Package ‘ShrinkageTrees’

July 21, 2025

Type Package

Title Regression Trees with Shrinkage Priors

Version 1.0.0

Date 2025-07-12

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

Description Bayesian regression tree models with shrinkage priors on step heights. Supports continuous, binary, and right-censored (survival) outcomes. Used for high-dimensional prediction and causal inference.

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

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

License MIT + file LICENSE

Depends R (>= 3.5.0)

Imports Rcpp

LinkingTo Rcpp (>= 1.0.11)

Suggests survival, afthd, testthat (>= 3.0.0)

RoxygenNote 7.3.2

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 2025-07-21 09:10:02 UTC

Contents

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.

Usage

```
CausalHorseForest(
  y,
  status = NULL,
  X_train_control,
  X_train_treat,
  treatment_indicator_train,
  X_test_control = NULL,
  X_test_treat = NULL,
  treatment_indicator_test = NULL,
  outcome_type = "continuous",
  timescale = "time",
  number_of_trees = 200,
  k = 0.1,
  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,
  seed = NULL,
  verbose = TRUE
)
```

Arguments

`y` Outcome vector. For survival, represents follow-up times (can be on original or log scale depending on `timescale`).

`status` Optional event indicator vector (1 = event occurred, 0 = censored). Required when `outcome_type = "right-censored"`.

`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.

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

`timescale` For survival outcomes: either `"time"` (original time scale, log-transformed internally) or `"log"` (already log-transformed).

`number_of_trees` Number of trees in each forest. Default is 200.

`k` Horseshoe prior scale hyperparameter. Default is 0.1. Controls global-local shrinkage on step heights.

`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.

seed
: Random seed for reproducibility. Default is NULL.

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 and right-censored survival outcomes.

Value

A list 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

HorseTrees, ShrinkageTrees, CausalShrinkageForest

Examples

```
# Example: Continuous outcome and homogenuous 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,
  seed = 1
)


## 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_treat[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 = 0.1,
  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 and right-censored survival outcomes.

Usage

```
CausalShrinkageForest(
  y,
  status = NULL,
  X_train_control,
  X_train_treat,
  treatment_indicator_train,
  X_test_control = NULL,
  X_test_treat = NULL,
  treatment_indicator_test = 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,
  local_hp_treat,
  global_hp_control = NULL,
  global_hp_treat = NULL,
  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,
  seed = NULL,
  verbose = TRUE
)
```

Arguments

y Outcome vector. Numeric. Represents continuous outcomes or follow-up times.

status Optional event indicator vector (1 = event occurred, 0 = censored). Required when outcome_type = "right-censored".

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.

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

timescale For survival outcomes: either "time" (original scale, log-transformed internally) or "log" (already log-transformed). Default is "time".

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", "horseshoe_fw", "horseshoe_EB", or "half-cauchy". 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 steps (control forest). Required for all prior types.

local_hp_treat Local hyperparameter for treatment forest.

global_hp_control Global hyperparameter for control forest. Required for horseshoe-type priors; ignored for "half-cauchy".

global_hp_treat Global hyperparameter for treatment forest.

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`.

`seed` Random seed for reproducibility. Default is `NULL`.

`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.

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 `horseshoe_EB` prior is an empirical Bayes variant of the `horseshoe` prior. Here, the global shrinkage parameter (τ) is not assigned a prior distribution but instead must be specified directly using `global_hp`, while local shrinkage parameters still follow half-Cauchy priors. Note: τ must be provided by the user; it is not estimated by the software.

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`.

Value

A list 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

`CausalHorseForest`, `ShrinkageTrees`, `HorseTrees`

Examples

```
# Example: Continuous outcome, homogenuous 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 = 0.1/sqrt(5),
                                       local_hp_treat = 0.1/sqrt(5),
                                       global_hp_control = 0.1/sqrt(5),
                                       global_hp_treat = 0.1/sqrt(5),
                                       N_post = 10,
                                       N_burn = 5,
                                       store_posterior_sample = TRUE,
                                       verbose = FALSE,
                                       seed = 1
)

