Package 'PSPI'

December 2, 2025

Type Package

Title Propensity Score Predictive Inference for Generalizability

Version 1.2

Date 2025-11-15

Author Jungang Zou [aut, cre],
Qixuan Chen [aut],
Joseph Schwartz [aut],
Nathalie Moise [aut],
Roderick Little [aut],
Robert McCulloch [ctb],
Rodney Sparapani [ctb],
Charles Spanbauer [ctb],
Robert Gramacy [ctb],
Jean-Sebastien Roy [ctb]

Maintainer Jungang Zou <jungang.zou@gmail.com>

Description Provides a suite of Propensity Score Predictive Inference (PSPI) methods to generalize treatment effects in trials to target populations. The package includes an existing model Bayesian Causal Forest (BCF) and four PSPI models (BCF-PS, FullBART, SplineBART, DSplineBART). These methods leverage Bayesian Additive Regression Trees (BART) to adjust for high-dimensional covariates and nonlinear associations, while SplineBART and DSplineBART further use propensity score based splines to address covariate shift between trial data and target population.

License GPL-2

Encoding UTF-8

Depends R (>= 4.1.0)

Imports Rcpp, arm, dplyr, mvtnorm, stringr, stats, nnet, methods

LinkingTo Rcpp, RcppArmadillo, RcppDist, RcppProgress, pg

RoxygenNote 7.3.2

SystemRequirements GNU make

Suggests knitr, rmarkdown, mitml

NeedsCompilation yes

Repository CRAN

Date/Publication 2025-12-02 07:50:21 UTC

Contents

`PSPI-package` *Propensity Score Predictive Inference for Generalizability*

Description

Provides a suite of Propensity Score Predictive Inference (PSPI) methods to generalize treatment effects in trials to target populations. The package includes an existing model Bayesian Causal Forest (BCF) and four PSPI models (BCF-PS, FullBART, SplineBART, DSplineBART). These methods leverage Bayesian Additive Regression Trees (BART) to adjust for high-dimensional covariates and nonlinear associations, while SplineBART and DSplineBART further use propensity score based splines to address covariate shift between trial data and target population.

Details

PSPI provides Bayesian methods for generalizing treatment effects from clinical trials to target populations. It implements five models-`BCF`, `BCF_P`, `FullBART`, `SplineBART`, and `DSplineBART`-built on Bayesian Additive Regression Trees (BART). Spline-based variants (`SplineBART` and `DSplineBART`) use propensity score transformations and spline terms to handle covariate shift between datasets. Core computations rely on efficient MCMC routines implemented in C++.

This package modifies and extends C++ code originally derived from the BART3 package, developed by Rodney Sparapani, which is licensed under the GNU General Public License version 2 (GPL-2).

The modified code is redistributed in accordance with the GPL-2 license. For more details on the modifications, see the package's documentation.

References

BART3 package: https://github.com/rsparapa/bnptools/tree/master, originally developed by Rodney Sparapani.

Examples

```
   sim <- sim_data(scenario = "linear", n_trial = 60)

 fit <- PSPI_generalizability(
   X = as.matrix(sim$trials[, paste0("X", 1:10)]),
   Y = sim$trials$Y,
   A = sim$trials$A,
   pi = sim$population$ps[sim$population$selected],
   X_pop = as.matrix(sim$population[, paste0("X", 1:10)]),
   pi_pop = sim$population$ps,
   model = "SplineBART",
   transformation = "InvGumbel",
   verbose = FALSE,
   nburn = 1, npost = 1
)
   str(fit)
```

`PSPI_generalizability` *Propensity Scores Predictive Inference for Generalizability*

Description

This is the main function of the **PSPI** package. It runs Bayesian models that generalize findings from a clinical trial to a target population, estimating the average treatment effects and potential outcomes. Propensity scores of trial participation play the central role for generalizability analysis. When covariate shift is an issue, we recommend PSPI-SplineBART and PSPI-DSplineBART, which leveraging Bayesian Additive Regression Trees (BART) to model high-dimensional covariates, and propensity scores based splines to extrapolate smoothly.

Users provide trial data (covariates, outcomes, treatment, and propensity scores) along with population-level covariates and propensity scores. Propensity scores can be the true values or estimated from some models. The function then performs Monte Carlo Markov chain (MCMC) for the posterior inference.

Usage

```
PSPI_generalizability(
  X,
  Y,
  A,
  pi,
  X_pop,
  pi_pop,
  model,
  transformation = "InvGumbel",
  nburn = 4000,
  npost = 4000,
  n_knots_main = NULL,
  n_knots_inter = NULL,
  order_main = 3,
  order_inter = 3,
  ntrees_s = 200,
  verbose = FALSE,
  seed = NULL
)
```

Arguments

`X` Matrix of covariates for the trial data.

`Y` Numeric vector of observed outcomes in the trial.

`A` Binary vector of treatment assignments (0 = control, 1 = intervention).

`pi` Numeric vector of trial propensity scores (probability of trial participation).

`X_pop` Matrix of covariates for the target population data.

`pi_pop` Numeric vector of the target population propensity scores.

`model` Character string specifying which PSPI model to use (see Details).

`transformation` Character string indicating the transformation applied to the propensity scores. Options are `"Identity"`, `"Logit"`, `"Cloglog"`, or `"InvGumbel"` (default).

