

Bayesian Additive Regression Trees for Counterfactual Prediction and Estimation

Michael Lamm, SAS Institute Inc.

ABSTRACT

Bayesian additive regression trees (BART) are a flexible and easy-to-use class of nonparametric models that are consistently cited as one of the best-performing tools for causal analysis of real-world data. Applications where BART models have shown great success include estimating causal effects, analyzing subgroups, and assisting in the incorporation of historical data. Properties of BART models that make them well suited for these tasks are their ability to capture potentially complex interactions without explicitly prespecifying and modeling them, the good generalization of the default model, and their ability to take a Bayesian approach to uncertainty quantification. This paper discusses the implementation of BART models in SAS® Visual Statistics software by using the BART procedure, and it demonstrates how you can use this procedure to estimate counterfactual outcomes and causal effects that are defined in a counterfactual framework.

INTRODUCTION

Shortly after Chipman, George, and McCulloch (2010) introduced the Bayesian additive regression trees (BART) model, it attracted attention for its potential for use in causal analysis and counterfactual estimation (Hill 2011). BART models have many properties that make them well suited for these tasks. In particular, their flexibility, when combined with the good generalization of the default model and the ability to assess uncertainty in model predictions from a Bayesian perspective, allows BART models to address some of the major challenges that arise in a regression-based approach to estimating counterfactual quantities. The usefulness of BART models for these tasks has indeed been observed in the literature. BART-based approaches were discussed as one of the top-performing methods at the 2016 Atlantic Causal Inference Conference Data Analysis Challenge (Dorie et al. 2019), and since then they have been used for numerous other tasks, such as subgroup analyses (Sivaganesan, Müller, and Huang 2017; Pan et al. 2022) and incorporating historical data (Zhou and Ji 2021). The growing body of literature highlights the role of BART models as a powerful tool for analyzing data that arise from real-world sources and nonrandomized observational studies.

This paper introduces BART models in SAS Visual Statistics and demonstrates how you can use the relatively new BART procedure to fit these models to estimate counterfactual quantities in the form of predictive margins. The paper is organized as follows. First, the background of BART models is presented, followed by an introduction to the estimation of counterfactual quantities in a potential outcomes framework and a discussion of why these models are well suited for this task. Features of the BART procedure are then discussed, with special consideration for computational concerns and issues of reproducibility. An example then illustrates how you can use PROC BART to fit models, as well as how you can use the fitted models to estimate counterfactual quantities in the form of predictive margins.

BACKGROUND

BART MODELS

BART models were originally inspired by the strong performance of tree-based gradient boosting models (Chipman, George, and McCulloch 2010). Like gradient boosting and random forest models, the BART algorithm uses a sum-of-trees ensemble to model a response variable. BART models therefore inherit many of the desirable properties of tree-based modeling techniques: namely, they can incorporate a mix of continuous and categorical predictors, they do not require the explicit specification of interactions between predictors, and they can naturally handle missing values in the predictors.

As described by Chipman, George, and McCulloch (2010), you fit BART models by using a Bayesian backfitting Markov chain Monte Carlo (MCMC) algorithm to produce samples of the sum-of-trees ensemble. At each iteration of the algorithm, a new sample of the ensemble is produced by sequentially

sampling updates to the individual trees. The fitted BART model consists of posterior samples of the ensemble that are saved after an initial burn-in period. You obtain predictions from the fitted model by averaging the prediction from each posterior sample of the ensemble. In addition to providing point estimates, the model enables you to assess uncertainty in the model predictions on the basis of variability of the posterior samples.

A key feature of BART models that makes them easy to use is the typically good predictive performance and good generalization of the default model. This is achieved by a well-designed regularization prior. The BART prior limits overfitting to the training data by placing greater prior probability on simpler tree structures, and it limits the contribution of a single tree to the ensemble prediction by applying shrinkage to the terminal node parameters. Although the regularization prior places greater prior probability on simpler tree structures, it does not impose hard constraints on the model, and complex tree structures are often sampled when they fit the training data well. Moreover, the performance of BART models typically does not change significantly if there are small changes in the prior parameters. As a result, fitting a BART model does not typically require hyperparameter tuning, and the default model fit tends to be comparable to that of other tree-based ensemble models, including random forests and gradient-boosted trees, which use hyperparameter tuning.

