Package ‘bartCause’

January 23, 2023

Version 1.0-6
Date 2023-01-23
Title Causal Inference using Bayesian Additive Regression Trees
Depends R (>= 3.1-0)
Imports dbarts (>= 0.9-16), methods, stats, graphics, parallel, utils, grDevices
Suggests testthat (>= 0.9-0), lme4, rpart, tmle, stan4bart
Description Contains a variety of methods to generate typical causal inference estimates using Bayesian Additive Regression Trees (BART) as the underlying regression model (Hill (2012) <doi:10.1198/jcgs.2010.08162>).
License GPL (>= 2)
NeedsCompilation no

URL https://github.com/vdorie/bartCause

BugReports https://github.com/vdorie/bartCause/issues

Author Vincent Dorie [aut, cre] (<https://orcid.org/0000-0002-9576-3064>), Jennifer Hill [aut] (<https://orcid.org/0000-0003-4983-2206>)
Maintainer Vincent Dorie <vdorie@gmail.com>
Repository CRAN
Date/Publication 2023-01-23 19:40:08 UTC

R topics documented:

bartc ................................................................. 2
bartc-generics .................................................... 8
bartc-plot .......................................................... 11
summary.bartcFit .................................................. 14

Index 17
Causal Inference using Bayesian Additive Regression Trees

Description

Fits a collection of treatment and response models using the Bayesian Additive Regression Trees (BART) algorithm, producing estimates of treatment effects.

Usage

```
bartc(response, treatment, confounders, parametric, data, subset, weights, 
method.rsp = c("bart", "tmle", "p.weight"), 
method.trt = c("bart", "glm", "none"), 
estimand = c("ate", "att", "atc"), 
group.by = NULL, 
commonSup.rule = c("none", "sd", "chisq"), 
commonSup.cut = c(NA_real_, 1, 0.05), 
args.rsp = list(), args.trt = list(), 
p.scoreAsCovariate = TRUE, use.ranef = TRUE, group.effects = FALSE, 
crossvalidate = FALSE, 
keepCall = TRUE, verbose = TRUE, 
seed = NA_integer_, 
...)
```

Arguments

- **response**: A vector of the outcome variable, or a reference to such in the data argument. Can be continuous or binary.
- **treatment**: A vector of the binary treatment variable, or a reference to data.
- **confounders**: A matrix or data frame of covariates to be used in estimating the treatment and response model. Can also be the right-hand-side of a formula (e.g., `x1 + x2 + ...`). The data argument will be searched if supplied.
- **parametric**: The right-hand-side of a formula (e.g., `x1 + x2 + (1 | g) ...`) giving the equation of a parametric form to be used for estimating the mean structure. See the details section below.
- **data**: An optional data frame or named list containing the response, treatment, and confounders.
- **subset**: An optional vector using to subset the data. Can refer to data if provided.
- **weights**: An optional vector of population weights used in model fitting and estimating the treatment effect. Can refer to data if provided.
- **method.rsp**: A character string specifying which method to use when fitting the response surface and estimating the treatment effect. Options are: "bart" - fit the response surface with BART and take the average of the individual treatment effect estimates, "p.weight" - fit the response surface with BART but compute the treatment effect estimate by using a propensity score weighted sum of individual
effects, and "tmle" - as above, but further adjust the individual estimates using the Targeted Minimum Loss based Estimation (TMLE) adjustment.

**method.trt**  
A character string specifying which method to use when fitting the treatment assignment mechanism, or a vector/matrix of propensity scores. Character string options are: "bart" - fit BART directly to the treatment variable, "glm" - fit a generalized linear model with a binomial response and all confounders added linearly, and "none" - do no propensity score estimation. Cannot be "none" if the response model requires propensity scores. When supplied as a matrix, it should be of dimensions equal to the number of observations times the number of samples used in any response model.

**estimand**  
A character string specifying which causal effect to target. Options are "ate" - average treatment effect, "att" - average treatment effect on the treated, and "atc" - average treatment effect on the controls.

**group.by**  
An optional factor that, when present, causes the treatment effect estimate to be calculated within each group.

