Package 'FusionForests'

October 25, 2026

Type Package

Title Bayesian Tree Ensembles for Data Fusion and Causal Inference

Version 1.1.1

Date 2026-10-24

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

Description Bayesian tree ensemble models for data fusion and causal inference. The main model FusionForest() combines data from a randomised controlled trial and an observational study using separate tree forests, allowing for unmeasured confounding in the observational data. Continuous and (interval-)censored survival outcomes are supported. Posterior summaries of treatment effect estimates and interpretable linear projections are provided.

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

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

License MIT + file LICENSE

Depends R (>= 3.5.0)

Imports Rcpp, ShrinkageTrees

LinkingTo Rcpp (>= 1.0.11)

Suggests knitr, rmarkdown, survival, testthat (>= 3.0.0)

VignetteBuilder knitr

Encoding UTF-8

LazyData true

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

RoxygenNote 7.3.3

NeedsCompilation yes

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

Repository CRAN

Date/Publication 2026-10-25 22:00:22 UTC

Contents

breast_fusion 2
breast_fusion_truth 3
FusionForest 4
fusion_estimand 9
fusion_projection 11
print.FusionForest 13
print.summary.FusionForest 14
summary.FusionForest 14

Index **16**

`breast_fusion` *Semi-synthetic RCT and real-world data for data fusion*

Description

A combined data set of a randomised controlled trial (RCT) and a real-world data (RWD) source with a right-censored time-to-event outcome, built to illustrate `FusionForest`. The covariates are real; the outcomes are simulated. The data say nothing about any real treatment.

Usage

```
breast_fusion
```

Format

A data frame with 3668 rows (686 RCT, 2982 RWD) and 13 columns:

id Row identifier, matching `breast_fusion_truth$id`.

source Factor, `"RCT"` or `"RWD"`.

S Source indicator (1 = RCT, 0 = RWD), as used by `FusionForest`.

A Treatment indicator (1 = treated, 0 = control).

time Observed relapse-free time in years.

status Event indicator (1 = event, 0 = censored).

age Age in years.

meno Menopausal status (0 = pre, 1 = post).

size Tumour size (0 = at most 20 mm, 1 = 20–50 mm, 2 = more than 50 mm).

grade Tumour grade (0 = grade 1 or 2, 1 = grade 3).

nodes Number of positive lymph nodes.

pgr Progesterone receptor (fmol/l).

er Oestrogen receptor (fmol/l).

Details

The covariates come from the German Breast Cancer Study Group 2 trial (gbsg; the RCT rows) and the Rotterdam tumour bank (rotterdam; the RWD rows), harmonised to a common set. The treatment A is a generic binary treatment. In the RCT it is randomised with probability 1/2. In the RWD it is the real, non-random treatment assignment of the Rotterdam cohort, so it is confounded. Part of that confounding acts through variables that are not included here (year of surgery and chemotherapy; see breast_fusion_truth).

Event times follow a Weibull accelerated failure time model whose baseline coefficients and scale were estimated from the real GBSG2 outcomes. The heterogeneous treatment effect and the effects of the unobserved variables are made up. Censoring times are drawn from each source's real censoring distribution, so follow-up in the RCT is shorter than in the RWD. The script that builds the data and an example analysis are installed with the package, in `system.file("scripts", package = "FusionForests")`.

Source

Covariates from the **survival** package data sets gbsg and rotterdam; outcomes simulated.

References

Royston, P. and Altman, D. G. (2013). External validation of a Cox prognostic model: principles and methods. *BMC Medical Research Methodology*, 13, 33.

Schumacher, M., Bastert, G., Bojar, H., et al. (1994). Randomized 2 x 2 trial evaluating hormonal treatment and the duration of chemotherapy in node-positive breast cancer patients. *Journal of Clinical Oncology*, 12, 2086–2093.

See Also

breast_fusion_truth for the simulation truth.

Examples

```
table(breast_fusion$source, breast_fusion$A)
```

---

breast_fusion_truth *Simulation truth for breast_fusion*

Description

The true treatment effects, uncensored event times and unobserved confounders behind breast_fusion. Rows are aligned with breast_fusion and share its id. Use these only to check estimates; a real analysis would not have them.

