Package 'mcmcensemble'

February 20, 2021

Title Ensemble Sampler for Affine-Invariant MCMC

Version 2.2.0

Description Provides ensemble samplers for
affine-invariant Monte Carlo Markov Chain, which allow a faster
convergence for badly scaled estimation problems. Two samplers are
proposed: the 'differential.evolution' sampler from ter Braak and
Vrugt (2008) <doi:10.1007/s11222-008-9104-9> and the 'stretch' sampler
from Goodman and Weare (2010) <doi:10.2140/camcos.2010.5.65>.

License GPL-2

URL https://github.com/Bisaloo/mcmcensemble,
https://bisaloo.github.io/mcmcensemble/

BugReports https://github.com/Bisaloo/mcmcensemble/issues

Depends R (>= 3.5)

Imports future.apply, progressr

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

Encoding UTF-8

RoxygenNote 7.1.1.9001

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Hugo Gruson [cre, aut, cph] (<https://orcid.org/0000-0002-4094-1476>),
Sanda Dejanic [aut, cph],
Andreas Scheidegger [aut, cph]
(<https://orcid.org/0000-0003-2575-2172>)

Maintainer Hugo Gruson <hugo.gruson+R@normalesup.org>

Repository CRAN

Date/Publication 2021-02-20 09:10:02 UTC

R topics documented:

`d.e.mcmc` *MCMC Ensemble sampler with the differential evolution jump move*

Description

Markov Chain Monte Carlo sampler: using the differential evolution jump move (implementation of the ter Braak differential evolution)

Usage

```
d.e.mcmc(f, lower.inits, upper.inits, max.iter, n.walkers, ...)
```

Arguments

`f`	function that returns a single scalar value proportional to the log probability density to sample from.
`lower.inits`	vector specifying for each parameter the lower value the initial distribution.
`upper.inits`	vector specifying for each parameter the upper value the initial distribution.
`max.iter`	maximum number of function evaluations
`n.walkers`	number of walkers (ensemble size)
`...`	further arguments passed to `f`

Value

List containing:

- `samples[n.walkers,chain.length,n.dim]`
- `log.p[n.walkers,chain.length]`

Author(s)

Sanda Dejanic

References

ter Braak, C. J. F. and Vrugt, J. A. (2008) Differential Evolution Markov Chain with snooker updater and fewer chains. Statistics and Computing, 18(4), 435–446, doi: 10.1007/s1122200891049

`MCMCEnsemble` *MCMC ensemble sampler*

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,
  lower.inits,
  upper.inits,
  max.iter,
  n.walkers = 10 * length(lower.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.
`lower.inits`	vector specifying for each parameter the lower value the initial distribution.
`upper.inits`	vector specifying for each parameter the upper value the initial distribution.
`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`)
`...`	further arguments passed to `f`

Value

- if `coda = FALSE` a list with:
  - *samples*: A three dimensional array of samples with dimensions `walker` x `generation` x `parameter`
  - *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

- ter Braak, C. J. F. and Vrugt, J. A. (2008) Differential Evolution Markov Chain with snooker updater and fewer chains. Statistics and Computing, 18(4), 435–446, doi: 10.1007/s11222-00891049
- Goodman, J. and Weare, J. (2010) Ensemble samplers with affine invariance. Communications in Applied Mathematics and Computational Science, 5(1), 65–80, doi: 10.2140/camcos.2010.5.65

Examples

```
## a log-pdf to sample from
p.log <- function(x) {
    B <- 0.03                           # controls 'bananacity'
    -x[1]^2/200 - 1/2*(x[2]+B*x[1]^2-100*B)^2
}

## use stretch move
res1 <- MCMCEnsemble(p.log, lower.inits=c(a=0, b=0), upper.inits=c(a=1, b=1),
                     max.iter=300, n.walkers=10, method="stretch")

attr(res1, "ensemble.sampler")

str(res1)


## use stretch move, return samples as 'coda' object
res2 <- MCMCEnsemble(p.log, lower.inits=c(a=0, b=0), upper.inits=c(a=1, b=1),
                     max.iter=300, n.walkers=10, method="stretch",
                     coda=TRUE)

attr(res2, "ensemble.sampler")

summary(res2$samples)
plot(res2$samples)


## use different evolution move, return samples as 'coda' object
res3 <- MCMCEnsemble(p.log, lower.inits=c(a=0, b=0), upper.inits=c(a=1, b=1),
                     max.iter=300, n.walkers=10,
                     method="differential.evolution", coda=TRUE)

attr(res3, "ensemble.sampler")

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

`s.m.mcmc` *MCMC Ensemble sampler with the stretch move (emcee)*

Description

Markov Chain Monte Carlo sampler: using the stretch move (implementation of the Goodman and Ware emcee)

Usage

```
s.m.mcmc(f, lower.inits, upper.inits, max.iter, n.walkers, ...)
```

Arguments

`f`	function that returns a single scalar value proportional to the log probability density to sample from.
`lower.inits`	vector specifying for each parameter the lower value the initial distribution.
`upper.inits`	vector specifying for each parameter the upper value the initial distribution.
`max.iter`	maximum number of function evaluations
`n.walkers`	number of walkers (ensemble size)
`...`	further arguments passed to `f`

Value

List containing:

- `samples[n.walkers,chain.length,n.dim]`
- `log.p[n.walkers,chain.length]`

Author(s)

Sanda Dejanic

References

Goodman, J. and Weare, J. (2010) Ensemble samplers with affine invariance. Communications in Applied Mathematics and Computational Science, 5(1), 65–80, doi: 10.2140/camcos.2010.5.65

Index