Bayesian Additive Regression Trees

Frank Rotiroti

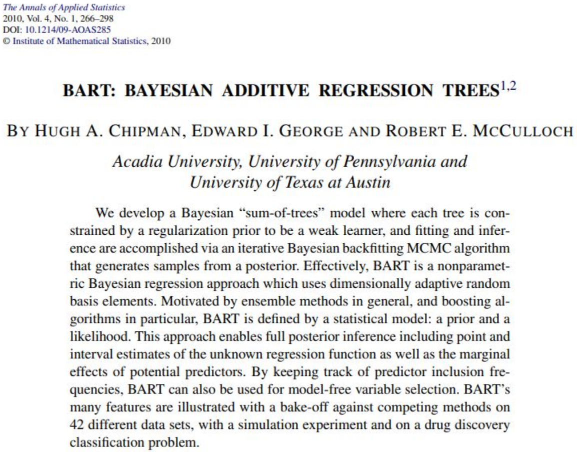
The University of Texas at Austin

Department of Statistics and Data Sciences

February 17, 2022

Table of Contents

1. Introduction
2. BART
3. Extensions
4. Time-varying model
5. Theory
6. Conclusion



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

We develop a Bayesian "sum-of-trees" model where each tree is constrained by a regularization prior to be a weak learner, and fitting and inference are accomplished via an iterative Bayesian backfitting MCMC algorithm that generates samples from a posterior. Effectively, BART is a nonparametric Bayesian regression approach which uses dimensionally adaptive random basis elements. Motivated by ensemble methods in general, and boosting algorithms in particular, BART is defined by a statistical model: a prior and a likelihood. This approach enables full posterior inference including point and interval estimates of the unknown regression function as well as the marginal effects of potential predictors. By keeping track of predictor inclusion frequencies, BART can also be used for model-free variable selection. BART's many features are illustrated with a bake-off against competing methods on 42 different data sets, with a simulation experiment and on a drug discovery classification problem.

Introduction

- The BART (Bayesian Additive Regression Trees) model begins with an unknown function f associated with output Y and input $x \in \mathbb{R}^p$ according to

$$Y = f(x) + \epsilon, \quad \epsilon \sim N(0, \sigma^2).$$

- Naturally, we are interested in making inference about the unknown f.
- BART assumes that $f(x) = E(Y|x)$ can modeled (or at least approximated) by a sum of m regression trees; i.e.,

$$f(x) \approx h(x) \equiv \sum_{i=1}^{m} g_j(x).$$

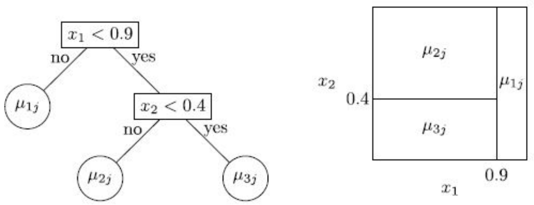
- With the imposition of a prior that regularizes the fit by keeping the individual tree effects small, the g_j functions can collectively be viewed as a dimensionally adaptive random basis of “weak learners,” each explaining a small and different portion of f .

- According to Robert McCulloch, "BART was inspired by the Boosting literature, in particular the work of Jerry Friedman," at least insofar as both are based on sums of trees and act on successive residuals.
  - See Appendix for further details on gradient tree boosting.
- A fundamental difference, of course, is the Bayesian framework in which BART is situated.
- Thus, a regularization prior is placed on (f, σ), and a Markov chain is constructed the stationary distribution of which is the desired posterior, $(f, \sigma)|$data.
- Assuming convergence of the chain, the draws of σ are directly available, whereas the draws of f can be used to produce an estimate of $\widehat{f}(x)$ for any x, i.e., an MCMC estimate of the posterior mean of $f(x)$.

Model specifics

- Letting T denote a binary tree equipped with a set of interior node decision rules as well as a set of terminal nodes, take $M = \{\mu_1, \dots, \mu_b\}$ to be a set of parameter values corresponding to the b terminal nodes of T.
- Following a sequence of decision rules, assign each $x = (x_1, \dots, x_p)$ to a single terminal node, where it is then given the particular value μ_h, $h \in \{1, \dots, b\}$, associated with that node.

