Package 'ShrinkageTrees'

September 2, 2026

Type Package

Title Bayesian Tree Ensembles for Survival Analysis and Causal Inference

Version 2.1.0

Date 2026-09-02

Maintainer Tijn Jacobs <t.jacobs@vu.nl>

Description Bayesian regression tree ensembles for survival analysis and causal inference. Implements BART, DART, Bayesian Causal Forests (BCF), and Horseshoe Forest models. Supports right-censored and interval-censored survival outcomes via accelerated failure time (AFT) formulations. Designed for high-dimensional prediction and heterogeneous treatment effect estimation.

URL https://github.com/tijn-jacobs/ShrinkageTrees

BugReports https://github.com/tijn-jacobs/ShrinkageTrees/issues

License MIT + file LICENSE

Depends R (>= 4.1.0)

Imports parallel, Rcpp, splines, stats, utils

LinkingTo Rcpp (>= 1.0.11)

Suggests coda, ggplot2, glmnet, knitr, rmarkdown, rpart, survival, testthat (>= 3.0.0)

VignetteBuilder knitr

RoxygenNote 7.3.3

Encoding UTF-8

LazyData true

LazyDataCompression xz

Config/testthat/edition 3

NeedsCompilation yes

Author Tijn Jacobs [aut, cre] (ORCID: <https://orcid.org/0009-0003-6188-9296>)

Repository CRAN

Date/Publication 2026-09-02 14:00:02 UTC

Contents

`as.mcmc.list.ShrinkageTrees`

Convert MCMC output to a coda mcmc.list

Description

Converts the posterior draws stored in a `ShrinkageTrees` object into a `mcmc.list` for use with the **coda** package's convergence diagnostics (Gelman–Rubin $\hat{R}$, effective sample size, Geweke test, etc.).

Usage

```
## S3 method for class 'ShrinkageTrees'
as.mcmc.list(x, ...)
```

Arguments

x	A fitted ShrinkageTrees object.
...	Currently unused.

Details

Requires the suggested package **coda**. For single-chain fits the returned object contains one chain.

Value

A mcmc.list object. Each chain is an mcmc object whose columns include:

sigma Posterior draws of the residual standard deviation (continuous and survival outcomes only).

See Also

summary.ShrinkageTrees which reports R-hat and ESS automatically when coda is available.

Examples

```
fit <- HorseTrees(y = rnorm(50), X_train = matrix(rnorm(250), 50, 5),
                  N_post = 200, N_burn = 100, n_chains = 2)
if (requireNamespace("coda", quietly = TRUE)) {
  mcmc_obj <- coda::as.mcmc.list(fit)
  coda::gelman.diag(mcmc_obj)
  coda::effectiveSize(mcmc_obj)
}
```

---

bayesian_bootstrap_ate

Bayesian bootstrap average treatment effect

Description

Post-hoc reweights the stored posterior CATE draws of a fitted causal model to produce credible intervals for the *population* ATE (PATE) that incorporate uncertainty in the covariate distribution F_X.

Usage

```
bayesian_bootstrap_ate(object, alpha = 0.05)
```

Arguments

`object`	Either a fitted `CausalShrinkageForest` (from `CausalShrinkageForest` or `CausalHorseForest` with `store_posterior_sample = TRUE`) or a `CausalShrinkageForestPrediction` (from `predict.CausalShrinkageForest`).
`alpha`	One minus the credible level. Default `0.05` (a 95 percent credible interval).

Details

At each MCMC iteration s the conditional treatment effects $\tau^{(s)}(x_i)$ are reweighted with $(w_1^{(s)}, \ldots, w_n^{(s)}) \sim \mathrm{Dir}(1, \ldots, 1)$ to give a draw

$$\widehat{\mathrm{PATE}}^{(s)} = \sum_{i=1}^{n} w_i^{(s)} \, \tau^{(s)}(x_i).$$

The collection $\{\widehat{\mathrm{PATE}}^{(s)}\}$ approximates the posterior of the PATE, integrating over $\tau(\cdot)$ *and* F_X.

The equal-weight mixed ATE (MATE), $\widehat{\mathrm{MATE}}^{(s)} = n^{-1} \sum_i \tau^{(s)}(x_i)$, is returned alongside for comparison.

For reproducibility, call `set.seed()` before invoking the function to fix the Dirichlet draws.

Value

A list with

pate_mean, pate_ci, pate_samples Posterior mean, credible interval (named `lower` and `upper`), and full vector of draws of the Bayesian-bootstrap PATE.

mate_mean, mate_ci, mate_samples Same quantities for the equal-weight mixed ATE.

n, S Number of observations and posterior draws used.

See Also

`summary.CausalShrinkageForest`, `plot.CausalShrinkageForest`, `predict.CausalShrinkageForest`

Examples

```
# Small toy causal model (binary outcome, for speed)
set.seed(1)
n <- 40; p <- 3
X <- matrix(runif(n * p), ncol = p)
trt <- rbinom(n, 1, 0.5)
y <- X[, 1] + trt * (0.5 + X[, 2]) + rnorm(n)

fit <- CausalShrinkageForest(
  y = y,
  X_train_control = X, X_train_treat = X,
  treatment_indicator_train = trt,
  outcome_type = "continuous",
  number_of_trees_control = 5, number_of_trees_treat = 5,
  prior_type_control = "horseshoe", prior_type_treat = "horseshoe",
  local_hp_control = 0.1, global_hp_control = 0.1,
```

```
  local_hp_treat = 0.1, global_hp_treat = 0.1,
  N_post = 20, N_burn = 10,
  store_posterior_sample = TRUE,
  verbose = FALSE
)

bb <- bayesian_bootstrap_ate(fit, alpha = 0.05)
bb$pate_mean
bb$pate_ci
```

`CausalHorseForest` *Causal Horseshoe Forests*

Description

This function fits a (Bayesian) Causal Horseshoe Forest. It can be used for estimation of conditional average treatments effects of survival data given high-dimensional covariates. The outcome is decomposed in a prognostic part (control) and a treatment effect part. For both of these, we specify a Horseshoe Trees regression function. Supports continuous, right-censored, and interval-censored outcomes.

Usage

```
CausalHorseForest(
  y = NULL,
  status = NULL,
  X_train_control,
  X_train_treat,
  treatment_indicator_train,
  X_test_control = NULL,
  X_test_treat = NULL,
  treatment_indicator_test = NULL,
  left_time = NULL,
  right_time = NULL,
  outcome_type = "continuous",
  timescale = "time",
  number_of_trees = 200,
  k = 1.5,
  power = 2,
  base = 0.95,
  p_grow = 0.4,
  p_prune = 0.4,
  nu = 3,
  q = 0.9,
  sigma = NULL,
  N_post = 5000,
  N_burn = 5000,
```

```
  delayed_proposal = 5,
  store_posterior_sample = FALSE,
  treatment_coding = "centered",
  propensity = NULL,
  propensity_test = NULL,
  n_chains = 1,
  verbose = TRUE
)
```

Arguments

y	Outcome vector. For survival, represents follow-up times (can be on original or log scale depending on timescale). Set to NULL when using outcome_type = "interval-censored", as values are derived from left_time and right_time.
status	Optional event indicator vector (1 = event occurred, 0 = censored). Required when outcome_type = "right-censored". For interval-censored outcomes, this is derived automatically from left_time and right_time.
X_train_control	Covariate matrix for the control forest. Rows correspond to samples, columns to covariates.
X_train_treat	Covariate matrix for the treatment forest. Rows correspond to samples, columns to covariates.
treatment_indicator_train	Vector indicating treatment assignment for training samples (1 = treated, 0 = control).
X_test_control	Optional test covariate matrix for control forest. If NULL, defaults to column means of X_train_control.
X_test_treat	Optional test covariate matrix for treatment forest. If NULL, defaults to column means of X_train_treat.
treatment_indicator_test	Optional vector indicating treatment assignment for test samples.
left_time	Optional numeric vector of left (lower) time boundaries. Required when outcome_type = "interval-censored". Exact events have left_time == right_time; right-censored observations have right_time = Inf; interval-censored observations have finite left_time < right_time.
right_time	Optional numeric vector of right (upper) time boundaries. Required when outcome_type = "interval-censored". Use Inf for right-censored observations.
outcome_type	Type of outcome: one of "continuous", "right-censored", or "interval-censored". Default is "continuous".
timescale	For survival outcomes: either "time" (original time scale, log-transformed internally) or "log" (already log-transformed). Used when outcome_type is "right-censored" or "interval-censored".
number_of_trees	Number of trees in each forest. Default is 200.

k — Horseshoe prior scale hyperparameter. Default is 1.5. Controls global-local shrinkage on the step heights of both forests, parameterized as $\alpha_{\text{local}} = \alpha_{\text{global}} = k/\sqrt{\text{number\_of\_trees}}$.

Choosing k. Values between 1 and 2 worked well across a range of simulated settings, and the default of 1.5 sits in the middle. Within that range:

- **k near 1 shrinks more aggressively** – tighter intervals on $\tau(x)$, and treatment-effect heterogeneity pulled harder towards a constant. Preferable when the effect is close to homogeneous, when censoring is heavy, or when the sample is small.
- **k near 2 is more conservative** – wider intervals and more freedom for $\tau(x)$ to vary. Preferable when the effect is genuinely heterogeneous and the sample supports estimating it.

If you would rather select it from the data, k is a natural target for cross-validation; see the "Choosing k" section of the package vignette.

The useful range is higher here than for the single-forest models (HorseTrees, ShrinkageTrees, roughly 0.5 to 1.5) because the treatment forest enters as $b\,\tau(x)$ with $b = \pm 1/2$, halving its contribution to the response. Measured on the scale of that contribution the two recommendations nearly coincide.

Changed in version 2.1.0. The default was 0.1 in earlier releases. See NEWS.md.

power — Power parameter for tree structure prior. Default is 2.0.

base — Base parameter for tree structure prior. Default is 0.95.

p_grow — Probability of proposing a grow move. Default is 0.4.

p_prune — Probability of proposing a prune move. Default is 0.4.

nu — Degrees of freedom for the error variance prior. Default is 3.

q — Quantile parameter for error variance prior. Default is 0.90.

sigma — Optional known standard deviation of the outcome. If NULL, estimated from data.

N_post — Number of posterior samples to store. Default is 5000.

N_burn — Number of burn-in iterations. Default is 5000.

delayed_proposal — Number of delayed iterations before proposal updates. Default is 5.

store_posterior_sample — Logical; whether to store posterior samples of predictions. Default is FALSE.

treatment_coding — Treatment coding scheme for the two-forest model. One of "centered" (default), "binary", "adaptive", or "invariant". "centered" uses $b_i \in \{-1/2, 1/2\}$; "binary" uses $b_i \in \{0, 1\}$; "adaptive" uses $b_i = A_i - \hat{e}(x_i)$ where $\hat{e}(x_i)$ is the estimated propensity score; "invariant" treats b_0, b_1 as parameters estimated within the Gibbs sampler with $b_j \sim N(0, 1/2)$ priors, yielding a parameterisation-invariant model (Hahn et al., 2020, Section 5.2).

propensity — Optional numeric vector of propensity scores $\hat{e}(x_i)$ for training observations. Required when treatment_coding = "adaptive".

propensity_test
: Optional numeric vector of propensity scores for test observations. Only used when treatment_coding = "adaptive". Defaults to 0.5 for all test observations if not provided.

n_chains
: Number of independent MCMC chains to run. Default is 1 (standard single-chain behaviour). When n_chains > 1 the chains are run in parallel via parallel::mclapply and their posterior samples are pooled into a single CausalShrinkageForest object, so all existing print and summary methods work without modification. On Windows, mclapply falls back to sequential execution.

verbose
: Logical; whether to print verbose output during sampling. Default is TRUE.

Details

The model separately regularizes the control and treatment trees using Horseshoe priors with global-local shrinkage on the step heights. This approach is designed for robust estimation of heterogeneous treatment effects in high-dimensional settings. It supports continuous, right-censored, and interval-censored survival outcomes. For interval-censored data, provide left_time and right_time instead of y and status; the event indicators are derived internally following the survival::Surv(type = "interval2") convention.

Value

An S3 object of class "CausalShrinkageForest" containing:

train_predictions Posterior mean predictions on training data (combined forest).

test_predictions Posterior mean predictions on test data (combined forest).

train_predictions_control Estimated control outcomes on training data.

test_predictions_control Estimated control outcomes on test data.

train_predictions_treat Estimated treatment effects on training data.

test_predictions_treat Estimated treatment effects on test data.

sigma Vector of posterior samples for the error standard deviation.

acceptance_ratio_control Average acceptance ratio in control forest.

acceptance_ratio_treat Average acceptance ratio in treatment forest.

train_predictions_sample_control Matrix of posterior samples for control predictions (if store_posterior_sample = TRUE).

test_predictions_sample_control Matrix of posterior samples for control predictions (if store_posterior_sample = TRUE).

train_predictions_sample_treat Matrix of posterior samples for treatment effects (if store_posterior_sample = TRUE).

test_predictions_sample_treat Matrix of posterior samples for treatment effects (if store_posterior_sample = TRUE).

See Also

Model family: HorseTrees (non-causal, horseshoe prior), ShrinkageTrees (non-causal, flexible prior), CausalShrinkageForest (causal, flexible prior).

Survival wrappers: SurvivalBCF, SurvivalShrinkageBCF.

S3 methods: print.CausalShrinkageForest, summary.CausalShrinkageForest, predict.CausalShrinkageForest, plot.CausalShrinkageForest.