**commonSup.rule**  
Rule for exclusion of observations lacking in common support. Options are "none" - no suppression, "sd" - exclude units whose predicted counterfactual standard deviation is extreme compared to the maximum standard deviation under those units’ observed treatment condition, where extreme refers to the distribution of all standard deviations of observed treatment conditions, "chisq" - exclude observations according to ratio of the variance of posterior predicted counterfactual to the posterior variance of the observed condition, having a Chi Squared distribution with one degree of freedom under the null hypothesis of have equal distributions.

**commonSup.cut**  
Cutoffs for commonSup.rule. Ignored for "none", when commonSup.rule is "sd", refers to how many standard deviations of the distribution of posterior variance for counterfactuals an observation can be above the maximum of posterior variances for that treatment condition. When commonSup.rule is "chisq", is the $p$ value used for rejection of the hypothesis of equal variances.

**p.scoreAsCovariate**  
A logical such that when TRUE, the propensity score is added to the response model as a covariate. When used, this is equivalent to the 'ps-BART' method described by described by Hahn, Murray, and Carvalho.

**use.ranef**  
Logical specifying if group.by variable - when present - should be included as a "random" or "fixed" effect. If true, rbart will be used for BART models. Using random effects for treatment assignment mechanisms of type "glm" require that the lme4 package be available.

**group.effects**  
Logical specifying if effects should be calculated within groups if the group.by variable is provided. Response methods of "tmle" and "p.weight" are such that if group effects are calculated, then the population effect is not provided.

**keepCall**  
A logical such that when FALSE, the call to bartc is not kept. This can reduce the amount of information printed by summary when passing in data as literals.

**crossvalidate**  
One of TRUE, FALSE, "trt", or "rsp". Enables code to attempt to estimate the optimal end-node sensitivity parameter. This uses a rudimentary Bayesian optimization routine and can be extremely slow.
verbose
Optional integer specifying the desired pRNG seed. It should not be needed when running single-threaded - set.seed will suffice, and can be used to obtain reproducible results when multi-threaded. See Reproducibility section of bart2.

seed
Further arguments to the treatment and response model fitting algorithms. Arguments passed to the main function as ... will be used in both models. args.rsp and args.trt can be used to set parameters in a single fit, and will override other values. See glm and bart2 for reference.

Details

bartc represents a collection of methods that primarily use the Bayesian Additive Regression Trees (BART) algorithm to estimate causal treatment effects with binary treatment variables and continuous or binary outcomes. This requires models to be fit to the response surface (distribution of the response as a function of treatment and confounders, \( p(Y(1), Y(0)|X) \)) and optionally for treatment assignment mechanism (probability of receiving treatment, i.e. propensity score, \( Pr(Z = 1|X) \)). The response surface model is used to impute counterfactuals, which may then be adjusted together with the propensity score to produce estimates of effects.

Similar to lm, models can be specified symbolically. When the data term is present, it will be added to the search path for the response, treatment, and confounders variables. The confounders must be specified devoid of any "left hand side", as they appear in both of the models.

Response Surface

The response surface methods included are:

- "bart" - use BART to fit the response surface and produce individual estimates \( \hat{Y}(1)_i \) and \( \hat{Y}(0)_i \). Treatment effect estimates are obtained by averaging the difference of these across the population of interest.
- "p.weight" - individual effects are estimated as in "bart", but treatment effect estimates are obtained by using a propensity score weighted average. For the average treatment effect on the treated, these weights are \( p(z_i|x_i)/(\sum z/n) \). For ATE, replace \( z \) with \( 1 - z \). For ATE, "p.weight" is equal to "bart".
- "tmle" - individual effects are estimated as in "bart" and a weighted average is taken as in "p.weight", however the response surface estimates and propensity scores are corrected by using the Targeted Minimum Loss based Estimation method.

Treatment Assignment

The treatment assignment models are:

- "bart" - fit a binary BART directly to the treatment using all the confounders.
- "none" - no modeling is done. Only applies when using response method "bart" and p.scoreAsCovariate is FALSE.
- "glm" - fit a binomial generalized linear model with logistic link and confounders included as linear terms.
- Finally, a vector or matrix of propensity scores can be supplied. Propensity score matrices should have a number of rows equal to the number of observations in the data and a number of columns equal to the number of posterior samples.
Parametrics

`bartc` uses the `stan4bart` package, when available, to fit semi-parametric surfaces. Equations can be given as to `lm`. Grouping structures are also permitted, and use the syntax of `lmer`.