Usage

```
breast_fusion_truth
```

Format

A data frame with 3668 rows and 6 columns:

id Row identifier, matching `breast_fusion$id`.

source Factor, `"RCT"` or `"RWD"`.

tau True conditional average treatment effect on the log-time scale; `exp(tau)` is the acceleration factor.

logT Uncensored log event time (log years).

chemo Chemotherapy (1 = yes, 0 = no); unobserved confounder. All RCT rows are 1.

year Year of surgery; unobserved confounder. Real for the RWD rows, drawn uniformly from 1984–1989 for the RCT rows.

Source

Simulated; see `breast_fusion`.

Examples

```
summary(breast_fusion_truth$tau)
```

FusionForest *Bayesian fusion forest*

Description

Fits the Bayesian fusion forest. The model combines a randomised controlled trial (RCT) with real-world data (RWD) to estimate heterogeneous treatment effects. The RWD may be confounded.

Usage

```
FusionForest(
  y,
  status = NULL,
  observed_left_time = NULL,
  observed_right_time = NULL,
  interval_censoring_indicator = NULL,
  X_train_control,
  X_train_treat,
  treatment_indicator_train,
  source_indicator_train,
  X_test_control = NULL,
  X_test_treat = NULL,
  X_test_deconf = NULL,
  treatment_indicator_test = NULL,
  source_indicator_test = NULL,
  outcome_type = "continuous",
```

```
  timescale = "time",
  number_of_trees_control = 200,
  number_of_trees_treat = 100,
  number_of_trees_deconf = 50,
  number_of_trees_deviation = 50,
  k_control = 1,
  k_treat = 0.5,
  k_deconf = 1,
  k_deviation = 1,
  power_control = 2,
  base_control = 0.95,
  power_deviation = 2,
  base_deviation = 0.95,
  power_treat = 3,
  base_treat = 0.95,
  power_deconf = 3,
  base_deconf = 0.25,
  p_grow = 0.4,
  p_prune = 0.4,
  nu = 3,
  q = 0.9,
  sigma = NULL,
  N_post = 5000,
  N_burn = 5000,
  treatment_coding = c("centered", "binary", "adaptive"),
  propensity_train = NULL,
  propensity_test = NULL,
  error_dist = c("source_hdp_scale", "gaussian", "source_gaussian", "shared_dp",
    "source_dp", "source_dp_scale", "source_hdp"),
  error_truncation_K = 50L,
  error_atom_scale = 0.5,
  error_mass_init = 1,
  store_posterior_sample = FALSE,
  verbose = TRUE
)
```

Arguments

`y`	Numeric vector of outcomes: survival times or a continuous response.
`status`	Event indicator (1 = event, 0 = censored). Required for right-censored outcomes. Use 0 for interval-censored rows.
`observed_left_time`, `observed_right_time`, `interval_censoring_indicator`	Optional vectors for interval censoring, on the same scale as `y`. Each row is one of three types: `status = 1` Event at `y`. The bounds are ignored. `status = 0`, **indicator 0** Right-censored at `observed_right_time`. `status = 0`, **indicator 1** Event in the interval (`observed_left_time`, `observed_right_time`].

Leave all three NULL for plain right-censoring at y. Only used when outcome_type = "right-censored".

X_train_control
: Covariate matrix for the m_0^{sh}, d and c forests. The d and c forests use the RWD rows only.

X_train_treat
: Covariate matrix for the τ forest, with the same rows as X_train_control.

treatment_indicator_train
: Treatment (1 = treated, 0 = control).

source_indicator_train
: Source (1 = RCT, 0 = RWD).