Examples

```
# Example: Continuous outcome and homogeneous treatment effect
n <- 50
p <- 3
X_control <- matrix(runif(n * p), ncol = p)
X_treat <- matrix(runif(n * p), ncol = p)
treatment <- rbinom(n, 1, 0.5)
tau <- 2
y <- X_control[, 1] + (0.5 - treatment) * tau + rnorm(n)

fit <- CausalHorseForest(
  y = y,
  X_train_control = X_control,
  X_train_treat = X_treat,
  treatment_indicator_train = treatment,
  outcome_type = "continuous",
  number_of_trees = 5,
  N_post = 10,
  N_burn = 5,
  store_posterior_sample = TRUE,
  verbose = FALSE
)


## Example: Right-censored survival outcome
# Set data dimensions
n <- 100
p <- 1000

# Generate covariates
X <- matrix(runif(n * p), ncol = p)
X_treat <- X
treatment <- rbinom(n, 1, pnorm(X[, 1] - 1/2))

# Generate true survival times depending on X and treatment
linpred <- X[, 1] - X[, 2] + (treatment - 0.5) * (1 + X[, 2] / 2 + X[, 3] / 3
                                                  + X[, 4] / 4)
true_time <- linpred + rnorm(n, 0, 0.5)

# Generate censoring times
censor_time <- log(rexp(n, rate = 1 / 5))

# Observed times and event indicator
```

```
time_obs <- pmin(true_time, censor_time)
status <- as.numeric(true_time == time_obs)

# Estimate propensity score using HorseTrees
fit_prop <- HorseTrees(
  y = treatment,
  X_train = X,
  outcome_type = "binary",
  number_of_trees = 200,
  N_post = 1000,
  N_burn = 1000
)

# Retrieve estimated probability of treatment (propensity score)
propensity <- fit_prop$train_probabilities

# Combine propensity score with covariates for control forest
X_control <- cbind(propensity, X)

# Fit the Causal Horseshoe Forest for survival outcome
fit_surv <- CausalHorseForest(
  y = time_obs,
  status = status,
  X_train_control = X_control,
  X_train_treat = X_treat,
  treatment_indicator_train = treatment,
  outcome_type = "right-censored",
  timescale = "log",
  number_of_trees = 200,
  k = 1.5,
  N_post = 1000,
  N_burn = 1000,
  store_posterior_sample = TRUE
)

## Evaluate and summarize results

# Evaluate C-index if survival package is available
if (requireNamespace("survival", quietly = TRUE)) {
  predicted_survtime <- fit_surv$train_predictions
 cindex_result <- survival::concordance(survival::Surv(time_obs, status) ~ predicted_survtime)
  c_index <- cindex_result$concordance
  cat("C-index:", round(c_index, 3), "\n")
} else {
  cat("Package 'survival' not available. Skipping C-index computation.\n")
}

# Compute posterior ATE samples
ate_samples <- rowMeans(fit_surv$train_predictions_sample_treat)
mean_ate <- mean(ate_samples)
ci_95 <- quantile(ate_samples, probs = c(0.025, 0.975))

cat("Posterior mean ATE:", round(mean_ate, 3), "\n")
```

```
cat("95% credible interval: [", round(ci_95[1], 3), ", ", round(ci_95[2], 3), "]\n", sep = "")

# Plot histogram of ATE samples
hist(
  ate_samples,
  breaks = 30,
  col = "steelblue",
  freq = FALSE,
  border = "white",
  xlab = "Average Treatment Effect (ATE)",
  main = "Posterior distribution of ATE"
)
abline(v = mean_ate, col = "orange3", lwd = 2)
abline(v = ci_95, col = "orange3", lty = 2, lwd = 2)
abline(v = 1.541667, col = "darkred", lwd = 2)
legend(
  "topright",
  legend = c("Mean", "95% CI", "Truth"),
  col = c("orange3", "orange3", "red"),
  lty = c(1, 2, 1),
  lwd = 2
)

## Plot individual CATE estimates

# Summarize posterior distribution per patient
posterior_matrix <- fit_surv$train_predictions_sample_treat
posterior_mean <- colMeans(posterior_matrix)
posterior_ci <- apply(posterior_matrix, 2, quantile, probs = c(0.025, 0.975))

df_cate <- data.frame(
  mean = posterior_mean,
  lower = posterior_ci[1, ],
  upper = posterior_ci[2, ]
)

# Sort patients by posterior mean CATE
df_cate_sorted <- df_cate[order(df_cate$mean), ]
n_patients <- nrow(df_cate_sorted)

# Create the plot
plot(
  x = df_cate_sorted$mean,
  y = 1:n_patients,
  type = "n",
  xlab = "CATE per patient (95% credible interval)",
  ylab = "Patient index (sorted)",
  main = "Posterior CATE estimates",
  xlim = range(df_cate_sorted$lower, df_cate_sorted$upper)
)

# Add CATE intervals
segments(
```

```
  x0 = df_cate_sorted$lower,
  x1 = df_cate_sorted$upper,
  y0 = 1:n_patients,
  y1 = 1:n_patients,
  col = "steelblue"
)

# Add mean points
points(df_cate_sorted$mean, 1:n_patients, pch = 16, col = "orange3", lwd = 0.1)

# Add reference line at 0
abline(v = 0, col = "black", lwd = 2)
```

`CausalShrinkageForest` *General Causal Shrinkage Forests*

Description

Fits a (Bayesian) Causal Shrinkage Forest model for estimating heterogeneous treatment effects. This function generalizes `CausalHorseForest` by allowing flexible global-local shrinkage priors on the step heights in both the control and treatment forests. It supports continuous, right-censored, and interval-censored survival outcomes.

Usage

```
CausalShrinkageForest(
  y = NULL,
  status = NULL,
  X_train_control,
  X_train_treat,
  treatment_indicator_train,
  X_test_control = NULL,
  X_test_treat = NULL,
  treatment_indicator_test = NULL,
  left_time = NULL,
  right_time = NULL,
  outcome_type = "continuous",
  timescale = "time",
  number_of_trees_control = 200,
  number_of_trees_treat = 200,
  prior_type_control = "horseshoe",
  prior_type_treat = "horseshoe",
  local_hp_control = NULL,
  local_hp_treat = NULL,
  global_hp_control = NULL,
```

```
  global_hp_treat = NULL,
  a_dirichlet_control = 0.5,
  a_dirichlet_treat = 0.5,
  b_dirichlet_control = 1,
  b_dirichlet_treat = 1,
  rho_dirichlet_control = NULL,
  rho_dirichlet_treat = NULL,
  power_control = 2,
  power_treat = 2,
  base_control = 0.95,
  base_treat = 0.95,
  p_grow = 0.5,
  p_prune = 0.5,
  nu = 3,
  q = 0.9,
  sigma = NULL,
  N_post = 5000,
  N_burn = 5000,
  delayed_proposal = 5,
  store_posterior_sample = FALSE,
  treatment_coding = "centered",
  propensity = NULL,
  propensity_test = NULL,
  n_chains = 1,
  verbose = TRUE
)
```

Arguments

`y`	Outcome vector. Numeric. Represents continuous outcomes or follow-up times. Set to `NULL` when using `outcome_type = "interval-censored"`, as values are derived from `left_time` and `right_time`.
`status`	Optional event indicator vector (1 = event occurred, 0 = censored). Required when `outcome_type = "right-censored"`. For interval-censored outcomes, this is derived automatically from `left_time` and `right_time`.
`X_train_control`	Covariate matrix for the control forest. Rows correspond to samples, columns to covariates.
`X_train_treat`	Covariate matrix for the treatment forest.
`treatment_indicator_train`	Vector indicating treatment assignment for training samples (1 = treated, 0 = control).
`X_test_control`	Optional covariate matrix for control forest test data. Defaults to column means of `X_train_control` if NULL.
`X_test_treat`	Optional covariate matrix for treatment forest test data. Defaults to column means of `X_train_treat` if NULL.
`treatment_indicator_test`	Optional vector indicating treatment assignment for test data.

`left_time` Optional numeric vector of left (lower) time boundaries. Required when `outcome_type = "interval-censored"`. Exact events have `left_time == right_time`; right-censored observations have `right_time = Inf`; interval-censored observations have finite `left_time < right_time`.

`right_time` Optional numeric vector of right (upper) time boundaries. Required when `outcome_type = "interval-censored"`. Use `Inf` for right-censored observations.

`outcome_type` Type of outcome: one of `"continuous"`, `"right-censored"`, or `"interval-censored"`. Default is `"continuous"`.

`timescale` For survival outcomes: either `"time"` (original scale, log-transformed internally) or `"log"` (already log-transformed). Default is `"time"`. Used when `outcome_type` is `"right-censored"` or `"interval-censored"`.

`number_of_trees_control`
Number of trees in the control forest. Default is 200.

`number_of_trees_treat`
Number of trees in the treatment forest. Default is 200.

`prior_type_control`
Type of prior on control forest step heights. One of `"horseshoe"` (global-local, global scale per tree), `"horseshoe_fw"` (global scale shared forest-wide), `"half-cauchy"` (local shrinkage only), `"standard"` (the classical BART normal prior), `"dirichlet"` (standard prior with a Dirichlet splitting rule), `"standard-halfcauchy"`, or `"dirichlet-halfcauchy"`. Default is `"horseshoe"`.

`prior_type_treat`
Type of prior on treatment forest step heights. Same options as `prior_type_control`.

`local_hp_control`
Local hyperparameter controlling shrinkage on individual step heights (control forest).

For `prior_type_control = "horseshoe"` this may be left `NULL`, in which case it defaults to `k / sqrt(number_of_trees_control)` with `k = 1.5`. Values between 1 and 2 worked well across a range of simulated settings: `k` near 1 shrinks more aggressively, `k` near 2 is more conservative. `k` is also a natural target for cross-validation; see the "Choosing k" section of the package vignette.

`local_hp_treat` Local hyperparameter for the treatment forest. Same default, using `number_of_trees_treat`.

`global_hp_control`
Global hyperparameter for the control forest. Ignored for `"half-cauchy"`. Same default as `local_hp_control`; only their product identifies the prior, so setting the two equal loses nothing.

`global_hp_treat`
Global hyperparameter for the treatment forest. Same default as `local_hp_treat`.

Changed in version 2.1.0. These previously had to be supplied for horseshoe priors, and the value used throughout the documentation was `0.1 / sqrt(number_of_trees)`. That value was calibrated against a defect which made the global hyperparameters inert (see `NEWS.md`); with the defect corrected it shrinks far too aggressively.

`a_dirichlet_control`
First shape parameter of the Beta prior used in the Dirichlet–Sparse splitting rule for the control forest. Together with `b_dirichlet_control`, it controls the expected sparsity level.

`a_dirichlet_treat`
: First shape parameter of the Beta prior used in the Dirichlet–Sparse splitting rule for the treatment forest.

`b_dirichlet_control`
: Second shape parameter of the Beta prior for the sparsity level in the control forest. Larger values shrink splitting probabilities more strongly toward uniform sparsity.

`b_dirichlet_treat`
: Second shape parameter of the Beta prior governing sparsity in the treatment forest.

`rho_dirichlet_control`
: Sparsity hyperparameter for the control forest. Represents the *expected number of active predictors*. If left NULL, it defaults to the number of covariates in the control forest.

`rho_dirichlet_treat`
: Sparsity hyperparameter for the treatment forest, interpreted as the expected number of active predictors. Defaults to the number of covariates in the treatment forest if not specified.

`power_control` Power parameter for the control forest tree structure prior splitting probability.

`power_treat` Power parameter for the treatment forest tree structure prior splitting probability.

`base_control` Base parameter for the control forest tree structure prior splitting probability.

`base_treat` Base parameter for the treatment forest tree structure prior splitting probability.

`p_grow` Probability of proposing a grow move. Default is 0.5. These are fixed at 0.5 for prior_type `"standard"` and `"dirichlet"`.

`p_prune` Probability of proposing a prune move. Default is 0.5. These are fixed at 0.5 for prior_type `"standard"` and `"dirichlet"`.

`nu` Degrees of freedom for the error variance prior. Default is 3.

`q` Quantile parameter for error variance prior. Default is 0.90.

`sigma` Optional known standard deviation of the outcome. If NULL, estimated from data.

`N_post` Number of posterior samples to store. Default is 5000.

`N_burn` Number of burn-in iterations. Default is 5000.

`delayed_proposal`
: Number of delayed iterations before proposal updates. Default is 5.

`store_posterior_sample`
: Logical; whether to store posterior samples of predictions. Default is `FALSE`.

`treatment_coding`
: Treatment coding scheme for the two-forest model. One of `"centered"` (default), `"binary"`, `"adaptive"`, or `"invariant"`. `"centered"` uses $b_i \in \{-1/2, 1/2\}$; `"binary"` uses $b_i \in \{0, 1\}$; `"adaptive"` uses $b_i = A_i - \hat{e}(x_i)$ where $\hat{e}(x_i)$ is the estimated propensity score; `"invariant"` treats b_0, b_1 as parameters estimated within the Gibbs sampler with $b_j \sim N(0, 1/2)$ priors, yielding a parameterisation-invariant model (Hahn et al., 2020, Section 5.2).

`propensity` Optional numeric vector of propensity scores $\hat{e}(x_i)$ for training observations. Required when `treatment_coding = "adaptive"`.

`propensity_test`
: Optional numeric vector of propensity scores for test observations. Only used when `treatment_coding = "adaptive"`. Defaults to `0.5` for all test observations if not provided.

`n_chains`
: Number of independent MCMC chains to run. Default is `1` (standard single-chain behaviour). When `n_chains > 1` the chains are run in parallel via `parallel::mclapply` and their posterior samples are pooled into a single `CausalShrinkageForest` object, so all existing `print` and `summary` methods work without modification. On Windows, `mclapply` falls back to sequential execution.

`verbose`
: Logical; whether to print verbose output. Default is `TRUE`.

Details

This function is a flexible generalization of `CausalHorseForest`. The Causal Shrinkage Forest model decomposes the outcome into a prognostic (control) and a treatment effect part. Each part is modeled by its own shrinkage tree ensemble, with separate flexible global-local shrinkage priors. It is particularly useful for estimating heterogeneous treatment effects in high-dimensional settings. Further methodological details on the Horseshoe Forest framework can be found in Jacobs, van Wieringen & van der Pas (2026).

The `horseshoe` prior is the fully Bayesian global-local shrinkage prior, where both the global and local shrinkage parameters are assigned half-Cauchy distributions with scale hyperparameters `global_hp` and `local_hp`, respectively. The global shrinkage parameter is defined separately for each tree, allowing adaptive regularization per tree.

The `horseshoe_fw` prior (forest-wide horseshoe) is similar to `horseshoe`, except that the global shrinkage parameter is shared across all trees in the forest simultaneously.

The `half-cauchy` prior considers only local shrinkage and does not include a global shrinkage component. It places a half-Cauchy prior on each local shrinkage parameter with scale hyperparameter `local_hp`.

The `dirichlet` prior implements the Dirichlet–Sparse splitting rule of Linero (2018), in which splitting probabilities follow a Dirichlet prior whose concentration is controlled by a Beta sparsity parameter (`a_dirichlet`, `b_dirichlet`) and an expected sparsity level `rho_dirichlet`.

