Package ‘geommc’

July 22, 2025

Title Geometric Markov Chain Sampling

Type Package

Version 0.1.1

Maintainer Vivekananda Roy <`vroy@iastate.edu`>

Date 2024-10-18

Description Simulates from discrete and continuous target distributions
using geometric Metropolis-Hastings (MH)
algorithms. Users specify the target distribution
by an R function that evaluates the log un-normalized pdf or pmf. The package
also contains a function implementing a specific geometric MH algorithm for
performing high dimensional Bayesian variable selection.

Encoding UTF-8

RoxygenNote 7.3.2

Imports Rcpp (>= 1.0.12), cubature, magrittr, Matrix, matrixcalc, mcmc

LinkingTo Rcpp

URL `https://github.com/vroys/geommc`

License GPL (>= 3)

NeedsCompilation yes

Author Vivekananda Roy [aut, cre] (ORCID:
<`https://orcid.org/0000-0002-2964-9503`>)

Repository CRAN

Date/Publication 2024-10-18 19:30:01 UTC

Contents

`geomc` *Markov chain Monte Carlo for discrete and continuous distributions using geometric MH algorithms.*

Description

geomc produces Markov chain samples from a target distribution. The target can be a pdf or pmf. Users specify the target distribution by an R function that evaluates the log un-normalized pdf or pmf. geomc uses the geometric approach of Roy (2024) to move an uninformed base density (e.g. a random walk proposal) towards different global/local approximations of the target density. The base density can be specified along with its mean, covariance matrix, and a function for sampling from it. Gaussian densities can be specified by mean and variance only, although it is preferred to supply its density and sampling functions as well. If either or both of the mean and variance arguments and any of the density and sampling functions is missing, then a base density corresponding to a random walk with an appropriate scale parameter is used. One or more approximate target densities can be specified along with their means, covariance matrices, and a function for sampling from the densities. Gaussian densities can be specified by mean and variance only, although it is preferred to supply their densities and sampling functions as well. If either or both of the mean and variance arguments and any of the density and sampling functions is missing for the approximate target density, then a normal distribution with mean computed from a pilot run of a random walk Markov chain and a diagonal covariance matrix with a large variance is used. If the Argument gaus is set as FALSE then both the base and the approximate target can be specified by their densities and functions for sampling from it. That is, if gaus=FALSE, the functions specifying the means and variances of both the base and the approximate target densities are not used. If the target is a pmf (discrete distribution), then gaus=FALSE and imp [1]=TRUE (not the default values) need to be specified.

Usage

```
geomc(
  log.target,
  initial,
  n.iter,
  eps = 0.5,
  ind = FALSE,
  gaus = TRUE,
  imp = c(FALSE, n.samp = 1000, samp.base = FALSE),
  a = 1,
  mean.base,
  var.base,
  dens.base,
  samp.base,
  mean.ap.tar,
  var.ap.tar,
  dens.ap.tar,
  samp.ap.tar
)
```

Arguments

`log.target`	is the logarithm of the (un-normalized) target density function, needs to be written as a function of the current value x.
`initial`	is the initial values.
`n.iter`	is the no. of samples needed.
`eps`	is the value for epsilon perturbation. Default is 0.5.
`ind`	is False if either the base density, f or the approximate target density, g depends on the current value x. Default is False.
`gaus`	is True if both f and g are normal distributions. Default is True.
`imp`	is a vector of three elements. If gaus is TRUE, then the imp argument is not used. imp [1] is False if numerical integration is used, otherwise, importance sampling is used to compute $\langle\sqrt{f}, \sqrt{g}\rangle$. Default is False. imp [2] (n.samp) is no of samples in importance sampling. imp [3] (samp.base) is True if samples from f is used, otherwise samples from g is used. Default is False.
`a`	is the probability vector for the mixture proposal density. Default is the uniform distribution.
`mean.base`	is the mean of the base density f, needs to be written as a function of the current value x.
`var.base`	is the covariance matrix of the base density f, needs to be written as a function of the current value x.
`dens.base`	is the density function of the base density f, needs to be written as a function (y, x) (in this order) of the proposed value y and the current value x, although it may not depend on x.
`samp.base`	is the function to draw from the base density f, needs to be written as a function of the current value x.
`mean.ap.tar`	is the vector of means of the densities $g_i(y\|x), i = 1, \ldots, k$. It needs to be written as a function of the current value x. It must have the same dimension as k times the length of initial.
`var.ap.tar`	is the matrix of covariance matrices of the densities $g_i(y\|x), i = 1, \ldots, k$ formed by column concatenation. It needs to be written as a function of the current value x. It must have the same dimension as the length of initial by k times the length of initial.
`dens.ap.tar`	is the vector of densities $g_i(y\|x), i = 1, \ldots, k$. It needs to be written as a function (y, x) (in this order) of the proposed value y and the current value x, although it may not depend on x.
`samp.ap.tar`	is the function to draw from the densities $g_i(y\|x), i = 1, \ldots, k$. It needs to be written as a function of (current value x, the indicator of mixing component kk). It must return a vector of the length of that of the initial.

