Package 'ordered'

November 17, 2026

Title 'parsnip' Engines and Wrappers for Ordinal Classification Models

Version 0.2.0

Description Bindings, methods, and tuners for using ordinal classification models with the 'parsnip' and 'dials' packages. These include the regularized elastic net ordinal regression of Wurm, Hanlon, and Rathouz (2021) <doi:10.18637/jss.v099.i06> in 'ordinalNet', the ordinal classification trees of Galimberti, Soffritti, and Di Maso (2012) <doi:10.18637/jss.v047.i10> in 'rpartScore', and the latent variable ordinal forests of Hornung (2020) <doi:10.1007/s00357-018-9302-x> in 'ordinalForest'.

License MIT + file LICENSE

Depends R (>= 4.1), parsnip (>= 1.6.1)

Imports cli, dials (>= 1.4.5), purrr, rlang (>= 1.1.4), scales (>= 1.3.0), tibble, tidyr

Suggests dplyr, MASS, ordinal, ordinalNet, rms (>= 6.8-0), glmnetcr, VGAM (>= 1.1-9), rpartScore, ordinalForest, orf, QSARdata, spelling, testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

Language en-US

URL https://corybrunson.github.io/ordered/

BugReports https://github.com/corybrunson/ordered/issues

Config/Needs/website rmarkdown

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Max Kuhn [aut] (ORCID: <https://orcid.org/0000-0003-2402-136X>), Jason Cory Brunson [aut, cre] (ORCID: <https://orcid.org/0000-0003-3126-9494>), Posit Software PBC [cph]

Maintainer Jason Cory Brunson <cornelioid@gmail.com>
Repository CRAN
Date/Publication 2026-11-17 17:30:02 UTC

Contents

ordinalForest_parameters 2
ordinalNet_parameters 3
orf_parameters 4
rpartScore_parameters 6

Index **7**

`ordinalForest_parameters`
Dials for `ordinalForest` *engine parameters*

Description

These parameters are auxiliary to random forest models that use the "ordinalForest" engine. They correspond to tuning parameters that would be specified using `set_engine("ordinalForest", ...)`.

Usage

```
naive_scores(values = c(FALSE, TRUE))

num_scores(range = c(100L, 2000L), trans = NULL)

num_score_perms(range = c(100L, 500L), trans = NULL)

num_score_trees(range = c(10L, 200L), trans = NULL)

num_scores_best(range = c(2L, 20L), trans = NULL)

ord_metric(values = values_ord_metric)

values_ord_metric
```

Arguments

`values`	A character string of possible values.
`range`	A two-element vector holding the *defaults* for the smallest and largest possible values, respectively. If a transformation is specified, these values should be in the *transformed units*.
`trans`	A `trans` object from the `scales` package, such as `scales::transform_log10()` or `scales::transform_reciprocal()`. If not provided, the default is used which matches the units used in `range`. If no transformation, `NULL`.

Details

The vector `values_ord_metric` provides options to the `ord_metric` dial.

These functions generate parameters for `parsnip::rand_forest()` models using the `"ordinalForest"` engine. See `?ordinalForest::ordfor()` for more details on the original parameters. These parameters are engine-specific, not general to decision tree models, so are provided here rather than in `dials`.

- `naive_scores()`: Whether to construct only a "naive" ordinal forest using the scores $1, 2, 3, \ldots$ for the ordinal values; tunes `naive`.
- `num_scores()`: The number of score sets tried prior to optimization; tunes `nsets`.
- `num_score_perms()`: The number of permutations of the class width ordering to try for each score set tried (after the first); tunes `npermtrial`.
- `num_score_trees()`: The number of trees in the score set–specific forests; tunes `ntreeperdiv`.
- `num_scores_best()`: The number of top-performing score sets used to calculate the optimized score set; tunes `nbest`.
- `ord_metric()`: The performance function used to evaluate score set–specific forests; tunes `perffunction`. See also `?ordinalForest::perff`.

Value

An object of S3 parent class `param` and primary class `qual_param` or `quant_param`; see `dials::new_qual_param()` and [dials::new_quant_param().

See Also

`dials::trees()`

Examples

```
naive_scores()
num_scores()
num_score_perms()
num_score_trees()
num_scores_best()
values_ord_metric
ord_metric()
```

`ordinalNet_parameters` *Dials for* `ordinalNet` *engine parameters*

Description

These parameters are auxiliary to ordinal regression models that use the "ordinalNet" engine. They correspond to tuning parameters that would be specified using `set_engine("ordinalNet", ...)`.

