Package ‘fastgbm’

November 3, 2026

Type Package

Title Fast Histogram Gradient Boosting for Regression, Classification, and Survival Analysis

Version 0.7.0

Description A fast gradient boosting machine covering four task types with one interface: regression (squared error), binary and multiclass classification (logistic and one-vs-rest), and right-censored survival analysis via Cox (Breslow ties), accelerated failure time (normal location-scale), or piecewise-exponential objectives, and competing risks via Fine-Gray subdistribution hazards, cause-specific hazards, or coupled cause-specific hazards, following 'CoxBoost' (Binder et al. 2009) <doi:10.1093/bioinformatics/btp088>. Provides native missing-value routing, baseline-hazard estimation and survival-probability prediction for the survival objectives, and deterministic multi-threaded training via 'RcppParallel'. Methods are described in Friedman (2001) <doi:10.1214/aos/1013203451>.

License MIT + file LICENSE

URL https://github.com/ielbadisy/fastgbm

BugReports https://github.com/ielbadisy/fastgbm/issues

Encoding UTF-8

Depends R (>= 4.1.0)

Imports stats, utils, Rcpp, RcppParallel

LinkingTo Rcpp, RcppParallel

Suggests testthat (>= 3.0.0), knitr, rmarkdown, survival, ggplot2, pdp, gbm, xgboost, ranger, CoxBoost (>= 1.5.2)

VignetteBuilder knitr

Config/testthat/edition 3

SystemRequirements C++17, GNU make

RoxygenNote 7.3.3

NeedsCompilation yes

Author Imad El Badisy [aut, cre]

Maintainer Imad El Badisy <elbadisyimad@gmail.com>

Repository CRAN

Date/Publication 2026-11-03 09:10:08 UTC

Contents

coef.fastgbm 2
fastgbm 3
importance 5
load_fastgbm 6
metrics 6
pdp 7
predict.fastgbm 8
predict.fastgbm_comrisk 9
predict.fastgbm_multiclass 9
print.fastgbm 10
print.fastgbm_comrisk 11
print.fastgbm_multiclass 11
print.summary.fastgbm 12
save_fastgbm 12
summary.fastgbm 13

Index **14**

`coef.fastgbm` *Tree ensembles have no ordinary regression coefficients*

Description

Tree ensembles have no ordinary regression coefficients

Usage

```
## S3 method for class 'fastgbm'
coef(object, ...)
```

Arguments

`object`	A fitted ‘fastgbm‘ object.
`...`	Unused.

Value

Always errors.

`fastgbm` *Fit a compact gradient boosting model for survival, regression, or classification*

Description

Histogram-based gradient boosting with a compiled backend, covering four task types: right-censored survival analysis (Cox with Breslow ties, AFT with a normal location-scale error, or piecewise-exponential hazard), regression (squared error), binary classification (logistic), and multiclass classification (one-vs-rest binary sub-models). Survival, regression, and binary classification share the same compiled tree-growing engine, missing-value routing, and early-stopping machinery; multiclass is a pure-R one-vs-rest wrapper around the binary objective.

Usage

```
fastgbm(
  x,
  time = NULL,
  status = NULL,
  y = NULL,
  objective = NULL,
  ntrees = 200L,
  learning_rate = 0.1,
  max_depth = 5L,
  min_node_size = 10L,
  max_leaves = NULL,
  max_bins = 255L,
  subsample = 0.8,
  colsample = 0.8,
  lambda = 1,
  gamma = 0,
  min_child_weight = 1,
  validation = NULL,
  early_stopping = NULL,
  pexp_bins = 10L,
  grow_policy = "depthwise",
  threads = 0L,
  seed = 1L,
  verbose = TRUE,
  comrisk = NULL,
  cause = NULL,
  ...
)
```

Arguments

`x`	Feature matrix, data frame, or a formula (‘Surv(time, status) ~ .‘ for survival, ‘y ~ .‘ otherwise).

`time` Survival time. Ignored unless ‘objective‘ is ‘"cox"‘, ‘"aft"‘, or ‘"pexp"‘; ignored if ‘y‘ is a ‘survival::Surv‘ object or if ‘x‘ is a formula.

`status` Event indicator (1 = event, 0 = censored). Ignored unless ‘objective‘ is ‘"cox"‘, ‘"aft"‘, or ‘"pexp"‘; ignored if ‘y‘ is a ‘survival::Surv‘ object or if ‘x‘ is a formula.