Figure: (Left) A binary tree T_j where terminal nodes are labeled with the corresponding scalar parameters μ_{hj} (Right) The corresponding partition of the sample space



- With m trees, the general model then takes the form

  $$Y = \sum_{j=1}^{m} g(x; T_j; M_j) + \epsilon, \quad \epsilon \sim N(0, \sigma^2),$$

  where for each binary regression tree T_j and its associated terminal node parameters M_j, $g(x; T_j, M_j)$ is the function that assigns $\mu_{hj} \in M_j$ to x.
- This model represents both main and interaction effects, and as the number of trees grows, this representation flexibility leads to impressive predictive performance.
- Finally, a regularization prior over the parameters $(T_1, M_1), \ldots, (T_m, M_m)$ and σ is imposed to facilitate the additive representation.

Specificer specifics

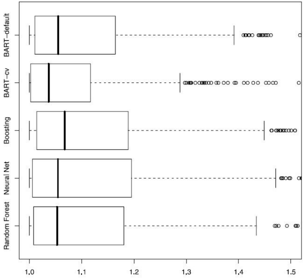
- The fits for the sum-of-trees model are obtained using a tailored version of Bayesian backfitting MCMC (Hastie and Tibshirani, 2000) which acts on successive residuals.
- In their regulizarization prior specification, Chipman et al. (2010) assume that the tree components (T_j, M_j) are independent of each other and of σ , and the terminal node parameters of every tree are independent.
- The prior over the tree space comprises a set of probabilities governing
  - whether a node at a given depth is a terminal node;
  - the choice of splitting covariate;
  - the choice of splitting value for each assigned covariate.
- The sampling algorithm proposes a new tree based on the current tree using one of four moves:
  - **grow**: randomly selects a terminal node and splits it into two child nodes;
  - **prune**: randomly selects an internal node with two children and no grandchildren and prunes the children, making the selected node a terminal node;
  - **change**: randomly selects an internal node and draws a new splitting rule;
  - **swap**: randomly selects a parent-child pair of internal nodes and swaps their decision rules.

- Chipman et al. (2010) recommend automatic default specifications for the priors, which they show to perform well across a number of different examples.
- A complete discussion of the priors and the backfitting MCMC algorithm can be found in Chipman et al. (2010).
- Notably, there are software packages in R that enable easy implementation of BART, such as *BART* (Sparapani et al., 2021) and *XBART* (He et al., 2019), with the latter offering an accelerated implementation, hence the 'X', with improved speed and reduced memory requirements.

Predictive comparisons

- Chipman et al. (2010) performed predictive comparisons on 42 data sets, which correspond to regression setups with between 3 and 28 numeric predictors and 0 to 6 categorical predictors.
- For each of the 42 data sets, 20 independent train/test splits were created by randomly selecting 5/6 of the data as a training set and the remaining 1/6 as a test set.
- After being trained on the training set, each method was then used to predict the corresponding test set and was evaluated on the basis of its predictive RMSE.
- Two versions of BART were considered: BART-default and BART-cv, the latter of which treats the prior hyperparameters as operational parameters to be tuned via 5-fold cross-validation within each training set.

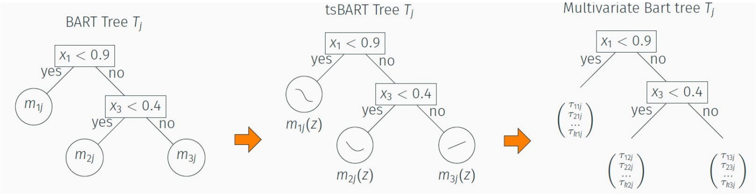
Figure: Each boxplot represents 840 out-of-sample predictions for a method.
Relative RMSE (RRMSE) is defined as the RMSE divided by the minimum RMSE obtained by any method for each test/train split.
An RRMSE of 1.2 means that the method performed 20% worse than the best method.



Boxplots of the RRMSE values for each method across the 840 test/train splits. Percentage RRMSE values larger than 1.5 for each method (and not plotted) were the following: random forests 16.2%, neural net 9.0%, boosting 13.6%, BART-cv 9.0% and BART-default 11.8%. The Lasso (not plotted because of too many large RRMSE values) had 29.5% greater than 1.5.

Extensions

- Extensions to the BART modeling framework include BART models for causal inference (Hahn et al., 2020), BART for log-linear models (Murray, 2021), and BART that can incorporate monotonicity in any predesignated subset of predictors (Chipman et al., 2021).
- One particular area of research involves the nature of the terminal node parameters.
- For example, tsBART (BART with Targeted Smoothing) of Starling et al. (2020) parameterizes the terminal nodes by a collection of Gaussian processes in t.
- This idea can be extended to the continuous time setting by using a spectral basis expansion to approximate a univariate Gaussian process.


