Package 'ino'

October 13, 2022

Title Initialization of Numerical Optimization

Version 0.2.0

Date 2022-09-28

Description Implementation of initialization strategies for the numerical optimization of real-valued functions, in particular likelihood functions of statistical models.

License GPL (>= 3)

Encoding UTF-8

LazyData true

LazyDataCompression xz

RoxygenNote 7.2.1

Imports ggplot2, rlang, mvtnorm, crayon, cli, progress, dplyr, foreach, doSNOW, ao (>= 0.2.3)

Suggests knitr, rmarkdown, testthat (>= 3.0.0), purrr, pracma, fHMM

Config/testthat/edition 3

URL https://github.com/loelschlaeger/ino

BugReports https://github.com/loelschlaeger/ino/issues

VignetteBuilder knitr

Depends R (>= 4.0.0), optimizeR

NeedsCompilation no

Author Lennart Oelschläger [aut, cre] (<https://orcid.org/0000-0001-5421-9313>), Marius Ötting [aut] (<https://orcid.org/0000-0002-9373-0365>)

Maintainer Lennart Oelschläger <oelschlaeger.lennart@gmail.com>

Repository CRAN

Date/Publication 2022-09-29 06:10:08 UTC

R topics documented:

`check_failed_runs` *Check for failed runs*

Description

This helper function checks for failed runs in an `ino` object.

Usage

```
check_failed_runs(x, verbose = getOption("ino_progress"))
```

Arguments

`x`	An object of class `ino`.
`verbose`	A boolean, which indicates whether progress should be printed. Set to `TRUE` (`FALSE`) to print (hide) progress. The default is `getOption("ino_progress")`, which is set to `TRUE` when the package is loaded.

Value

No return value, called for side effects.

clear_ino *Clear records*

Description

This function clears initialization records saved in an `ino` object.

Usage

```
clear_ino(x, which)
```

Arguments

`x`	An object of class `ino`.
`which`	Either • `"all"` to clear all records, • or `"fails"` to clear all failed records, • or a numeric vector of row numbers from `summary(x)`.

Value

The updated input `x`.

earthquakes *Earthquake data*

Description

This data set includes the number of yearly measured earthquakes from 1900 to 2006.

Usage

```
data(earthquakes)
```

Format

The data set is a `data.frame` with two integer columns, `year` for the year and `obs` for the number of measured earthquakes.

Source

The data was obtained from http://hmms-for-time-series.de/second/data/earthquakes.txt on 2022-03-25.

`fixed_initialization` *Fixed initialization*

Description

This function is an implementation of the fixed initialization strategy.

Usage

```
fixed_initialization(
  x,
  at,
  ncores = getOption("ino_ncores"),
  verbose = getOption("ino_progress"),
  label = "fixed"
)
```

Arguments

`x`	An object of class `ino`.
`at`	A numeric vector of length `npar(x)` with the (fixed) initial values.
`ncores`	The number of cores for parallel computation over parameters and optimizers. The default is `getOption("ino_ncores")`, which is set to `1` when the package is loaded.
`verbose`	A boolean, which indicates whether progress should be printed. Set to `TRUE` (`FALSE`) to print (hide) progress. The default is `getOption("ino_progress")`, which is set to `TRUE` when the package is loaded.
`label`	A character, the label for the initialization strategy.

Value

The updated `ino` object.

See Also

`npar()` to extract the number `npar` from an `ino` object.

get_fails *Get failure messages*

Description

This function extracts failure messages from an ino object.

Usage

```
get_fails(x, runs = NULL)
```

Arguments

x	An object of class ino.
runs	An integer vector, specifying the optimization runs of interest. Can be NULL (default), in which case all runs are considered.

Value

A list of failure messages for the optimization run.

See Also

get_vars() for extracting any available variable.

get_vars *Get variables*

Description

This function extracts available variables from an ino object.

Usage

```
get_vars(x, runs = NULL, vars = NULL)
```

Arguments

x	An object of class ino.
runs	An integer vector, specifying the optimization runs of interest. Can be NULL (default), in which case all runs are considered.
vars	A character vector, specifying the variables of interest. Can be NULL (default), in which case all variables are considered.