`y` Response. For survival objectives, an optional ‘survival::Surv(time, status)‘ object, as an alternative to passing ‘time‘/‘status‘ separately. For ‘objective = "regression"‘, a numeric vector. For ‘objective = "binary"‘, a numeric/logical 0-1 vector or a two-level factor. For ‘objective = "multiclass"‘, a factor or character vector with 3+ levels.

`objective` One of ‘"cox"‘, ‘"aft"‘, ‘"pexp"‘ (piecewise exponential: the ensemble models the log hazard rate jointly over covariates and time, via a person-time expansion; see [fastgbm_pexp_cutpoints()] and ‘vignette("survival", package = "fastgbm")‘), ‘"regression"‘ (squared error), ‘"binary"‘ (logistic classification), or ‘"multiclass"‘ (one-vs-rest binary classification). Defaults to ‘"cox"‘ when ‘time‘/‘status‘/a ‘Surv‘ response is supplied, otherwise inferred from ‘y‘ (a two-level 0/1 response defaults to ‘"binary"‘, a factor/character response with 3+ levels defaults to ‘"multiclass"‘, anything else numeric to ‘"regression"‘).

`ntrees` Number of boosting rounds (an upper bound when ‘early_stopping‘ is used). Defaults to ‘200‘, matched to ‘learning_rate‘/‘max_depth‘/ ‘min_node_size‘ below in the benchmark diagnostics (‘inst/benchmarks/error-analysis/‘). **Without ‘validation‘/ ‘early_stopping‘, training all ‘ntrees‘ rounds reliably overfits** on small-to-medium survival datasets – test-set C-index was found to peak between 10 and 100 rounds and then degrade with further training on every one of the 6 benchmark datasets. Supplying ‘validation‘/ ‘early_stopping‘ is strongly recommended for any dataset where held-out performance matters.

`learning_rate` Shrinkage parameter. Defaults to ‘0.1‘.

`max_depth` Maximum tree depth. Defaults to ‘5‘.

`min_node_size` Minimum rows in a node before splitting. Defaults to ‘10‘.

`max_leaves` Unused placeholder for compatibility.

`max_bins` Maximum number of histogram bins.

`subsample` Row subsampling fraction. Defaults to ‘0.8‘; a repeat of the regularization diagnostic with early stopping active showed this gives a further, mostly-positive C-index gain over ‘subsample = 1‘ on top of the ‘colsample‘ default below.

`colsample` Feature subsampling *fraction, resampled at every node* (like ‘ranger‘’s ‘mtry‘, not xgboost’s per-tree ‘colsample_bytree‘) – diagnostics found per-node resampling decorrelates sibling splits more effectively than per-tree sampling, which is a large part of why random forests have lower variance than boosting on the same data. Defaults to ‘0.8‘; diagnostics on the benchmark datasets showed this gives a small, consistently non-negative gain in C-index over ‘colsample = 1‘, though a much smaller value close to ‘sqrt(p)/p‘ (matching ‘ranger‘’s actual ‘mtry‘ default) did better still on higher-‘p‘ datasets and worse on low-‘p‘ ones – there is no universally optimal single value found so far.

`lambda` L2 regularization.

`gamma` Split penalty.

`min_child_weight`
: Minimum child Hessian sum.

`validation`
: A list with ‘x‘ (in the same matrix representation as the training data), and either ‘time‘/‘status‘/a ‘survival::Surv‘ ‘y‘ (survival objectives) or ‘y‘ (regression/binary), used for early stopping. Must be supplied together with ‘early_stopping‘.

`early_stopping`
: Number of boosting rounds without validation-loss improvement before stopping. Must be supplied together with ‘validation‘. When active, ‘fit$trees‘ is truncated to the best-validation-loss iteration after training, so predictions use the best model by default; ‘fit$n_trees_grown‘, ‘fit$validation_history‘, ‘fit$best_iteration‘, and ‘fit$stopping_reason‘ record the full run.

`pexp_bins`
: Number of piecewise-exponential time intervals, used only when ‘objective = "pexp"‘. Cutpoints are quantiles of the observed event times (see [fastgbm_pexp_cutpoints()]).

`grow_policy`
: Tree growth policy; only ‘"depthwise"‘ is implemented.

`threads`
: Number of threads for the RcppParallel split search (‘0‘ = automatic).

`seed`
: Random seed.

`verbose`
: Whether to print progress.

