Package ‘CEoptim’

January 20, 2025

Type Package

Title Cross-Entropy R Package for Optimization

Version 1.3

Date 2023-10-04

Author Tim Benham and Qibin Duan and Dirk P. Kroese and Benoit Liquet

Maintainer Benoit Liquet <b.liquet@uq.edu.au>

Depends MASS, msm, stats, sna

Description Optimization solver based on the Cross-Entropy method.

License GPL (>= 2.0)

NeedsCompilation no

Repository CRAN

Date/Publication 2023-10-04 04:00:02 UTC

Contents

CEoptim-package *Cross-Entropy R package for optimization*

Description

The CEoptim package provides an optimization solver based on the Cross-Entropy method. The main function `CEoptim` can be used to solve multi-extremal optimization problems involving discrete, continuous, and mixed variables. In addition, `CEoptim` implements linear constraints for continuous optimization.

Author(s)

Tim Benham, Qibin Duan, Dirk P. Kroese, Benoit Liquet <b.liquet@uq.edu.au>

References

Benham T., Duan Q., Kroese D.P., Liquet B. (2017) **CEoptim**: Cross-Entropy R package for optimization. *Journal of Statistical Software*, 76(8), 1-29.

See Also

`CEoptim`

CEoptim *Cross-Entropy optimizer*

Description

`CEopt` is an optimization function based on the Cross-Entropy method

Usage

```
CEoptim(f, f.arg=NULL, maximize=FALSE, continuous=NULL, discrete=NULL,
    N=100L, rho=0.1, iterThr=1e4L, noImproveThr=5, verbose=FALSE)
```

Arguments

`f`	Function to be optimized. Can have continuous and discrete arguments
`f.arg`	List of additional fixed arguments passed to function f.
`maximize`	Logical value determining whether to maximize or minimize the objective function
`continuous`	List of arguments for the continuous optimization part consisting of:

- `mean` Vector of initial means.
- `sd` Vector of initial standard deviations.

- `smoothMean` Smoothing parameter for the vector of means. Default value 1 (no smoothing).
- `smoothSd` Smoothing parameter for the standard deviations. Default value 1 (no smoothing).
- `sdThr` Positive numeric convergence threshold. Check whether the maximum standard deviation is smaller than `sdThr`. Default value 0.001.
- `conMat` Coefficient matrix of linear constraint `conMat` $x \le$ `conVec`.
- `conVec` Value vector of linear constraint `conMat` $x \le$ `conVec`.

`discrete` List of arguments for the discrete optimization part, consisting of:

- `categories` Integer vector which defines the allowed values of the categorical variables. The `i`th categorical variable takes values in the set {0,1,. . . ,`categories(i)-1`}.
- `probs` List of initial probabilities for the categorical variables. Defaults to equal (uniform) probabilities.
- `smoothProb` Smoothing parameter for the probabilities of the categorical sampling distribution. Default value 1 (no smoothing).
- `ProbThr` Positive numeric convergence threshold. Check whether all probabilities in the categorical sampling distributions deviate less than `ProbThr` from either 0 or 1. Default value 0.001.

`N` Integer representing the CE sample size.

`rho` Value between 0 and 1 representing the elite proportion.

`iterThr` Termination threshold on the largest number of iterations.

`noImproveThr` Termination threshold on the largest number of iterations during which no improvement of the best function value is found.

`verbose` Logical value set for CE progress output.

Value

`CEoptim` returns an object of class "CEoptim" which is a list with the following components.

- **optimum** Optimal value of `f`.
- **optimizer** List of the location of the optimal value, consisting of:
  - **continuous** Continuous part of the optimizer.
  - **discrete** Discrete part of the optimizer.
- **termination** List of termination information consisting of:
  - **niter** Total number of iterations upon termination.
  - **convergence** One of the following statements:
    - `Not converged`, if the number of iterations reaches `iterThr`;
    - `The optimum did not change for noImproveThr iterations`, if the best value has not improved for `noImproveThr` iterations;
    - `Variances converged`, otherwise.
- **states** List of intermediate results computed at each iteration. It consists of the iteration number (`iter`), the best overall value (`optimum`) and the worst value of the elite samples, (`gammat`). The means (`mean`) and maximum standard deviations (`maxSd`) of the elite set are also included for continuous cases, and the maximum deviations (`maxProbs`) of the sampling probabilities to either 0 or 1 are included for discrete cases.

- **states.probs** List of categorical sampling probabilities computed at each iteration. Will only be returned for discrete and mixed cases.

Note

Although partial parameter passing is allowed outside lists, it is recommended that parameters names are specified in full. Parameters inside lists have to specified completely.

Because `CEoptim` is a random function it is useful to (1) set the seed for the random number generator (for testing purposes), and (2) investigate the quality of the results by repeating the optimization a number of times.

Author(s)

Tim Benham, Qibin Duan, Dirk P. Kroese, Benoit Liquet

References

Benham T., Duan Q., Kroese D.P., Liquet B. (2017) **CEoptim**: Cross-Entropy R package for optimization. *Journal of Statistical Software*, 76(8), 1-29.

Rubinstein R.Y. and Kroese D.P. (2004). *The Cross-Entropy Method.* Springer, New York.