Details

Using a geometric Metropolis-Hastings algorithm geom.mc produces Markov chains with the target as its stationary distribution. The details of the method can be found in Roy (2024).

Value

The function returns a list with the following elements:

`samples`	A matrix containing the MCMC samples. Each column is one sample.
`acceptance.rate`	The acceptance rate.

Author(s)

Vivekananda Roy vroy@iastate.edu

References

Roy, V.(2024) A geometric approach to informative MCMC sampling https://arxiv.org/abs/2406.09010

Examples

```
result <- geomc(log.target=function(y) dnorm(y,log=TRUE),initial=0,n.iter=500)
#target is univariate normal
result$samples # the MCMC samples.
result$acceptance.rate # the acceptance rate.
result<-geomc(log.target=function(y) log(0.5*dnorm(y)+0.5*dnorm(y,mean=10,sd=1.4)),
initial=0,n.iter=500) #target is mixture of univariate normals, default choices
hist(result$samples)
result<-geomc(log.target=function(y) log(0.5*dnorm(y)+0.5*dnorm(y,mean=10,sd=1.4)),
initial=0,n.iter=500, mean.base = function(x) x,var.base= function(x) 4,
dens.base=function(y,x) dnorm(y, mean=x,sd=2),samp.base=function(x) x+2*rnorm(1),
mean.ap.tar=function(x) c(0,10),var.ap.tar=function(x) c(1,1.4^2),
dens.ap.tar=function(y,x) c(dnorm(y),dnorm(y,mean=10,sd=1.4)),
samp.ap.tar=function(x,kk=1){if(kk==1){return(rnorm(1))} else{return(10+1.4*rnorm(1))}})
#target is mixture of univariate normals, random walk base density, an informed
#choice for dens.ap.tar
hist(result$samples)
samp.ap.tar=function(x,kk=1){s.g=sample.int(2,1,prob=c(.5,.5))
if(s.g==1){return(rnorm(1))
}else{return(10+1.4*rnorm(1))}}
result<-geomc(log.target=function(y) log(0.5*dnorm(y)+0.5*dnorm(y,mean=10,sd=1.4)),
initial=0,n.iter=500,gaus=FALSE,imp=c(TRUE,n.samp=100,samp.base=TRUE),
dens.base=function(y,x) dnorm(y, mean=x,sd=2),samp.base=function(x) x+2*rnorm(1),
dens.ap.tar=function(y,x) 0.5*dnorm(y)+0.5*dnorm(y,mean=10,sd=1.4),
samp.ap.tar=samp.ap.tar)
#target is mixture of univariate normals, random walk base density, another
#informed choice for dens.ap.tar
hist(result$samples)
result <- geomc(log.target=function(y) -0.5*crossprod(y),initial=rep(0,4),
n.iter=500) #target is multivariate normal, default choices
rowMeans(result$samples)
size=5
result <- geomc(log.target = function(y) dbinom(y, size, 0.3, log = TRUE),
initial=0,n.iter=500,ind=TRUE,gaus=FALSE,imp=c(TRUE,n.samp=1000,samp.base=TRUE),
dens.base=function(y,x) 1/(size+1), samp.base= function(x) sample(seq(0,size,1),1),
dens.ap.tar=function(y,x) dbinom(y, size, 0.7),samp.ap.tar=function(x,kk=1) rbinom(1, size, 0.7))
```

```
#target is binomial
table(result$samples)
```

`geomc.vs` *Markov chain Monte Carlo for Bayesian variable selection using a geometric MH algorithm.*

Description

geomc.vs uses a geometric approach to MCMC for performing Bayesian variable selection. It produces MCMC samples from the posterior density of a Bayesian hierarchical feature selection model.

Usage

```
geomc.vs(
  X,
  y,
  initial = NULL,
  n.iter = 50,
  burnin = 1,
  eps = 0.5,
  symm = TRUE,
  move.prob = c(0.4, 0.4, 0.2),
  lam0 = 0,
  a0 = 0,
  b0 = 0,
  lam = nrow(X)/ncol(X)^2,
  w = sqrt(nrow(X))/ncol(X),
  model.summary = FALSE,
  model.threshold = 0.5
)
```

Arguments

`X`	The $n \times p$ covariate matrix without intercept. The following classes are supported: `matrix` and `dgCMatrix`. No need to center or scale this matrix manually. Scaling is performed implicitly and regression coefficients are returned on the original scale.
`y`	The response vector of length n. No need to center or scale.
`initial`	is the initial model (the set of active variables). Default: Null model.
`n.iter`	is the no. of samples needed. Default: 50.
`burnin`	is the value of burnin used to compute the median probability model. Default: 1.
`eps`	is the value for epsilon perturbation. Default: 0.5.