`comrisk`
: Competing-risks model, following the ‘cmprsk‘ options of ’CoxBoost’ (Binder et al. 2009): ‘"sh"‘ (Fine-Gray subdistribution hazard for one cause, with inverse probability of censoring weighted risk sets), ‘"csh"‘ (one independent cause-specific Cox booster per cause), or ‘"ccsh"‘ (coupled cause-specific boosters: one shared tree per round, splits ranked by Fisher’s combination of the per-cause score p-values, with leaf values fitted per cause). Competing risks are used when ‘comrisk‘ is supplied, when ‘y‘ is a multi-state ‘survival::Surv(time, factor(status))‘ response, or when ‘status‘ has 0 (censored) and at least two event codes; ‘comrisk‘ then defaults to ‘"sh"‘, as in ’CoxBoost’. Returns a ‘fastgbm_comrisk‘ object, see [predict.fastgbm_comrisk()].

`cause`
: For ‘comrisk = "sh"‘, the cause of interest (a status code or factor level). Defaults to the first cause.

`...`
: Additional arguments (e.g. ‘data‘ for the formula interface).

Value

A fitted ‘fastgbm‘ object.

`importance` *Feature importance (total split gain)*

Description

Feature importance (total split gain)

Usage

```
importance(object, ...)
```

Arguments

object	A fitted ‘fastgbm‘ object.
...	Unused.

Value

A ‘data.frame‘ with columns ‘feature‘ and ‘gain‘, sorted by decreasing gain.

---

load_fastgbm *Load a serialized fastgbm model*

Description

Load a serialized fastgbm model

Usage

```
load_fastgbm(path)
```

Arguments

path	Path to a serialized model, as written by [save_fastgbm()].

Value

A ‘fastgbm‘ object.

---

metrics *Evaluation metric for a fitted fastgbm model*

Description

Harrell’s C-index for survival objectives (‘"cox"‘, ‘"aft"‘, ‘"pexp"‘), RMSE for ‘"regression"‘, log loss for ‘"binary"‘, and accuracy plus multiclass log loss for ‘"multiclass"‘.

Usage

```
metrics(object, newdata = NULL, y = NULL, type = c("response", "link"))
```

Arguments

object	A fitted ‘fastgbm‘ object.
newdata	Optional new data.
y	Optional observed outcomes: a ‘survival::Surv‘ object for survival objectives, a numeric/logical/two-level-factor vector for ‘"regression"‘/‘"binary"‘, or a factor for ‘"multiclass"‘. If omitted, returns the requested predictions instead of a metric.
type	Prediction type used when ‘y‘ is omitted.

Value

A list with ‘objective‘, ‘metric‘, and ‘value‘, or a numeric vector of predictions if ‘y‘ is omitted.

---

pdp *Partial dependence for a fitted fastgbm model*

Description

Computes Friedman’s partial dependence of the model’s prediction on a single feature: for each value on a grid, the feature column is replaced across every row of ‘data‘ and the resulting predictions are averaged.

Usage

```
pdp(object, feature, data, grid_resolution = 20L, type = NULL)
```

Arguments

object	a fitted ‘fastgbm‘ model.
feature	name of the feature (column of ‘data‘) to profile.
data	the data used to compute the partial dependence average; typically the training data, in the same representation (matrix or data frame) used to fit ‘object‘.
grid_resolution	number of grid points for numeric features (fewer are used if the feature has fewer distinct values). Ignored for factor or character features, where every level is used.
type	‘"response"‘ or ‘"link"‘; defaults to ‘"link"‘ (the risk/location score).

Details

Partial dependence is computed on the *linear predictor* (‘type = "link"‘) by default, i.e. the Cox log-risk score or the AFT log-time location parameter. This is a risk-score (or location-score) partial dependence, not a survival-probability partial dependence; use ‘predict(object, ..., type = "survival")‘ directly if a probability-scale summary at specific horizons is needed.

Value

a ‘data.frame‘ with class ‘fastgbm_pdp‘ and columns ‘feature‘, ‘x‘ (grid value) and ‘yhat‘ (average prediction).

`predict.fastgbm` *Predict from a fitted fastgbm model*

Description

For ‘objective = "pexp"‘, there is no single scalar "linear predictor" the way Cox/AFT have one, since the model is a function of both covariates and time. ‘type = "link"‘/‘"response"‘ instead use the cumulative hazard at the model's full fitted time horizon (‘max(object$pexp_cutpoints)‘) as a fixed, well-defined risk score (higher = more risk, same ranking convention as Cox), and ‘type = "survival"‘ evaluates the hazard-over-time surface directly at the requested ‘times‘.

Usage

```
## S3 method for class 'fastgbm'
predict(
  object,
  newdata,
  type = c("response", "link", "survival"),
  times = NULL,
  ...
)
```

Arguments

`object`	A fitted ‘fastgbm‘ object.
`newdata`	New data to predict on, in the same representation used to fit.
`type`	‘"response"‘ (survival: exp of the linear predictor; ‘"regression"‘: raw prediction; ‘"binary"‘: predicted probability), ‘"link"‘ (raw linear predictor, all objectives), or ‘"survival"‘ (survival probability at ‘times‘; survival objectives only).
`times`	Required when ‘type = "survival"‘: a vector of times at which to evaluate the survival function.
`...`	Unused.