POTENTIAL OUTCOMES

In studying treatment effects, analysts commonly use potential outcomes, also referred to as counterfactual outcomes, to define causal estimands and to state the assumptions that are required for their unbiased estimation. Within the Neyman-Rubin potential outcomes framework (Rubin 1980, 1990), for each subject, a potential outcome is defined for each possible level of the treatment variable. In this framework, you interpret the potential outcomes as the outcome that would occur in a treatment scenario that might be contrary to the observed facts. That is, for a binary treatment variable T that takes the values 0 and 1, and outcome variable Y , there would be two potential outcomes, Y_1 and Y_0 . You can then naturally define the total causal effect of treatment T on outcome Y by using the potential outcomes Y_1 and Y_0 .

The potential outcome framework is commonly used for analyzing nonrandomized or imperfectly randomized data, which might arise from observational studies or from analyzing real-world data. In these settings, there are likely to be pretreatment variables that affect both the treatment decision for subjects and the outcome of interest. Pretreatment variables that are common causes of the treatment and the outcome are typically referred to as confounding variables. When confounding variables are present, the unbiased estimation of potential outcomes and the treatment effects that they are used to define require some form of adjustment. You can make adjustments by modeling the treatment variable, the outcome variable, or doubly robust methods that combine models for both. For an overview of different estimation methods and a more detailed discussion of the required assumptions, see Hernán and Robins (2020).

For this paper, the regression adjustment, or response surface, method is of the most interest. In this approach, you fit a model for the outcome variable by using the treatment variable T and the set of confounding variables X . If the set of confounding variables X satisfies an ignorability assumption, also known as an assumption of no unmeasured confounding, you can use the fitted model to obtain unbiased estimates for the potential outcome means. To do so, you predict counterfactual outcomes for each subject, assuming the treatment level of interest. You then average these predictions to obtain estimates for the potential outcome means in the entire population or in some well-defined subset.

The regression adjustment approach to estimating potential outcome means has a few important practical challenges to consider. First, in addition to assuming that the set of confounders X represents a valid adjustment set, there can still be concerns about model misspecification. These concerns are related to the second challenge for regression-based estimation of potential outcomes. Namely, it can be difficult to diagnose model misspecification, because you cannot use residual diagnostics and fit statistics to assess the model's performance for predicting counterfactual outcomes that correspond to unobserved levels of the treatment. The third challenge is then how to assess uncertainty in the counterfactual predictions.

These three challenges of regression-based approaches to counterfactual estimation and causal effect estimation motivate the use of BART models for these tasks. Concerns about model misspecification are minimized when you use a BART model, because the sum-of-trees ensemble can capture potentially

complex interactions without the need for explicit prespecification. For predicting the unobserved counterfactual outcomes, the effectiveness of the default regularization prior that is used to train a BART model is important, because it typically results in good generalization without requiring difficult hyperparameter tuning. Finally, because a BART model is a Bayesian model, you can naturally assess uncertainty in the counterfactual estimates on the basis of variability in predictions from the posterior samples.

Many simulation studies and data analysis competitions have observed these benefits of using BART models for counterfactual prediction, and BART-based approaches are typically among the top-performing methods (Dorie et al. 2019).

THE BART PROCEDURE

FEATURES

Starting in the 2022.1.1 release of SAS Visual Statistics, you can fit BART models of normally distributed response variables by using either the BART procedure or the `bartGauss` action in the Bayesian Additive Regression Trees (Bart) action set. PROC BART uses a standard statistical modeling syntax, where categorical predictors are identified in the CLASS statement, and the outcome and independent variables are identified in the MODEL statement. The MODEL statement does not require explicit modeling of interaction effects because of the nonparametric nature of BART models and their use of a sum-of-trees ensemble for model predictions. Both the procedure and action enable you to specify the parameters that the BART prior uses, parameters that control the size of the sum-of-trees ensemble, and parameters to control MCMC sampling.

The models that you train using either the procedure or the action can be saved as an analytic store. The analytic store lets you score new observations by using the `bartScore` action in the Bart action set, the `ASTORE` procedure, or the `score` action in the Astore action set. PROC ASTORE and the Astore action set are both part of SAS® Viya® machine learning. In addition to scoring new observations, you can use the analytic store to compute predictive margins by using the `bartScoreMargin` action in the Bart action set.

Predictive margins (Lane and Nelder 1982) are obtained by fixing the value of one or more input variables and averaging the predicted values over the distribution of covariates in an input data table. When the appropriate set of causal assumptions are satisfied, the computation of predictive margins corresponds to the regression-based approach to estimating the counterfactual potential outcome means. In addition to computing predictive margins, the `bartScoreMargin` action can compute the difference in predictive margins and create an output data table that has predicted marginal means from each saved sample of the ensemble for each predictive margin and difference that you request. When the predictive margin computations are used in a counterfactual framework for causal effect estimation, you can use the output data table with the marginal mean for each sample of the ensemble to compute effect estimates on a scale other than the difference scale.