Generics

For a fitted model, the easiest way to analyze the resulting fit is to use the generics `fitted`, `extract`, and `predict` to analyze specific quantities and `summary` to aggregate those values into targets (e.g. ATE, ATT, or ATC).

Common Support Rules

Common support, or that the probability of receiving all treatment conditions is non-zero within every area of the covariate space ($P(Z = 1|X = x) > 0$ for all $x$ in the inferential sample), can be enforced by excluding observations with high posterior uncertainty. `bartc` supports two common support rules through `commonSup.rule` argument:

- "sd" - observations are cut from the inferential sample if: $s_i^{(1-z)} > m_z + a \times sd(s_j^{f(z)})$, where $s_i^{(1-z)}$ is the posterior standard deviation of the predicted counterfactual for observation $i$, $s_j^{f(z)}$ is the posterior standard deviation of the prediction for the observed treatment condition of observation $j$, $sd(s_j^{f(z)})$ is the empirical standard deviation of those quantities, and $m_z = \max_j \{s_j^{f(z)}\}$ for all $j$ in the same treatment group, i.e. $Z_j = z$. $a$ is a constant to be passed in using `commonSup.cut` and its default is 1.

- "chisq" - observations are cut from the inferential sample if: $(s_i^{(1-z)}/s_i^{f(z)})^2 > q_\alpha$, where $s_i$ are as above and $q_\alpha$, is the upper $\alpha$ percentile of a $\chi^2$ distribution with one degree of freedom, corresponding to a null hypothesis of equal variance. The default for $\alpha$ is 0.05, and it is specified using the `commonSup.cut` parameter.

Special Arguments

Some default arguments are unconventional or are passed in a unique fashion.

- If `n.chains` is missing, unlike in `bart2` a default of 10 is used.
- For `method.rsp == "tmle"`, a special `arg.trt` of `posteriorOfTMLE` determines if the TMLE correction should be applied to each posterior sample (`TRUE`), or just the posterior mean (`FALSE`).

Missing Data

Missingness is allowed only in the response. If some response values are `NA`, the BART models will be trained just for where data are available and those values will be used to make predictions for the missing observations. Missing observations are not used when calculating statistics for assessing common support, although they may still be excluded on those grounds. Further, missing observations may not be compatible with response method "tmle".

Value

`bartc` returns an object of class `bartcFit`. Information about the object can be derived by using methods `summary`, `plot_sigma`, `plot_est`, `plot_indiv`, and `plot_support`. Numerical quantities are recovered with the `fitted` and `extract` generics. Predictions for new observations are obtained with `predict`.

Objects of class `bartcFit` are lists containing items:
method.rsp character string specifying the method used to fit the response surface
method.trt character string specifying the method used to fit the treatment assignment mechanism
estimand character string specifying the targeted causal effect
fit.rsp object containing the fitted response model
data.rsp `dbartsData` object used when fitting the response model
fit.trt object containing the fitted treatment model
group.by optional factor vector containing the groups in which treatment effects are estimated
est matrix or array of posterior samples of the treatment effect estimate
p.score the vector of propensity scores used as a covariate in the response model, when applicable
samples.p.score matrix or array of posterior samples of the propensity score, when applicable
mu.hat.obs samples from the posterior of the expected value for individual responses under the observed treatment regime
mu.hat.cf samples from the posterior of the expected value for individual responses under the counterfactual treatment
name.trt character string giving the name of the treatment variable in the data of `fit.rsp`
trt vector of treatment assignments
call how `bartc` was called
n.chains number of independent posterior sampler chains in response model
commonSup.rule common support rule used for suppressing observations
commonSup.cut common support parameter used to set cutoff when suppressing observations
sd.obs vector of standard deviations of individual posterior predictors for observed treatment conditions
sd.cf vector of standard deviations of individual posterior predictors for counterfactuals
commonSup.sub logical vector expressing which observations are used when estimating treatment effects
use.ranef logical for whether ranef models were used; only added when true
group.effects logical for whether group-level estimates are made; only added when true
seed a random seed for use when drawing from the posterior predictive distribution

Author(s)

