The Journal of Open Source Software

# BART-Survival: A Bayesian machine learning approach to survival analyses in Python

**Jacob Tiegs** [1,2], **Julia Raykin**[1¶], **and Ilia Rochlin**[1¶]

**1** Inform and Disseminate Division, Office of Public Health Data, Surveillance, and Technology, Centers for Disease Control and Prevention, Atlanta, GA, USA **2** Metas Solutions, Atlanta, Georgia ¶ Corresponding author

**DOI:** 10.xxxxxx/draft

**Software**

- Review
- Repository
- Archive

**Editor:** Open Journals

**Reviewers:**

- @openjournals

**Submitted:** 01 January 1970
**Published:** unpublished



## Summary

`BART-Survival` is a Python package that allows time-to-event (survival) analyses in discrete-time using the non-parametric machine learning algorithm, Bayesian Additive Regression Trees (BART). `BART-Survival` combines the performance of the BART algorithm with the complementary data and model structural formatting required to complete the survival analyses. The library contains a convenient application programming interface (API) that allows a simple approach when using the library for survival analyses, while maintaining capabilities for added complexity when desired. The package is intended for analysts exploring use of flexible non-parametric alternatives to traditional (semi-)parametric survival analyses.

## Statement of need

Survival analyses are a cornerstone of public health and clinical research in such diverse fields as cancer, cardiovascular disease, and infectious diseases (Altman & Bland, 1998; Bradburn et al., 2003). Traditional parametric and semi-parametric statistical methods, such as the Cox proportional hazards model, are commonly employed for survival analyses (Cox, 1972). However, these methods have several limitations, particularly when applied to complex data. One major issue is the need for restrictive assumptions, such as proportional hazards and predefined functional forms, which may not hold true in complex, real-world healthcare data (Harrell, 2015; Ishwaran et al., 2008). Additionally, these methods often struggle with high-dimensional datasets, leading to problems with overfitting, multicollinearity, and dealing with complex interactions (Ishwaran et al., 2008; Joffe et al., 2013).

More recently, non-parametric machine learning approaches have been introduced to address these limitations by reducing the need for restrictive assumptions and providing increased capabilities for more accurately modeling underlying distributions and complex interactions (Harrell, 2015; Ishwaran et al., 2008). BART is one such machine learning method that has demonstrated utility in the survival setting through its performance in identifying underlying statistical distributions (Chipman et al., 2010; R. Sparapani et al., 2021). BART offers flexibility in modeling complex relationships and interactions within the data without requiring the specification of a particular functional form (R. A. Sparapani et al., 2016).

Currently, the only BART survival algorithm readily available exists as part of the `BART` R package, which contains a library of various BART-based approaches in addition to a BART survival analysis application (R. Sparapani et al., 2021; R. A. Sparapani et al., 2016). BART-Survival package described here combines the survival analysis approach outlined in the `BART` R package with the foundational Python-based probabilistic programming language library, `PyMC` (Abril-Pla et al., 2023), and the accompanying BART algorithm from the `PyMC-BART` library (Quiroga et al., 2023). Our aim in developing `BART-Survival` is to provide accessibility

to the BART survival algorithm within the Python programing language. This contribution is beneficial for analysts when Python is the preferred programming language, the analytic workflow is Python-based, or when the R language is unavailable for analyses. Additionally, `BART-Survival` package abstracts away the complexities of the `PyMC` and `PyMC-BART` libraries through use of a pre-specified core model and generalized functionality that can accommodate analyses across various survival settings. The `BART-Survival` package is intended for public health and clinical professionals and students who are looking for non-parametric alternatives to traditional (semi-)parametric survival analysis, especially for use in large, complex healthcare data and machine learning applications.

# Methods

The following sections provides details on the methods employed by the BART-Survival library, focusing specifically on the discrete-time Survival algorithm used. For review of the BART algorithm we refer to associated PyMC-BART publication (Quiroga et al., 2023).

## Background

The `BART-Survival` package provides a discrete-time Survival method which aims to model Survival as a function of a series of probabilities (indicating risk of event) that can be determined from the sequence of discrete-time intervals examined. In combination with a structural configuration of the data, the discrete-time algorithm allows for flexible modeling of the risk probabilities as a non-parametric function of time and observed covariates. The series of probability risks can then be used in deriving Survival probabilities, along with other estimates of interest.

The foundation of the method is simple.