Enhancements to PROC BART procedure and the Bart action set were recently made in the 2024.04 release of SAS Visual Statistics. In particular, probit models for binary response data can now be fit by using PROC BART or by using the `bartProbit` action in the Bart action set. Along with supporting probit models, starting in the 2024.04 release, PROC BART now supports the computations of predictive margins and the ability to score new data or compute predictive margins by using a previously fit model saved in an analytic store. Before the 2024.04 release, PROC BART could not use a saved model, and scoring new data or computing predictive margins required the use of PROC CAS to invoke either the `bartScore` or `bartScoreMargin` action. How to use PROC BART to compute predictive margins in training a model and with a restored model is demonstrated in the Example section of this paper.

COMPUTATIONAL CONSIDERATIONS

Training a BART model can be a computationally intensive process, because at each iteration, sampling an update for an individual tree requires at least one pass through the training data. The BART procedure supports options that can be used to reduce the time that it takes to train a model, at the cost of increased memory usage. The memory that these options require is proportional to the size of the training data, so

they might not be suitable for very large data sets. The TRAININMEM option in the PROC BART statement stores the data in memory when the model is trained. This option can reduce the time required to access the data and to assess a sampled modification for a tree. The MAPINMEM option in the PROC BART statement stores in memory a mapping of each observation to terminal nodes, or leaves, during model training. Storing this mapping in memory removes the need to route an observation through a tree when you sample an update to the tree. By default, the training data and the observation-to-leaf mapping are not stored in memory.

Because the BART procedure runs on SAS Cloud Analytic Services (CAS), you can run it on a cluster of machines. When you do that, by default, the procedure trains the model by running multiple chains and dividing the MCMC samples across worker nodes in the cluster. This distributed mode requires each worker node to have a full copy of the training data. In addition to allowing for multiple chains to be run in parallel, distributed mode eliminates the need for the large number of communications across the cluster of machines that are required when a single chain is run and each worker node is assigned a portion of the data. By running in distributed mode, you can use the NDISTCHAINS= option to specify the number of distributed chains to use. When NDISTCHAINS=0, a single chain is run, and each worker node is assigned a portion of the training data.

In addition to being computationally intensive to train, a BART model can also take more time to score new data. The default model that is fit by PROC BART consists of 1,000 saved samples of a 200-tree ensemble, and scoring new data requires you to route each observation through all 200,000 trees. Contrast this with the size of tree-based ensembles that are produced by gradient boosting or random forest algorithms, which might consist of a few hundred trees. The large number of posterior samples of the ensemble that the default BART model saves is required to assess uncertainty in the predictions. If the model is used as a purely predictive model and the Bayesian aspects of the model are not of interest, then a BART model that saves fewer samples of the ensemble for prediction can in some cases provide comparable fit, while taking less time to train and score new data. Instances where a BART model might be used as a purely predictive model include its use as a candidate algorithm in a super-learner (Van der Laan and Rose 2011) or as an initial outcome model for targeted maximum likelihood estimation (Gruber and Van der Laan 2009; Van der Laan 2010a, 2010b; Van der Laan and Rose 2011).

REPRODUCIBILITY

Model reproducibility is an important consideration in many settings. In general, models that you fit by using PROC BART depend on the number of chains that are run and the seed that you use for random number generation. In theory, BART models of normally distributed response data do not depend on the order of the observations. However, because of the finite precision of floating-point arithmetic, slight differences might occur in edge cases where the computations are sensitive to data ordering. In most cases, though, a BART model for normal data that is fit using the same initial random number seed and the same number of chains should be reproducible across multiple runs of the procedure.

The algorithm for training the probit BART model involves the simulation of latent variables for observations at each iteration. The simulation of these latent variables makes the probit BART model sensitive to the order of the data across nodes in a grid and to the order of the data across threads on a node. The distribution of data in a table on a CAS server can change between calls to a procedure, and you cannot use PROC SORT to sort a data table on a CAS server. However, you can use the Partition action in the Table action set to create partition and order-by keys based on variables in the table. All observations with the same partition key are assigned to the same worker node and thread, and the order-by key determines the ordering of observations in a partition. If the order-by key value is unique for each observation in the partitions, then when you run the procedure on the same distributed system, the row ordering is reproducible when the partition and order-by operations are applied.