Examples

```
## Maximizing the Peaks Function


fun <- function(x){
return(3*(1-x[1])^2*exp(-x[1]^2 - (x[2]+1)^2)
-10*(x[1]/5-x[1]^3 - x[2]^5)*exp(-x[1]^2 - x[2]^2)
-1/3*exp(-(x[1]+1)^2 - x[2]^2))}

set.seed(1234)

mu0 <- c(-3,-3); sigma0 <- c(10,10)

res <- CEoptim(fun,continuous=list(mean=mu0, sd=sigma0), maximize=TRUE)

## To extract the Optimal value of fun
res$optimum
## To extract the location of the optimal value
res$optimizer$continuous
## print function gives the following default values
print(res)
```

dirichletrnd *Dirichlet generator*

Description

Random generation for the Dirichlet distribution

Usage

```
dirichletrnd(a, n)
```

Arguments

a	numeric vector for the concentration parameters
n	number of observations

Value

dirichletrnd generates n random observations from a Dirichlet distribution

Author(s)

Tim Benham, Qibin Duan, Dirk P. Kroese, Benoit Liquet

References

Kroese D.P., Taimre T., Botev Z.I. (2011), *Handbook of Monte Carlo Methods*, John Wiley & Sons.

Examples

```
## Generation from the Dirichlet distribution
## with parameter a=(1,2,3,4,5)

set.seed(12345)
a <- 1:5
n <- 10

y <- dirichletrnd(a,n)
y
```

FitzHugh *Simulated data from FitzHugh-Nagumo differential equations*

Description

The data correspond to the values V(t) of the FitzHugh-Nagumo differential equations

V'(t) = c*(V(t) - (V(t)^3)/3 + R(t))

R'(t) = -(1/c)*(V(t) - a + b*R(t))

at times 0, 0.05,..,20.0, with parameters a = 0.2, b = 0.2, c = 3 and initial conditions V(0) = -1, R(0)=1, and adding gaussian noise with standard deviation 0.5.

Usage

```
data(FitzHugh)
```

Format

A numeric vetor of length 401

References

Nagumo, J. and Arimoto, S. and Yoshizawa, S. (1962) An active pulse ransmission line simulating nerve axon, *Proceedings of the IRE*, **50** (10), 2061–2070.

Ramsay, J.O. and Hooker, G. and Campbell, D. and Cao J. (2007) Parameter estimation for differential equations: A generalized smoothing approach, *Journal of the Royal Statistical Society, Series B* **69** (5) 741–796.

Benham T., Duan Q., Kroese D.P., Liquet B. (2017) **CEoptim**: Cross-Entropy R package for optimization. *Journal of Statistical Software*, 76(8), 1-29.

Examples

```
## Plot the data
data(FitzHugh)
plot(FitzHugh,col="blue")
```

lesmis *Network data from Les Miserables*

Description

An R implementation of Donald Knuth's social network graph describing the interaction of characters in Vicor Hugo's novel Les Miserables. Each node represents a character, and edges connect any pair of characters that coappear. The weights of the edges are the number of such coapperances.

Usage

```
data(lesmis)
```

Format

Matrix of weights (77x77)

References

Knuth, D.E. (1993) *The Stanford GraphBase: A Platform for Combinatorial Computing*, ACM Press: Reading MA

Benham T., Duan Q., Kroese D.P., Liquet B. (2017) **CEoptim**: Cross-Entropy R package for optimization. *Journal of Statistical Software*, 76(8), 1-29.

Examples

```
## Display the social network graph
data(lesmis)
gplot(lesmis,gmode="graph")
```

`print` *Print method for the CEoptim object*

Description

Produce `print` method for class `"CEoptim"`

Usage

```
## S3 method for class 'CEoptim'
print(x,...)
```

Arguments

`x`	object of class inheriting from `"CEoptim"`
`...`	additional arguments: `optimizer`; `optimum`; `termination`;`states`;`states.probs`

Details

`print` method for `"CEoptim"` class, returns by default the main description of the `x` object including: `optimizer`; `optimum`; `termination`. To get the `states` and `states.probs` outputs, one should specify the corresponding argument to "TRUE".

Author(s)

Tim Benham, Qibin Duan, Dirk P. Kroese, Benoit Liquet

References

Benham T., Duan Q., Kroese D.P., Liquet B. (2017) **CEoptim**: Cross-Entropy R package for optimization. *Journal of Statistical Software*, 76(8), 1-29.

See Also

CEoptim

Examples

```
## Maximizing the Peaks Function


fun <- function(x){
return(3*(1-x[1])^2*exp(-x[1]^2 - (x[2]+1)^2)
-10*(x[1]/5-x[1]^3 - x[2]^5)*exp(-x[1]^2 - x[2]^2)
-1/3*exp(-(x[1]+1)^2 - x[2]^2))}

set.seed(1234)

mu0 <- c(-3,-3); sigma0 <- c(10,10)

res <- CEoptim(fun,continuous=list(mean=mu0, sd=sigma0), maximize=TRUE)

## Print method provides by default
## optimizer; optimum and termination.
print(res)
## To print only the Optimal value of fun
print(res,optimum=TRUE)
## To print only the location of the optimal value
print(res,optimizer=TRUE)
## To print only termination information
print(res,termination=TRUE)
```

yt *Simulated cumulative data from an AR(1) model with regime switching*

Description

yt represents the added value of a stock at time t, at day t=1,2,...,300; that is, the increase (which may be negative) in stock price relative to the price at time t=0.

Usage

```
data(yt)
```

Format

Numeric vector of length 300

References

Benham T., Duan Q., Kroese D.P., Liquet B. (2017) **CEoptim**: Cross-Entropy R package for optimization. *Journal of Statistical Software*, 76(8), 1-29.

Examples

```
## Plot the yt data
data(yt)
plot(yt,type="l",col="blue")
```

Index