Value

An S3 object of class `"CausalShrinkageForest"` containing:

train_predictions Posterior mean predictions on training data (combined forest).

test_predictions Posterior mean predictions on test data (combined forest).

train_predictions_control Estimated control outcomes on training data.

test_predictions_control Estimated control outcomes on test data.

train_predictions_treat Estimated treatment effects on training data.

test_predictions_treat Estimated treatment effects on test data.

sigma Vector of posterior samples for the error standard deviation.

acceptance_ratio_control Average acceptance ratio in control forest.

acceptance_ratio_treat Average acceptance ratio in treatment forest.

train_predictions_sample_control Matrix of posterior samples for control predictions (if `store_posterior_sample = TRUE`).

test_predictions_sample_control Matrix of posterior samples for control predictions (if `store_posterior_sample = TRUE`).

train_predictions_sample_treat Matrix of posterior samples for treatment effects (if `store_posterior_sample = TRUE`).

test_predictions_sample_treat Matrix of posterior samples for treatment effects (if `store_posterior_sample = TRUE`).

References

Jacobs, T., van Wieringen, W. N., & van der Pas, S. L. (2026). *Horseshoe Forests for High-Dimensional Causal Survival Analysis.* Bayesian Analysis, advance publication, 1-30. https://doi.org/10.1214/26-BA1603

Chipman, H. A., George, E. I., & McCulloch, R. E. (2010). *BART: Bayesian additive regression trees.* Annals of Applied Statistics.

Linero, A. R. (2018). *Bayesian regression trees for high-dimensional prediction and variable selection.* Journal of the American Statistical Association.

See Also

Model family: `CausalHorseForest` (causal, horseshoe prior), `ShrinkageTrees` (non-causal, flexible prior), `HorseTrees` (non-causal, horseshoe prior).

Survival wrappers: `SurvivalBCF`, `SurvivalShrinkageBCF`.

S3 methods: `print.CausalShrinkageForest`, `summary.CausalShrinkageForest`, `predict.CausalShrinkageForest`, `plot.CausalShrinkageForest`.

Examples

```
# Example: Continuous outcome, homogeneous treatment effect, two priors
n <- 50
p <- 3
X <- matrix(runif(n * p), ncol = p)
X_treat <- X_control <- X
treat <- rbinom(n, 1, X[,1])
tau <- 2
y <- X[, 1] + (0.5 - treat) * tau + rnorm(n)

# Fit a standard Causal Horseshoe Forest
fit_horseshoe <- CausalShrinkageForest(y = y,
                                       X_train_control = X_control,
                                       X_train_treat = X_treat,
                                       treatment_indicator_train = treat,
                                       outcome_type = "continuous",
                                       number_of_trees_treat = 5,
                                       number_of_trees_control = 5,
                                       prior_type_control = "horseshoe",
                                       prior_type_treat = "horseshoe",
                                       local_hp_control = 1.5/sqrt(5),
```

```
                                        local_hp_treat = 1.5/sqrt(5),
                                        global_hp_control = 1.5/sqrt(5),
                                        global_hp_treat = 1.5/sqrt(5),
                                        N_post = 10,
                                        N_burn = 5,
                                        store_posterior_sample = TRUE,
                                        verbose = FALSE
)

# Fit a Causal Shrinkage Forest with half-cauchy prior
fit_halfcauchy <- CausalShrinkageForest(y = y,
                                         X_train_control = X_control,
                                         X_train_treat = X_treat,
                                         treatment_indicator_train = treat,
                                         outcome_type = "continuous",
                                         number_of_trees_treat = 5,
                                         number_of_trees_control = 5,
                                         prior_type_control = "half-cauchy",
                                         prior_type_treat = "half-cauchy",
                                         local_hp_control = 1/sqrt(5),
                                         local_hp_treat = 1/sqrt(5),
                                         N_post = 10,
                                         N_burn = 5,
                                         store_posterior_sample = TRUE,
                                         verbose = FALSE
)

# Posterior mean CATEs
CATE_horseshoe <- colMeans(fit_horseshoe$train_predictions_sample_treat)
CATE_halfcauchy <- colMeans(fit_halfcauchy$train_predictions_sample_treat)

# Posteriors of the ATE
post_ATE_horseshoe <- rowMeans(fit_horseshoe$train_predictions_sample_treat)
post_ATE_halfcauchy <- rowMeans(fit_halfcauchy$train_predictions_sample_treat)

# Posterior mean ATE
ATE_horseshoe <- mean(post_ATE_horseshoe)
ATE_halfcauchy <- mean(post_ATE_halfcauchy)

# Example: Interval-censored causal survival outcome
n <- 50; p <- 3
X_ic <- matrix(rnorm(n * p), ncol = p)
treat_ic <- rbinom(n, 1, 0.5)
true_t <- rexp(n, rate = exp(-X_ic[, 1] - 0.5 * treat_ic))
left_t  <- true_t * runif(n, 0.5, 1)
right_t <- true_t * runif(n, 1, 1.5)
exact <- sample(n, 15)
left_t[exact] <- true_t[exact]; right_t[exact] <- true_t[exact]
rc <- sample(setdiff(seq_len(n), exact), 10); right_t[rc] <- Inf

fit_ic <- CausalShrinkageForest(
  left_time = left_t, right_time = right_t,
  X_train_control = X_ic, X_train_treat = X_ic,
```

```
  treatment_indicator_train = treat_ic,
  outcome_type = "interval-censored",
  number_of_trees_control = 5, number_of_trees_treat = 5,
  prior_type_control = "horseshoe", prior_type_treat = "horseshoe",
  local_hp_control = 1.5/sqrt(5), local_hp_treat = 1.5/sqrt(5),
  global_hp_control = 1.5/sqrt(5), global_hp_treat = 1.5/sqrt(5),
  N_post = 10, N_burn = 5,
  store_posterior_sample = TRUE, verbose = FALSE)
```

`HorseTrees` *Horseshoe Regression Trees (HorseTrees)*

Description

Fits a Bayesian Horseshoe Trees model with a single learner. Implements regularization on the step heights using a global-local Horseshoe prior, controlled via the parameter `k`. Supports continuous, binary, right-censored, and interval-censored (survival) outcomes.

Usage

```
HorseTrees(
  y = NULL,
  status = NULL,
  X_train,
  X_test = NULL,
  left_time = NULL,
  right_time = NULL,
  outcome_type = "continuous",
  timescale = "time",
  number_of_trees = 200,
  k = 1,
  power = 2,
  base = 0.95,
  p_grow = 0.4,
  p_prune = 0.4,
  nu = 3,
  q = 0.9,
  sigma = NULL,
  N_post = 1000,
  N_burn = 1000,
  delayed_proposal = 5,
  store_posterior_sample = TRUE,
  n_chains = 1,
  verbose = TRUE
)
```

Arguments

y	Outcome vector. Numeric. Can represent continuous outcomes, binary outcomes (0/1), or follow-up times for survival data. Set to NULL (default) when using outcome_type = "interval-censored", as values are derived from left_time and right_time.
status	Optional censoring indicator vector (1 = event occurred, 0 = censored). Required if outcome_type = "right-censored". For interval-censored outcomes, this is derived automatically from left_time and right_time.
X_train	Covariate matrix for training. Each row corresponds to an observation, and each column to a covariate.
X_test	Optional covariate matrix for test data. If NULL, defaults to the mean of the training covariates.
left_time	Optional numeric vector of left (lower) time boundaries. Required when outcome_type = "interval-censored". Exact events have left_time == right_time; right-censored observations have right_time = Inf; interval-censored observations have finite left_time < right_time.
right_time	Optional numeric vector of right (upper) time boundaries. Required when outcome_type = "interval-censored". Use Inf for right-censored observations.
outcome_type	Type of outcome. One of "continuous", "binary", "right-censored", or "interval-censored".
timescale	Indicates the scale of follow-up times. Options are "time" (nonnegative follow-up times, will be log-transformed internally) or "log" (already log-transformed). Only used when outcome_type = "right-censored" or "interval-censored".
number_of_trees	Number of trees in the ensemble. Default is 200.
k	Horseshoe scale hyperparameter. Default is 1.0. Controls the overall level of shrinkage by setting the scale of both the local and the global component, parameterized as $\alpha_{\text{local}} = \alpha_{\text{global}} = k/\sqrt{\text{number\_of\_trees}}$. **Choosing k.** Values between roughly 0.5 and 1.5 worked well across a wide range of simulated settings. Within that range: • **smaller k shrinks more aggressively** – tighter credible intervals, more leaves pulled to zero. Prefer it when the signal is sparse, the dimension is high, or the sample is small. • **larger k is more conservative** – wider intervals, more freedom for individual leaves. Prefer it when the function is complex or the signal is strong. The default sits in the middle. If you would rather select it from the data, k is a natural target for cross-validation; see the "Choosing k" section of the package vignette. **Changed in version 2.1.0.** The default was 0.1 in earlier releases. That value was calibrated against a defect in the global update which made global_hp inert (see NEWS.md); with the defect corrected it shrinks far too aggressively. Pass k = 0.1 explicitly only if you need to reproduce pre-2.1.0 output, and note that this does not reproduce it exactly, since the sampler itself has been corrected.
power	Power parameter for tree structure prior. Default is 2.0.

base	Base parameter for tree structure prior. Default is 0.95.
p_grow	Probability of proposing a grow move. Default is 0.4.
p_prune	Probability of proposing a prune move. Default is 0.4.
nu	Degrees of freedom for the error distribution prior. Default is 3.
q	Quantile hyperparameter for the error variance prior. Default is 0.90.
sigma	Optional known value for error standard deviation. If NULL, estimated from data.
N_post	Number of posterior samples to store. Default is 1000.
N_burn	Number of burn-in iterations. Default is 1000.
delayed_proposal	Number of delayed iterations before proposal. Only for reversible updates. Default is 5.
store_posterior_sample	Logical; whether to store posterior samples for each iteration. Default is TRUE.
n_chains	Number of independent MCMC chains to run. Default is 1 (standard single-chain behaviour). When n_chains > 1 the chains are run in parallel via parallel::mclapply and their posterior samples are pooled into a single ShrinkageTrees object, so all existing print, summary, and predict methods work without modification. On Windows, mclapply falls back to sequential execution.
verbose	Logical; whether to print verbose output. Default is TRUE.

Details

For continuous outcomes, the model centers and optionally standardizes the outcome using a prior guess of the standard deviation. For binary outcomes, the function uses a probit link formulation. For right-censored outcomes (survival data), the function can handle follow-up times either on the original time scale or log-transformed. For interval-censored outcomes, provide `left_time` and `right_time` instead of `y` and `status`; the event indicators are derived internally following the `survival::Surv(type = "interval2")` convention. Generalized implementation with multiple prior possibilities is given by ShrinkageTrees.

Value

An S3 object of class `"ShrinkageTrees"` with the following elements:

train_predictions Vector of posterior mean predictions on the training data.

test_predictions Vector of posterior mean predictions on the test data (or on mean covariate vector if `X_test` not provided).

sigma Vector of posterior samples of the error variance.

acceptance_ratio Average acceptance ratio across trees during sampling.

train_predictions_sample Matrix of posterior samples of training predictions (iterations in rows, observations in columns). Present only if `store_posterior_sample = TRUE`.

test_predictions_sample Matrix of posterior samples of test predictions. Present only if `store_posterior_sample = TRUE`.

train_probabilities Vector of posterior mean probabilities on the training data (only for `outcome_type = "binary"`).

test_probabilities Vector of posterior mean probabilities on the test data (only for `outcome_type = "binary"`).

train_probabilities_sample Matrix of posterior samples of training probabilities (only for `outcome_type = "binary"` and if `store_posterior_sample = TRUE`).

test_probabilities_sample Matrix of posterior samples of test probabilities (only for `outcome_type = "binary"` and if `store_posterior_sample = TRUE`).

See Also

Model family: `ShrinkageTrees` (flexible prior choice), `CausalHorseForest` (causal inference), `CausalShrinkageForest` (causal, flexible prior).

Survival wrappers: `SurvivalBART`, `SurvivalDART`.

S3 methods: `print.ShrinkageTrees`, `summary.ShrinkageTrees`, `predict.ShrinkageTrees`, `plot.ShrinkageTrees`.

Examples

```
# Minimal example: continuous outcome
n <- 25
p <- 5
X <- matrix(rnorm(n * p), ncol = p)
y <- X[, 1] + rnorm(n)
fit1 <- HorseTrees(y = y, X_train = X, outcome_type = "continuous",
                   number_of_trees = 5, N_post = 75, N_burn = 25,
                   verbose = FALSE)

# Minimal example: binary outcome
X <- matrix(rnorm(n * p), ncol = p)
y <- ifelse(X[, 1] + rnorm(n) > 0, 1, 0)
fit2 <- HorseTrees(y = y, X_train = X, outcome_type = "binary",
                   number_of_trees = 5, N_post = 75, N_burn = 25,
                   verbose = FALSE)

# Minimal example: right-censored outcome
X <- matrix(rnorm(n * p), ncol = p)
time <- rexp(n, rate = 0.1)
status <- rbinom(n, 1, 0.7)
fit3 <- HorseTrees(y = time, status = status, X_train = X,
                   outcome_type = "right-censored", number_of_trees = 5,
                   N_post = 75, N_burn = 25, verbose = FALSE)

# Minimal example: interval-censored outcome
X <- matrix(rnorm(n * p), ncol = p)
true_t <- rexp(n, rate = 0.1)
left_t  <- true_t * runif(n, 0.5, 1)
right_t <- true_t * runif(n, 1, 1.5)
# Mark some as exact, some as right-censored
exact <- sample(n, 8); left_t[exact] <- true_t[exact]; right_t[exact] <- true_t[exact]
```

```
rc <- sample(setdiff(seq_len(n), exact), 5); right_t[rc] <- Inf
fit4 <- HorseTrees(left_time = left_t, right_time = right_t, X_train = X,
                   outcome_type = "interval-censored", number_of_trees = 5,
                   N_post = 75, N_burn = 25, verbose = FALSE)

# Larger continuous example (not run automatically)

n <- 100
p <- 100
X <- matrix(rnorm(100 * p), ncol = p)
X_test <- matrix(rnorm(50 * p), ncol = p)
y <- X[, 1] + X[, 2] - X[, 3] + rnorm(100, sd = 0.5)

fit5 <- HorseTrees(y = y,
                   X_train = X,
                   X_test = X_test,
                   outcome_type = "continuous",
                   number_of_trees = 200,
                   N_post = 2500,
                   N_burn = 2500,
                   store_posterior_sample = TRUE,
                   verbose = TRUE)

plot(fit4$sigma, type = "l", ylab = expression(sigma),
     xlab = "Iteration", main = "Sigma traceplot")

hist(fit4$train_predictions_sample[, 1],
     main = "Posterior distribution of prediction outcome individual 1",
     xlab = "Prediction", breaks = 20)
```

`ovarian` *Semi-synthesised TCGA Ovarian Cancer Dataset*

Description

Gene expression and clinical covariates for ovarian cancer patients from The Cancer Genome Atlas (TCGA-OV), combined with semi-synthetic survival outcomes and treatment assignment. Real covariates (age, FIGO stage, tumor grade, gene expression) are retained; survival times, event indicator, and treatment assignment are simulated from a known data-generating process so that the true treatment effect is available for validation (see `ovarian_truth`).