For probit models, you can use the APPLYROWORDER option in the PROC BART statement to apply the partition and order-by keys to the input data table. These keys must have been previously created using the Partition action, and if they are not present for the data, you get a warning message. If the partition and order-by keys are present and you use the APPLYROWORDER option, information about the variables that define the keys is displayed in the "Model Information" table. Note that even when you apply the APPLYROWORDER option with unique keys, the distribution of data can differ across systems

that have a different number of threads or a different number of workers. Also, the distribution of data across threads can differ, depending on whether or not you use the TRAININMEM option. However, if you train the probit BART model on the same system, with the same unique partition and order-by keys, the same initial random number seed, and the same data access method, then the model fit should be reproducible up to differences in floating-point arithmetic.

EXAMPLE

This example demonstrates how you can use the BART procedure to compute counterfactual outcomes and predictive margins. The example uses the MARGINS statement, which was first added to the procedure in the 2023.04 release of SAS Visual Statistics. In prior versions of the software, you can perform the same computations by using the bartScoreMargin action in the Bart action set. For an example of how to use the bartScoreMargin action, see the example in Lamm (2022) or the example in the bartScoreMargin action documentation.

The data in this example come from the SmokingWeight data set that is used in the documentation examples for the CAUSALTRT procedure in SAS/STAT software. (You can access the data in the *SAS/STAT User's Guide* sample program GitHub repository.) The data are a subset of the NHANES 1 Epidemiological Follow-Up Study (NHEFS) that is used in Hernán and Robins (2020). For this example, the study population consists of individuals who reported being smokers in the baseline year, and the output of interest is the individuals' change in weight over a 10-year period. Counterfactual outcomes are estimated to study the effect that quitting smoking has on the individuals' weight change. The variables that the analysis uses are summarized in Table 1.

Variable	Description
Activity	Level of daily activity, with values of 0, 1, and 2
Age	Age in 1971
Change	Change in weight (in kg)
Education	Level of education, with values of 0, 1, 2, 3, and 4
Exercise	Amount of regular recreational exercise, with values of 0, 1 and 2
PerDay	Number of cigarettes smoked per day in 1971
Quit	Indicator of smoking cessation during the 10-year period
Race	0 for white; 1 otherwise
Sex	0 for male; 1 for female
YearsSmoke	Number of years a subject has smoked

Table 1: Variables in the SmokingWeight Data Used for Analysis

The following statements invoke the BART procedure to fit a model for the variable Change, compute predictive margins defined by the variable Quit, request a difference in the predictive margins, and create an analytic store to save the fitted model:

```
proc bart data=mycas.smokingWeight seed=1976;
  class Sex Race Education Exercise Activity Quit;
  model Change = Quit Sex Education Exercise Activity
    YearsSmoke PerDay Age;
  store mycas.SmokingFit;
  margin "Cessation" Quit = '1';
  margin "No Cessation" Quit = '0';
  margindiff ref= "No Cessation" event="Cessation" /
    label="Cessation Difference";
run;
```

The model is fit using the default values for the sum-of-trees ensemble and the BART prior. These values are summarized in the “Model Information” table and the “Prior Information” table, displayed in Figure 1. The default model uses an initial burn-in period of 100 iterations and consists of 1,000 saved samples of a 200-tree ensemble.

Model Information	
Data Source	SMOKINGWEIGHT
Target Variable	Change
Distribution	Normal
Number of Trees	200
Number of Bins	50
Bin Method	Quantiles
Missing Rule	Separate
Burn-In Size	100
Simulation Size	1000
Thinning Rate	1
Random Number Seed	1976

Prior Information	
Depth Zero Node Split Probability	0.95
Node Depth Power	2
Split Operation Probability	0.5
Prune Operation Probability	0.5
Leaf Variance Multiplier	2
Variance Prior Degrees of Freedom	3
Variance Prior Scale	10.69453
Variance Prior Quantile Level	0.9
Initial Variance Estimate	55.08177

Figure 1: Model Information and Prior Information

Figure 2 displays the “Sample Information” table, which summarizes basic information about the saved samples of the sum-of-trees ensemble. Although most of the trees in samples of the ensemble that is saved for prediction are simple, with an average of 3.33375 nodes per tree, more complex tree structures

are also saved; the largest tree consists of 15 nodes. Note that the example is trained in single-machine mode; the fit will be different if you fit the model on a grid of machines in distributed mode.

