.76	0.95
	ranger	16.01	144.28	659.05	0.95	606.64	0.94
500	xgboost	10.61	97.95	511.03	0.97	371.63	0.95
	SoftBART	2.95	30.75	66.69	0.98	94.66	0.94

Table: Nonlinear results for tree-based methods across different sample sizes, showing Bootstrap and Conformal intervals.

Time Results



Simulation Considerations

- The calibration set was necessary to accurately estimate the residual standard deviation for constructing bootstrap prediction intervals. That is, inside the bootstrap replicates we compute

$$\hat{y}_{\text{boot}} = \hat{f}_{\text{boot}}(X) + \gamma, \quad \gamma \sim N(0, \hat{\sigma})$$

  where $\hat{\sigma}$ is estimated from the calibration set.

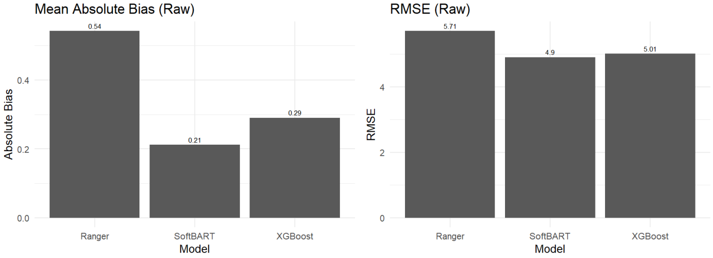
- No tuning parameters were optimized for XGBoost or ranger; hyperparameters were chosen arbitrarily following the simulation design in (Chipman et al., 2010).

- Our data-generating process is very smooth (large α), which favors methods that assume smoothness, such as SoftBART.

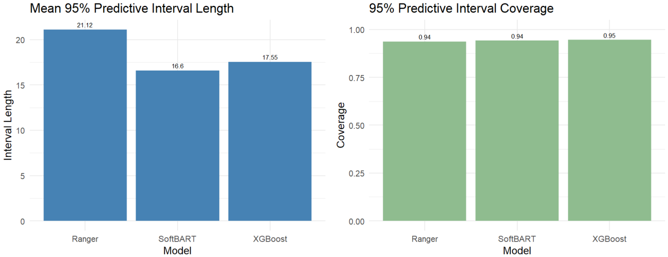
- We used only $J = 20$ replications, making the results susceptible to Monte Carlo noise.

Real World Use Case

Bias and RMSE



Coverage and Interval length



Conclusion

SoftBART outperforms both random forest and XGBoost across all sample sizes:

- In both linear and nonlinear cases, SoftBART had lowest Bias and RMSE
- SoftBART had significantly smaller interval lengths with comparable coverage
- In the applied data set, SoftBART had the lowest Bias and comparable RMSE and coverage
- SoftBART and ranger have worse computational complexity compared to xgboost

Responsibilities

Name	Role
Tyler Schmidt	Wrote simulation code and applied BART concepts
Camden Foster	Applied simulation to real-world data and analyzed results
Ashwin Dervesh	Managed development workflow and created presentation slides
Dylan Day	Developed and refined presentation materials

References I

Tianqi Chen and Carlos Guestrin. Xgboost: Extreme gradient boosting. In Arthur Samuel, Charu C. Aggarwal, et al., editors, *The Encyclopedia of Machine Learning and Data Mining*, pages 1–6. Springer, 2018. doi: 10.1007/978-1-4899-7502-7_1283-1. URL https://doi.org/10.1007/978-1-4899-7502-7_1283-1.

Hugh A. Chipman, Edward I. George, and Robert E. McCulloch. Bart: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298, March 2010. doi: 10.1214/09-AOAS285. URL https://doi.org/10.1214/09-AOAS285.

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: With Applications in R*. Springer, Berlin, 2 edition, 2023. doi: 10.1007/978-1-0716-1418-1. URL https://doi.org/10.1007/978-1-0716-1418-1.

References II

Antonio R. Linero and Yun Yang. Bayesian regression tree ensembles that adapt to smoothness and sparsity. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80 (5):1087–1110, November 2018. doi: 10.1111/rssb.12293. URL https://doi.org/10.1111/rssb.12293.

Link to Github repo with code: https://github.com/adervesh03/Uncertainty-Quantification