Usage

```
ovarian
```

Format

A data frame with 357 rows (patients) and 1004 columns:

- **OS_time**: Numeric. Observed survival time in days (simulated).
- **OS_event**: Integer. Event indicator (simulated). 1 = event observed, 0 = right-censored.
- **treatment**: Integer. Simulated treatment assignment. 1 = carboplatin, 0 = cisplatin. Driven primarily by `year_of_diagnosis`, a confounder (cisplatin era pre-~2000, carboplatin after).
- **age**: Integer. Age at initial pathologic diagnosis in years.
- **figo_stage**: Integer. FIGO stage coded as 2 = Stage II, 3 = Stage III, 4 = Stage IV.
- **tumor_grade**: Integer. Histologic tumor grade coded as 2 = G2, 3 = G3, 4 = G4. Rows with GX (unknown grade) were excluded.
- **year_of_diagnosis**: Integer. Year of initial pathologic diagnosis (approx. 1992–2013). A confounder in the DGP: it drives treatment assignment and also has a direct prognostic effect.
- **ENSG...**: Numeric. Log2(TPM + 1) normalised gene expression levels for 997 Ensembl genes (columns named by versioned Ensembl gene IDs, e.g. `ENSG00000270372.1`). Genes were selected as the most variable transcripts across TCGA-OV samples, ranked by median absolute deviation (MAD).

Details

Every column except `OS_time`, `OS_event` and `treatment` is a baseline covariate, so the design matrix is

```
drop <- c("OS_time", "OS_event", "treatment")
X <- as.matrix(ovarian[, setdiff(names(ovarian), drop)])
```

Select by name rather than position. Earlier versions of this dataset carried outcome-derived `left_time` and `right_time` columns together with a duplicated `year_of_diagnosis`; positional selection silently admitted them to the design matrix, leaking the outcome. They have been removed. To demonstrate the interval-censored interface, build the bounds from the outcome instead:

```
left  <- ovarian$OS_time
right <- ifelse(ovarian$OS_event == 1, ovarian$OS_time, Inf)
```

RNA-seq data were downloaded from the GDC portal using the `TCGAbiolinks` package (STAR - Counts workflow). Expression values were normalised to TPM and log2-transformed as log2(TPM + 1). Genes with median TPM <= 1 across all samples were removed prior to MAD filtering. Clinical data were obtained from the BCR Biotab clinical supplement. Treatment assignment was derived from the drug table (`clinical_drug_ov`), restricted to adjuvant (first-line) treatment records. Samples were matched between expression and clinical data using the 12-character TCGA patient barcode.

Source

https://portal.gdc.cancer.gov/projects/TCGA-OV

References

Cancer Genome Atlas Research Network (2011). Integrated genomic analyses of ovarian carcinoma. *Nature*, 474, 609–615. doi:10.1038/nature10166

Colaprico, A. et al. (2016). TCGAbiolinks: an R/Bioconductor package for integrative analysis with GDC data. *Nucleic Acids Research*, 44(8). doi:10.1093/nar/gkv1507

Examples

```
data(ovarian)

# Dimensions: patients x (6 clinical + 2000 gene columns)
dim(ovarian)

# Survival outcome
head(ovarian[, c("OS_time", "OS_event", "treatment")])

# KM plot by treatment
if (requireNamespace("survival", quietly = TRUE)) {
  library(survival)
  fit <- survfit(Surv(OS_time, OS_event) ~ treatment, data = ovarian)
  plot(fit, col = c("blue", "red"), xlab = "Time (days)", ylab = "Survival")
  legend("topright", c("Carboplatin", "Cisplatin"), col = c("blue", "red"), lty = 1)
}
```

---

ovarian_truth *Ground-truth quantities for the semi-synthesised ovarian dataset*

Description

The simulated quantities that correspond to the `ovarian` dataset. Because the ovarian outcomes and treatment assignment are generated from a known data-generating process, the underlying potential outcomes, prognostic function, conditional treatment effect, and propensity score are available for validating estimators of treatment effects under right- and interval-censored survival.

Usage

```
ovarian_truth
```

Format

A data frame with one row per patient in `ovarian` and the following columns:

mu Numeric. Prognostic function $\mu(x)$: expected log survival under the control treatment.

tau Numeric. Conditional average treatment effect $\tau(x)$ on the log-survival scale. This is the CATE; $\exp(\tau)$ is the acceleration factor.

f Numeric. The true regression function $f(x) = \mu(x) + A\,\tau(x)$, that is $E[\log T \mid x, A]$ under the treatment actually assigned. This is what a prediction model estimates when treatment is included as an ordinary covariate, so it is the target for prediction error.

propensity Numeric. True propensity $e(x) = P(A = 1 \mid x)$ used to simulate the observed assignment.

log_time Numeric. True (uncensored) survival time on the log scale, under the treatment actually assigned.

time Numeric. The same on the original scale, in months.

censoring_time Numeric. The censoring time drawn for this patient, so it is visible why an observation was censored.

Details

The data-generating process is

$$\log T = \mu(x) + A\,\tau(x) + \sigma W,$$

with W standard normal. The residual scale σ is stored as the attribute `"sigma"`, and the indices of the covariates that enter μ and τ as `"active_prog"` and `"active_tau"`. Attributes are dropped by row subsetting, so read them before subsetting.

`f` is the target for a model that uses treatment as a covariate. A model fitted without it estimates $\mu(x) + e(x)\,\tau(x)$ instead, by the law of total expectation. Score prediction error against whichever matches the design matrix that was used.

See Also

`ovarian` for the observed semi-synthesised data.

Examples

```
data(ovarian)
data(ovarian_truth)
stopifnot(nrow(ovarian) == nrow(ovarian_truth))

# True average treatment effect on the log-survival scale
mean(ovarian_truth$tau)

# log T decomposes as f plus noise
stopifnot(all.equal(ovarian_truth$f,
                    ovarian_truth$mu + ovarian$treatment * ovarian_truth$tau))

# Target for a model fitted without treatment
eta <- ovarian_truth$mu + ovarian_truth$propensity * ovarian_truth$tau

# Best out-of-sample concordance the process allows: no model can beat this
sigma <- attr(ovarian_truth, "sigma")
rho   <- sd(eta) / sqrt(var(eta) + sigma^2)
0.5 + asin(rho) / pi
```

pdac *Processed TCGA PAAD dataset (pdac)*

Description

A reduced and cleaned subset of the TCGA pancreatic ductal adenocarcinoma (PAAD) dataset, derived from The Cancer Genome Atlas (TCGA) PAAD cohort. This version, pdac, is smaller and simplified for practical analyses and package examples.

Usage

```
pdac
```

Format

A data frame with rows corresponding to patients and columns as described above.

Details

This dataset was originally compiled and curated in the open-source pdacR package by Torre-Healy et al. (2023), which harmonized and integrated the TCGA PAAD gene expression and clinical data. The current version further reduces and simplifies the data for efficient modeling demonstrations and survival analyses.

The data frame includes:

- **time**: Overall survival time in months.
- **status**: Event indicator; 1 = event occurred, 0 = censored.
- **treatment**: Binary treatment indicator; 1 = radiation therapy, 0 = control.
- **age**: Age at initial pathologic diagnosis (numeric).
- **sex**: Binary sex indicator; 1 = male, 0 = female.
- **grade**: Tumor differentiation grade (ordinal; 1 = well, 2 = moderate, 3 = poor, 4 = undifferentiated).
- **tumor.cellularity**: Tumor cellularity estimate (numeric).
- **tumor.purity**: Tumor purity class (binary; 1 = high, 0 = low).
- **absolute.purity**: Absolute purity estimate (numeric).
- **moffitt.cluster**: Moffitt transcriptional subtype (binary; 1 = basal-like, 0 = classical).
- **meth.leukocyte.percent**: DNA methylation leukocyte estimate (numeric).
- **meth.purity.mode**: DNA methylation purity mode (numeric).
- **stage**: Nodal stage indicator (binary; 1 = n1, 0 = n0).
- **lymph.nodes**: Number of lymph nodes examined (numeric).
- **Driver gene columns**: Expression values of key driver genes (e.g., KRAS, TP53, CDKN2A, SMAD4, BRCA1, BRCA2).
- **Other gene columns**: Expression values of ~3,000 most variable non-driver genes (based on median absolute deviation).

Source

doi:10.1016/j.ccell.2017.07.007

References

- Raphael BJ, et al. "Integrated genomic characterization of pancreatic ductal adenocarcinoma." Cancer Cell. 2017 Aug 14;32(2):185–203.e13. PMID: 28810144.
- Torre-Healy LA, Kawalerski RR, Oh K, et al. "Open-source curation of a pancreatic ductal adenocarcinoma gene expression analysis platform (pdacR) supports a two-subtype model." Communications Biology. 2023; https://doi.org/10.1038/s42003-023-04461-6.
- The Cancer Genome Atlas (TCGA), PAAD project, DbGaP: phs000178.

---

plot.CausalShrinkageForest

Plot diagnostics for a CausalShrinkageForest model

Description

Visualises posterior draws using **ggplot2**. Requires the suggested package **ggplot2**.

Usage

```
## S3 method for class 'CausalShrinkageForest'
plot(
  x,
  type = c("trace", "density", "ate", "cate", "vi"),
  forest = c("both", "control", "treat"),
  n_vi = 10,
  bayesian_bootstrap = TRUE,
  ...
)
```

Arguments

x	A `CausalShrinkageForest` object.
type	Character; one of:
	`"trace"` Sigma traceplot (chain mixing).
	`"density"` Overlaid posterior density of sigma per chain.
	`"ate"` Posterior density of the average treatment effect (ATE) with 95 percent credible region. Requires `store_posterior_sample = TRUE`.
	`"cate"` Point estimates and 95 percent credible intervals for the CATE of each training observation, sorted by posterior mean. Requires `store_posterior_sample = TRUE`.
	`"vi"` Posterior credible intervals for variable inclusion probabilities (Dirichlet prior only). Controlled by `forest`.

`forest` For `type = "vi"`: which forest to display. One of `"both"` (default), `"control"`, or `"treat"`. When `"both"`, a named list of two **ggplot2** objects is returned.

`n_vi` Integer; number of top variables for `type = "vi"`. Default `10`.

`bayesian_bootstrap` Logical; only used when `type = "ate"`. If `TRUE` (default), the ATE posterior is computed by reweighting each iteration's CATE vector with Dirichlet(1, ..., 1) weights, giving the population ATE (PATE). If `FALSE`, equal $1/n$ weights are used, giving the mixed ATE (MATE).

`...` Additional arguments (currently unused).

Value

A **ggplot2** object, or (for `type = "vi"` with `forest = "both"`) a named list with elements `control` and `treat`.

Examples

```
if (requireNamespace("ggplot2", quietly = TRUE)) {
  set.seed(1)
  n <- 60; p <- 5
  X <- matrix(rnorm(n * p), ncol = p)
  w <- rbinom(n, 1, 0.5)
  y <- X[, 1] + w * 1.5 * (X[, 2] > 0) + rnorm(n, sd = 0.5)

  fit <- CausalShrinkageForest(
    y = y,
    X_train_control = X, X_train_treat = X,
    treatment_indicator_train = w,
    prior_type_control = "horseshoe", prior_type_treat = "horseshoe",
    local_hp_control = 0.1, global_hp_control = 0.1,
    local_hp_treat  = 0.1, global_hp_treat  = 0.1,
    number_of_trees_control = 5, number_of_trees_treat = 5,
    N_post = 50, N_burn = 25,
    store_posterior_sample = TRUE,
    verbose = FALSE
  )

  plot(fit, type = "trace")
  plot(fit, type = "ate")
  plot(fit, type = "cate")
}
```

`plot.PosteriorProjection` *Plot a posterior projection*

Description

Plot a posterior projection

Usage

```
## S3 method for class 'PosteriorProjection'
plot(
  x,
  type = c("coefficients", "rho2", "path", "effects", "tree"),
  n_max = 25,
  ...
)
```

Arguments

x	A PosteriorProjection object.
type	"coefficients" is a caterpillar plot of the projection coefficients; "rho2" the posterior of the summary R^2; "path" the cross-validation curve against $\log \lambda$, marking lambda.min and lambda.1se (any penalty whose lambda was cross-validated); "effects" the per-covariate partial effect curves with credible bands (family = "additive" only); "tree" the CART partition, with the posterior interval for each leaf (family = "tree" only).
n_max	Maximum number of coefficients to show, type = "coefficients". The largest by absolute posterior mean are kept.
...	Currently unused.

Value

A **ggplot2** object.

---

plot.ShrinkageTrees *Plot diagnostics for a ShrinkageTrees model*

Description

Visualises posterior draws using **ggplot2**. Requires the suggested package **ggplot2**.

Usage

```
## S3 method for class 'ShrinkageTrees'
plot(
  x,
  type = c("trace", "density", "vi", "survival"),
  n_vi = 10,
  obs = NULL,
  t_grid = NULL,
```

```
  level = 0.95,
  km = FALSE,
  ...
)
```

Arguments

x	A ShrinkageTrees object.
type	Character; one of:
	"trace" Sigma traceplot across MCMC iterations (one line per chain). Useful for assessing chain mixing.
	"density" Overlaid posterior density of sigma, one curve per chain.
	"vi" Posterior credible intervals for variable inclusion probabilities (top n_vi predictors). Only available for Dirichlet prior models.
	"survival" Posterior survival curves $S(t\|x_i) = 1 - \Phi((\log t - \mu_i)/\sigma)$ with pointwise credible bands, derived from the AFT log-normal model. Only available for survival outcome types ("right-censored" or "interval-censored"). **Population-averaged curve** (default, obs = NULL): computes $\bar{S}(t) = n^{-1} \sum_i S(t\|x_i)$ at each MCMC iteration. The credible band reflects posterior uncertainty in both μ_i and σ when store_posterior_sample = TRUE, or sigma-only uncertainty otherwise. **Individual curves** (obs = c(1, 5, ...)): one curve per selected training observation with its own credible band. Set km = TRUE to overlay the Kaplan–Meier estimate as a non-parametric reference (population-averaged plot only).
n_vi	Integer; number of top variables to display when type = "vi". Default 10.
obs	Integer vector of training-set observation indices for individual survival curves, or NULL (default) for the population-averaged curve. Indices must be between 1 and the number of training observations. Used only when type = "survival".
t_grid	Optional numeric vector of time points (on the original time scale) at which to evaluate the survival function. If NULL (default), a grid of 200 equally spaced points is generated from the range of observed training times. Used only when type = "survival".
level	Width of the pointwise credible band for type = "survival". Default 0.95 (a 95 percent credible interval at each time point).
km	Logical; if TRUE and type = "survival" with obs = NULL, overlay the Kaplan–Meier curve (dashed black step function) as a non-parametric reference. Default FALSE. Requires the **survival** package. Ignored with a message when obs is not NULL.
...	Additional arguments (currently unused).