`nburn` Number of burn-in iterations (default = 4000).

`npost` Number of posterior iterations saved after burn-in (default = 4000).

`n_knots_main`, `n_knots_inter`
Number of spline knots for main and interaction effects. If `NULL`, defaults are chosen automatically. `n_knots_inter` is available for `SplineBART` and `DSplineBART`; `n_knots_main` is available only for `DSplineBART`.

`order_main`, `order_inter`
Order of spline basis functions (default = 3). `order_inter` applies to both `SplineBART` and `DSplineBART`; `order_main` applies only to `DSplineBART`.

`ntrees_s` Number of trees used for the BART component (default = 200).

`verbose` Logical; if TRUE, prints progress messages.

`seed` Optional random seed for reproducibility.

Details

Model choices

The `model` argument selects the type of PSPI model to be fitted:

- `"BCF"` – Bayesian Causal Forests (Hahn et al., 2020).
- `"BCF_P"` – BCF with the propensity score as an additional predictor.
- `"FullBART"` – Uses three BARTs to estimate treatment effects.
- `"SplineBART"` – Incorporates a natural cubic spline for heterogeneous treatment effects.
- `"DSplineBART"` – Adds another natural cubic spline for the prognostic score.

Propensity score transformations

Since splines are sensitive to scales of predictor, robust transformation is needed. The propensity scores (`pi` for trial, `pi_pop` for population) can be optionally transformed before modeling using one of the following:

- `"Identity"` – uses the raw propensity scores directly (no transformation).
- `"Logit"` – applies the logit transform: $g(p) = \log(p/(1-p))$.
- `"Cloglog"` – complementary log–log transform: $g(p) = \log(-\log(1-p))$.
- `"InvGumbel"` – inverse Gumbel transform: $g(p) = -\log(-\log(p))$. Default choice.

Users can experiment with different transformations to assess model sensitivity.

Spline settings

Spline-based models (`"SplineBART"` and `"DSplineBART"`) allow flexible extrapolation to address covariate shift. The number and order of spline basis functions can be customized through the following parameters:

- `n_knots_inter`, `order_inter`: number and order of spline knots for treatment-interaction effects. Available for both `SplineBART` and `DSplineBART`.
- `n_knots_main`, `order_main`: number and order of spline knots for main effects. Available only for `DSplineBART`.

If any of these are left as `NULL`, default values are chosen automatically based on the cube root of the sample size (ensuring a reasonable smoothness level).

Value

A list containing posterior samples and model summaries produced by the C++ sampler. Typical elements include:

post_outcome1 Each row is a posterior draw for individual potential outcome under treatment

post_outcome0 Each row is a posterior draw for individual potential outcome under control

post_te Each row is a posterior draw for individual treatment effects

Note

This function utilizes modified C++ code originally derived from the BART3 package (Bayesian Additive Regression Trees). The original package was developed by Rodney Sparapani and is licensed under GPL-2. Modifications were made by Jungang Zou, 2024. For more information about the original BART3 package, see: https://github.com/rsparapa/bnptools/tree/master/BART3

Examples

```
# Example with simulated data
sim <- sim_data(scenario = "linear", n_trial = 60)

fit <- PSPI_generalizability(
  X = as.matrix(sim$trials[, paste0("X", 1:10)]),
  Y = sim$trials$Y,
  A = sim$trials$A,
  pi = sim$population$ps[sim$population$selected],
  X_pop = as.matrix(sim$population[, paste0("X", 1:10)]),
  pi_pop = sim$population$ps,
  model = "SplineBART",
  transformation = "InvGumbel",
  verbose = FALSE,
  nburn = 1, npost = 1
)

str(fit)
```

`sim_data` *Simulate a population and a randomized trial under PSPI scenarios*

Description

Generates a finite population of size 1000 with seven continuous and three binary covariates, constructs potential outcomes `Y0` and `Y1` according to the chosen scenario, simulates trial participation through a logistic selection model calibrated to target `n_trial` = 200 or 60, and returns both the target population and the randomized trial (with treatment assigned at probability `prop`).

Usage

```
sim_data(n_trial = 200, scenario = "linear", seed = NULL, prop = 0.5)
```

Arguments

`n_trial`	Integer. Target trial size; must be `200` or `60`.
`scenario`	Character. One of `"linear"`, `"linear+covariate shift"`, `"nonlinear"`, `"nonlinear+covariate shift"`.
`seed`	Optional integer seed for reproducibility. If `NULL`, the current RNG state is used.
`prop`	Numeric in `[0,1]`. Randomization probability $P(A = 1)$ within the trial.

Value

A `list` with two data frames:

population columns `X1:X10`, potential outcomes `Y1` and `Y0`, `selected` (logical), and `ps` (true propensity scores of trial participation).

trials columns `X1:X10`, `A`, and observed `Y`.

Examples

```
set.seed(2025)
sim <- sim_data(n_trial = 200, scenario = "nonlinear", prop = 0.5)
str(sim$population)
table(sim$trials$A)            # treatment allocation
mean(sim$population$selected)   # selection rate

# A smaller trial size and linear scenario with covariate shift
sim2 <- sim_data(n_trial = 60, scenario = "linear+covariate shift", seed = 1, prop = 0.6)
nrow(sim2$trials)
```

Index