Package: ttcg (via r-universe)

October 9, 2024

Type Package

Title Three-Term Conjugate Gradient for Unconstrained Optimization

Version 1.1.1

Date 2021-10-14

Maintainer Hao Chi Kiang <hello@hckiang.com>

Description Some accelerated three-term conjugate gradient algorithms implemented purely in R with the same user interface as optim(). The search directions and acceleration scheme are described in Andrei, N. (2013) <doi:10.1016/j.amc.2012.11.097>, Andrei, N. (2013) <doi:10.1016/j.cam.2012.10.002>, and Andrei, N (2015) <doi:10.1007/s11075-014-9845-9>. Line search is done by a hybrid algorithm incorporating the ideas in Oliveia and Takahashi (2020) <doi:10.1145/3423597> and More and Thuente (1994) <doi:10.1145/192115.192132>.

License GPL-3

RoxygenNote 7.1.2

Encoding UTF-8

URL https://git.sr.ht/~hckiang/ttcg

Depends R (>= 3.3.0)

Imports numDeriv

Suggests testthat

Repository https://hckiang.r-universe.dev

RemoteUrl https://github.com/hckiang/ttcg

RemoteRef HEAD

RemoteSha 33608c6f040bb541a368a1f030518d2bc1967c02

Contents

ttcg-package *ttcg: Three-term conjugate gradient minimization algorithms*

Description

Some accelerated three-term conjugate gradient algorithms implemented purely in R with the same user interface as optim(). The search directions and acceleration scheme are described in Andrei, N. (2013) <doi:10.1016/j.amc.2012.11.097>, Andrei, N. (2013) <doi:10.1016/j.cam.2012.10.002>, and Andrei, N (2015) <doi:10.1007/s11075-014-9845-9>. Line search is done by a hybrid algorithm incorporating the ideas in Oliveia and Takahashi (2020) <doi:10.1145/3423597> and More and Thuente (1994) <doi:10.1145/192115.192132>.

Author(s)

Hao Chi Kiang, <hello@hckiang.com>

See Also

Useful links:

- https://git.sr.ht/~hckiang/ttcg

ttcg *Accelerated three-term conjugate gradient optimization with restart*

Description

The ttcg function minimizes a given function using several Neculai Andrei's three-term conjugate gradient algorithms.

Usage

```
ttcg(par, fn, gr = NULL, method = "TTCG", control = list(), ...)
```

Arguments

par	A numerical vector containing the initial parameters.
fn	A function to be minimized. It should accept a numerical vector as its sole argument and return a scaler.
gr	The gradient function of fn. It should accept the same argument as fn and return a vector of the same length as par. If it is NULL then numerical finite difference is used to obtain the gradient.
method	A character string, one of 'TTDES', 'TTCG', 'THREECG'. This determines how the line search direction is computed. 'TTCG' is the default method.
control	A list of control parameters. See Details.
...	Extra arguments to be passed to fn

Details

By default, the algorithm stops when any one of the following three convergence tests is satisfied. (1) The squared Euclidean norm of the squared Euclidean norm of the gradient is smaller than a tolerance; (2) The infinity norm of the gradient is smaller than a tolerance; (3) $|f_{k+1} - f_k| < \epsilon * (1 + |f_k|)$. These three tolerances can be set in the `control` argument, and turnt off by setting them to any negative values. If all three were turnt off, the algorithm may never stop.

For minimization problems, in which `fnscale` is positive, the objective function can return `NaN` or `Inf` but `NA` or `-Inf` will results in the algorithm being stopped immediately, because `-Inf` means the function is unbounded below and any arithmetic error, such as dividing by zero, should be coded as `NaN`; while `NA` should signify programming error. For maximization problems, `-Inf` instead of `Inf` is allowed. The gradient function, similarly, can contain `NaN`, `Inf`, or `-Inf`, but returning `NA` will stop the algorithm.

The `method` argument specifies how the search direction in each step is computed. Please see the three Neculai Andrei's three papers in the citation section for more detailed description. An acceleration scheme and a restart procedure are implemented according to his three papers. Line search is done by a bisection-like weak-Wolfe search described in Oliveira and Takahashi's (2020) interpolate-truncate-project algorithm, but replacing their gradient-secant interpolation with some of More-Thuente's (1994) cubic interpolation idea.

The `control` argument is a list that can contain any of the following named element:

maxit The maximal number of iteration. Default is 500.

fnscale Scalar to divide fn and gr during optimization. If negative, turns the problem into a maximization problem. Optimization is performed on `fn(par)/fnscale`.

parscale A vector of scaling values for the parameters. Optimization is performed on `par/parscale` and these should be comparable in the sense that a unit change in any element produces about a unit change in the scaled value.

gl2tol A positive small number. The iteration will be terminated if the squared Euclidean norm of the gradient is smaller than this number. Default is `min(1e-9, length(par)*1e-10)`. To turn off this test, set it to any negative values.

gmaxtol A positive small number. The iteration will be terminated if the infinity norm of the graident is smaller than `gmaxtol`. Default is 1e-6. To turn off this test, set it to any negative values.

ftol A positive small number. The iteration will be terminated if $|f_{k+1} - f_k| < ftol * (1 + |f_k|)$. To turn off this test, set it to any negative values.

c1 The line search parameter for the sufficient descent condition. Default is 1e-3.

c2 The line search parameter for the curvature condition. Default is 0.08.

trace Either TRUE or FALSE, indicating whether or not details should be printed to the terminal during the optimization. Default is FALSE.

Value

A list containing the following named elements.

par The optimal parameter.

value The optimal function value.

counts An integer vector containing the number of function and gradient calls used during the optimization.

convergence An integer indicating convergence status. '0' means successful convergence; '1' means `maxit` has been reached; '2' means a line search failure in which a point that satisfies the weak Wolfe condition is not found. Among other possibilities, this may happen when the function is unbounded below or the function is non-differentiable.)

message A character string giving additional message.

References

Andrei, N. (2013). On three-term conjugate gradient algorithms for unconstrained optimization. Applied Mathematics and Computation, 219(11), 6316-6327.

Andrei, N. (2013). A simple three-term conjugate gradient algorithm for unconstrained optimization. Journal of Computational and Applied Mathematics, 241, 19-29.

Andrei, N. (2015). A new three-term conjugate gradient algorithm for unconstrained optimization. Numerical Algorithms, 68(2), 305-321.

Oliveira, I. F., & Takahashi, R. H. (2020). An Enhancement of the Bisection Method Average Performance Preserving Minmax Optimality. ACM Transactions on Mathematical Software (TOMS), 47(1), 1-24.

More, J. J., & Thuente, D. J. (1994). Line search algorithms with guaranteed sufficient decrease. ACM Transactions on Mathematical Software (TOMS), 20(3), 286-307.

Examples

```
nqm = rnorm(500)*2
fn  = function (x,nqm1) sum((x - nqm1)^2.)
gr  = function (x,nqm1) 2.*(x - nqm1)
r   = ttcg(par = rnorm(500)*4., fn = fn, gr = gr, method='TTDES', nqm=nqm)
all.equal(r$value, 0.0)
```

Index