Value

A list, each element is a list of variables of an optimization run.

hmm_ino *Example application of ino to HMM likelihood*

Description

TBA

Usage

```
data(hmm_ino)
```

Format

TBA

nfails *Number of failed optimization runs*

Description

This function returns the number of failed optimization runs saved in an `ino` object.

Usage

```
nfails(x)
```

Arguments

`x`	An object of class `ino`.

Value

An integer, the number of failed recorded optimization runs.

ngrid *Number of grid elements*

Description

This function returns the number of grid elements of an `ino` object.

Usage

```
ngrid(x)
```

Arguments

`x`	An object of class `ino`.

Value

An integer, the number of grid elements.

npar *Number of parameters*

Description

This function extracts the `npar` element from an `ino` object, which is the length of the parameter vector over which the target function is optimized.

Usage

```
npar(x)
```

Arguments

`x`	An object of class `ino`.

Value

An integer, the number of parameters.

nruns *Number of optimization runs*

Description

This function returns the number of optimization runs saved in an `ino` object.

Usage

```
nruns(x)
```

Arguments

`x`	An object of class `ino`.

Value

An integer, the number of recorded optimization runs.

overview_optima *Optima overview*

Description

This function provides an overview of the identified optima.

Usage

```
overview_optima(x, digits = 2)
```

Arguments

`x`	An object of class `ino`.
`digits`	The number of decimal places of the optima values. The default is `2`.

Value

A data frame with columns

optimum the unique optima (with respect to `digits`)

frequency the number of runs the optima was reached

plot.ino *Visualization of optimization time*

Description

This function plots boxplots of optimization times in an `ino` object.

Usage

```
## S3 method for class 'ino'
plot(x, by = NULL, time_unit = "secs", nrow = NULL, ...)
```

Arguments

`x`	An object of class `ino`.
`by`	A character vector of variables to group by. Can be `NULL` (default).
`time_unit`	The time unit, see `difftime`.
`nrow`	Passed to `facet_wrap`.
`...`	Ignored.

Value

A `ggplot` object.

probit_ino *Example application of ino to probit likelihood*

Description

See the vignette about the probit likelihood for details: `vignette("example_probit", package = "ino")`

Usage

```
data(probit_ino)
```

Format

An object of class `ino`.

`random_initialization` *Random initialization*

Description

This function is an implementation of the random initialization strategy.

Usage

```
random_initialization(
  x,
  runs = 1L,
  sampler = function() stats::rnorm(npar(x)),
  ncores = getOption("ino_ncores"),
  verbose = getOption("ino_progress"),
  label = "random"
)
```

Arguments

`x`	An object of class `ino`.
`runs`	An integer, the number of random initializations. The default is `1`.
`sampler`	A function without any arguments which returns a numeric vector of length `npar(x)` with (random) initial values. Per default, `sampler = function() stats::rnorm(npar(x))`, i.e. random initial values from a standard normal distribution.
`ncores`	The number of cores for parallel computation over parameters and optimizers. The default is `getOption("ino_ncores")`, which is set to `1` when the package is loaded.
`verbose`	A boolean, which indicates whether progress should be printed. Set to `TRUE` (`FALSE`) to print (hide) progress. The default is `getOption("ino_progress")`, which is set to `TRUE` when the package is loaded.
`label`	A character, the label for the initialization strategy.

Value

The updated `ino` object.

See Also

`npar()` to extract the number `npar` from an `ino` object.

setup_ino *Setup*

Description

Use this function to specify the numerical optimization problem. The function returns an object of class `ino` that contains all specifications.

