Overview

`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 from the PyMC-BART library with the complementary data and model structural formatting required to provide a convenient approach to conducting high performance, non-parametric survival analysis.

This repository contains the source code and documentation for the BART_SURVIVAL package as well as user-guides/example notebooks. We additionally provide the code used in conducting the validation study of the algorithm.

Background

Survival analysis methods are statistical methods used to describe the risk of an event occurrence over a period of time. The BART-Survival package provides a discrete-time survival method which aims to model survival as a function of the cumulative risk of event occurence over the series of discrete time intervals.

Using discrete-time intervals provides a convenient approach to flexibly model the latent probability of event as a non-parametric function of the distinct time interval and a set of observation covariates. The latent probabilities can then be used for deriving survival probability or other estimates.

The foundation of the method is simple. First create a sequence of time intervals, denoted as t_j with $(j = 1, ..., k)$, from the range of observed event times. 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. Finally, the risk of event occurence within each interval t_j can naively be derived as: $

$$P_{t_j} = \frac{\text{n events}_{t_j}}{\text{n at risk}_{t_j}} \tag{1}$$

$

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

$

$$S(t_q) = \prod_{j=1}^{q} (1 - P_{t_j}) \tag{2}$$

$

where $q \in j$.

BART-Survival builds off this simple foundation by replacing P_t with a probability risk estimate, $p_{t_j|x_i}$ yielded from a BART regression model for each distinct observation in the dataset and survival can be estimated as:

$

$$S(t_q|x_i) = \prod_{j=1}^{q}(1 - p_{t_j|x_i}) \tag{3}$$

$

To properly model $p_{t|x}$, the data requires an transformation from the standard dataset to a *augmented* dataset. Standard survival data is given as a paired (**event status**, **event time**) outcome and set of covariates for each observation. **Event status** is typically a binary variable (1=event; 0=censored) and **event time** is some continous 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 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 times** = 12), then the observation will be represented in the *augmented* dataset as the sequence of observations:

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

Each row in the *augmented* dataset is 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}).

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:

$

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

$

where Φ is the CDF of the Normal distribution.

A trained BART-Survival model yields the p_{ij} esimates which can be used to derive survival as decribed in equation (3).

Inference A typical goal of survival regression models is to derive a statistical estimate for the effect of a variable on the outcome. The classic example being Hazard Ratios simply derived from the coefficients of Cox Proportional Hazard Models.

With BART-Survival, the underlying BART algorithm does not rely on a linear equation and does not produce coefficient that can be treated as measures of effects. Instead, variable effects are summarized as marginial effect estimates which can derived through use of partial depence 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 $x_{i[O]}$ remain as observed. Each observation is then expanded over the time-intervals $1, ..., j_{T_{max}}$ to create the discrete-time datasets.

Mulitple APD datasets can be created, each with different values of the specific variable of interest (i.e. $x_{[I]_1}$, $x_{[I]_2}$). The p_{ij} values from each predicted dataset ($p_{[1]}$, $p_{[2]}$), can then be contrasted.

Common marginal effect estimates derived from these predicted values include:

- Marginal difference is survival probability at time j:

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

- 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):

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

Uncertaintity intervals for the estimates are additionally generated from the posterior predictive distributions and naturally accompany the estimated point values.

Summary The BART-Survival package provide the algorithms necessary to complete the 3 major steps of the survival analysis.

1. Generate augmented dataset
2. Train-predict-transform the BART model and estimates
3. Generate *augmented partial dependece* datasets and generate estimates.

Installation

Bart-Survival can be accessed directly from PyPi: `pip install bart-survival==0.1.1`

Additonaly the whl/tar.gz and src code is accessible in the github repo: https://github.com/CDCgov/BART-Survival/dist https://github.com/CDCgov/BART-Survival/src

Demonstration

API

https://cdcgov.github.io/BART-Survival/build/html/index.html

User Guide/Example Notebooks

https://github.com/CDCgov/BART-Survival/blob/main/examples/example1.ipynb

https://github.com/CDCgov/BART-Survival/blob/main/examples/example2.ipynb

Validation study links

COMING SOON

CDCgov General Disclaimers

This repository was created for use by CDC programs to collaborate on public health related projects in support of the CDC mission. GitHub is not hosted by the CDC, but is a third party website used by CDC and its partners to share information and collaborate on software. CDC use of GitHub does not imply an endorsement of any one particular service, product, or enterprise.

Related Documents

- Open Practices
- Rules of Behavior
- Disclaimer
- Contribution Notice
- Code of Conduct
- Licence

Public Domain Standard Notice

This repository constitutes a work of the United States Government and is not subject to domestic copyright protection under 17 USC § 105. This repository is in the public domain within the United States, and copyright and related rights in the work worldwide are waived through the CC0 1.0 Universal public domain dedication. All contributions to this repository will be released under the CC0 dedication. By submitting a pull request you are agreeing to comply with this waiver of copyright interest.

License Standard Notice

The repository utilizes code licensed under the terms of the Apache Software License and therefore is licensed under ASL v2 or later.

This source code in this repository is free: you can redistribute it and/or modify it under the terms of the Apache Software License version 2, or (at your option) any later version.

This source code in this repository is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the Apache Software License for more details.

You should have received a copy of the Apache Software License along with this program. If not, see http://www.apache.org/licenses/LICENSE-2.0.html

The source code forked from other open source projects will inherit its license.

Privacy Standard Notice

This repository contains only non-sensitive, publicly available data and information. All material and community participation is covered by the Disclaimer and Code of Conduct. For more information about CDC's privacy policy, please visit http://www.cdc.gov/other/privacy.html.

Contributing Standard Notice

Anyone is encouraged to contribute to the repository by forking and submitting a pull request. (If you are new to GitHub, you might start with a basic tutorial.) By contributing to this project, you grant a world-wide, royalty-free, perpetual, irrevocable, non-exclusive, transferable license to all users under the terms of the Apache Software License v2 or later.

All comments, messages, pull requests, and other submissions received through CDC including this GitHub page may be subject to applicable federal law, including but not limited to the Federal Records Act, and may be archived. Learn more at http://www.cdc.gov/other/privacy.html.

Records Management Standard Notice

This repository is not a source of government records, but is a copy to increase collaboration and collaborative potential. All government records will be published through the CDC web site.

Additional Standard Notices

Please refer to CDC's Template Repository for more information about contributing to this repository, public domain notices and disclaimers, and code of conduct.