X_test_control, X_test_treat, X_test_deconf
: Optional covariate matrices for prediction. X_test_deconf defaults to X_test_control. Without test matrices, the function predicts at the column means of the training data.

treatment_indicator_test, source_indicator_test
: Treatment and source of the test rows. Both default to 1.

outcome_type
: "continuous" or "right-censored". Use "right-censored" also for interval-censored data.

timescale
: "time" if y holds survival times, "log" if it holds log survival times. With "time", the predictions are returned on the time scale.

number_of_trees_control, number_of_trees_treat, number_of_trees_deconf
: Number of trees J_f in the m_0^{sh}, τ and c forests. Defaults: 200, 100 and 50.

number_of_trees_deviation
: Number of trees in the d forest. Default 50.

k_control, k_treat, k_deconf, k_deviation
: Step-height scales k_f of the m_0^{sh}, τ, c and d forests; see Details. Larger values shrink less. The defaults are 1, except k_treat = 0.5. Set k_deviation = 0 to switch off d, so that both sources share one baseline. Under a source-specific error model the reported d is then the constant $\mu_0 - \mu_1$, the difference between the two source-specific error means absorbed by the centring (see error_dist).

power_control, base_control, power_deviation, base_deviation, power_treat, base_treat, power_deconf, base_deconf
: Tree-structure prior of each forest. A node at depth δ splits with probability $\alpha_f(1+\delta)^{-\beta_f}$, with α_f = base and β_f = power. A larger power or a smaller base gives shallower trees. Defaults: $(\alpha_f, \beta_f) = (0.95, 2)$ for m_0^{sh} and d, $(0.95, 3)$ for τ and $(0.25, 3)$ for c.

p_grow, p_prune
: Probabilities of a grow and a prune proposal.

nu
: Degrees of freedom of the scaled inverse chi-squared prior on σ^2.

q
: Quantile that calibrates the scale λ of this prior.

sigma
: Fixed error standard deviation. If NULL (default), σ is estimated.

N_post, N_burn
: Number of posterior and burn-in iterations.

treatment_coding
: How the treatment enters the model. The coded treatment b_i replaces A in front of τ and c. One of:

"centered" **(default)** $b_i = \pm 1/2$. Both arms inform τ. The baseline forests describe the average of the two arms.

"binary" $b_i = A_i$, as in the paper. The baseline forests describe the control arm.

"adaptive" $b_i = A_i - \pi_i$, the propensity score residual (Hahn et al., 2020). Requires propensity scores.

propensity_train, propensity_test

Propensity scores in (0, 1) for the training and test rows. Only used with treatment_coding = "adaptive".

error_dist Prior for the error distribution. All mixtures are centred, so the error has mean zero in each source. The sampler works with the uncentred mixture, whose full conditionals are all conjugate, and centres every stored draw afterwards: the mixture mean of the RCT, μ_1, is added to m_0^{sh} and the difference $\mu_0 - \mu_1$ to d, while the atoms are shifted to mean zero (Yang et al., 2010; Henderson et al., 2020). τ and c are unaffected. One of:

"gaussian" Normal errors with one scale σ.

"source_gaussian" Normal errors with a separate scale σ_s per source.

"shared_dp" One Dirichlet process mixture of normals for both sources (Henderson et al., 2020).

"source_dp" A separate Dirichlet process mixture per source, with a shared scale σ.

"source_dp_scale" As "source_dp", with a separate scale σ_s per source.

"source_hdp" Hierarchical Dirichlet process mixture. The sources share the mixture components but have their own weights. Shared scale σ.

"source_hdp_scale" **(default)** As "source_hdp", with a separate scale σ_s per source. This is the error model of the paper.

error_truncation_K

Truncation level of the mixture. Default 50. Increase it if the number of occupied components gets close to this level.

error_atom_scale

Prior standard deviation of the mixture locations, on the standardised scale. Default 0.5.

error_mass_init

Starting value of the concentration parameters. They have a $\mathrm{Gamma}(2, 0.1)$ prior. Default 1.

store_posterior_sample

If TRUE, return all posterior draws of each forest. Required by fusion_estimand and fusion_projection.

verbose If TRUE, print progress.

Details

The model decomposes the log survival time as

$$\log T = m_0^{\mathrm{sh}}(X) + (1 - S)\, d(X) + A\, \tau(X) + (1 - S)\, A\, c(X) + \varepsilon,$$

with treatment A (1 = treated, 0 = control) and source S (1 = RCT, 0 = RWD). The error ε has mean zero; see `error_dist`. For a continuous outcome, the outcome replaces $\log T$.

Each component has its own BART forest. The arguments of a forest carry the suffix in brackets.