Usage

```
setup_ino(
  f,
  npar,
  global = NULL,
  ...,
  mpvs = character(),
  opt = set_optimizer_nlm(),
 test_par = list(validate = TRUE, init_rest = list(lower = -1, upper = 1), init_digits =
    2, f_checks = 10, f_checks_time = 1),
  verbose = getOption("ino_progress")
)
```

Arguments

`f`	An object of class `function`, the function to be optimized.
`npar`	The length of the first argument of `f`, i.e. the argument over which `f` is optimized.
`global`	Either `NULL` (default) or the point where `f` obtains its global optimum (i.e., a numeric vector of length `npar`).
`...`	Additional and named arguments to be passed to `f` (optional).
`mpvs`	A character vector of the argument names with multiple parameter values. None per default.
`opt`	The output of `set_optimizer`, which is an object of class `optimizer`. Per default, `opt = set_optimizer_nlm()`, which specifies the `nlm` optimizer. Can also be a (named) list of multiple `optimizer` objects.
`test_par`	A list of test parameters for an `ino` object: • `validate`, a Boolean, set to `TRUE` (`FALSE`) to (not) validate the `ino` object. Per default, `validate = TRUE`. • `init_rest`, a list of two elements, `lower` and `upper`, with lower and upper limits, respectively, for test values for `f`. Can be single values (for joint limits) or numeric vectors of length `npar` (for individual limits). Per default, `lower = -1` and `upper = 1`. • `init_digits`, the number of decimal places for the test initial values. Per default, `init_digits = 2`. • `f_checks`, the number of checks for `f` with random input values (that fulfill the `init_rest` restrictions). Per default, `f_checks = 10`.

- f_check_time, the maximum number of seconds for a single check for f. A check is considered to be successful, if no error occurred within f_check_time seconds. Per default, f_check_time = 1.

verbose — A boolean, which indicates whether progress should be printed. Set to TRUE (FALSE) to print (hide) progress. The default is getOption("ino_progress"), which is set to TRUE when the package is loaded.

Format

The format of an ino object is documented in new_ino.

Details

Specifying a function:

One real-valued function f must be specified per ino object. The function is optimized over its first argument, the target argument, which must be a numeric vector of length npar, followed by any other arguments specified via the ... argument.

Specifying multiple parameter values:

You can specify multiple values for each ... parameter for comparison. Such arguments must be in list format, where each list element must be a valid parameter value. The list elements can be named. The names of the ... parameters with multiple values must be added to the mpvs input.

Specifying an optimizer:

The numerical optimizer must be specified via the opt argument as an optimizer object. Such optimizer objects can be created via the function set_optimizer. You can specify multiple optimizer objects for comparison by passing a (named) list of optimizers to opt.

An example:

Let nll be a negative log-likelihood function. Its first argument is a numeric vector of length 2. The global optimum is obtained in the origin. The function has the additional argument data. Say that you want to conduct an experiment of the initialization effect for nll for two different data sets. And say that you want to compare the nlm and the optim optimizer. Then, specify

```
setup_ino(
  f = nll,
  npar = 2,
  global = c(0,0),
  data = list("data1" = <data set 1>,
              "data2" = <data set 2>),
  mpvs = "data",
  opt = list("nlm"   = set_optimizer_nlm(),
             "optim" = set_optimizer_optim())
)
```

Value

An object of class ino.

See Also

`set_optimizer()` to specify an optimizer.

Examples

```
setup_ino(
  f = f_ll_hmm,
  npar = 4,
  data = earthquakes,
  N = 2,
  neg = TRUE
)
```

`standardize_initialization`
Standardize initialization

Description

This function is an implementation of the standardize initialization strategy.

Usage

```
standardize_initialization(
  x,
  arg = "data",
  by_col = TRUE,
  center = TRUE,
  scale = TRUE,
  ind_ign = integer(),
  initialization = random_initialization(),
  ncores = getOption("ino_ncores"),
  verbose = getOption("ino_progress"),
  label = "standardize"
)
```

Arguments

`x`	An object of class `ino`.
`arg`	A character, the name of the argument to be standardized. The argument must be of class `matrix` or `data.frame`. Per default, `arg = "data"`.
`by_col`	A boolean, set to `TRUE` (the default) to standardize column-wise, set to `FALSE` to standardize by rows.
`center`	A boolean, set to `TRUE` (the default) for mean standardization.
`scale`	A boolean, set to `TRUE` (the default) for variance standardization.