# 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,
                                        seed = 1
)

# 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)
```

censored_info *Compute mean estimate for censored data*

Description

Estimates the mean and standard deviation for right-censored survival data. Uses the afthd package if available (placeholder), else survival, and otherwise falls back to the naive mean among observed events.

Usage

```
censored_info(y, status)
```

Arguments

y	Numeric vector of (log-transformed) survival times.
status	Numeric vector; event indicator (1 = event, 0 = censored).

Value

A list with elements:

`mu` Estimated mean of survival times.

`sd` Estimated standard deviation of survival times.

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, and right-censored (survival) outcomes.

Usage

```
HorseTrees(
  y,
  status = NULL,
  X_train,
  X_test = NULL,
  outcome_type = "continuous",
  timescale = "time",
  number_of_trees = 200,
  k = 0.1,
  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,
  seed = NULL,
  verbose = TRUE
)
```

Arguments

`y` Outcome vector. Numeric. Can represent continuous outcomes, binary outcomes (0/1), or follow-up times for survival data.

`status` Optional censoring indicator vector (1 = event occurred, 0 = censored). Required if `outcome_type = "right-censored"`.

`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.

`outcome_type` Type of outcome. One of `"continuous"`, `"binary"`, or `"right-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"`.

`number_of_trees`
Number of trees in the ensemble. Default is 200.

`k` Horseshoe scale hyperparameter (default 0.1). This parameter controls the overall level of shrinkage by setting the scale for both global and local shrinkage components. The local and global hyperparameters are parameterized as $\alpha = \frac{k}{\sqrt{\text{number\_of\_trees}}}$ to ensure adaptive regularization across trees.

`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.

`seed` Random seed for reproducibility.

`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. Generalized implementation with multiple prior possibilities is given by `ShrinkageTrees`.

Value

A named list 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

`ShrinkageTrees`, `CausalHorseForest`, `CausalShrinkageForest`

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)

# 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)

fit4 <- 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)
```

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.

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.

Usage

```
ShrinkageTrees(
  y,
  status = NULL,
  X_train,
  X_test = NULL,
  outcome_type = "continuous",
  timescale = "time",
  number_of_trees = 200,
  prior_type = "horseshoe",
  local_hp = NULL,
  global_hp = 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,
  seed = NULL,
  verbose = TRUE
)
```

Arguments

`y` Outcome vector. Numeric. Can represent continuous outcomes, binary outcomes (0/1), or follow-up times for survival data.

`status` Optional censoring indicator vector (1 = event occurred, 0 = censored). Required if `outcome_type = "right-censored"`.

`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.

`outcome_type` Type of outcome. One of `"continuous"`, `"binary"`, or `"right-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"`.

`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"`, `"horseshoe_EB"`, and `"half-cauchy"`.

`local_hp` Local hyperparameter controlling shrinkage on individual step heights. Should typically be set smaller than 1 / sqrt(number_of_trees).

`global_hp` Global hyperparameter controlling overall shrinkage. Must be specified for Horseshoe-type priors; ignored for `prior_type = "half-cauchy"`.

`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.

`seed` Random seed for reproducibility.

`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. Currently, four priors have been implemented.

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 horseshoe_EB prior is an empirical Bayes variant of the horseshoe prior. Here, the global shrinkage parameter (τ) is not assigned a prior distribution but instead must be specified directly using global_hp, while local shrinkage parameters still follow half-Cauchy priors. Note: τ must be provided by the user; it is not estimated by the software.

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.

Value

A named list 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

HorseTrees, CausalHorseForest, CausalShrinkageForest

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 = 0.1 / sqrt(5),
                                global_hp = 0.1 / sqrt(5),
                                N_post = 10,
                                N_burn = 5,
                                store_posterior_sample = TRUE,
                                verbose = FALSE,
                                seed = 1)

# 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,
                                 seed = 1)

# 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)
```

Index