

Package ‘ao’

November 8, 2023

Title Alternating Optimization

Version 0.3.0

Description Alternating optimization of (high-dimensional) functions is an iterative procedure for optimizing jointly over all parameters by alternately optimizing parameter subsets.

URL <https://loelschlaeger.de/ao/>, <https://github.com/loelschlaeger/ao/>

BugReports <https://github.com/loelschlaeger/ao/issues>

License GPL-3

Encoding UTF-8

RoxygenNote 7.2.3

Imports checkmate

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

Depends R (>= 4.0.0), optimizeR (>= 1.0.1)

NeedsCompilation no

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

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

Repository CRAN

Date/Publication 2023-11-08 21:50:02 UTC

R topics documented:

ao	2
Index	4

Description

This function performs alternating optimization of the function f .

Usage

```
ao(
  f,
  p,
  ...,
  partition = as.list(1:length(p)),
  base_optimizer = optimizerR::Optimizer$new("stats::optim"),
  iterations = 10,
  tolerance = 1e-06,
  f_partition = vector(mode = "list", length = length(partition)),
  joint_end = FALSE,
  verbose = FALSE
)
```

Arguments

<code>f</code>	A function to be optimized, returning a single numeric. The first argument of <code>f</code> must be a numeric of the same length as <code>p</code> followed by any other arguments specified by the <code>...</code> argument.
<code>p</code>	A numeric vector, the starting parameter values for the optimization.
<code>...</code>	Additional arguments to be passed to <code>f</code> .
<code>partition</code>	A list of vectors of indices of <code>p</code> , specifying the partition in the alternating optimization process. The default is <code>as.list(1:length(p))</code> , i.e. each parameter is optimized separately. Parameter indices can be members of multiple subsets.
<code>base_optimizer</code>	An <code>Optimizer</code> object, which can be created via Optimizer . It numerically solves the optimization problems in the partitions.
<code>iterations</code>	An integer, the maximum number of iterations through partitions before the alternating optimization process is terminated. Can also be <code>Inf</code> , in which case <code>tolerance</code> is responsible for the termination. The default is 10.
<code>tolerance</code>	A non-negative numeric. The alternating optimization terminates prematurely (i.e., before <code>iterations</code> is reached) if the euclidean distance between the current estimate and the one from the last iteration is smaller than <code>tolerance</code> . The default is <code>1e-6</code> .
<code>f_partition</code>	A list of the same length as <code>partition</code> . The i -th element can be a function that computes the value of <code>f</code> for the i -th parameter set defined in <code>partition</code> . The function must be of the form <code>function(theta_part, theta_rest, ...)</code> , where

- `theta_part` receives the parameter set for the current partition (this argument can be named differently),
 - `theta_rest` receives the remaining parameters (this argument must be named `theta_rest`),
 - `...` receives the additional arguments to `f`. Alternatively, it can be `NULL`, in which case `f` is used.
- `joint_end` If `TRUE`, the parameter set is optimized jointly after the alternating optimization process is terminated. The default is `FALSE`.
- `verbose` If `TRUE`, full tracing details are printed during the alternating optimization process. The default is `FALSE`.

Value

A list with the elements

- `estimate`, the optimal parameter vector found,
- `value`, the value of `f` at `estimate`,
- `sequence`, a `data.frame` of the function values, estimates and computation times in the single iterations and partitions,
- `and seconds`, the overall computation time in seconds.

Examples

```
# definition of the Himmelblau function
himmelblau <- function(x) (x[1]^2 + x[2] - 11)^2 + (x[1] + x[2]^2 - 7)^2

# alternating minimization separately for x_1 and x_2
# parameter restriction: -5 <= x_1, x_2 <= 5
ao(
  f = himmelblau, p = c(0, 0), partition = list(1, 2), iterations = Inf,
  base_optimizer = optimizeR::Optimizer$new(
    which = "stats::optim", lower = -5, upper = 5, method = "L-BFGS-B"
  )
)
```

Index

ao, [2](#)

Optimizer, [2](#)