Package: elhmc (via r-universe)

August 26, 2024

Type Package

Title Sampling from a Empirical Likelihood Bayesian Posterior of Parameters Using Hamiltonian Monte Carlo

Version 1.2.0

Date 2024-08-20

Description A tool to draw samples from a Empirical Likelihood Bayesian posterior of parameters using Hamiltonian Monte Carlo.

Imports emplik, plyr, stats, MASS, graphics

License GPL-2

RoxygenNote 7.3.2

NeedsCompilation no

Author Dang Trung Kien [aut], Sanjay Chaudhuri [aut, cre], Neo Han Wei, [aut]

Maintainer Sanjay Chaudhuri <schaudhuri2@unl.edu>

Repository CRAN

Date/Publication 2024-08-25 20:00:02 UTC

Contents

`ELHMC` *Empirical Likelihood Hamiltonian Monte Carlo Sampling*

Description

This function draws samples from a Empirical Likelihood Bayesian posterior distribution of parameters using Hamiltonian Monte Carlo.

Usage

```
ELHMC(
  initial,
  data,
  fun,
  dfun,
  prior,
  dprior,
  n.samples = 100,
  lf.steps = 10,
  epsilon = 0.05,
  p.variance = 1,
  tol = 10^-5,
  detailed = FALSE,
  print.interval = 1000,
  plot.interval = 0,
  which.plot = NULL,
  FUN,
  DFUN
)
```

Arguments

`initial`	a vector containing the initial values of the parameters
`data`	a matrix containing the data
`fun`	the estimating function g. It takes in a parameter vector `params` as the first argument and a data point vector `x` as the second parameter. This function returns a vector.
`dfun`	a function that calculates the gradient of the estimating function g. It takes in a parameter vector `params` as the first argument and a data point vector `x` as the second argument. This function returns a matrix.
`prior`	a function with one argument `x` that returns the prior densities of the parameters of interest
`dprior`	a function with one argument `x` that returns the gradients of the log densities of the parameters of interest
`n.samples`	number of samples to draw
`lf.steps`	number of leap frog steps in each Hamiltonian Monte Carlo update
`epsilon`	the leap frog step size(s). This has to be a single numeric value or a vector of the same length as `initial`.
`p.variance`	the covariance matrix of a multivariate normal distribution used to generate the initial values of momentum p in Hamiltonian Monte Carlo. This can also be a single numeric value or a vector. See Details.
`tol`	EL tolerance
`detailed`	If this is set to `TRUE`, the function will return a list with extra information.

`print.interval`	the frequency at which the results would be printed on the terminal. Defaults to 1000.
`plot.interval`	the frequency at which the drawn samples would be plotted. The last half of the samples drawn are plotted after each plot.interval steps. The acceptance rate is also plotted. Defaults to 0, which means no plot.
`which.plot`	the vector of parameters to be plotted after each plot.interval. Defaults to NULL, which means no plot.
`FUN`	the same as `fun` but takes in a matrix `X` instead of a vector `x` and returns a matrix so that `FUN(params, X)[i, ]` is the same as `fun(params, X[i, ])`. Only one of `FUN` and `fun` should be provided. If both are then `fun` is ignored.
`DFUN`	the same as `dfun` but takes in a matrix `X` instead of a vector `x` and returns an array so that `DFUN(params, X)[, , i]` is the same as `dfun(params, X[i, ])`. Only one of `DFUN` and `dfun` should be provided. If both are then `dfun` is ignored.

Details

Suppose there are data $x = (x_1, x_2, ..., x_n)$ where x_i takes values in R^p and follow probability distribution F. Also, F comes from a family of distributions that depends on a parameter $\theta = (\theta_1, ..., \theta_d)$ and there is a smooth function $g(x_i, \theta) = (g_1(x_i, \theta), ..., g_q(x_i, \theta))^T$ that satisfies $E_F[g(x_i, \theta)] = 0$ for $i = 1, ..., n$.

`ELHMC` draws samples from a Empirical Likelihood Bayesian posterior distribution of the parameter θ, given the data x as `data`, the smoothing function g as `fun`, and the gradient of g as `dfun` or $G(X) = (g(x_1), g(x_2), ..., g(x_n))^T$ as `FUN` and the gradient of G as `DFUN`.

Value

The function returns a list with the following elements:

`samples`	A matrix containing the parameter samples
`acceptance.rate`	The acceptance rate
`call`	The matched call

If `detailed = TRUE`, the list contains these extra elements:

`proposed`	A matrix containing the proposed values at `n.samaples - 1` Hamiltonian Monte Carlo updates
`acceptance`	A vector of `TRUE/FALSE` values indicates whether each proposed value is accepted
`trajectory`	A list with 2 elements `trajectory.q` and `trajectory.p`. These are lists of matrices contraining position and momentum values along trajectory in each Hamiltonian Monte Carlo update.

References

Chaudhuri, S., Mondal, D. and Yin, T. (2017) Hamiltonian Monte Carlo sampling in Bayesian empirical likelihood computation. *Journal of the Royal Statistical Society: Series B*.

Neal, R. (2011) MCMC for using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo* (eds S. Brooks, A.Gelman, G. L.Jones and X.-L. Meng), pp. 113-162. New York: Taylor and Francis.

Examples

```
## Not run:
## Suppose there are four data points (1, 1), (1, -1), (-1, -1), (-1, 1)
x = rbind(c(1, 1), c(1, -1), c(-1, -1), c(-1, 1))
## If the parameter of interest is the mean, the smoothing function and
## its gradient would be
f <- function(params, x) {
 x - params
}
df <- function(params, x) {
 rbind(c(-1, 0), c(0, -1))
}
## Draw 50 samples from the Empirical Likelihood Bayesian posterior distribution
## of the mean, using initial values (0.96, 0.97) and standard normal distributions
## as priors:
normal_prior <- function(x) {
   exp(-0.5 * x[1] ^ 2) / sqrt(2 * pi) * exp(-0.5 * x[2] ^ 2) / sqrt(2 * pi)
}
normal_prior_log_gradient <- function(x) {
   -x
}
set.seed(1234)
mean.samples <- ELHMC(initial = c(0.96, 0.97), data = x, fun = f, dfun = df,
                     n.samples = 50, prior = normal_prior,
                     dprior = normal_prior_log_gradient)
plot(mean.samples$samples, type = "l", xlab = "", ylab = "")

## End(Not run)
```

Index