BART Tree T_j
$x_1 < 0.9$
yes
no
m_{1j}
$x_3 < 0.4$
yes
no
m_{2j}
m_{3j}
tsBART Tree T_j
$x_1 < 0.9$
yes
no
$m_{1j}(z)$
$x_3 < 0.4$
yes
no
$m_{2j}(z)$
$m_{3j}(z)$
Multivariate Bart tree T_j
$x_1 < 0.9$
yes
no
$\begin{pmatrix} \tau_{11j} \\ \tau_{21j} \\ \cdots \\ \tau_{k1j} \end{pmatrix}$
$x_3 < 0.4$
yes
no
$\begin{pmatrix} \tau_{12j} \\ \tau_{22j} \\ \cdots \\ \tau_{k2j} \end{pmatrix}$
$\begin{pmatrix} \tau_{13j} \\ \tau_{23j} \\ \cdots \\ \tau_{k3j} \end{pmatrix}$

Time-varying predictions

- While BART yields the "non-linear" aspect of predictions, of, say, stock returns, we might also expect the mean-response function to vary smoothly over time.
- Given that the time periods of interest for such data can span hundreds of time points, models like tsBART (BART with Targeted Smoothing) of Starling et al. (2020), which similarly parameterizes the terminal nodes by a collection of Gaussian processes in *t* but is aimed at predicting responses over several time points, will not suffice.
- Accordingly, to model the time dependence of such observations, we can extend the BART framework by introducing a stationary AR(1) process into the leaves.

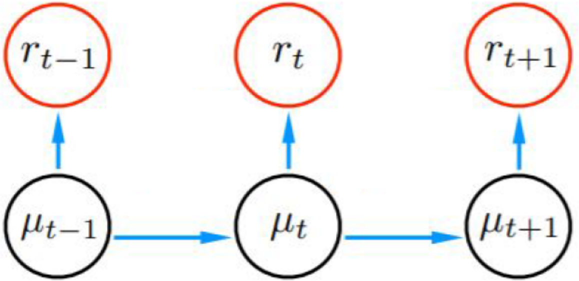
Model

- Let $R_{hj} = (R_{h1j}, \dots, R_{hn_hj})$, where R_{hkj} denotes the kth observation (partial residual) in terminal node $h \in \{1, 2, \dots, b_j\}$ of tree j out of m total trees.
- Let $\mu_{\cdot,hj} = (\mu_{1,hj}, \dots, \mu_{T,hj})$.
- Then, ordering the data in R_{hj} by time, ascending from $t = 1$ to $t = T$, and denoting by $R_{t,hj}$ each subvector of size n_t, where n_t denotes the number of data points observed at time t, we have a model that takes the form of an AR(1) process observed with noise.
- That is, for the given tree j, we have that

$$\begin{aligned} R_{t,hj} &= \mu_{t,hj} 1_{n_t} + v_t, \\ \mu_{t,hj} &= \alpha + \beta \mu_{t-1,hj} + u_t, \end{aligned}$$

where $v_t \sim N(0, \sigma^2 I)$ and $u_t \sim N(0, \sigma_\mu^2)$ are independent sequences, for $t = 1, \dots, T$.

Figure: Dependence structure over time



Theory?

- While regression trees can offer impressive predictive performance, they are susceptible to over-fitting.
- As we have seen, the Bayesian framework offers a remedy against overfitting through regularization priors.
- Nevertheless, the theory supporting the Bayesian framework is still being developed.
- In particular, Ročková and van der Pas (2020) seek to provide some theoretical justification for why BART has been resilient to overfitting in practice.
- Studying the speed at which the posterior concentrates around the true smooth regression function, they show that BART achieves near minimax-rate optimal performance when approximating a single smooth function and that it is certifiably optimal when the true function is an actual sum of smooth functions, again concentrating at a near minimax rate.

Conclusion

- BART is a Bayesian nonparametric, ensemble modeling method that can accommodate continuous, binary, categorical and time-to-event outcomes.
- Each tree in the “sum-of-trees” model is constrained by a regularization prior to be a weak learner, and by use of an iterative Bayesian backfitting MCMC algorithm, posterior samples are generated, thus allowing full posterior inference.
- In addition to its excellent predictive performance, as demonstrated by Chipman et al. (2010), extensions to the BART modeling framework for applications in areas like causal inference and the availability of software packages for easy implementation make BART a valuable modeling method.

Reference I

Hugh A Chipman, Edward I George, and Robert E McCulloch. Bart: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298, 2010.

Hugh A Chipman, Edward I George, Robert E McCulloch, and Thomas S Shively. mbart: Multidimensional monotone bart. *Bayesian Analysis*, 1(1):1–30, 2021.