Usage

```
parallel_penalty_factor(
  range = c(-Inf, Inf),
  trans = scales::transform_log10()
)
```

Arguments

range	A two-element vector holding the *defaults* for the smallest and largest possible values, respectively. If a transformation is specified, these values should be in the *transformed units*.
trans	A trans object from the scales package, such as scales::transform_log10() or scales::transform_reciprocal(). If not provided, the default is used which matches the units used in range. If no transformation, NULL.

Details

These functions generate parameters for parsnip::ordinal_reg() models using the "ordinalNet" engine. See ?ordinalNet::ordinalNet() for more details on the original parameters. These parameters are engine-specific, not general to ordinal regression models, so are provided here rather than in dials.

- parallel_penalty_factor(): A nonnegative numeric factor that scales the penalty on all parallel terms; tunes parallelPenaltyFactor. Only used when parallelTerms = TRUE.

Value

An object of S3 parent class param and primary class quant_param; see dials::new_quant_param().

See Also

dials::penalty()

Examples

```
parallel_penalty_factor()
```

---

orf_parameters *Dials for* orf *engine parameters*

Description

These parameters are auxiliary to random forest models that use the "orf" engine. They correspond to tuning parameters that would be specified using set_engine("orf", ...).

Usage

```
sample_fraction(range = c(0.1, 1), trans = NULL)

honesty(values = c(TRUE, FALSE))

honesty_fraction(range = c(0.1, 0.9), trans = NULL)
```

Arguments

range	A two-element vector holding the *defaults* for the smallest and largest possible values, respectively. If a transformation is specified, these values should be in the *transformed units*.
trans	A trans object from the scales package, such as scales::transform_log10() or scales::transform_reciprocal(). If not provided, the default is used which matches the units used in range. If no transformation, NULL.
values	A logical vector of possible values. See values_ord_metric.

Details

These functions generate parameters for parsnip::rand_forest() models using the "orf" engine. See ?orf::orf() for more details on the original parameters. These parameters are engine-specific, not general to random forest models, so are provided here rather than in dials.

- sample_fraction(): The fraction of observations to be subsampled; tunes sample.fraction.
- honesty(): Whether to use honest splitting (sample splitting); tunes honesty.
- honesty_fraction(): The fraction of observations reserved for the honest (estimation) sample; tunes honesty.fraction.

Value

An object of S3 parent class param and primary class qual_param or quant_param; see dials::new_qual_param() and dials::new_quant_param().

See Also

dials::trees()

Examples

```
sample_fraction()
honesty()
honesty_fraction()
```

`rpartScore_parameters` *Dials for* `rpartScore` *engine parameters*

Description

These parameters are auxiliary to decision tree models that use the "rpartScore" engine. They correspond to tuning parameters that would be specified using `set_engine("rpartScore", ...)`.

Usage

```
split_func(values = c("abs", "quad"))

prune_func(values = c("mr", "mc"))
```

Arguments

`values`	A character string of possible values.

Details

`split_func` and `prune_func` are dials for `split` and `prune`, respectively. See `?rpartScore::rpartScore` for more details on the original parameters. These parameters are engine-specific, not general to decision tree models, so are provided here rather than in `dials`.

Value

An object of S3 parent class `param` and primary class `qual_param`; see `dials::new_qual_param()`.

See Also

`dials::trees()`

Examples

```
split_func()
prune_func()
```

Index

dials::new_qual_param(), *3*, *5*, *6*
dials::new_quant_param(), *4*, *5*
dials::penalty(), *4*
dials::trees(), *3*, *5*, *6*

honesty *(*orf_parameters*)*, 4
honesty_fraction *(*orf_parameters*)*, 4

naive_scores
        *(*ordinalForest_parameters*)*, 2
num_score_perms
        *(*ordinalForest_parameters*)*, 2
num_score_trees
        *(*ordinalForest_parameters*)*, 2
num_scores *(*ordinalForest_parameters*)*, 2
num_scores_best
        *(*ordinalForest_parameters*)*, 2

ord_metric *(*ordinalForest_parameters*)*, 2
ordinalForest_parameters, 2
ordinalNet_parameters, 3
orf_parameters, 4

parallel_penalty_factor
        *(*ordinalNet_parameters*)*, 3
parsnip::ordinal_reg(), *4*
parsnip::rand_forest(), *3*, *5*
prune_func *(*rpartScore_parameters*)*, 6

rpartScore_parameters, 6

sample_fraction *(*orf_parameters*)*, 4
split_func *(*rpartScore_parameters*)*, 6

values_ord_metric
        *(*ordinalForest_parameters*)*, 2