Package ‘smco’

February 20, 2015

Type Package

Title A simple Monte Carlo optimizer using adaptive coordinate sampling

Version 0.1

Date 2011-06-05

Author Prof. Juan David Velasquez H. Universidad Nacional de Colombia

Maintainer Prof. Juan David Velasquez H. <jdvelasq@unal.edu.co>

Description This package is for optimizing non-linear complex functions based on Monte Carlo random sampling.

License GPL (>= 2)

LazyLoad yes

Repository CRAN

Date/Publication 2012-10-29 08:59:45

NeedsCompilation no

R topics documented:

`smco-package` *A simple Monte Carlo optimizer using adaptive coordinate sampling*

Description

This package introduces a novel approach to optimize non-linear complex functions based on three simple ideas: first, the thus sampling of each component of the solution vector, one at a time, based on a truncated normal distribution; second, the evolution of the standard deviation of the sampling distribution in each iteration, as a mechanism of self-adaptation; and third, the restart of the algorithm for escaping of local optima.

Details

Package:	smco
Type:	Package
Version:	1.0
Date:	2011-06-05
License:	GPL (>= 2)
LazyLoad:	yes

Unique function:

smco():	Simple Monte Carlo optimizer

Author(s)

Prof. Juan D. Velasquez, Ph.D.
Grupo de Computacion Aplicada
Univesidad Nacional de Colombia
jdvelasq@unal.edu.co

References

Velasquez, J. D. (2011). *A Simple Monte Carlo optimizer based on Adaptive Coordinate Sampling*. Submitted to Operation Research Letters.

Examples

```
SphereModel.fcn <-
function( x ) {
  return(sum(x ^ 2))
}

f = SphereModel.fcn;
ndim = 10;
LB = rep( -600.000, ndim);
UB = rep(  600.000, ndim);
maxiter = 100;

s = smco(par = NULL, fn = SphereModel.fcn, N = ndim, LB = LB,
    UB = UB, maxiter = maxiter, Co = 0.01, Cmin = 0.0001,
    Cmax = 0.5, trc = TRUE, lambda = 20,
    useBFGS = TRUE, control = list(maxit = 10))
```

smco *simple Monte Carlo optimizer using adaptive coordinate sampling*

Description

The function smco is use to optimize non-linear complex functions based on three simple ideas: first, the sampling of each component of the solution vector, one at a time, based on a truncated normal distribution; second, the evolution of the standard deviation of the sampling distribution in each iteration, as a mechanism of self-adaptation; and third, the restart of the algorithm for escaping of local optima.

Usage

```
smco(par = NULL, fn, gr = NULL, ..., N = length(par),
    LB, UB, maxiter = 1000, Co = 0.01, Cmin = 0.0001,
    Cmax = 0.5, trc = FALSE, lambda = 20, useBFGS = FALSE,
    control = list(), hessian = FALSE)
```

Arguments

`par` Initial values for the parameters to be optimized over. When it is ’NULL’ the user must be specify the value for the ’N’ parameter and the algorithm it is initializated in a random point.

`fn` A function to be minimized (or maximized), with first argument the vector of parameters over which minimization is to take place. It should return a scalar result.

`gr` A function to return the gradient for the ’"L-BFGS-B"’ method when the option `trc` is set to TRUE. If it is ’NULL’, a finite-difference approximation will be used.

`...` Further arguments to be passed to ’fn’ and ’gr’.

`N` The number of dimensions of the vector of solution variables

`LB, UB` Lower and upper bounds on the variables.

`maxiter` The maximum number of iterations. The number of function calls to ’"L-BFGS-B"’ implementation is not taking into count.

`Co, Cmin, Cmax` Initial, minimum and maximum values for the standard deviation of the truncated normal distribution

`trc` Logical. When it is ’TRUE’ the progress of the optimization algorithm is reported.

`lambda` Restarting parameter. When the algorithm reaches ’lambda’ * ’N’ iterations without found a new local optimum the restart ocurrs.

`useBFGS` Logical. When it value is ’TRUE’, first, a new random value for the current coordinate is generated using a truncated normal distribution; second, the one-dimensional optimization of the value of the current coordinate is made using the ’"L-BFGS-B"’ method implemented in the `optim` function `control` A list of control parameters for the ’"L-BFGS-B"’ implemented in the `optim` function.

`hessian` A control parameter for the ’"L-BFGS-B"’ implemented in the `optim` function.

Details

The algorithm tries to improve the current solution by changing the i-th coordinate by a random value generated using a truncated normal distribution. Only a random value for each coordinate is generated each time. When the parameter ’useBFGS’ is ’TRUE’, the new random value is used as the initial point for the one-dimensional optimization of the i-th coordinate using ’"L-BFGS-B"’ method implemented in the `optim` function.

The standard deviation of the truncated random deviation changes in each iteration in a random way. The values of the standard deviation are bounded to the interval defined by ’Cmin’ and ’Cmax’.

When the algorithm reaches ’lambda’ * ’N’ iterations without found a new local optimum the restart ocurrs. For this, a new random current solution is generated.

Value

A list with components:

`par` The best set of parameters found.

`value` The value of ’fn’ corresponding to ’par’.

`f.opt, f.min, f.curr`
Vectors of size ’maxiter’ containing the values of the global optimum, local optimum and current value of the function ’fn’ for the current iteration.

`call` The call to the function.

Author(s)

Prof. Juan D. Velasquez, Ph.D.
Grupo de Computacion Aplicada
Univesidad Nacional de Colombia
jdvelasq@unal.edu.co

References

Velasquez, J. D. (2011). *A Simple Monte Carlo optimizer based on Adaptive Coordinate Sampling*. Submitted to Operation Research Letters.

See Also

`optim`

Examples

```
SphereModel.fcn <-
function( x ) {
  return(sum(x ^ 2))
}

f = SphereModel.fcn;
ndim = 10;
LB = rep( -600.000, ndim);
UB = rep(  600.000, ndim);
maxiter = 100;

s = smco(par = NULL, fn = SphereModel.fcn, N = ndim, LB = LB,
    UB = UB, maxiter = maxiter, Co = 0.01, Cmin = 0.0001,
    Cmax = 0.5, trc = TRUE, lambda = 20,
    useBFGS = TRUE, control = list(maxit = 10))
```

Index