Vincent Dorie: <vdorie@gmail.com>.
References


See Also

bart2

Examples

```r
## fit a simple linear model
n <- 100L
beta.z <- c(.75, -0.5, 0.25)
beta.y <- c(.5, 1.0, -1.5)
sigma <- 2

set.seed(725)
x <- matrix(rnorm(3 * n), n, 3)
tau <- rgamma(1L, 0.25 * 16 * rgamma(1L, 1 * 32, 32), 16)
p.score <- pnorm(x %*% beta.z)
z <- rbinom(n, 1, p.score)
mu.0 <- x %*% beta.y
mu.1 <- x %*% beta.y + tau
y <- mu.0 * (1 - z) + mu.1 * z + rnorm(n, 0, sigma)

# low parameters only for example
fit <- bartc(y, z, x, n.samples = 100L, n.burn = 15L, n.chains = 2L)
summary(fit)

## example to show refitting under the common support rule
fit2 <- refit(fit, commonSup.rule = "sd")
fit3 <- bartc(y, z, x, subset = fit2$commonSup.sub,
n.samples = 100L, n.burn = 15L, n.chains = 2L)
```
**bartc-generics**

**Generic Methods for bartcFit Objects**

**Description**

Visual exploratory data analysis and model fitting diagnostics for causal inference models fit using the `bartc` function.

**Usage**

```r
## S3 method for class 'bartcFit'
fitted(object, 
       type = c("pate", "sate", "cate", "mu.obs", "mu.cf", "mu.0", 
               "mu.1", "y.cf", "y.0", "y.1", "icate", "ite", 
               "p.score", "p.weights"), 
       sample = c("inferential", "all"),
       ...) 

extract(object, ...)

## S3 method for class 'bartcFit'
extract(object, 
         type = c("pate", "sate", "cate", "mu.obs", "mu.cf", "mu.0", 
                  "mu.1", "y.cf", "y.0", "y.1", "icate", "ite", 
                  "p.score", "p.weights", "sigma"), 
         sample = c("inferential", "all"), 
         combineChains = TRUE, 
         ...) 

## S3 method for class 'bartcFit'
predict(object, newdata, 
         group.by, 
         type = c("mu", "y", "mu.0", "mu.1", "y.0", "y.1", "icate", "ite", 
                  "p.score"), 
         combineChains = TRUE, 
         ...) 

toest(object, newresp, ...)

## S3 method for class 'bartcFit'
toest(object, 
       newresp = NULL, 
       commonSup.rule = c("none", "sd", "chisq"), 
       commonSup.cut = c(NA_real_, 1, 0.05), 
       ...) 
```
**Arguments**

- **object**: Object of class `bartcFit`.
- **type**: Which quantity to return. See details for a description of possible values.
- **sample**: Return information for either the "inferential" (e.g. treated observations when the estimand is att) or "all" observations.
- **combineChains**: If the models were fit with more than one chain, results retain the chain structure unless `combineChains` is TRUE.
- **newresp**: Not presently used, but provided for compatibility with other definitions of the `refit` generic.
- **newdata**: Data corresponding to the confounders in a `bartc` fit.
- **group.by**: Optional grouping variable. See definition of `group.by` in `bartc`.
- **commonSup.rule, commonSup.cut**: As in `bartc`.
- **...**: Additional parameters passed up the generic method chain.

**Details**

`fitted` returns the values that would serve as predictions for an object returned by the `bartc` function, while `extract` instead returns the full matrix or array of posterior samples. The possible options are:

- "pate", "sate", "cate" - various target quantities; see `summary`
- "mu" - predict only: expected value; requires user-supplied treatment variable in `newdata`
- "y" - predict only: sample of the response; requires user-supplied treatment variable in `newdata`
- "mu.obs" - (samples from the posterior of) the expected value under the observed treatment condition, i.e. \( \hat{\mu}_i(1) \ast z_i + \hat{\mu}_i(0) \ast (1 - z_i) \)
- "mu.cf" - the expected value under the counterfactual treatment condition, i.e. \( \hat{\mu}_i(1) \ast (1 - z_i) + \hat{\mu}_i(0) \ast z_i \)
- "mu.0" - the expected value under the control condition
- "mu.1" - the expected value under the treated condition
- "y.cf" - samples of the response under the the counterfactual treatment condition, i.e. \( \hat{y}_i(1 - z_i) \); values are obtained by adding noise to `mu.cf` using the posterior predictive distribution
- "y.0" - observed responses under the control together with predicted under the treated, i.e. \( \hat{y}_i(1) \ast z_i + y(0) \ast (1 - z_i) \)
- "y.1" - observed responses under the treatment together with predicted under the control, i.e. \( y_i(1) \ast z_i + \hat{y}_i(0) \ast (1 - z_i) \)
- "ite" - (sample) individual treatment effect estimates, i.e. \( (y_i(z_i) - y_i(1 - z_i)) \ast (2z_i - 1) \); uses observed responses and posterior predicted counterfactuals
- "icate" - individual conditional average treatment effect estimates, i.e. \( \hat{\mu}_i(1) - \hat{\mu}_i(0) \)
- "p.score" - probability that each observation is assigned to the treatment group
- "p.weights" - weights assigned to each individual difference if the response method is "p.weight"
- "sigma" - residual standard deviation from continuous response models