Value

A **ggplot2** object.

Examples

```
if (requireNamespace("ggplot2", quietly = TRUE)) {
  set.seed(1)
  n <- 50; p <- 5
  X <- matrix(rnorm(n * p), ncol = p)
  y <- X[, 1] + rnorm(n)

  # Fit a small continuous model
  fit <- ShrinkageTrees(
    y = y, X_train = X,
    prior_type = "horseshoe",
    local_hp = 0.1, global_hp = 0.1,
    number_of_trees = 5,
    N_post = 50, N_burn = 25,
    verbose = FALSE
  )

  # Sigma traceplot -- check chain mixing
  plot(fit, type = "trace")

  # Overlaid posterior densities of sigma per chain
  plot(fit, type = "density")
}
```

`plot.ShrinkageTreesPrediction` *Plot posterior predictive survival curves*

Description

Plots posterior predictive survival curves for new observations from a `ShrinkageTreesPrediction` object. Only available for survival outcome types (`"right-censored"` or `"interval-censored"`).

Usage

```
## S3 method for class 'ShrinkageTreesPrediction'
plot(x, type = "survival", obs = NULL, t_grid = NULL, level = 0.95, ...)
```

Arguments

`x`	A `ShrinkageTreesPrediction` object returned by `predict.ShrinkageTrees` for a survival model.
`type`	Character; currently only `"survival"` is supported.
`obs`	Integer vector of predicted-observation indices for individual survival curves, or `NULL` (default) for the population-averaged curve across all predicted observations.

t_grid	Optional numeric vector of time points (on the original time scale) at which to evaluate the survival function. If NULL (default), a grid of 200 equally spaced points is generated automatically.
level	Width of the pointwise credible band. Default 0.95.
...	Additional arguments (currently unused).

Value

A **ggplot2** object.

See Also

predict.ShrinkageTrees, plot.ShrinkageTrees

Examples

```
if (requireNamespace("ggplot2", quietly = TRUE)) {
  set.seed(1)
  n <- 40; p <- 3
  X <- matrix(rnorm(n * p), ncol = p)
  X_test <- matrix(rnorm(10 * p), ncol = p)
  time <- rexp(n, rate = exp(0.5 * X[, 1]))
  status <- rbinom(n, 1, 0.7)

  fit_surv <- SurvivalBART(
    time = time, status = status, X_train = X,
    number_of_trees = 5, N_post = 50, N_burn = 25,
    store_posterior_sample = TRUE, verbose = FALSE
  )

  pred <- predict(fit_surv, newdata = X_test)
  plot(pred, type = "survival")
}
```

---

posterior_projection *Lower-dimensional posterior projection*

Description

Projects every posterior draw of the fitted function onto a simpler model, following Woody, Carvalho and Murray (2021), and returns the resulting posterior distribution over projections.

Usage

```
posterior_projection(object, ...)

## S3 method for class 'ShrinkageTrees'
posterior_projection(object, ...)

## S3 method for class 'CausalShrinkageForest'
posterior_projection(object, target = c("treatment", "prognostic"), ...)

## Default S3 method:
posterior_projection(
  object,
  X,
  family = c("linear", "additive", "tree"),
  covariates = NULL,
  penalty = c("none", "ridge", "lasso", "elastic_net"),
  alpha = 0.5,
  lambda = NULL,
  df_spline = 4,
  max_leaves = 6,
  weights = NULL,
  level = 0.95,
  scale_label = "response",
  target = "f",
  ...
)
```

Arguments

object	A fitted ShrinkageTrees or CausalShrinkageForest object, or (default method) an S x n matrix of posterior draws on the additive scale.
...	Passed on to the default method.
target	Which function to project: "treatment" for the CATE surface $\tau(x)$, or "prognostic" for $\mu(x)$.
X	(default method) The n x p design the supplied draws correspond to; column names label the projection. Not a newdata argument – it must come from somewhere when there is no fitted object to read it from.
family	"linear" projects onto a linear model, "additive" onto a natural-spline additive model (per-covariate effect curves with posterior bands), "tree" onto a CART partition.
covariates	Optional character or integer vector selecting the covariates to project onto. Default NULL uses all of them. No selection is performed: choosing this subset is the user's job.
penalty	Regularisation for the linear projection: "none" for ordinary least squares, or "ridge", "lasso", "elastic_net". The penalised options are the ones that stay defined when the covariate set is larger than n, which is the regime this

	package targets, and they summarise the posterior mean without uncertainty quantification. `family = "linear"` only.
`alpha`	Elastic-net mixing parameter, `penalty = "elastic_net"`. `alpha = 1` is the lasso, `alpha = 0` the ridge.
`lambda`	Penalty level for the penalised projections. `NULL` or `"1se"` cross-validates on the posterior mean and takes `lambda.1se`, `"min"` takes `lambda.min`, and a number is used directly.
`df_spline`	Degrees of freedom per natural spline, `family = "additive"`.
`max_leaves`	Maximum number of leaves, `family = "tree"`.
`weights`	Optional non-negative weights over the observations, defining the population the projection is taken over. Default uniform.
`level`	Credible level for all reported intervals. Unused when `penalty != "none"`, which reports no intervals.
`scale_label`	Provenance label carried into the printed output (default method).

Details

The projection is taken over the design the model was fitted to; there is no `newdata` argument. Nor is there any covariate selection: by default the projection uses every covariate, and a subset can be supplied through `covariates`. Choosing that subset is the user's job.

With `penalty = "none"` the projection is solved separately for every posterior draw, so the result is a posterior over projections: the coefficients, the partial effects and the summary R^2 all come with credible intervals.

The penalised projections behave differently. They are solved **once**, on the posterior mean $\bar{f}$, and report point estimates only. The reason is that a per-draw penalised solution does not yield an interval anyone can interpret: the penalised estimator is biased and non-smooth in its input, its spread across draws has no coverage guarantee for the projection of the true surface, and the implied inclusion frequencies read as selection probabilities while being driven by the penalty. A single sparse or shrunken summary of the Bayes estimate states exactly what it is. Use `penalty = "none"` when the uncertainty in the summary is the point.

`lambda` is chosen by cross-validating on $\bar{f}$. Holding out observations asks how well a sparse linear summary generalises across the covariate space, which is a real question even though a posterior draw carries no observation noise.

With `penalty = "ridge"` nothing is set to zero, so the result is a regularised projection rather than a lower-dimensional summary.

Value

An object of class `"PosteriorProjection"`, a list with `coefficients` (`S` x `q` draws of the projection coefficients, or a single row when `penalty != "none"`), `coef_summary`, `uq` (whether uncertainty is quantified), `rho2` (an `S`-vector of summary R^2, or one number when `penalty != "none"`), `covariates`, `structure`, and provenance fields.

References

Woody, S., Carvalho, C. M., & Murray, J. S. (2021). Model Interpretation Through Lower-Dimensional Posterior Summarization. *Journal of Computational and Graphical Statistics*, 30(1), 144-161.

See Also

print.PosteriorProjection(), plot.PosteriorProjection()

Examples

```
n <- 80; p <- 30
X <- matrix(rnorm(n * p), n, p,
            dimnames = list(NULL, paste0("x", 1:p)))
y <- 2 * X[, 1] - X[, 2] + rnorm(n)

fit <- HorseTrees(y = y, X_train = X, outcome_type = "continuous",
                  number_of_trees = 5, N_post = 50, N_burn = 25,
                  store_posterior_sample = TRUE, verbose = FALSE)

# Linear projection onto two covariates
pp <- posterior_projection(fit, family = "linear",
                           covariates = c("x1", "x2"))
pp

# Additive projection with per-covariate effect curves
pp_add <- posterior_projection(fit, family = "additive",
                               covariates = c("x1", "x2"))


# Sparse summary of the posterior mean; lambda cross-validated
if (requireNamespace("glmnet", quietly = TRUE)) {
  pp_las <- posterior_projection(fit, family = "linear",
                                 penalty = "lasso")
}

# Subgroup summary
if (requireNamespace("rpart", quietly = TRUE)) {
  pp_tree <- posterior_projection(fit, family = "tree", max_leaves = 4)
}
```

---

predict.CausalShrinkageForest

Posterior predictive inference for a CausalShrinkageForest model

---

Description

Re-runs the MCMC sampler on new covariate data using the stored training data and hyperparameters, returning posterior mean predictions and credible interval bounds for three quantities: the **prognostic function** (control-forest prediction $\mu(X)$), the **Conditional Average Treatment Effect** (CATE, $\tau(X)$), and the **total predicted outcome** ($\mu(X) + \tau(X)$).

Usage

```
## S3 method for class 'CausalShrinkageForest'
predict(
  object,
  newdata_control,
  newdata_treat,
  level = 0.95,
  bayesian_bootstrap = TRUE,
  ...
)
```

Arguments

`object`	A fitted `CausalShrinkageForest` model object.
`newdata_control`	A matrix of new covariates for the control forest, with the same number of columns as `X_train_control` at fit time.
`newdata_treat`	A matrix of new covariates for the treatment forest, with the same number of columns as `X_train_treat` at fit time. Must have the same number of rows as `newdata_control`.
`level`	Credible interval width. Default `0.95`.
`bayesian_bootstrap`	Logical; if `TRUE` (default), the ATE over `newdata` is computed by reweighting each iteration's CATE vector with Dirichlet(1, ..., 1) weights (population ATE, PATE). If `FALSE`, equal $1/n$ weights are used (mixed ATE, MATE).
`...`	Currently unused.

Details

The causal forest decomposes the expected outcome as

$$E[Y \mid X] = \mu(X) + \tau(X) \cdot W,$$

where $\mu(X)$ is the prognostic function (control forest), $\tau(X)$ is the CATE (treatment forest), and W is the treatment indicator.

For **continuous** outcomes and survival with `timescale = "log"`, all three components are on the response scale: `prognostic` and `total` include the intercept shift ($+\bar{y}$), while `cate` is the pure additive treatment effect with no intercept.

For **survival with** `timescale = "time"`, predictions are back-transformed to the original time scale:

- `prognostic`: posterior expected baseline survival time $E[\exp(\mu(X))]$.
- `cate`: multiplicative effect on survival time $\exp(\tau(X))$; a value greater than 1 means treatment prolongs survival.
- `total`: posterior expected survival time under the observed treatment $E[\exp(\mu(X) + \tau(X))]$.

Value

A `CausalShrinkageForestPrediction` object with elements:

prognostic List with `mean`, `lower`, `upper`: posterior summaries of the prognostic function $\mu(X_{\text{new}})$.

cate List with `mean`, `lower`, `upper`: posterior summaries of the CATE $\tau(X_{\text{new}})$.

total List with `mean`, `lower`, `upper`: posterior summaries of the total outcome $\mu(X_{\text{new}}) + \tau(X_{\text{new}})$.

ate List with `mean`, `lower`, `upper`: posterior summary of the average treatment effect over `newdata`. For survival with `timescale = "time"`, reported as a multiplicative time ratio on the original scale.

cate_samples $S \times n_{\text{new}}$ matrix of posterior CATE draws on the scale reported in `cate`.

bayesian_bootstrap Flag indicating whether the reported ATE CI used Dirichlet reweighting.

n Number of test observations.

level Credible level used.

outcome_type Outcome type inherited from the fitted model.

timescale Timescale inherited from the fitted model.

See Also

`CausalHorseForest`, `CausalShrinkageForest`, `print.CausalShrinkageForestPrediction`, `summary.CausalShrinkag`

---

predict.ShrinkageTrees

Posterior predictive inference for a ShrinkageTrees model

Description

Re-runs the MCMC sampler on new covariate data using the stored training data and hyperparameters, returning posterior mean predictions and credible interval bounds.

Usage

```
## S3 method for class 'ShrinkageTrees'
predict(object, newdata, level = 0.95, ...)
```

Arguments

`object`	A fitted `ShrinkageTrees` model object.
`newdata`	A matrix (or object coercible to one) of new covariates with the same number of columns as the training data.
`level`	Credible interval width. Default `0.95`.
`...`	Currently unused.

Value

A `ShrinkageTreesPrediction` object with elements:

mean Posterior mean predictions (length `nrow(newdata)`).

lower Lower credible interval bound.

upper Upper credible interval bound.

n Number of test observations.

level Credible level used.

outcome_type Outcome type inherited from the fitted model.

timescale Timescale inherited from the fitted model (survival only).

predictions_sample (Survival only) `N_post` x `n` matrix of posterior predictive draws on the original scale.

sigma (Survival only) Posterior draws of sigma on the log-time scale (length `N_post`).

See Also

`HorseTrees`, `ShrinkageTrees`, `print.ShrinkageTreesPrediction`, `summary.ShrinkageTreesPrediction`, `plot.ShrinkageTreesPrediction`

`print.CausalShrinkageForest` *Print a CausalShrinkageForest model*

Description

Displays a concise summary of a fitted `CausalShrinkageForest` model with per-forest columns for priors, tree counts, feature counts, and MCMC acceptance ratios.

Usage

```
## S3 method for class 'CausalShrinkageForest'
print(x, ...)
```

Arguments

`x`	A fitted `CausalShrinkageForest` model object.
`...`	Currently unused.

Value

Invisibly returns `x`.

See Also

`summary.CausalShrinkageForest`, `CausalHorseForest`, `CausalShrinkageForest`

`print.CausalShrinkageForestPrediction` *Print a CausalShrinkageForestPrediction object*

Description

Displays a formatted table of posterior mean predictions and credible interval bounds for the first `n_head` observations, with separate sections for the prognostic function $\mu(X)$, the CATE $\tau(X)$, and the total outcome $\mu(X) + \tau(X)$.