1. Create a sequence of time intervals, denoted as $t_j$ with $(j = 1, ..., k)$, from the range of observed event times.
2. Then for each interval $t_j$ obtain the number of observations with an event, along with the total number of observations at risk for having an event.
3. Finally, the risk of event occurrence within each interval $t_j$ can naively be derived as:

$$P_{t_j} = \frac{\text{n events}_{t_j}}{\text{n at risk}_{t_j}}$$

and the Survival probability $S(t)$ at a time $q$, can be derived as:

$$S(t_q) = \prod_{j=1}^{q} (1 - P_{t_j})$$

`BART-Survival` builds off this simple foundation by replacing $P_t$ with a probability risk estimate, $p_{t_j|x_i}$ yielded from the BART regression model. The predicted values $p_{t_j|x_i}$ are generated for each observation, at each time interval from the set $j$. Downstream targets can then be derived from these predicted values. For example the Survival probability curve for a single observation can be derived as:

$$S(t_q|x_i) = \prod_{j=1}^{q} (1 - p_{t_j|x_i})$$

## Data Preparation

To properly model $p_{t|x}$ the data requires an transformation from a standard Survival dataset to a *augmented* dataset. Survival data is typically given as a paired (**event status**, **event time**) outcome, along with a set of covariates for each observation. In this setup **event status** is typically a binary variable (1=event; 0=censored) and **event time** is a representation of time.

The *augmented* dataset transforms the generic dataset from a single, paired (**event status**, **event time**) outcome per observation, to a sequence of single (**event status**) outcomes over the series of discrete-time intervals, up to the given **event time** for the observation.

For example if the unique set of a dataset's **event times** is **{4,6,7,8,12,14}**, and a single observation's paired outcome is (**event status** $= 1$, **event time** $= 12$), then the single observation will be represented in the *augmented* dataset as the sequence of rows:

| event status | time |
|---|---|
| 0 | 4 |
| 0 | 6 |
| 0 | 7 |
| 0 | 8 |
| 1 | 12 |

Each row in the *augmented* dataset can then be treated as an independent observation, with **event status** as the outcome $Y$ and the **event time** $T$ as an added covariate. Now each of the original observations are represented by $j$ rows ($j = 1, ..., i_{\text{event time}}$) and the corresponding variables can be denoted as ($y_{ij}$, $t_j$, $x_{ij}$).

## Model

Using the new *augmented* dataset, the model is simplified to a probit regression of $y_{ij}$ on time $t_j$ and covariates $x_{ij}$, which yields a latent value $p_{ij}$ corresponding to $P(y_{ij} = 1)$. Explicitly the model is defined as:

$$
\begin{aligned}
y_{ij}|p_{ij} &\sim Bernoulli(p_{ij}) \\
p_{ij}|\mu_{ij} &= \Phi(\mu_{ij}) \\
\mu_{ij} &\sim \text{BART}(j, x_i)
\end{aligned}
$$

A trained `BART-Survival` model can then be used to yield the $p_{ij}$ predictions, which can be used to derive Survival as described above.

## Inference

A common goal of Survival regression models is to derive a statistical estimate for the adjusted effect of a variable on the outcome. A classic example of these estimates are Hazard Ratios derived from the coefficients of Cox Proportional Hazard Models.

With `BART-Survival`, the underlying BART algorithm does not rely on a linear equation and therefore does not produce coefficient that can be treated as conditional measures of effects. Instead, variable effects are summarized as marginal effect estimates which can derived through use of partial dependence functions.

The partial dependence function method is relatively simple. It involves generating predictions of $p$ from a trained `BART-Survival` model using a *augmented partial dependence* (*APD*) dataset as input.

The *APD* dataset is generated so that a specific variable $x_{[I]}$ is deterministically set to a specific value for all observations while the other covariates $x_{[O]_i}$ remain as observed. Each observation is then expanded over the time-intervals $1, ..., j_{T_{max}}$ to create the discrete-time datasets. Here $j_{T_{max}}$ is the maximum time across all **event times**.

To estimate the effect of an variable on the outcome, multiple *APD* datasets are created. Each *APD* dataset varies the value of the specific variable of interest (i.e. $x_{[I]_1}$, $x_{[I]_2}$ ), allowing evaluation of the outcome under different conditions of that variable. The $p_{ij}$ values from each predicted dataset ($p_{[1]}$, $p_{[2]}$), can be used to obtain the marginal estimates of a specific outcome.

