Package ‘optimg’

October 7, 2021

Type Package

Title General-Purpose Gradient-Based Optimization

Version 0.1.2

Date 2021-10-07

Author Vithor Rosa Franco <`vithorfranco@gmail.com`>

Maintainer Vithor Rosa Franco <`vithorfranco@gmail.com`>

Description Provides general purpose tools for helping users to implement steepest gradient descent methods for function optimization; for details see Ruder (2016) <arXiv:1609.04747v2>. Currently, the Steepest 2-Groups Gradient Descent and the Adaptive Moment Estimation (Adam) are the methods implemented. Other methods will be implemented in the future.

Imports ucminf (>= 1.1-4)

License GPL-3

Encoding UTF-8

URL `https://github.com/vthorrf/optimg`

NeedsCompilation no

Repository CRAN

Date/Publication 2021-10-07 16:30:05 UTC

R topics documented:

optimg — *General-Purpose Gradient-Based Optimization*

Description

General-purpose optimization based on gradient descent algorithms. It is greatly inspired on the stats::optim function, aiming at increased usability and adaptability for optim's users.

Usage

```
optimg(par, fn, gr=NULL, ..., method=c("STGD","ADAM"),
       Interval=1e-6, maxit=100, tol=1e-8, full=F, verbose=F)
```

Arguments

`par`	Initial values for the parameters to be optimized over.
`fn`	A function to be minimized. It should return a scalar result.
`gr`	A function to return the gradient. Should include "Interval" as an argument, even if not used. If it is NULL, a finite-difference approximation will be used.
`...`	Further arguments to be passed to fn and gr.
`method`	The method to be used. See 'Details'.
`Interval`	The epsilon difference to be used by gr.
`maxit`	The maximum number of iterations. Defaults to 100.
`tol`	Relative convergence tolerance. The algorithm stops if it is unable to reduce the value by a factor of reltol * (abs(val) + reltol) at a step. Defaults to 1e-8.
`full`	Boolean argument for returning all results, or only the basics (see Value). Defaults to FALSE.
`verbose`	Boolean argument for printing update status. Defaults to FALSE.

Details

As with the optim function, ... must be matched exactly. Also as with optim, optimg performs minimization. All the methods implemented are the "steepest" variation of the original methods. This means that tuning parameters are optimized at each step automatically. This makes the algorithms slower, but also more effective, especially for complex models.

The default method (unless the length of par is equal to 1; in which case, the default is "ADAM") is an implementation of the Steepest 2-Group Gradient Descent ("STGD") algorithm. This algorithm is a variation of the Steepest Gradient Descent method which optimizes different step sizes for two groups of gradients: those within a standard deviation (below or above), and those beyond a standard deviation (below or above).

Method "ADAM" is the Adaptive Moment Estimation method. This method computes adaptive learning rates for each parameter. Adam stores both exponentially decaying average of past squared gradients, as well as measures of momentum.

Value

If full = FALSE, a list with components:

`par`	The best set of parameters found.
`value`	The value of fn corresponding to par.
`counts`	A scalar giving the number of iterations before convergence is reached.
`convergence`	An integer code. 0 indicates successful completion. 1 indicates that the iteration limit maxit had been reached.

If full = TRUE, apart from the above, components regarding the changes of par, gradient, value, and auxiliary parameters will also be returned.

Examples

```
### Fit a simple linear regression with RMSE as cost function
require(optimg)

# Predictor
x <- seq(-3,3,len=100)
# Criterion
y <- rnorm(100, 2 + {1.2*x}, 1)

# RMSE cost function
fn <- function(par, X) {
  mu <- par[1] + {par[2] * X}
  rmse <- sqrt(mean({y-mu}^2))
  return(rmse)
}

# Compare optimization methods
optim(c(0,0),fn,X=x,method="Nelder-Mead")
optim(c(0,0),fn,X=x,method="BFGS")
optim(c(0,0),fn,X=x,method="CG")
optim(c(0,0),fn,X=x,method="L-BFGS-B")
optimg(c(0,0),fn,X=x,method="ADAM")
optimg(c(0,0),fn,X=x,method="STGD")
```

Index