Usage

```
## S3 method for class 'CausalShrinkageForestPrediction'
print(x, n_head = 6, digits = 3, ...)
```

Arguments

`x`	A `CausalShrinkageForestPrediction` object.
`n_head`	Number of observations to display per section. Default `6`.
`digits`	Number of decimal places. Default `3`.
`...`	Currently unused.

Value

Invisibly returns `x`.

See Also

`predict.CausalShrinkageForest`, `summary.CausalShrinkageForestPrediction`

`print.PosteriorProjection` *Print a posterior projection*

Description

Laid out like `summary.lm()`: the residual distribution, the coefficient table, and the residual scale and R^2. Uncertainty is reported as posterior credible intervals throughout.

Usage

```
## S3 method for class 'PosteriorProjection'
print(x, digits = 4, n_max = 25, ...)
```

Arguments

x	A PosteriorProjection object.
digits	Significant digits for the coefficient table.
n_max	Maximum number of coefficients to display.
...	Currently unused.

Value

Invisibly returns x.

---

print.ShrinkageTrees *Print a ShrinkageTrees model*

Description

Displays a concise summary of a fitted ShrinkageTrees model, including outcome type, prior, MCMC settings, acceptance ratio, and posterior mean sigma.

Usage

```
## S3 method for class 'ShrinkageTrees'
print(x, ...)
```

Arguments

x	A fitted ShrinkageTrees model object.
...	Currently unused.

Value

Invisibly returns x.

See Also

summary.ShrinkageTrees, HorseTrees, ShrinkageTrees

print.ShrinkageTreesPrediction

Print a ShrinkageTreesPrediction object

Description

Displays a formatted table of posterior mean predictions and credible interval bounds for the first `n_head` observations.

Usage

```
## S3 method for class 'ShrinkageTreesPrediction'
print(x, n_head = 6, digits = 3, ...)
```

Arguments

`x`	A `ShrinkageTreesPrediction` object.
`n_head`	Number of observations to display. Default `6`.
`digits`	Number of decimal places. Default `3`.
`...`	Currently unused.

Value

Invisibly returns `x`.

See Also

`predict.ShrinkageTrees`, `summary.ShrinkageTreesPrediction`

print.summary.CausalShrinkageForest

Print a CausalShrinkageForest model summary

Description

Displays a detailed summary of a `CausalShrinkageForest` model, including model specification, treatment effect estimates, prognostic function, posterior sigma, variable importance for each forest, and MCMC diagnostics.

Usage

```
## S3 method for class 'summary.CausalShrinkageForest'
print(x, n_vi = 10, ...)
```

Arguments

x	A summary.CausalShrinkageForest object.
n_vi	Maximum number of variables to display per variable importance table. Default 10.
...	Currently unused.

Value

Invisibly returns x.

See Also

summary.CausalShrinkageForest

---

print.summary.CausalShrinkageForestPrediction

Print a CausalShrinkageForestPrediction summary

Description

Displays distributional summaries (min, Q1, median, max) of the posterior mean predictions and credible interval bounds, separately for the prognostic function, CATE, and total outcome.

Usage

```
## S3 method for class 'summary.CausalShrinkageForestPrediction'
print(x, digits = 3, ...)
```

Arguments

x	A summary.CausalShrinkageForestPrediction object.
digits	Number of decimal places. Default 3.
...	Currently unused.

Value

Invisibly returns x.

See Also

summary.CausalShrinkageForestPrediction

print.summary.ShrinkageTrees

Print a ShrinkageTrees model summary

Description

Displays a detailed summary of a `ShrinkageTrees` model, including model specification, posterior sigma, prediction summaries, variable importance, and MCMC diagnostics.

Usage

```
## S3 method for class 'summary.ShrinkageTrees'
print(x, n_vi = 10, ...)
```

Arguments

`x`	A `summary.ShrinkageTrees` object.
`n_vi`	Maximum number of variables to display in the variable importance table. Default `10`.
`...`	Currently unused.

Value

Invisibly returns `x`.

See Also

`summary.ShrinkageTrees`

print.summary.ShrinkageTreesPrediction

Print a ShrinkageTreesPrediction summary

Description

Displays distributional summaries (min, Q1, median, max) of the posterior mean predictions and credible interval bounds.

Usage

```
## S3 method for class 'summary.ShrinkageTreesPrediction'
print(x, digits = 3, ...)
```

Arguments

x	A summary.ShrinkageTreesPrediction object.
digits	Number of decimal places. Default 3.
...	Currently unused.

Value

Invisibly returns x.

See Also

summary.ShrinkageTreesPrediction

ShrinkageTrees *General Shrinkage Regression Trees (ShrinkageTrees)*

Description

Fits a Bayesian Shrinkage Tree model with flexible global-local priors on the step heights. This function generalizes HorseTrees by allowing different global-local shrinkage priors on the step heights. Supports continuous, binary, right-censored, and interval-censored outcomes.

Usage

```
ShrinkageTrees(
  y = NULL,
  status = NULL,
  X_train,
  X_test = NULL,
  left_time = NULL,
  right_time = NULL,
  outcome_type = "continuous",
  timescale = "time",
  number_of_trees = 200,
  prior_type = "horseshoe",
  local_hp = NULL,
  global_hp = NULL,
  a_dirichlet = 0.5,
  b_dirichlet = 1,
  rho_dirichlet = NULL,
  power = 2,
  base = 0.95,
  p_grow = 0.4,
  p_prune = 0.4,
  nu = 3,
  q = 0.9,
```

```
  sigma = NULL,
  N_post = 1000,
  N_burn = 1000,
  delayed_proposal = 5,
  store_posterior_sample = TRUE,
  n_chains = 1,
  verbose = TRUE
)
```

Arguments

y	Outcome vector. Numeric. Can represent continuous outcomes, binary outcomes (0/1), or follow-up times for survival data. Set to NULL when using outcome_type = "interval-censored", as values are derived from left_time and right_time.
status	Optional censoring indicator vector (1 = event occurred, 0 = censored). Required if outcome_type = "right-censored". For interval-censored outcomes, this is derived automatically from left_time and right_time.
X_train	Covariate matrix for training. Each row corresponds to an observation, and each column to a covariate.
X_test	Optional covariate matrix for test data. If NULL, defaults to the mean of the training covariates.
left_time	Optional numeric vector of left (lower) time boundaries. Required when outcome_type = "interval-censored". Exact events have left_time == right_time; right-censored observations have right_time = Inf; interval-censored observations have finite left_time < right_time.
right_time	Optional numeric vector of right (upper) time boundaries. Required when outcome_type = "interval-censored". Use Inf for right-censored observations.
outcome_type	Type of outcome. One of "continuous", "binary", "right-censored", or "interval-censored".
timescale	Indicates the scale of follow-up times. Options are "time" (nonnegative follow-up times, will be log-transformed internally) or "log" (already log-transformed). Used when outcome_type is "right-censored" or "interval-censored".
number_of_trees	Number of trees in the ensemble. Default is 200.
prior_type	Type of prior on the step heights. Options include "horseshoe", "horseshoe_fw", "half-cauchy", "standard" and "dirichlet".
local_hp	Local hyperparameter controlling shrinkage on individual step heights. Required for prior_type = "standard". For prior_type = "horseshoe" this may be left NULL, in which case it defaults to k / sqrt(number_of_trees) with k = 1.0. Values of k between roughly 0.5 and 1.5 worked well across a wide range of simulation settings; k is also a natural target for cross-validation. See the "Choosing k" section of the package vignette.

`global_hp` Global hyperparameter controlling overall shrinkage. Ignored for `prior_type = "half-cauchy"` or `"standard"`.

For `prior_type = "horseshoe"` this may be left `NULL`, in which case it takes the same default as `local_hp`. Only their product identifies the prior, so setting the two equal loses nothing.

Changed in version 2.1.0. Both arguments previously had to be supplied for horseshoe priors, and the value used throughout the documentation was `0.1 / sqrt(number_of_trees)`. That value was calibrated against a defect which made `global_hp` inert (see `NEWS.md`); with the defect corrected it shrinks far too aggressively.

`a_dirichlet` First shape parameter of the Beta prior used in the Dirichlet-Sparse splitting rule. Together with `b_dirichlet`, it controls the expected sparsity level. Only when `prior_type = "dirichlet"`.

`b_dirichlet` Second shape parameter of the Beta prior for the sparsity level. Larger values shrink splitting probabilities more strongly toward uniform sparsity. Only when `prior_type = "dirichlet"`.

`rho_dirichlet` Sparsity hyperparameter. If left NULL, it defaults to the number of covariates. Only when `prior_type = "dirichlet"`.

`power` Power parameter for the tree structure prior. Default is 2.0.

`base` Base parameter for the tree structure prior. Default is 0.95.

`p_grow` Probability of proposing a grow move. Default is 0.4.

`p_prune` Probability of proposing a prune move. Default is 0.4.

`nu` Degrees of freedom for the error distribution prior. Default is 3.

`q` Quantile hyperparameter for the error variance prior. Default is 0.90.

`sigma` Optional known value for error standard deviation. If NULL, estimated from data.

`N_post` Number of posterior samples to store. Default is 1000.

`N_burn` Number of burn-in iterations. Default is 1000.

`delayed_proposal`
Number of delayed iterations before proposal. Only for reversible updates. Default is 5.

`store_posterior_sample`
Logical; whether to store posterior samples for each iteration. Default is TRUE.

`n_chains` Number of independent MCMC chains to run. Default is `1` (standard single-chain behaviour). When `n_chains > 1` the chains are run in parallel via `parallel::mclapply` and their posterior samples are pooled into a single `ShrinkageTrees` object, so all existing `print`, `summary`, and `predict` methods work without modification. On Windows, `mclapply` falls back to sequential execution.

`verbose` Logical; whether to print verbose output. Default is TRUE.

Details

This function is a flexible generalization of `HorseTrees`. Instead of using a single Horseshoe prior, it allows specifying different global–local shrinkage configurations for the tree step heights. Further

methodological details on the Horseshoe Forest framework can be found in Jacobs, van Wieringen & van der Pas (2026).

The `horseshoe` prior is the fully Bayesian global-local shrinkage prior, where both the global and local shrinkage parameters are assigned half-Cauchy distributions with scale hyperparameters `global_hp` and `local_hp`, respectively. The global shrinkage parameter is defined separately for each tree, allowing adaptive regularization per tree.

The `horseshoe_fw` prior (forest-wide horseshoe) is similar to `horseshoe`, except that the global shrinkage parameter is shared across all trees in the forest simultaneously.

The `half-cauchy` prior considers only local shrinkage and does not include a global shrinkage component. It places a half-Cauchy prior on each local shrinkage parameter with scale hyperparameter `local_hp`.

The `standard` prior (Chipman, George & McCulloch, 2010) corresponds to the classical BART specification, where step heights are given a normal prior with variance scaled by the number of trees. This prior does not introduce a global shrinkage parameter and does not use global–local structure.

The `dirichlet` prior implements the Dirichlet–Sparse splitting rule of Linero (2018), in which splitting probabilities follow a Dirichlet prior whose concentration is controlled by a Beta sparsity parameter (`a_dirichlet`, `b_dirichlet`) and an expected sparsity level `rho_dirichlet`.

Value

An S3 object of class "ShrinkageTrees" containing the following elements:

train_predictions Vector of posterior mean predictions on the training data.

test_predictions Vector of posterior mean predictions on the test data (or on mean covariate vector if `X_test` not provided).

sigma Vector of posterior samples of the error variance.

acceptance_ratio Average acceptance ratio across trees during sampling.

train_predictions_sample Matrix of posterior samples of training predictions (iterations in rows, observations in columns). Present only if `store_posterior_sample = TRUE`.

test_predictions_sample Matrix of posterior samples of test predictions. Present only if `store_posterior_sample = TRUE`.

train_probabilities Vector of posterior mean probabilities on the training data (only for `outcome_type = "binary"`).

test_probabilities Vector of posterior mean probabilities on the test data (only for `outcome_type = "binary"`).

train_probabilities_sample Matrix of posterior samples of training probabilities (only for `outcome_type = "binary"` and if `store_posterior_sample = TRUE`).

test_probabilities_sample Matrix of posterior samples of test probabilities (only for `outcome_type = "binary"` and if `store_posterior_sample = TRUE`).

References

Jacobs, T., van Wieringen, W. N., & van der Pas, S. L. (2026). *Horseshoe Forests for High-Dimensional Causal Survival Analysis.* Bayesian Analysis, advance publication, 1-30. https://doi.org/10.1214/26-BA1603

Chipman, H. A., George, E. I., & McCulloch, R. E. (2010). *BART: Bayesian additive regression trees.* Annals of Applied Statistics.

Linero, A. R. (2018). *Bayesian regression trees for high-dimensional prediction and variable selection.* Journal of the American Statistical Association.

See Also

Model family: `HorseTrees` (horseshoe prior), `CausalHorseForest` (causal inference), `CausalShrinkageForest` (causal, flexible prior).

Survival wrappers: `SurvivalBART`, `SurvivalDART`.

S3 methods: `print.ShrinkageTrees`, `summary.ShrinkageTrees`, `predict.ShrinkageTrees`, `plot.ShrinkageTrees`.

Examples

```
# Example: Continuous outcome with ShrinkageTrees, two priors
n <- 50
p <- 3
X <- matrix(runif(n * p), ncol = p)
X_test <- matrix(runif(n * p), ncol = p)
y <- X[, 1] + rnorm(n)

# Fit ShrinkageTrees with standard horseshoe prior
fit_horseshoe <- ShrinkageTrees(y = y,
                                X_train = X,
                                X_test = X_test,
                                outcome_type = "continuous",
                                number_of_trees = 5,
                                prior_type = "horseshoe",
                                local_hp = 1.0 / sqrt(5),
                                global_hp = 1.0 / sqrt(5),
                                N_post = 10,
                                N_burn = 5,
                                store_posterior_sample = TRUE,
                                verbose = FALSE)

# Fit ShrinkageTrees with half-Cauchy prior
fit_halfcauchy <- ShrinkageTrees(y = y,
                                 X_train = X,
                                 X_test = X_test,
                                 outcome_type = "continuous",
                                 number_of_trees = 5,
                                 prior_type = "half-cauchy",
                                 local_hp = 1 / sqrt(5),
                                 N_post = 10,
                                 N_burn = 5,
```

```
                                store_posterior_sample = TRUE,
                                verbose = FALSE)

# Posterior mean predictions
pred_horseshoe <- colMeans(fit_horseshoe$train_predictions_sample)
pred_halfcauchy <- colMeans(fit_halfcauchy$train_predictions_sample)

# Posteriors of the mean (global average prediction)
post_mean_horseshoe <- rowMeans(fit_horseshoe$train_predictions_sample)
post_mean_halfcauchy <- rowMeans(fit_halfcauchy$train_predictions_sample)

# Posterior mean prediction averages
mean_pred_horseshoe <- mean(post_mean_horseshoe)
mean_pred_halfcauchy <- mean(post_mean_halfcauchy)

# Example: Interval-censored survival outcome
n <- 50; p <- 3
X_ic <- matrix(rnorm(n * p), ncol = p)
true_t <- rexp(n, rate = exp(-X_ic[, 1]))
left_t  <- true_t * runif(n, 0.5, 1)
right_t <- true_t * runif(n, 1, 1.5)
exact <- sample(n, 15)
left_t[exact] <- true_t[exact]; right_t[exact] <- true_t[exact]
rc <- sample(setdiff(seq_len(n), exact), 10); right_t[rc] <- Inf

fit_ic <- ShrinkageTrees(left_time = left_t, right_time = right_t,
                         X_train = X_ic,
                         outcome_type = "interval-censored",
                         prior_type = "horseshoe",
                         local_hp = 1.0 / sqrt(5),
                         global_hp = 1.0 / sqrt(5),
                         number_of_trees = 5,
                         N_post = 10, N_burn = 5,
                         verbose = FALSE)
```

`summary.CausalShrinkageForest` *Summarise a CausalShrinkageForest model*

Description

Returns an inspectable list with treatment effect estimates, prognostic function summaries, posterior sigma, variable importance for each forest, and MCMC diagnostics.

