

Package: simpleMH (via r-universe)

September 6, 2024

Title Simple Metropolis-Hastings MCMC Algorithm

Version 0.1.1.9000

Description A very bare-bones interface to use the Metropolis-Hastings Monte Carlo Markov Chain algorithm. It is suitable for teaching and testing purposes.

Imports mvtnorm

Suggests coda, mockery, testthat (>= 3.0.0), knitr, rmarkdown

License GPL-3

Config/testthat/edition 3

Encoding UTF-8

Language en-GB

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1.9001

VignetteBuilder knitr

URL <https://github.com/Bisaloo/simpleMH>

BugReports <https://github.com/Bisaloo/simpleMH/issues>

Repository <https://bisaloo.r-universe.dev>

RemoteUrl <https://github.com/bisaloo/simpleMH>

RemoteRef HEAD

RemoteSha 3f99c5b2ab26fa4297b79a1ac87a1f870dc23510

Contents

simpleMH	2
Index	3

simpleMH

*Simple Metropolis-Hastings MCMC***Description**

Simple Metropolis-Hastings MCMC

Usage

```
simpleMH(f, inits, theta.cov, max.iter, coda = FALSE, ...)
```

Arguments

f	function that returns a single scalar value proportional to the log probability density to sample from.
inits	numeric vector with the initial values for the parameters to estimate
theta.cov	covariance matrix of the parameters to estimate.
max.iter	maximum number of function evaluations
coda	logical. Should the samples be returned as <code>coda::mcmc</code> object? (defaults to FALSE)
...	further arguments passed to f

Value

- if coda = FALSE a list with:
 - *samples*: A two dimensional array of samples with dimensions generation x parameter
 - *log.p*: A numeric vector with the log density evaluate at each generation.
- if coda = TRUE a list with:
 - *samples*: A object of class `coda::mcmc` containing all samples.
 - *log.p*: A numeric vector with the log density evaluate at each generation.

Examples

```
p.log <- function(x) {
  B <- 0.03
  return(-x[1]^2/200 - 1/2*(x[2]+B*x[1]^2-100*B)^2)
}

simpleMH(p.log, inits=c(0, 0), theta.cov = diag(2), max.iter=3000)
```

Index

`coda::mcmc`, 2

`simpleMH`, 2