$m_0^{\mathrm{sh}}(X)$ **(`control`)** Baseline log survival time under control, shared by both sources.

$d(X)$ **(`deviation`)** Difference between the RWD and RCT baselines. The baseline is $m_0(X, S) = m_0^{\mathrm{sh}}(X) + (1 - S)\, d(X)$.

$\tau(X)$ **(`treat`)** Treatment effect on the log-time scale.

$c(X)$ **(`deconf`)** Confounding function: the bias in the RWD treatment contrast.

The step heights of forest $f \in \{\mathrm{sh}, d, \tau, c\}$ have prior $N(0, \sigma_{h,f}^2)$ with

$$\sigma_{h,f} = k_f/(2\sqrt{J_f}),$$

where J_f is the number of trees and k_f the scale. A smaller k_f shrinks the forest more strongly toward zero. All defaults follow the paper.

The model above uses the binary treatment indicator A, as in the paper (`treatment_coding = "binary"`). The default `"centered"` coding replaces A by $A - 1/2$. The functions τ and c keep their meaning. The baseline forests then describe the average of the two arms instead of the control arm.

Value

A `FusionForest` object: a list with

train_predictions, test_predictions Posterior mean of the fitted values.

train_predictions_control, test_predictions_control Posterior mean of $m_0^{\mathrm{sh}}(X)$.

train_predictions_treat, test_predictions_treat Posterior mean of $\tau(X)$.

train_predictions_deconf, test_predictions_deconf Posterior mean of $c(X)$. The training values cover the RWD rows only.

train_predictions_deviation, test_predictions_deviation Posterior mean of $d(X)$. The training values cover the RWD rows only.

sigma Posterior draws of σ on the standardised scale. For the error models with a scale per source, this is a pooled scale. The per-source scales are in `dp_sigma_g`.

acceptance_ratio_control, acceptance_ratio_treat, acceptance_ratio_deconf, acceptance_ratio_deviation Acceptance rates of the tree proposals.

train_predictions_sample_control, ... All posterior draws. Only with `store_posterior_sample = TRUE`.

dp_mix_prop, dp_locations, dp_mass, dp_mu_g Posterior draws of the error distribution, one list element per source in the order RWD, RCT (a single element for `"shared_dp"`). Rows are draws. `dp_mix_prop` holds the weights π_{sk}, `dp_locations` the centred atoms $\theta_k - \mu_s$ and `dp_mu_g` the mean μ_s of the uncentred working mixture, both on the response scale. Only for error models other than `"gaussian"`.

dp_sigma_g Posterior draws of the per-source scales σ_s, same order. Only for the error models with a scale per source.

dp_locations_shared, dp_beta, dp_gamma Posterior draws of the shared atoms θ_k^* (response scale), the top-level weights and the top-level concentration. Only for the hierarchical error models.

meta Settings used by the summary functions.

Examples

```
# Small simulated fusion example combining an RCT and RWD
set.seed(1)
n <- 100
X <- matrix(rnorm(n * 3), n, 3)
s <- rbinom(n, 1, 0.5)  # 1 = RCT, 0 = RWD
a <- rbinom(n, 1, 0.5)  # treatment
y <- X[, 1] + 0.5 * a + rnorm(n)

fit <- FusionForest(
  y = y,
  X_train_control = X,
  X_train_treat = X,
  treatment_indicator_train = a,
  source_indicator_train = s,
  N_post = 50, N_burn = 25,
  verbose = FALSE
)
print(fit)
summary(fit)
```

---

fusion_estimand *Causal survival estimands from a FusionForest fit*

Description

Computes posterior draws of the survival difference or the acceleration factor at the test rows of a FusionForest fit.

Usage

```
fusion_estimand(
  fit,
  estimand = c("SD", "AF"),
  time = NULL,
  target_source = c("rwd", "rct"),
  population_average = FALSE,
  bayesian_bootstrap = TRUE,
  seed = NULL
)
```

Arguments

fit	A FusionForest fit with store_posterior_sample = TRUE.
estimand	"SD" or "AF".
time	Time point for "SD", on the time scale. Ignored for "AF".

`target_source` Target population: `"rwd"` (default) or `"rct"`. Ignored for `"AF"`.