Usage

```
## S3 method for class 'CausalShrinkageForest'
summary(object, bayesian_bootstrap = TRUE, ...)
```

Arguments

`object`	A fitted `CausalShrinkageForest` model object.
`bayesian_bootstrap`	Logical; if `TRUE` (default), the ATE posterior is computed by reweighting the per-iteration CATE vector with Dirichlet(1, ..., 1) weights (Bayesian bootstrap). This gives a draw from the posterior of the *population* ATE (PATE), with a credible interval that accounts for uncertainty in the covariate distribution. If `FALSE`, equal $1/n$ weights are used, giving the mixed ATE (MATE), which conditions on the observed covariates. Ignored when posterior samples are not stored.
`...`	Currently unused.

Value

A `summary.CausalShrinkageForest` object with elements:

call The original model call.

outcome_type Outcome type.

timescale Timescale for survival outcomes.

prior Prior specification for control and treatment forests.

mcmc MCMC settings.

data_info Training and test data dimensions.

treatment_effect List with `ate` (posterior mean ATE), `cate_sd` (SD of individual CATEs), and optionally `ate_lower`, `ate_upper` (95 percent credible interval; requires `store_posterior_sample = TRUE`) and `bayesian_bootstrap` (the flag used to produce the CI).

prognostic Summary of the prognostic function (mean, SD, range).

sigma Named vector with posterior mean, SD, and 95 percent credible interval of sigma (if estimated).

variable_importance_control Variable importance for the control forest (if available).

variable_importance_treat Variable importance for the treatment forest (if available).

acceptance_ratios List with acceptance ratios for each forest.

See Also

`print.summary.CausalShrinkageForest`, `CausalHorseForest`, `CausalShrinkageForest`

summary.CausalShrinkageForestPrediction

Summarise a CausalShrinkageForestPrediction object

Description

Returns distributional summaries (min, Q1, median, max) of the posterior mean predictions and credible interval bounds across all test observations, separately for the prognostic function, CATE, and total outcome.

Usage

```
## S3 method for class 'CausalShrinkageForestPrediction'
summary(object, ...)
```

Arguments

object	A CausalShrinkageForestPrediction object.
...	Currently unused.

Value

A summary.CausalShrinkageForestPrediction object.

See Also

predict.CausalShrinkageForest, print.summary.CausalShrinkageForestPrediction

summary.ShrinkageTrees

Summarise a ShrinkageTrees model

Description

Returns an inspectable list with posterior sigma summaries, prediction summaries, variable importance (posterior inclusion probabilities), and MCMC diagnostics.

Usage

```
## S3 method for class 'ShrinkageTrees'
summary(object, ...)
```

Arguments

object	A fitted ShrinkageTrees model object.
...	Currently unused.

Value

A `summary.ShrinkageTrees` object with elements:

call The original model call.

outcome_type Outcome type (`"continuous"`, `"binary"`, `"right-censored"`, or `"interval-censored"`).

timescale Timescale for survival outcomes (`"time"` or `"log"`).

prior Prior specification.

mcmc MCMC settings (trees, draws, burn-in).

data_info Training and test data dimensions.

sigma Named vector with posterior mean, SD, and 95 percent credible interval of sigma (continuous and survival outcomes only).

predictions List with `train` (and optionally `test`) prediction summaries (mean, SD, range).

variable_importance Named vector of posterior inclusion probabilities, sorted decreasingly (if available).

acceptance_ratio MCMC acceptance ratio vector.

diagnostics (When **coda** is installed) A list with `ess` (effective sample size) and, for multi-chain fits, `rhat` (Gelman–Rubin $\hat{R}$).

See Also

`print.summary.ShrinkageTrees`, `as.mcmc.list.ShrinkageTrees`, `HorseTrees`, `ShrinkageTrees`

---

summary.ShrinkageTreesPrediction

Summarise a ShrinkageTreesPrediction object

Description

Returns distributional summaries (min, Q1, median, max) of the posterior mean predictions and credible interval bounds across all observations.

Usage

```
## S3 method for class 'ShrinkageTreesPrediction'
summary(object, ...)
```

Arguments

`object`	A `ShrinkageTreesPrediction` object.
`...`	Currently unused.

Value

A `summary.ShrinkageTreesPrediction` object.

See Also

predict.ShrinkageTrees, print.summary.ShrinkageTreesPrediction

---

SurvivalBART *SurvivalBART*

Description

Fits an Accelerated Failure Time (AFT) model using the classical Bayesian Additive Regression Trees (BART) prior: $\log(Y) = f(x) + \varepsilon$. Supports both right-censored and interval-censored survival outcomes.

Usage

```
SurvivalBART(
  time = NULL,
  status = NULL,
  X_train,
  X_test = NULL,
  timescale = "time",
  number_of_trees = 200,
  k = 2,
  N_post = 1000,
  N_burn = 1000,
  store_posterior_sample = TRUE,
  verbose = TRUE,
  left_time = NULL,
  right_time = NULL,
  ...
)
```

Arguments

time	Outcome vector of (non-negative) survival times. Required for right-censored outcomes; set to NULL when using interval censoring.
status	Event indicator (1 = event, 0 = censored). Required for right-censored outcomes; derived automatically for interval censoring.
X_train	Design matrix for training data.
X_test	Optional test matrix. If NULL, predictions are computed at the column means of X_train.
timescale	Either "time" (log-transform internally) or "log" (already log-transformed).
number_of_trees	Number of trees in the ensemble. Default is 200.
k	Scaling constant used to calibrate the prior variance of the step heights.
N_post	Number of posterior samples to store.

`N_burn`	Number of burn-in iterations.
`store_posterior_sample`	Logical; if `TRUE` (default), store the full $N_{\text{post}} \times n$ matrix of posterior predictions. Required for `predict()`, `plot(type = "survival")` with full posterior credible bands, and custom posterior analyses. Set to `FALSE` to save memory when only posterior means are needed.
`verbose`	Logical; print sampling progress.
`left_time`	Optional numeric vector of left (lower) time boundaries for interval-censored data. Exact events have `left_time == right_time`; right-censored observations have `right_time = Inf`; interval-censored observations have finite `left_time < right_time`. When provided together with `right_time`, the model is fitted with `outcome_type = "interval-censored"` and `time`/`status` are ignored.
`right_time`	Optional numeric vector of right (upper) time boundaries. Use `Inf` for right-censored observations.
`...`	Additional arguments passed to `ShrinkageTrees` to override default hyperparameters.

Details

This function provides a survival-specific interface for classical BART under an AFT formulation for right-censored or interval-censored outcomes.

For right-censored data, supply `time` and `status`. For interval-censored data, supply `left_time` and `right_time` instead; event indicators are derived internally following the `survival::Surv(type = "interval2")` convention.

Structural regularisation is induced through the standard Gaussian leaf prior and tree depth prior of Chipman, George & McCulloch (2010).

Users requiring alternative shrinkage priors (e.g., Horseshoe or Dirichlet splitting priors) should use `ShrinkageTrees` directly.

Value

An object of class `"ShrinkageTrees"` fitted under a classical BART prior within an AFT formulation.

See `ShrinkageTrees` for a full description of returned components

References

Chipman, H. A., George, E. I., & McCulloch, R. E. (2010). Bayesian Additive Regression Trees. Annals of Applied Statistics.

See Also

Related models: `SurvivalDART` (Dirichlet sparsity), `HorseTrees` (horseshoe prior), `ShrinkageTrees` (general shrinkage priors).

S3 methods: `print.ShrinkageTrees`, `summary.ShrinkageTrees`, `predict.ShrinkageTrees`, `plot.ShrinkageTrees`.

Examples

```
set.seed(1)
n <- 30; p <- 5
X <- matrix(rnorm(n * p), ncol = p)
time <- rexp(n, rate = exp(0.5 * X[, 1]))
status <- rbinom(n, 1, 0.7)

fit <- SurvivalBART(time = time, status = status, X_train = X,
                    number_of_trees = 5, N_post = 50, N_burn = 25,
                    verbose = FALSE)

# S3 methods
print(fit)
smry <- summary(fit)

# Posterior predictions on new data
X_new <- matrix(rnorm(10 * p), ncol = p)
pred <- predict(fit, newdata = X_new)
print(pred)

# Diagnostic plot (requires ggplot2)
if (requireNamespace("ggplot2", quietly = TRUE)) {
  plot(fit, type = "trace")

  # Posterior survival curves for training data
  plot(fit, type = "survival")

  # Posterior predictive survival curves for new data
  plot(pred, type = "survival")
  plot(pred, type = "survival", obs = c(1, 5))
}

# Interval-censored example
set.seed(11)
n <- 30; p <- 5
X <- matrix(rnorm(n * p), ncol = p)
true_t <- rexp(n, rate = exp(0.5 * X[, 1]))
left_t  <- true_t * runif(n, 0.5, 1)
right_t <- true_t * runif(n, 1, 1.5)
exact <- sample(n, 10); left_t[exact] <- true_t[exact]; right_t[exact] <- true_t[exact]
rc <- sample(setdiff(seq_len(n), exact), 5); right_t[rc] <- Inf

fit_ic <- SurvivalBART(left_time = left_t, right_time = right_t,
                       X_train = X, number_of_trees = 5,
                       N_post = 50, N_burn = 25, verbose = FALSE)
```

SurvivalBCF *SurvivalBCF (Bayesian Causal Forest for survival data)*

Description

Fits an Accelerated Failure Time (AFT) version of Bayesian Causal Forest (BCF): $Y = \mu(x) + W\tau(x) + \varepsilon$, where separate forests are used for the prognostic (control) function $\mu(x)$ and the treatment effect function $\tau(x)$.

Usage

```
SurvivalBCF(
  time = NULL,
  status = NULL,
  X_train,
  treatment,
  timescale = "time",
  propensity = NULL,
  treatment_coding = "centered",
  number_of_trees_control = 200,
  number_of_trees_treat = 50,
  power_control = 2,
  base_control = 0.95,
  power_treat = 3,
  base_treat = 0.25,
  N_post = 1000,
  N_burn = 1000,
  store_posterior_sample = TRUE,
  verbose = TRUE,
  left_time = NULL,
  right_time = NULL,
  ...
)
```

Arguments

`time`	Outcome vector of (non-negative) survival times. Required for right-censored outcomes; set to `NULL` when using interval censoring.
`status`	Event indicator (1 = event, 0 = censored). Required for right-censored outcomes; derived automatically for interval censoring.
`X_train`	Design matrix for training data.
`treatment`	Treatment indicator (0/1) for training data.
`timescale`	Either `"time"` (log-transform internally) or `"log"` (already log-transformed).
`propensity`	Optional vector of propensity scores. If provided, it is appended to the control forest design matrix. Required when `treatment_coding = "adaptive"`.
`treatment_coding`	Character string specifying how the treatment indicator enters the model. One of `"centered"` (default, maps to $-1/2$ and $1/2$), `"binary"` (maps to 0 and 1), `"adaptive"` (maps to $z_i - \hat{e}(x_i)$, where $\hat{e}(x_i)$ is the propensity score), or `"invariant"` (parameter-expanded coding with $b_0, b_1 \sim N(0, 1/2)$ estimated within the Gibbs sampler; Hahn et al., 2020, Section 5.2).

number_of_trees_control
: Number of trees in the control forest. Default is 200.

number_of_trees_treat
: Number of trees in the treatment forest. Default is 50.

power_control, base_control
: Tree-structure prior parameters for the control forest.

power_treat, base_treat
: Tree-structure prior parameters for the treatment forest.

N_post
: Number of posterior samples to store.

N_burn
: Number of burn-in iterations.

store_posterior_sample
: Logical; if TRUE (default), store the full $N_{\text{post}} \times n$ matrix of posterior predictions. Required for `predict()`, `plot(type = "survival")` with full posterior credible bands, and custom posterior analyses. Set to FALSE to save memory when only posterior means are needed.

verbose
: Logical; print sampling progress.

left_time
: Optional numeric vector of left (lower) time boundaries for interval-censored data. Exact events have `left_time == right_time`; right-censored observations have `right_time = Inf`; interval-censored observations have finite `left_time < right_time`. When provided together with `right_time`, the model is fitted with `outcome_type = "interval-censored"` and `time`/`status` are ignored.

right_time
: Optional numeric vector of right (upper) time boundaries. Use `Inf` for right-censored observations.

...
: Additional arguments passed to `CausalShrinkageForest` to override default hyperparameters.

Details

This wrapper provides a survival-specific implementation using classical BART-style priors for both forests. Supports both right-censored and interval-censored survival outcomes.

This function implements a simplified AFT-BCF model for right-censored or interval-censored survival outcomes. Structural regularisation is induced through classical BART priors on the tree structure and leaf parameters.

For right-censored data, supply `time` and `status`. For interval-censored data, supply `left_time` and `right_time` instead; event indicators are derived internally following the `survival::Surv(type = "interval2")` convention.

Users requiring alternative shrinkage priors (e.g., Horseshoe or Dirichlet splitting priors) should use `SurvivalShrinkageBCF` or call `CausalShrinkageForest` directly.

Value

An object of class `"CausalShrinkageForest"` corresponding to a survival BCF model under classical BART priors.

