

Package ‘aMTM’

December 15, 2018

Type Package

Title Adaptive Multiple-Try Metropolis Algorithm

Version 0.1.0

Author person(`Simon`, ``Fontaine`, role = ``aut`, email = ``fontaines@dms.umontreal.ca`")

Maintainer Simon Fontaine <fontaines@dms.umontreal.ca>

Description Produces a Monte Carlo sample from a continuous distribution of a random vector using a Markov Chain Monte Carlo (MCMC) algorithm. In particular, an adaptive version of the Multiple-Try Metropolis algorithm of Liu et al. (2001) is implemented: details of the algorithm can be found in Fontaine and Bedard (2019). The sample can then be used to perform a Monte Carlo estimation of the expectation of a function of the random vector and standard MCMC techniques can be done (standard error estimation, diagnostic of convergence, etc.).

License GPL-2

URL <https://github.com/fontaine618/aMTM/>

Repository GitHub

Depends BH, RcppArmadillo

NeedsCompilation Yes

Encoding UTF-8

LazyData true

Imports Rcpp (>= 0.12.15), RcppArmadillo, BH

LinkingTo Rcpp, RcppArmadillo, BH

RoxygenNote 6.0.1

Archs x64

R topics documented:

aMTM 2

Index 4

aMTM

*Adaptive Multiple-Try Metropolis algorithm***Description**

This function performs the Adaptive Multiple-Try Metropolis algorithm as described in Fontaine and Bedard (2019). The sampling step is performed via a MTM algorithm with K gaussian proposals which may be correlated (see argument `proposal`) and using either weights that are proportional to the target density or importance weights. The adaptation step is performed via one of AM, ASWAM or RAM updates of the selected proposal density; a global component may also be adapted at each iteration (see argument `global`) and the scale parameters may be adapted at each iteration (see argument `scale`). The AM and ASWAM update may be done using local steps rather than global steps (see argument `local`).

Usage

```
aMTM(target, N, K, x0, ...)
```

Arguments

<code>target</code>	Target log-density which must be vectorized, i.e. take input of dimension $K \times d$. Additional parameters must be passed as a list through the <code>parms</code> argument.
<code>N</code>	Size of MCMC sample.
<code>K</code>	Number of proposals in the MTM sampling.
<code>x0</code>	A vector of dimension d corresponding to the initial state of the chain.
<code>sig0</code>	An array of dimension $d \times d \times K$ containing the K initial covariance of the instrumental gaussian distributions. Default is K identity matrices.
<code>mu0</code>	A matrix of dimension $d \times K$ containing the K initial mean parameters for AM and ASWAM updates. Default is K zero vectors.
<code>lam0</code>	A vector of dimension K containing the K scale parameters. Default is $(2.38)^2/d$ for AM and AWSAM updates and 1 for RAM updates.
<code>adapt</code>	f
<code>global</code>	f
<code>scale</code>	f
<code>local</code>	f
<code>proposal</code>	f
<code>accrate</code>	f
<code>gamma</code>	f
<code>parms</code>	f
<code>weight</code>	f
<code>burnin</code>	f

Value

A list containing the following elements:

`x` The MCMC sample in a matrix of dimension $N \times d$.

`acc.rate` The MCMC sample in a matrix.

`sel.prop` The MCMC sample in a matrix.

`mu` The MCMC sample in a matrix.

`lam` The MCMC sample in a matrix.

`Sig` The MCMC sample in a matrix.

`sel` The MCMC sample in a matrix.

`time` The MCMC sample in a matrix.

Author(s)

Simon Fontaine, <fontaines@dms.umontreal.ca>

References

Andrieu, C. and Thoms, J. (2008). "A tutorial on adaptive MCMC". *Statistics and computing*, 18:4, 343-373.

Fontaine, S. and Bedard, M. (2019). "An Adaptive Multiple-Try Metropolis algorithm". To be submitted.

Haario, H., Saksman, E., Tamminen, J. et al. (2001). "An adaptive Metropolis algorithm". *Bernoulli*, 7:2, 223-242.

Liu, J.S., Liang, F. and Wong, W.H. (2000). "The Multiple-Try Method and Local Optimization in Metropolis Sampling". *Journal of the American Statistical Association*, 95:449, 121-134.

Vihola, M. (2012). "Robust adaptive Metropolis algorithm with coerced acceptance rate". *Statistics and Computing*, 22:5, 997-1008.

Index

aMTM, [2](#)