Value

A numeric vector (‘"response"‘/‘"link"‘) or a matrix of survival probabilities, ‘nrow(newdata)‘ by ‘length(times)‘ (‘"survival"‘).

`predict.fastgbm_comrisk` *Predict from a fitted competing-risks fastgbm model*

Description

Predict from a fitted competing-risks fastgbm model

Usage

```
## S3 method for class 'fastgbm_comrisk'
predict(
  object,
  newdata,
  type = c("cif", "link", "response"),
  times = NULL,
  ...
)
```

Arguments

`object`	A fitted ‘fastgbm_comrisk‘ object (from [fastgbm()] with ‘comrisk‘).
`newdata`	New data to predict on, in the same representation used to fit.
`type`	‘"cif"‘ (cumulative incidence at ‘times‘), ‘"link"‘ (the boosted log subdistribution hazard ratio for ‘"sh"‘, or the log cause-specific hazard ratios for ‘"csh"‘/‘"ccsh"‘), or ‘"response"‘ (exp of ‘"link"‘).
`times`	Required when ‘type = "cif"‘: times at which to evaluate the cumulative incidence.
`...`	Unused.

Value

For ‘"cif"‘, a named list with one ‘nrow(newdata)‘ by ‘length(times)‘ matrix per modelled cause (a single cause for ‘"sh"‘). Otherwise a matrix with one column per modelled cause.

`predict.fastgbm_multiclass` *Predict from a fitted multiclass fastgbm model*

Description

‘objective = "multiclass"‘ fits one binary (one-vs-rest) ‘fastgbm‘ model per class, all sharing the same hyperparameters. Predicted class probabilities are each class’s own binary probability, renormalized to sum to 1 across classes.

Usage

```
## S3 method for class 'fastgbm_multiclass'
predict(object, newdata, type = c("prob", "class"), ...)
```

Arguments

object	A fitted ‘fastgbm_multiclass‘ object.
newdata	New data to predict on, in the same representation used to fit.
type	‘"prob"‘ (a ‘nrow(newdata)‘ x ‘nlevels(y)‘ matrix of class probabilities), or ‘"class"‘ (a factor of predicted class labels, the ‘argmax‘ of ‘"prob"‘).
...	Unused.

Value

A matrix (‘"prob"‘) or factor (‘"class"‘).

---

print.fastgbm *Print a fitted fastgbm model*

Description

Print a fitted fastgbm model

Usage

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

Arguments

x	A fitted ‘fastgbm‘ object.
...	Unused.

Value

‘x‘, invisibly.

print.fastgbm_comrisk *Print a fitted competing-risks fastgbm model*

Description

Print a fitted competing-risks fastgbm model

Usage

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

Arguments

x	A fitted ‘fastgbm_comrisk‘ object.
...	Unused.

Value

‘x‘, invisibly.

print.fastgbm_multiclass *Print a fitted multiclass fastgbm model*

Description

Print a fitted multiclass fastgbm model

Usage

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

Arguments

x	A fitted ‘fastgbm_multiclass‘ object.
...	Unused.

Value

‘x‘, invisibly.

print.summary.fastgbm *Print a fastgbm model summary*

Description

Print a fastgbm model summary

Usage

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

Arguments

x	A ‘summary.fastgbm‘ object.
...	Unused.

Value

‘x‘, invisibly.

save_fastgbm *Save a fitted fastgbm model*

Description

Save a fitted fastgbm model

Usage

```
save_fastgbm(object, path)
```

Arguments

object	A fitted ‘fastgbm‘ object.
path	Path to write the serialized model to.

Value

‘path‘, invisibly.

`summary.fastgbm` *Summarize a fitted fastgbm model*

Description

Summarize a fitted fastgbm model

Usage

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

Arguments

`object`	A fitted ‘fastgbm‘ object.
`...`	Unused.

Value

A ‘summary.fastgbm‘ object.

Index

coef.fastgbm, 2

fastgbm, 3

importance, 5

load_fastgbm, 6

metrics, 6

pdp, 7
predict.fastgbm, 8
predict.fastgbm_comrisk, 9
predict.fastgbm_multiclass, 9
print.fastgbm, 10
print.fastgbm_comrisk, 11
print.fastgbm_multiclass, 11
print.summary.fastgbm, 12

save_fastgbm, 12
summary.fastgbm, 13