`population_average`
: If `TRUE`, average over the test rows. Default `FALSE`.

`bayesian_bootstrap`
: If `TRUE` (default), the population average uses Bayesian bootstrap weights. If `FALSE`, it uses equal weights.

`seed` Optional seed for the Bayesian bootstrap weights.

Details

The survival function under treatment a in target source s_t is

$$S_a(t \mid x, s_t) = 1 - F_{s_t}\{\log t - \eta_a(x, s_t)\},$$

with F_{s_t} the error distribution in source s_t and

$$\eta_a(x, s_t) = m_0(x, s_t) + a\,\tau(x).$$

The baseline $m_0(x, s_t)$ is the mean log survival time of untreated patients in source s_t. Under the binary coding it equals $m_0^{\mathrm{sh}}(x) + (1 - s_t)\,d(x)$; for the `"centered"` and `"adaptive"` codings the function recovers it from the fit. The treatment effect is the causal effect $\tau(x)$ in both target sources. The confounding function $c(x)$ does not enter: it is the bias of the treated arm in the RWD, not part of the causal effect. The survival difference is thus causal in both target populations.

Two estimands are available:

`"SD"` Survival difference $S_1(t \mid x, s_t) - S_0(t \mid x, s_t)$.

`"AF"` Acceleration factor $\exp\{\tau(x)\}$. It does not depend on the error distribution or the target source.

The population average weights the test rows. By default, the weights are drawn from a Dirichlet(1, ..., 1) distribution at each iteration: the Bayesian bootstrap (Rubin, 1981).

Value

A matrix of posterior draws with one row per iteration and one column per test row. With `population_average = TRUE`, a vector with one draw per iteration. The result carries the attributes `estimand`, `time` and `target_source`.

References

Rubin, D.~B. (1981). The Bayesian bootstrap. *The Annals of Statistics*, 9, 130–134.

Examples

```
# Right-censored fusion fit that stores the posterior draws
set.seed(1)
n <- 100
X <- matrix(rnorm(n * 3), n, 3)
s <- rbinom(n, 1, 0.5)
a <- rbinom(n, 1, 0.5)
```

```
true_time <- exp(1 + X[, 1] + 0.5 * a + 0.3 * rnorm(n))
cens_time <- rexp(n, rate = 1 / (2 * mean(true_time)))
time   <- pmin(true_time, cens_time)
status <- as.integer(true_time <= cens_time)

fit <- FusionForest(
  y = time, status = status,
  X_train_control = X, X_train_treat = X,
  treatment_indicator_train = a, source_indicator_train = s,
  X_test_control = X, X_test_treat = X,
  treatment_indicator_test = a, source_indicator_test = s,
  outcome_type = "right-censored",
  N_post = 50, N_burn = 25,
  store_posterior_sample = TRUE, verbose = FALSE
)

# Posterior draws of the acceleration factor per test row
af <- fusion_estimand(fit, estimand = "AF")
quantile(colMeans(af), c(0.25, 0.5, 0.75))
```

`fusion_projection` *Posterior linear projection of the treatment effect*

Description

Summarises the heterogeneous treatment effect by a linear model in a few covariates. Each posterior draw of the effect is projected onto a linear basis by least squares (Woody, Carvalho and Murray, 2020). The result is a posterior sample of the projection coefficients. No refitting is needed.

Usage

```
fusion_projection(
  fit,
  basis,
  X_eval,
  target = c("cate", "af"),
  af_scale = c("multiplicative", "log"),
  which = c("test", "train"),
  weights = c("uniform", "bayesian_bootstrap"),
  center = TRUE,
  scale = FALSE,
  seed = NULL
)
```

Arguments

`fit`	A `FusionForest` fit with `store_posterior_sample = TRUE`.

basis	One-sided formula for the basis, such as ~ x1 + x2 + x1:x2.
X_eval	Data frame with the covariates of the evaluation rows: the test or training rows of the fit (see which). It must contain the variables in basis.
target	"cate" projects $\tau(x)$ on the log-time scale. "af" projects the acceleration factor; see af_scale.
af_scale	For target = "af" only. "multiplicative" (default) projects $\exp\tau$, so a coefficient is the change in the acceleration factor per unit of the covariate. "log" projects τ, as target = "cate" does.
which	Evaluation rows: "test" (default) or "train". These rows define the target population.
weights	"uniform" (default) or "bayesian_bootstrap".
center, scale	Centre or scale the basis columns other than the intercept. Defaults: TRUE and FALSE.
seed	Optional seed for the Bayesian bootstrap weights.