`refit` exists to allow the same regressions to be used to calculate estimates under different common support rules. To refit those models on a subset, see the examples in `bartc`.

`predict` allows the fitted model to be used to make predictions on an out-of-sample set. Requires model to be fit with `keepTrees` equal to TRUE. As ‘y’ values are all considered out of sample, the posterior predictive distribution is always used when relevant.

**Value**

For `fitted`, `extract`, and `predict`, a matrix, array, or vector depending on the dimensions of the result and the number of chains. For the following, when `n.chains` is one the dimension is dropped.

- "pate", "sate", or "cate" - with `fitted`, a scalar; with `extract`, `n.chains x n.samples`
- "p.score" - depending on the fitting method, samples may or not be present; when samples are absent, a vector is returned for both functions; when present, the same as "y".
- all other types - with `fitted`, a vector of length equal to the number of observations (`n.obs`); with `extract` or `predict`, a matrix or array of dimensions `n.chains x n.samples x n.obs`.

For `refit`, an object of class `bartcFit`.

**Author(s)**

Vincent Dorie: <vdorie@gmail.com>.

**See Also**

`bartc`

**Examples**

```r
## fit a simple linear model
n <- 100L
beta.z <- c(.75, -0.5, 0.25)
beta.y <- c(.5, 1.0, -1.5)
sigma <- 2
set.seed(725)
x <- matrix(rnorm(3 * n), n, 3)
tau <- rgamma(1L, 0.25 * 16 * rgamma(1L, 1 * 32, 32), 16)
p.score <- pnorm(x %*% beta.z)
z <- rbinom(n, 1, p.score)
mu.0 <- x %*% beta.y
mu.1 <- x %*% beta.y + tau
y <- mu.0 * (1 - z) + mu.1 * z + rnorm(n, 0, sigma)

# low parameters only for example
fit <- bartc(y, z, x, n.samples = 100L, n.burn = 15L, n.chains = 2L)
```
# compare fit to linear model
lm.fit <- lm(y ~ z + x)

plot(fitted(fit, type = "mu.obs"), fitted(lm.fit))

# rank order sample individual treatment effect estimates and plot
ites <- extract(fit, type = "ite")
ite.m <- apply(ites, 2, mean)
ite.sd <- apply(ites, 2, sd)
ite.lb <- ite.m - 2 * ite.sd
ite.ub <- ite.m + 2 * ite.sd
ite.o <- order(ite.m)

plot(NULL, type = "n",
    xlim = c(1, length(ite.m)), ylim = range(ite.lb, ite.ub),
    xlab = "effect order", ylab = "individual treatment effect")
lines(rbind(seq_along(ite.m), seq_along(ite.m), NA),
    rbind(ite.lb[ite.o], ite.ub[ite.o], NA), lwd = 0.5)
points(seq_along(ite.m), ite.m[ite.o], pch = 20)

---

**bartc-plot**

*Plot methods for bartc*

**Description**

Visual exploratory data analysis and model fitting diagnostics for causal inference models fit using the **bartc** function.