ind_ign	A numeric vector of column indices (or row indices if by_col = FALSE) that are ignored when standardizing.
initialization	An object of class strategy_call which determines the initialization. The strategy_call can be generated by one of the strategy functions (any function with the name *_initialization), when the x argument is unspecified. Per default, initialization = random_initialization(), i.e. random initialization.
ncores	The number of cores for parallel computation over parameters and optimizers. The default is getOption("ino_ncores"), which is set to 1 when the package is loaded.
verbose	A boolean, which indicates whether progress should be printed. Set to TRUE (FALSE) to print (hide) progress. The default is getOption("ino_progress"), which is set to TRUE when the package is loaded.
label	A character, the label for the initialization strategy.

Value

The updated ino object.

---

subset_initialization *Subset initialization*

Description

This function is an implementation of the subset initialization strategy.

Usage

```
subset_initialization(
  x,
  arg = "data",
  by_row = TRUE,
  how = "random",
  prop = 0.5,
  ind_ign = integer(),
  kmeans_arg = list(centers = 2),
  initialization = random_initialization(),
  ncores = getOption("ino_ncores"),
  verbose = getOption("ino_progress"),
  label = paste0("subset(", how, ",", prop, ")")
)
```

Arguments

x	An object of class ino.
arg	A character, the name of the argument to be subsetted. Only an argument of class matrix or data.frame can be chosen. Per default, arg = "data".
by_row	A boolean, set to TRUE (the default) to subset by row, set to FALSE to subset by column.
how	A character, specifying how to select the subset. Can be one of "random" (default), "first", and "kmeans".
prop	A numeric between 0 and 1, specifying the proportion of the subset.
ind_ign	A numeric vector of column indices (or row indices if by_row = FALSE) that are ignored when clustering. Only relevant if how = "kmeans".
kmeans_arg	A list of additional arguments for kmeans. Per default, kmeans_arg = list(centers = 2), which sets the number of clusters to 2. Only relevant if how = "kmeans".
initialization	An object of class strategy_call which determines the initialization. The strategy_call can be generated by one of the strategy functions (any function with the name *_initialization), when the x argument is unspecified. Per default, initialization = random_initialization(), i.e. random initialization.
ncores	The number of cores for parallel computation over parameters and optimizers. The default is getOption("ino_ncores"), which is set to 1 when the package is loaded.
verbose	A boolean, which indicates whether progress should be printed. Set to TRUE (FALSE) to print (hide) progress. The default is getOption("ino_progress"), which is set to TRUE when the package is loaded.
label	A character, the label for the initialization strategy.

Value

The updated ino object.

---

summary.ino *Summary of initialization runs*

Description

This function gives an overview of the initialization runs in an ino object.

Usage

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

Arguments

object	An object of class ino.
...	Expressions of variables from var_names as characters.

Details

The following variables are available for each ino object:

.strategy the name of the initialization strategy

.time the optimization time

.optimum the function value at the optimum

.optimizer the identifier of the optimizer

Value

A tibble, optimization runs as rows and variables as columns.

---

update_opt *Update optimizer*

Description

Use this function to update the optimizer functions for an ino object.

Usage

```
update_opt(x, opt, verbose = getOption("ino_progress"))
```

Arguments

x	An object of class ino.
opt	The output of set_optimizer, which is an object of class optimizer. Per default, opt = set_optimizer_nlm(), which specifies the nlm optimizer. Can also be a (named) list of multiple optimizer objects.
verbose	A boolean, which indicates whether progress should be printed. Set to TRUE (FALSE) to print (hide) progress. The default is getOption("ino_progress"), which is set to TRUE when the package is loaded.

Value

An object of class ino.

See Also

set_optimizer() to specify an optimizer.

var_names *Variable names*

Description

This function returns the names of the available variables in an `ino` object.

Usage

```
var_names(x)
```

Arguments

`x`	An object of class `ino`.

Value

A character vector.

Index