See `CausalShrinkageForest` for returned components.

References

Hahn, P. R., Murray, J. S., & Carvalho, C. M. (2020). Bayesian regression tree models for causal inference: Regularization, confounding, and heterogeneous effects. Bayesian Analysis.

See Also

Related models: SurvivalShrinkageBCF (Dirichlet sparsity), CausalHorseForest (horseshoe prior), CausalShrinkageForest (general shrinkage priors).

S3 methods: print.CausalShrinkageForest, summary.CausalShrinkageForest, predict.CausalShrinkageForest, plot.CausalShrinkageForest.

Examples

```
set.seed(3)
n <- 30; p <- 5
X <- matrix(rnorm(n * p), ncol = p)
treatment <- rbinom(n, 1, 0.5)
log_T <- X[, 1] + treatment * (-0.5) + rnorm(n)
time <- exp(log_T)
status <- rbinom(n, 1, 0.7)

fit <- SurvivalBCF(time = time, status = status, X_train = X,
                   treatment = treatment,
                   number_of_trees_control = 5,
                   number_of_trees_treat = 5,
                   N_post = 50, N_burn = 25,
                   verbose = FALSE)

# S3 methods
print(fit)
smry <- summary(fit)

# Posterior ATE
cat("ATE:", round(smry$treatment_effect$ate, 3), "\n")

# Diagnostic and treatment-effect plots (requires ggplot2)
if (requireNamespace("ggplot2", quietly = TRUE)) {
  plot(fit, type = "trace")
  plot(fit, type = "ate")
  plot(fit, type = "cate")
}

# Interval-censored causal example
set.seed(13)
n <- 30; p <- 5
X <- matrix(rnorm(n * p), ncol = p)
treatment <- rbinom(n, 1, 0.5)
true_t <- exp(X[, 1] + treatment * (-0.5) + rnorm(n))
left_t  <- true_t * runif(n, 0.5, 1)
right_t <- true_t * runif(n, 1, 1.5)
exact <- sample(n, 10); left_t[exact] <- true_t[exact]; right_t[exact] <- true_t[exact]
```

```
rc <- sample(setdiff(seq_len(n), exact), 5); right_t[rc] <- Inf

fit_ic <- SurvivalBCF(left_time = left_t, right_time = right_t,
                      X_train = X, treatment = treatment,
                      number_of_trees_control = 5,
                      number_of_trees_treat = 5,
                      N_post = 50, N_burn = 25, verbose = FALSE)
```

SurvivalDART *SurvivalDART*

Description

Fits an Accelerated Failure Time (AFT) model using the Dirichlet splitting prior (DART), which induces structural sparsity through a Beta-Dirichlet hierarchy on splitting probabilities. Supports both right-censored and interval-censored survival outcomes.

Usage

```
SurvivalDART(
  time = NULL,
  status = NULL,
  X_train,
  X_test = NULL,
  timescale = "time",
  number_of_trees = 200,
  a_dirichlet = 0.5,
  b_dirichlet = 1,
  rho_dirichlet = NULL,
  k = 2,
  N_post = 1000,
  N_burn = 1000,
  store_posterior_sample = TRUE,
  verbose = TRUE,
  left_time = NULL,
  right_time = NULL,
  ...
)
```

Arguments

time	Outcome vector of (non-negative) survival times. Required for right-censored outcomes; set to NULL when using interval censoring.
status	Event indicator (1 = event, 0 = censored). Required for right-censored outcomes; derived automatically for interval censoring.
X_train	Design matrix for training data.

X_test	Optional test matrix. If NULL, predictions are computed at the column means of X_train.
timescale	Either "time" (log-transform internally) or "log" (already log-transformed).
number_of_trees	Number of trees in the ensemble. Default is 200.
a_dirichlet, b_dirichlet	Beta hyperparameters controlling sparsity in the Dirichlet splitting rule.
rho_dirichlet	Expected number of active predictors. If NULL, defaults to the number of covariates in X_train.
k	Scaling constant used to calibrate the prior variance of the step heights.
N_post	Number of posterior samples to store.
N_burn	Number of burn-in iterations.
store_posterior_sample	Logical; if TRUE (default), store the full $N_{\text{post}} \times n$ matrix of posterior predictions. Required for predict(), plot(type = "survival") with full posterior credible bands, and custom posterior analyses. Set to FALSE to save memory when only posterior means are needed.
verbose	Logical; print sampling progress.
left_time	Optional numeric vector of left (lower) time boundaries for interval-censored data. Exact events have left_time == right_time; right-censored observations have right_time = Inf; interval-censored observations have finite left_time < right_time. When provided together with right_time, the model is fitted with outcome_type = "interval-censored" and time/status are ignored.
right_time	Optional numeric vector of right (upper) time boundaries. Use Inf for right-censored observations.
...	Additional arguments passed to ShrinkageTrees to override default hyperparameters.

Details

This function provides a survival-specific wrapper for DART under an AFT formulation for right-censored or interval-censored outcomes.

For right-censored data, supply time and status. For interval-censored data, supply left_time and right_time instead; event indicators are derived internally following the survival::Surv(type = "interval2") convention.

Structural regularisation is induced through a Dirichlet prior on splitting probabilities, encouraging sparse feature usage in high-dimensional settings.

Users requiring alternative shrinkage priors on the leaf parameters (e.g., Horseshoe or half-Cauchy priors) should use ShrinkageTrees directly.

Value

An object of class "ShrinkageTrees" fitted under a Dirichlet splitting prior (DART) within an AFT formulation.

See ShrinkageTrees for a full description of returned components.

See Also

Related models: SurvivalBART (standard BART prior), ShrinkageTrees (general shrinkage priors).

S3 methods: print.ShrinkageTrees, summary.ShrinkageTrees, predict.ShrinkageTrees, plot.ShrinkageTrees.

Examples

```
set.seed(2)
n <- 30; p <- 5
X <- matrix(rnorm(n * p), ncol = p)
time <- rexp(n, rate = exp(0.5 * X[, 1]))
status <- rbinom(n, 1, 0.7)

fit <- SurvivalDART(time = time, status = status, X_train = X,
                    number_of_trees = 5, N_post = 50, N_burn = 25,
                    verbose = FALSE)

# S3 methods
print(fit)
smry <- summary(fit)

# Posterior predictions on new data
X_new <- matrix(rnorm(10 * p), ncol = p)
pred <- predict(fit, newdata = X_new)
print(pred)

# Variable importance and survival plots (requires ggplot2)
if (requireNamespace("ggplot2", quietly = TRUE)) {
  plot(fit, type = "vi", n_vi = 5)

  # Posterior survival curves for training data
  plot(fit, type = "survival")

  # Posterior predictive survival curves for new data
  plot(pred, type = "survival")
  plot(pred, type = "survival", obs = c(1, 5))
}

# Interval-censored example
set.seed(12)
n <- 30; p <- 5
X <- matrix(rnorm(n * p), ncol = p)
true_t <- rexp(n, rate = exp(0.5 * X[, 1]))
left_t  <- true_t * runif(n, 0.5, 1)
right_t <- true_t * runif(n, 1, 1.5)
exact <- sample(n, 10); left_t[exact] <- true_t[exact]; right_t[exact] <- true_t[exact]
rc <- sample(setdiff(seq_len(n), exact), 5); right_t[rc] <- Inf

fit_ic <- SurvivalDART(left_time = left_t, right_time = right_t,
                       X_train = X, number_of_trees = 5,
```

```
                    N_post = 50, N_burn = 25, verbose = FALSE)
```

`SurvivalShrinkageBCF` *SurvivalShrinkageBCF (Shrinkage Bayesian Causal Forest for survival data)*

Description

Fits a survival version of a Bayesian Causal Forest (BCF) under an accelerated failure time (AFT) model, combining Dirichlet splitting priors with global-local shrinkage. Supports both right-censored and interval-censored survival outcomes.

Usage

```
SurvivalShrinkageBCF(
  time = NULL,
  status = NULL,
  X_train,
  treatment,
  timescale = "time",
  propensity = NULL,
  treatment_coding = "centered",
  a_dir = 0.5,
  b_dir = 1,
  number_of_trees_control = 200,
  number_of_trees_treat = 50,
  power_control = 2,
  base_control = 0.95,
  power_treat = 3,
  base_treat = 0.25,
  N_post = 1000,
  N_burn = 1000,
  store_posterior_sample = TRUE,
  verbose = TRUE,
  left_time = NULL,
  right_time = NULL,
  ...
)
```

Arguments

`time`	Outcome vector of (non-negative) survival times. Required for right-censored outcomes; set to `NULL` when using interval censoring.
`status`	Event indicator (1 = event, 0 = censored). Required for right-censored outcomes; derived automatically for interval censoring.
`X_train`	Design matrix for training data.

`treatment` Treatment indicator (0/1) for training data.

`timescale` Either `"time"` (log-transform internally) or `"log"` (already log-transformed).

`propensity` Optional vector of propensity scores. If provided, it is appended to the control forest design matrix. Required when `treatment_coding = "adaptive"`.

`treatment_coding` Character string specifying how the treatment indicator enters the model. One of `"centered"` (default, maps to $-1/2$ and $1/2$), `"binary"` (maps to 0 and 1), `"adaptive"` (maps to $z_i - \hat{e}(x_i)$, where $\hat{e}(x_i)$ is the propensity score), or `"invariant"` (parameter-expanded coding with $b_0, b_1 \sim N(0, 1/2)$ estimated within the Gibbs sampler; Hahn et al., 2020, Section 5.2).

`a_dir` First shape parameter of the Beta prior controlling the sparsity level in the Dirichlet splitting rule.

`b_dir` Second shape parameter of the Beta prior controlling the sparsity level in the Dirichlet splitting rule.

`number_of_trees_control` Number of trees in the control forest. Default is 200.

`number_of_trees_treat` Number of trees in the treatment forest. Default is 50.

`power_control, base_control` Tree-structure prior parameters for the control forest.

`power_treat, base_treat` Tree-structure prior parameters for the treatment forest.

`N_post` Number of posterior samples to store.

`N_burn` Number of burn-in iterations.

`store_posterior_sample` Logical; if `TRUE` (default), store the full $N_{\text{post}} \times n$ matrix of posterior predictions. Required for `predict()`, `plot(type = "survival")` with full posterior credible bands, and custom posterior analyses. Set to `FALSE` to save memory when only posterior means are needed.

`verbose` Logical; print sampling progress.

`left_time` Optional numeric vector of left (lower) time boundaries for interval-censored data. Exact events have `left_time == right_time`; right-censored observations have `right_time = Inf`; interval-censored observations have finite `left_time < right_time`. When provided together with `right_time`, the model is fitted with `outcome_type = "interval-censored"` and `time`/`status` are ignored.

`right_time` Optional numeric vector of right (upper) time boundaries. Use `Inf` for right-censored observations.

`...` Additional arguments passed to `CausalShrinkageForest` to override default hyperparameters.

Details

This wrapper extends `SurvivalBCF` by incorporating Dirichlet sparsity in both the prognostic (control) and treatment forests, while applying additional shrinkage to the control forest via a half-Cauchy prior.

The SurvivalShrinkageBCF model decomposes the outcome as

$$\log T = \mu(x) + a \cdot \tau(x) + \varepsilon,$$

where $\mu(x)$ represents the prognostic (control) component and $\tau(x)$ the heterogeneous treatment effect.

In contrast to `SurvivalBCF`, this function:

- Applies a Dirichlet splitting prior to both forests, inducing structural sparsity in variable selection.
- Combines Dirichlet sparsity with additional half-Cauchy shrinkage in the control forest.

The Dirichlet prior follows the sparse splitting framework of Linero (2018), where splitting probabilities are governed by a Beta-Dirichlet hierarchy. The sparsity level is controlled by `a_dir` and `b_dir`.

Survival outcomes are modeled using an AFT formulation with censoring handled via data augmentation. Both right-censored and interval-censored data are supported. For interval-censored data, supply `left_time` and `right_time` instead of `time` and `status`.

Value

An object of class `"CausalShrinkageForest"` fitted with Dirichlet splitting priors and additional shrinkage.

References

Caron, A., Baio, G., & Manolopoulou, I. (2022). Shrinkage Bayesian Causal Forests for Heterogeneous Treatment Effects Estimation. Journal of Computational and Graphical Statistics, 31(4), 1202–1214. https://doi.org/10.1080/10618600.2022.2067549

See Also

Related models: `SurvivalBCF` (standard BART priors), `CausalShrinkageForest` (general shrinkage priors), `CausalHorseForest` (horseshoe prior).

S3 methods: `print.CausalShrinkageForest`, `summary.CausalShrinkageForest`, `predict.CausalShrinkageForest`, `plot.CausalShrinkageForest`.

Examples

```
set.seed(4)
n <- 30; p <- 5
X <- matrix(rnorm(n * p), ncol = p)
treatment <- rbinom(n, 1, 0.5)
log_T <- X[, 1] + treatment * (-0.5) + rnorm(n)
time <- exp(log_T)
status <- rbinom(n, 1, 0.7)

fit <- SurvivalShrinkageBCF(time = time, status = status, X_train = X,
                            treatment = treatment,
                            number_of_trees_control = 5,
```

```
                              number_of_trees_treat = 5,
                              N_post = 50, N_burn = 25,
                              verbose = FALSE)

# S3 methods
print(fit)
smry <- summary(fit)

# Posterior ATE with 95% credible interval
cat("ATE:", round(smry$treatment_effect$ate, 3), "\n")

# Diagnostic and treatment-effect plots (requires ggplot2)
if (requireNamespace("ggplot2", quietly = TRUE)) {
  plot(fit, type = "trace")
  plot(fit, type = "cate")
}

# Interval-censored causal example
set.seed(14)
n <- 30; p <- 5
X <- matrix(rnorm(n * p), ncol = p)
treatment <- rbinom(n, 1, 0.5)
true_t <- exp(X[, 1] + treatment * (-0.5) + rnorm(n))
left_t  <- true_t * runif(n, 0.5, 1)
right_t <- true_t * runif(n, 1, 1.5)
exact <- sample(n, 10); left_t[exact] <- true_t[exact]; right_t[exact] <- true_t[exact]
rc <- sample(setdiff(seq_len(n), exact), 5); right_t[rc] <- Inf

fit_ic <- SurvivalShrinkageBCF(left_time = left_t, right_time = right_t,
                               X_train = X, treatment = treatment,
                               number_of_trees_control = 5,
                               number_of_trees_treat = 5,
                               N_post = 50, N_burn = 25, verbose = FALSE)
```

Index