**Usage**

```r
plot_sigma(x, main = "Traceplot sigma",
           xlab = "iteration", ylab = "sigma",
           lty = 1:x$n.chains,
           ...)

plot_est(x, main = paste("Traceplot", x$estimand),
         xlab = "iteration", ylab = x$estimand,
         lty = 1:x$n.chains, col = NULL,
         ...)

plot_indiv(x, main = "Histogram Individual Quantities",
           type = c("icate", "mu.obs", "mu.cf", "mu.0",
                      "mu.1", "y.cf", "y.0", "y.1", "ite"),
           xlab = "treatment effect",
           breaks = 20,
           ...)
```
plot_support(x, main = "Common Support Scatterplot",
    xvar = "tree.1", yvar = "tree.2",
    sample = c("inferential", "all"),
    xlab = NULL, ylab = NULL,
    pch.trt = 21, bg.trt = "black",
    pch.ctl = pch.trt, bg.ctl = NA,
    pch.sup = pch.trt, bg.sup = NA, col.sup = "red", cex.sup = 1.5,
    legend.x = "topleft", legend.y = NULL,
    ...)  

Arguments

x Object of class bartcFit.
main Character title of plot.
xlab Character label of $x$ axis. For plot_support, if NULL a default will be used.
ylab Character label of $y$ axis. For plot_support, if NULL a default will be used.
lty For line plots (plot_sigma, plot_est), models use the values of lty to visually distinguish each chain.
col For line plots, use col vector to distinguish between groups (if any).
breaks Argument to codehist.
type The individual quantity to be plotted. See fitted.
xvar Variable for use on $x$ axis. Can be one of "tree.XX", "pca.XX", "css", any individual level quantity accepted by fitted, the number or name of a column used to fit the response model, or a given vector. See below for details.
sample Return information for either the "inferential" (e.g. treated observations when the estimand is att) or "all" observations.
yvar Variable for use on the $y$ axis, of the same form as xvar.
pch.trt pch point value used when plotting treatment observations.
bg.trt bg background value used when plotting treatment observations.
pch.ctl pch point value used when plotting control observations.
bg.ctl bg background value used when plotting treatment observations.
pch.sup pch point value used when plotting suppressed observations.
bg.sup bg background value used when plotting suppressed observations.
col.sup col color value used when plotting suppressed observations.
cex.sup cex size value used when plotting suppressed observations.
legend.x x value passed to legend. If NULL, legend plotting is skipped.
legend.y Optional y value passed to legend
... Optional graphical parameters.
Details

Produces various plots using objects fit by `bartc`. `plot_sigma` and `plot_est` are traditional parameter trace plots that can be used to diagnose the convergence of the posterior sampler. If the `bartc` model is fit with `n.chains` greater than one, by default each chain will be plotted with its own line type.

`plot_indiv` produces a simple histogram of the distribution of the estimates of the individual effects, taken as the average of their posterior samples.

`plot_support` is used to visualize the common support diagnostic in the form of a scatterplot. Points that the diagnostic excludes are outlined in red. The contents of the x and y axes are controlled by the `xvar` and `yvar` arguments respectively and can be of the form:

- "tree.XX" - Variable number "XX" as selected by variable importance in predicting individual treatment effects using a tree fit by `rpart`. "XX" must be a number not exceeding the number of continuous variables used to fit the response model.
- "pca.XX" - "XX"th principal component axis.
- "css" - The common support statistic.
- "y" - Observed response variable.
- "y0" - Predicted values for the response under the control as obtained by `fitted`.
- "y1" - Predicted values for the response under the treatment `fitted`.
- "indiv.diff" - Individual treatment effect estimates, or \( \hat{y}(1) - \hat{y}(0) \).
- "p.score" - Propensity score used to fit the response model (if applicable).
- "p.weights" - Weights used when taking average of individual differences for response method "p.weight".
- predictor columns - Given by name or number.
- supplied vector - Any vector of length equal to the number of observations.

Value

None, although plotting occurs as a side-effect.

Author(s)

Vincent Dorie: <vdorie@gmail.com>.

See Also

`bartc`

Examples

```r
# generate fake data using a simple linear model
n <- 100L
beta.z <- c(0.75, -0.5, 0.25)
beta.y <- c(0.5, 1.0, -1.5)
sigma <- 2
```
```r
set.seed(725)
x <- matrix(rnorm(3 * n), n, 3)
tau <- rgamma(1L, 0.25 * 16 * rgamma(1L, 1 * 32, 32), 16)
p.score <- pnorm(x %*% beta.z)
z <- rbinom(n, 1, p.score)
mu.0 <- x %*% beta.y
mu.1 <- x %*% beta.y + tau
y <- mu.0 * (1 - z) + mu.1 * z + rnorm(n, 0, sigma)

# run with low parameters only for example
fit <- bartc(y, z, x, n.samples = 100L, n.burn = 15L, n.chains = 2L,
n.threads = 1L,
commonSup.rule = "sd")

## plot specific functions

## sigma plot can be used to access convergence of chains
plot_sigma(fit)

## show points lacking common support
plot_support(fit, xvar = "tree.1", yvar = "css", legend.x = NULL)

# see example in ?"bartc-generics" for rank-ordered individual effects plot
```

---

**summary.bartcFit**

**Summary for bartcFit Objects**

**Description**

Summarizes bartc fits by calculating target quantities and uncertainty estimates.

