R topics documented:

MCMCEnsemble .............................................................................. 2

Description

Ensemble Markov Chain Monte Carlo sampler with different strategies to generate proposals. Either the stretch move as proposed by Goodman and Weare (2010), or a differential evolution jump move similar to Braak and Vrugt (2008).

Usage

MCMCEnsemble(
  f,
  inits,
  max.iter,
  n.walkers = 10 * ncol(inits),
  method = c("stretch", "differential.evolution"),
  coda = FALSE,
  ...
)

Arguments

f function that returns a single scalar value proportional to the log probability density to sample from.
inits A matrix (or data.frame) containing the starting values for the walkers. Each column is a variable to estimate and each row is a walker
max.iter maximum number of function evaluations
n.walkers number of walkers (ensemble size)
method method for proposal generation, either "stretch", or "differential.evolution". This argument will be saved as an attribute in the output (see examples).
coda logical. Should the samples be returned as coda::mcmc.list object? (defaults to FALSE)
...

Value

• if coda = FALSE a list with:
  – samples: A three dimensional array of samples with dimensions walker x generation x parameter
MCMCEmsemble

- $\log p$: A matrix with the log density evaluate for each walker at each generation.
- if coda = TRUE a list with:
  - samples: A object of class coda::mcmc.list containing all samples.
  - $\log p$: A matrix with the log density evaluate for each walker at each generation.

In both cases, there is an additional attribute (accessible via attr(res, "ensemble.sampler")) recording which ensemble sampling algorithm was used.

References


Examples

```r
## a log-pdf to sample from
p.log <- function(x) {
  B <- 0.03 # controls 'bananacity'
}

## set options and starting point
n_walkers <- 10
unif_inits <- data.frame(
  "a" = runif(n_walkers, 0, 1),
  "b" = runif(n_walkers, 0, 1)
)

## use stretch move
res1 <- MCMCEmsemble(p.log, inits = unif_inits, max.iter = 300, n.walkers = n_walkers, method = "stretch")
attr(res1, "ensemble.sampler")
str(res1)

## use stretch move, return samples as 'coda' object
res2 <- MCMCEmsemble(p.log, inits = unif_inits, max.iter = 300, n.walkers = n_walkers, method = "stretch", coda = TRUE)
attr(res2, "ensemble.sampler")
summary(res2$samples)
plot(res2$samples)
```
## use different evolution move, return samples as 'coda' object

```r
res3 <- MCMCEnsemble(p.log, inits = unif_inits,
                      max.iter = 300, n.walkers = n_walkers,
                      method = "differential.evolution", coda = TRUE)

attr(res3, "ensemble.sampler")

summary(res3$samples)
plot(res3$samples)
```
Index

coda::mcmc.list, 2, 3

MCMCEnsemble, 2