P Richard Hahn, Jared S Murray, and Carlos M Carvalho. Bayesian regression tree models for causal inference: Regularization, confounding, and heterogeneous effects (with discussion). *Bayesian Analysis*, 15(3):965–1056, 2020.

Trevor Hastie and Robert Tibshirani. Bayesian backfitting (with comments and a rejoinder by the authors. *Statistical Science*, 15(3): 196–223, 2000.

Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

Reference II

Jingyu He, Saar Yalov, and P Richard Hahn. Xbart: Accelerated bayesian additive regression trees. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1130–1138. PMLR, 2019.

Jared S Murray. Log-linear bayesian additive regression trees for multinomial logistic and count regression models. *Journal of the American Statistical Association*, 116(534):756–769, 2021.

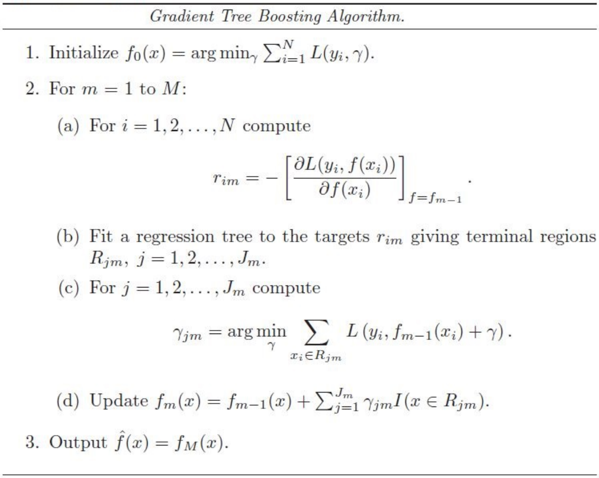
Veronika Ročková and Stéphanie van der Pas. Posterior concentration for bayesian regression trees and forests. *The Annals of Statistics*, 48(4):2108–2131, 2020.

Rodney Sparapani, Charles Spanbauer, and Robert McCulloch. Nonparametric machine learning and efficient computation with bayesian additive regression trees: the bart r package. *Journal of Statistical Software*, 97:1–66, 2021.

Reference III

Jennifer E Starling, Jared S Murray, Carlos M Carvalho, Radek K Bukowski, and James G Scott. Bart with targeted smoothing: An analysis of patient-specific stillbirth risk. *The Annals of Applied Statistics*, 14(1):28–50, 2020.

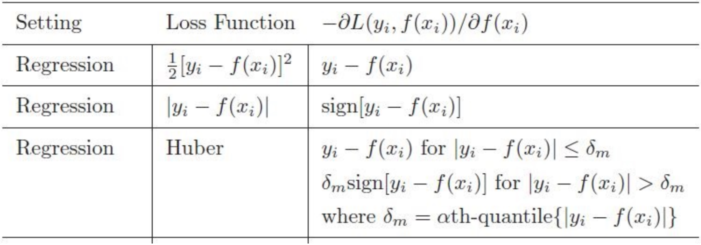
Appendix (copied from *Elements of Statistical Learning (Hastie et al., 2009)*)



Gradient Tree Boosting Algorithm.

```
1. Initialize $f_0(x) = \arg\min_\gamma \sum_{i=1}^{N} L(y_i, \gamma)$.
2. For $m = 1$ to $M$:
    (a) For $i = 1, 2, \ldots, N$ compute
            $r_{im} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f_{m-1}}$.
    (b) Fit a regression tree to the targets $r_{im}$ giving terminal regions
        $R_{jm},\ j = 1, 2, \ldots, J_m$.
    (c) For $j = 1, 2, \ldots, J_m$ compute
            $\gamma_{jm} = \arg\min_{\gamma} \sum_{x_i \in R_{jm}} L(y_i, f_{m-1}(x_i) + \gamma)$.
    (d) Update $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$.
3. Output $\hat{f}(x) = f_M(x)$.
```

Gradients for commonly used loss functions.



Setting	Loss Function	$-\partial L(y_i, f(x_i))/\partial f(x_i)$
Regression	$\frac{1}{2}[y_i - f(x_i)]^2$	$y_i - f(x_i)$
Regression	$\lvert y_i - f(x_i)\rvert$	$\text{sign}[y_i - f(x_i)]$
Regression	Huber	$y_i - f(x_i)$ for $\lvert y_i - f(x_i)\rvert \leq \delta_m$ $\delta_m \text{sign}[y_i - f(x_i)]$ for $\lvert y_i - f(x_i)\rvert > \delta_m$ where $\delta_m = \alpha\text{th-quantile}\{\lvert y_i - f(x_i)\rvert\}$