**Usage**

```r
# S3 method for class 'bartcFit'
summary(object,
    target = c("pate", "sate", "cate"),
    ci.style = c("norm", "quant", "hpd"),
    ci.level = 0.95,
    pate.style = c("ppd", "var.exp"),
    ...)
```

**Arguments**

- `object` Object of class bartcFit.
- `target` Treatment effect to calculate. One of "pate" - population average treatment effect, "sate" - sample average treatment effect, and "cate" - conditional average treatment effect.
ci.style: Means of calculating confidence intervals (posterior credible regions). Options include "norm" - use a normal approximation, "quart" - the empirical quantities of the posterior samples, and "hpd" - region of highest posterior density.

ci.level: Level of confidence for intervals.

pate.style: For target "pate", calculate uncertainty by using "ppd" - the posterior predictive distribution or "var.exp" a variance expansion formula.

...: Not used at moment, but present to match summary generic signature.

Details

summary produces a numeric and qualitative summary of a bartc fit.

Target Types

The SATE and PATE involve calculating predicted response values under different treatment conditions. When using extract or fitted, these values are simulated directly from the posterior predictive distribution. However, since these quantities all have the same expected value, in order to provide consistent results summary only uses those simulations to derive credible intervals. Thus the estimates for CATE, SATE, and PATE all should be reported as the same but with increasing degrees of uncertainty.

Grouped Data

If a model is fit with a supplied grouping variable and group.effects = TRUE, the estimates will also be reported within groups. When possible, the last line corresponds to the population. Within group estimates for response methods such as "tmle" cannot easily be extrapolated to the population at large - the means will combine based on the sample sizes but the uncertainty estimates will lack correlations.

Value

An object of class bartcFit.summary equivalent to a list with named items:

call: how bartc was called
method.rsp: character string specifying the method used to fit the response surface
method.trt: character string specifying the method used to fit the treatment assignment mechanism

ci.info: a named list with items target, ci.style, ci.level, and pate.style as passed to summary

n.obs: total number of observations
n.samples: number of samples within any one chain
n.chains: total number of chains

commonSup.rule: common support rule used when fitting object to produce estimates

estimates: a data.frame with columns "estimate" - the target, "sd" - standard deviation of posterior of estimate, "ci.lower" - lower bound of credible region, "ci.upper" - upper bound of credible region, and optionally "n.cut" - how many observations were dropped using common support rule
Author(s)

Vincent Dorie: <vdorie@gmail.com>.
Index

* **bart**
  - bartc, 2
  - bartc-generics, 8
  - bartc-plot, 11
* **causal**
  - bartc, 2
  - bartc-generics, 8
  - bartc-plot, 11
* **plot**
  - bartc-plot, 11
* **tmle**
  - bartc, 2
  - bartc-generics, 8

bart2, 4, 5, 7
bartc, 2, 8–11, 13, 15
bartc-generics, 8
bartc-plot, 11

dbartsData, 6

extract, 5, 15
extract (bartc-generics), 8

fitted, 5, 12, 13
fitted.bartcFit (bartc-generics), 8

generics, 5
glm, 4

hist, 12

legend, 12
lm, 4, 5
lme4, 3
lmer, 5

plot_est, 5
plot_est (bartc-plot), 11
plot_indiv, 5
plot_indiv (bartc-plot), 11
plot_sigma, 5
plot_sigma (bartc-plot), 11
plot_support, 5
plot_support (bartc-plot), 11
predict, 5
predict.bartcFit (bartc-generics), 8

rbart, 3
refit (bartc-generics), 8
rpart, 13

seed, 4
set.seed, 4
summary, 3, 5, 9
summary.bartcFit, 14