Common marginal effect estimates derived from these predicted values include:

- Marginal difference is Survival probability at time $j$:

$$\text{Surv Diff}_{marg} = E_i[S_{p_{[2]}}(t_j)] - E_i[S_{p_{[1]}}(t_j)]$$

- Marginal Risk Ratio at time $j$:

$$\text{RR}_{marg} = \frac{E_i[p_{[2]_j}]}{E_i[p_{[1]_j}]}$$

- Marginal Hazard Ratio (assuming constant hazard rates):

$$\text{HR}_{marg} = \frac{E_{ij}[p_{[2]}]}{E_{ij}[p_{[1]}]}$$

In addition to providing point estimates, the `BART-Survival` naturally generates Bayesian credible intervals as a product of the posterior predictive distribution. The credible intervals can provide useful measure of uncertainty and allows for bayesian variants of hypothesis testing and statistical inference to be made on the estimates.

## Conclusion

`BART-Survival` provides the computational methods required for completing non-parametric discrete-time Survival analysis. This approach can have several advantages over alternative Survival methods. These advantages include capabilities to incorporate non-linear and interaction effects into the model, naturally ability to regularize the model (which reduces the risk of over-fitting) and of being robust to issues of multi-collinearity. The `BART-Survival` approach is especially useful when the assumptions of alternative Survival methods are violated.

Our library provides a convenient API for completing discrete-time Survival analysis, along with the functionality to customize the methodology as needed. The associated API documentation can be found here, along with the associated github repository `BART-Survival`.

## Acknowledgements

We thank Oscar Rincón-Guevara for helpful suggestions and review. We also thank Tegan Boehmer, Sachin Agnihotri, and Matt Ritchey for supporting the project throughout its development.

## References

Abril-Pla, O., Andreani, V., Carroll, C., Dong, L., Fonnesbeck, C. J., Kochurov, M., Kumar, R., Lao, J., Luhmann, C. C., Martin, O. A., Osthege, M., Vieira, R., Wiecki, T., & Zinkov,

R. (2023). PyMC: A modern, and comprehensive probabilistic programming framework in Python. *PeerJ Computer Science*, *9*, e1516. https://doi.org/10.7717/peerj-cs.1516

Altman, D. G., & Bland, J. M. (1998). Statistics Notes: Time to event (survival) data. *BMJ*, *317*(7156), 468–469. https://doi.org/10.1136/bmj.317.7156.468

Bradburn, M. J., Clark, T. G., Love, S. B., & Altman, D. G. (2003). Survival Analysis Part II: Multivariate data analysis – an introduction to concepts and methods. *British Journal of Cancer*, *89*(3), 431–436. https://doi.org/10.1038/sj.bjc.6601119

Chipman, H. A., George, E. I., & McCulloch, R. E. (2010). BART: Bayesian additive regression trees. *The Annals of Applied Statistics*, *4*(1). https://doi.org/10.1214/09-AOAS285

Cox, D. R. (1972). Regression Models and Life-Tables. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *34*(2), 187–202. https://doi.org/10.1111/j.2517-6161.1972.tb00899.x

Harrell, F. E. (2015). *Regression Modeling Strategies: With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis*. Springer International Publishing. https://doi.org/10.1007/978-3-319-19425-7

Ishwaran, H., Kogalur, U. B., Blackstone, E. H., & Lauer, M. S. (2008). Random survival forests. *The Annals of Applied Statistics*, *2*(3). https://doi.org/10.1214/08-AOAS169

Joffe, E., Coombes, K. R., Qiu, Y. H., Yoo, S. Y., Zhang, N., Bernstam, E. V., & Kornblau, S. M. (2013). Survival Prediction In High Dimensional Datasets – Comparative Evaluation Of Lasso Regularization and Random Survival Forests. *Blood*, *122*(21), 1728–1728. https://doi.org/10.1182/blood.V122.21.1728.1728

Quiroga, M., Garay, P. G., Alonso, J. M., Loyola, J. M., & Martin, O. A. (2023). *Bayesian additive regression trees for probabilistic programming* (No. arXiv:2206.03619). arXiv. https://doi.org/10.48550/arXiv.2206.03619

Sparapani, R. A., Logan, B. R., McCulloch, R. E., & Laud, P. W. (2016). Nonparametric survival analysis using Bayesian Additive Regression Trees (BART). *Statistics in Medicine*, *35*(16), 2741–2753. https://doi.org/10.1002/sim.6893

Sparapani, R., Spanbauer, C., & McCulloch, R. (2021). Nonparametric Machine Learning and Efficient Computation with Bayesian Additive Regression Trees: The **BART** *R* Package. *Journal of Statistical Software*, *97*(1). https://doi.org/10.18637/jss.v097.i01