Sample Information	
Number of Samples Kept	1000
Proportion of Burn-In Operations Accepted	0.2102
Proportion of Sampling Operations Accepted	0.197705
Minimum Number of Tree Nodes	1
Maximum Number of Tree Nodes	15
Average Number of Tree Nodes	3.33375
Minimum Tree Depth	0
Maximum Tree Depth	6
Minimum Number of Leaves	1
Maximum Number of Leaves	8
Average Number of Leaves	2.166875
Minimum Leaf Size	5
Maximum Leaf Size	1679
Average Leaf Size	774.8486

Figure 2: Sample Information

You specify the predictive margins by using the MARGIN statement; you define each predictive margin in a separate statement. The MARGIN statement requires that you specify a name for the predictive margin, the variables that define the predictive margin, and the values that they are set to. This example defines two predictive margins by using the variable Quit. The predictive margin named “Cessation” sets the value of Quit to 1, which corresponds to the counterfactual scenario where subjects quit smoking; and the predictive margin named “No Cessation” sets the value of Quit to 0, which corresponds to the counterfactual scenario where subjects do not quit smoking. You request a difference in the predictive margins by using the MARGINDIFF statement. This statement specifies the event level and reference level for the comparison by using the predictive margin names that you specify in the MARGIN statements. Figure 3 displays the “Predictive Margins” and “Predictive Margin Differences” tables, which show point estimates for the predictive margins and the requested difference, along with 95% equal-tail credible intervals for the values. You can adjust the level that is used to construct the equal-tail credible intervals by specifying the ALPHA= option in the PROC BART statement.

Predictive Margins			
Description	Estimate	95% Equal-Tail Interval	
Cessation	5.18502	4.45435	5.88656
No Cessation	1.93216	1.50406	2.34517

Predictive Margin Differences			
Description	Estimate	95% Equal-Tail Interval	
Cessation Difference	3.2529	2.3898	4.0648

Figure 3: Predictive Margins and Differences for the Training Data

When you use both the MARGIN statement and a MODEL statement in a PROC BART call, the predictive margins are computed relative to the covariate distribution in the training data. In this case, if you were to assume that the covariates in the model make up a valid adjustment set for studying the effect of quitting

smoking on an individual's change in weight, then predictive margins would correspond to unbiased estimates for the corresponding potential outcomes. In that case, the difference in predictive margins would also correspond to an estimate for the average treatment effect, or ATE.

The STORE statement saves the results of the fitted model in an analytic store named SmokingFit. By using the saved model, you can also compute predictive margins relative to a covariate distribution other than the distribution in the training data. One alternative distribution that causal analysis motivates is to compute the predictive margins only for the subjects in a certain level of the designated treatment variable. If the appropriate causal assumptions are satisfied, these estimates correspond to the potential outcome means conditioned on a subject being in the designated treatment condition. You can use these conditional potential outcome mean estimates to estimate the so-called average treatment effect for the treated, or ATT, which is interpreted as the average effect of treatment among those who actually receive it. The following code computes predictive margins by using the saved model:

```
proc bart data=sascasl.smokingWeight (where=(Quit=1))
    restore=sascasl.SmokingFit;
    margin "Cessation" Quit = '1';
    margin "No Cessation" Quit = '0';
    margindiff ref= "No Cessation" event="Cessation" /
        label="Treated Cessation Difference";
run;
```

You specify the model that is saved in the analytic store by using the RESTORE= option in the PROC BART statement, and you specify the data that are used to compute the predictive margins in the DATA= option. This example uses a WHERE= option to compute the predictive margins by using only the observations for subjects in the designated treatment condition of QUIT=1.

When you use a restored model, the procedure produces a “Scoring Information” table, which provides information about the saved model. The table, displayed in Figure 4, includes the name of the data table that is used to train the model, information about the size of the ensemble, and the number of observations that are used in the predictive margin computations.

Scoring Information	
Store	SMOKINGFIT
Scoring Table	SMOKINGWEIGHT
Training Table	SMOKINGWEIGHT
Target Variable	Change
Missing Rule	Separate
MCMC Sample Size	1000
Number of Observations	464

Figure 4: Scoring Information

The predictive margins that are computed relative to the covariate distribution are displayed in Figure 5. For subjects who quit smoking, the predictive margins that are computed relative to the covariate distribution are less than the estimates relative to the entire population, though the difference in the predictive margins is still about 3 kilograms.