Details

At iteration r, the coefficients solve

$$\gamma^{(r)} = \arg\min_{\gamma} \sum_i w_i^{(r)} \{\theta^{(r)}(x_i) - \phi(x_i)^\top \gamma\}^2.$$

Here $\theta^{(r)}$ is $\tau^{(r)}$ or $\exp\tau^{(r)}$, depending on target and af_scale, and ϕ is the basis given by basis.

The weights are equal by default. With weights = "bayesian_bootstrap", they are drawn from a Dirichlet(1, ..., 1) distribution at each iteration (Rubin, 1981). This adds the uncertainty about the covariate distribution.

The basis columns other than the intercept are centred on X_eval, and optionally scaled. The centring and scaling constants are returned as attributes.

Value

A matrix of posterior draws of the coefficients, with one row per iteration and one column per basis term. The attributes record target, af_scale, which, weights, the centring and scaling constants, and the formula.

References

Woody, S., Carvalho, C.~M., and Murray, J.~S. (2020). Bayesian inference with posterior projection. *Journal of Computational and Graphical Statistics*, 29(4), 798–808.

Rubin, D.~B. (1981). The Bayesian bootstrap. *The Annals of Statistics*, 9, 130–134.

Examples

```
# Continuous fusion fit that stores the posterior draws
set.seed(1)
n <- 100
```

```
X <- matrix(rnorm(n * 3), n, 3)
colnames(X) <- paste0("x", 1:3)
s <- rbinom(n, 1, 0.5)
a <- rbinom(n, 1, 0.5)
y <- X[, 1] + (0.5 + 0.3 * X[, 2]) * a + rnorm(n)

fit <- FusionForest(
  y = y,
  X_train_control = X, X_train_treat = X,
  treatment_indicator_train = a, source_indicator_train = s,
  X_test_control = X, X_test_treat = X,
  treatment_indicator_test = a, source_indicator_test = s,
  N_post = 50, N_burn = 25,
  store_posterior_sample = TRUE, verbose = FALSE
)

# Summarise the treatment effect by a linear model in x1, x2 and x3
proj <- fusion_projection(fit, basis = ~ x1 + x2 + x3,
                          X_eval = as.data.frame(X))
colMeans(proj)
```

`print.FusionForest` *Print a FusionForest fit*

Description

Prints the main settings of a `FusionForest()` fit: outcome type, treatment coding, error model, sample sizes and number of draws.

Usage

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

Arguments

`x`	A `FusionForest` object.
`...`	Ignored.

Value

`x`, invisibly.

print.summary.FusionForest *Print a FusionForest summary*

Description

Print a FusionForest summary

Usage

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

Arguments

x	A summary.FusionForest object.
digits	Number of significant digits to print.
...	Ignored.

Value

x, invisibly.

summary.FusionForest *Summarise a FusionForest fit*

Description

Summarises a FusionForest() fit: the error standard deviation, the estimated treatment effects of the training rows and the acceptance rates.

Usage

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

Arguments

object	A FusionForest object.
...	Ignored.

Value

A `summary.FusionForest` object: a list with

- `meta`: the settings of the fit;
- `sigma`: posterior draws of the error standard deviation on the standardised scale;
- `treatment_effects`: posterior mean treatment effect of each training row;
- `acceptance_ratios`: mean acceptance rate per forest.

Index

∗ **datasets**
breast_fusion, 2
breast_fusion_truth, 3

breast_fusion, 2, *3*, *4*
breast_fusion_truth, *3*, 3

fusion_estimand, *7*, 9
fusion_projection, *7*, 11
FusionForest, *2*, 4, *9*
FusionForest(), *13*, *14*

gbsg, *3*

print.FusionForest, 13
print.summary.FusionForest, 14

rotterdam, *3*

summary.FusionForest, 14