Predictive Margins			
Description	Estimate	95% Equal-Tail Interval	
Cessation	4.38144	3.69144	5.00959
No Cessation	1.27046	0.75982	1.74417

Predictive Margin Differences			
Description	Estimate	95% Equal-Tail Interval	
Treated Cessation Difference	3.1110	2.2858	3.9372

Figure 5: Predictive Margins and Difference for the Treated

When you compute predictive margins, you can use the OUTPUTMARGINS statement to create an output data table that contains the predicted marginal mean for each saved sample of the ensemble. With these data, you can compute effect estimates on scales other than the difference scale, or you can compute measures of uncertainty in the estimates other than the credible intervals that are provided in the procedure output.

CONCLUSION

BART models are a flexible, easy-to-use, and powerful tool for predictive modeling. The use of a sum-of-trees ensemble for prediction, combined with the good generalization of BART models and the ability to assess uncertainty in the model predictions from a Bayesian perspective, makes BART-based models well suited to estimating counterfactual outcomes and performing causal analyses. In SAS Visual Statistics, you can use the BART procedure to fit models for normally distributed data and binary data. Moreover, if you use BART models to estimate counterfactual outcomes, you can do so by using either the MARGIN statement in PROC BART or the bartScoreMargin action in the Bart action set.

REFERENCES

- Chipman, H. A., George, E. I., and McCulloch, R. E. (2010). "BART: Bayesian Additive Regression Trees." *Annals of Applied Statistics* 4:266–298. <https://doi.org/10.1214/09-AOAS285>.
- Dorie, V., Hill, J., Shalit, U., Scott, M., and Cervone, D. (2019). "Automated versus Do-It-Yourself Methods for Causal Inference: Lessons Learned from a Data Analysis Competition." *Statistical Science* 34:43–68. <https://doi.org/10.1214/18-STS667>.
- Gruber, S., and Van der Laan, M. J. (2009). "Targeted Maximum Likelihood Estimation: A Gentle Introduction." Working paper no. 252, University of California, Berkeley.
- Hernán, M. A., and Robins, J. M. (2020). *Causal Inference: What If*. Boca Raton, FL: Chapman & Hall/CRC.
- Hill, J. L. (2011). "Bayesian Nonparametric Modeling for Causal Inference." *Journal of Computational and Graphical Statistics* 20:201–204. <https://doi.org/10.1198/jcgs.2010.08162>.
- Lamm, M. (2022). "Introducing Bayesian Additive Regression Trees in SAS Viya." SAS Statistics Research and Applications paper no. 2, SAS Institute Inc., Cary, NC.
- Lane, P. W., and Nelder, J. A. (1982). "Analysis of Covariance and Standardization as Instances of Prediction." *Biometrics* 38:613–621.
- Pan, P., Bunn, V., Hupf, B., Lin, J. (2022). "Bayesian Additive Regression Trees (BART) with Covariate Adjusted Borrowing in Subgroup Analyses." *Journal of Biopharmaceutical Statistics* 32:613–626. <https://doi.org/10.1080/10543406.2022.2089160>.
- Rubin, D. B. (1980). "Comment on D. Basu, 'Randomization Analysis of Experimental Data: The Fisher Randomization Test'." *Journal of the American Statistical Association* 75:591–593.
- Rubin, D. B. (1990). "Comment: Neyman (1923) and Causal Inference in Experiments and Observational Studies." *Statistical Science* 5:472–480.
- Sivaganesan, S., Müller, P., and Huang, B. (2017). "Subgroup Finding via Bayesian Additive Regression Trees." *Statistics in Medicine*, 36: 2391–2403. <https://doi.org/10.1002/sim.7276>.

Van der Laan, M. J. (2010a). “Targeted Maximum Likelihood Based Causal Inference: Part I.” *International Journal of Biostatistics* 6:2.

Van der Laan, M. J. (2010b). “Targeted Maximum Likelihood Based Causal Inference: Part II.” *International Journal of Biostatistics* 6:3.

Van der Laan, M. J., and Rose, S. (2011). *Targeted Learning: Causal Inference for Observational and Experimental Data*. New York: Springer.

Zhou, T., and Ji, Y. (2021). “Incorporating External Data into the Analysis of Clinical Trials via Bayesian Additive Regression Trees.” *Statistics in Medicine*. 40: 6421–6442. <https://doi.org/10.1002/sim.9191>.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author:

Michael Lamm
SAS Institute
Michael.Lamm@sas.com

Any brand and product names are trademarks of their respective companies.