`symm`	indicates if the base density is of symmetric RW-MH. Default: True.
`move.prob`	is the vector of ('addition', 'deletion', 'swap') move probabilities. Default: (0.4,0.4,0.2). move.prob is used only when symm is set to False.
`lam0`	The precision parameter for β_0. Default: 0 (corresponding to improper uniform prior).
`a0`	The shape parameter for prior on σ^2. Default: 0.
`b0`	The scale parameter for prior on σ^2. Default: 0.
`lam`	The slab precision parameter. Default: n/p^2 as suggested by the theoretical results of Li, Dutta, Roy (2023).
`w`	The prior inclusion probability of each variable. Default: $\sqrt{n}/p$.
`model.summary`	If true, additional summaries are returned. Default: FALSE.
`model.threshold`	The threshold probability to select the covariates for the median model (median.model) and the weighted average model (wam). A covariate will be included in median.model (wam) if its marginal inclusion probability (weighted marginal inclusion probability) is greater than the threshold. Default: 0.5.

Details

geomc.vs provides MCMC samples using the geometric MH algorithm of Roy (2024) for variable selection based on a hierarchical Gaussian linear model with priors placed on the regression coefficients as well as on the model space as follows:

$$y|X,\beta_0,\beta,\gamma,\sigma^2,w,\lambda \sim N(\beta_0 1 + X_\gamma\beta_\gamma, \sigma^2 I_n)$$

$$\beta_i|\beta_0,\gamma,\sigma^2,w,\lambda \overset{indep.}{\sim} N(0,\gamma_i\sigma^2/\lambda),\ i=1,\ldots,p,$$

$$\beta_0|\gamma,\sigma^2,w,\lambda \sim N(0,\sigma^2/\lambda_0)$$

$$\sigma^2|\gamma,w,\lambda \sim Inv-Gamma(a_0,b_0)$$

$$\gamma_i|w,\lambda \overset{iid}{\sim} Bernoulli(w)$$

where X_γ is the $n\times|\gamma|$ submatrix of X consisting of those columns of X for which $\gamma_i = 1$ and similarly, β_γ is the $|\gamma|$ subvector of β corresponding to γ. The density $\pi(\sigma^2)$ of $\sigma^2 \sim Inv - Gamma(a_0,b_0)$ has the form $\pi(\sigma^2)\propto(\sigma^2)^{-a_0-1}\exp(-b_0/\sigma^2)$. The functions in the package also allow the non-informative prior $(\beta_0,\sigma^2)|\gamma,w \sim 1/\sigma^2$ which is obtained by setting $\lambda_0 = a_0 = b_0 = 0$. geomc.vs provides the empirical MH acceptance rate and MCMC samples from the posterior pmf of the models $P(\gamma|y)$, which is available up to a normalizing constant. If `model.summary` is set TRUE, geomc.vs also returns other model summaries. In particular, it returns the marginal inclusion probabilities (mip) computed by the Monte Carlo average as well as the weighted marginal inclusion probabilities (wmip) computed with weights

$$w_i = P(\gamma^{(i)}|y)/\sum_{k=1}^{K}P(\gamma^{(k)}|y), i=1,2,...,K$$

where $\gamma^{(k)}, k = 1, 2, ..., K$ are the distinct models sampled. Thus, if N_k is the no. of times the kth distinct model $\gamma^{(k)}$ is repeated in the MCMC samples, the mip for the jth variable is

$$mip_j = \sum_{k=1}^{K} N_k I(\gamma_j^{(k)} = 1)/n.iter$$

and wmip for the jth variable is

$$wmip_j = \sum_{k=1}^{K} w_k I(\gamma_j^{(k)} = 1).$$

The median.model is the model containing variables j with $mip_j >$ `model.threshold` and the wam is the model containing variables j with $wmip_j >$ `model.threshold`. Note that $E(\beta|\gamma, y)$, the conditional posterior mean of β given a model γ is available in closed form (see Li, Dutta, Roy (2023) for details). geomc.vs returns two estimates (beta.mean, beta.wam) of the posterior mean of β computed as

$$beta.mean = \sum_{k=1}^{K} N_k E(\beta|\gamma^{(k)}, y)/n.iter$$

and

$$beta.wam = \sum_{k=1}^{K} w_k E(\beta|\gamma^{(k)}, y),$$

respectively.

Value

A list with components

`samples`	MCMC samples from $P(\gamma\|y)$ returned as a $p\times$n.iter sparse `lgCMatrix`.
`acceptance.rate`	The acceptance rate based on all samples.
`mip`	The p vector of marginal inclusion probabilities of all variables based on post burnin samples.
`median.model`	The median probability model based on post burnin samples.
`beta.mean`	The Monte Carlo estimate of posterior mean of β (the $p + 1$ vector c(intercept, regression coefficients)) based on post burnin samples.
`wmip`	The p vector of weighted marginal inclusion probabilities of all variables based on post burnin samples.
`wam`	The weighted average model based on post burnin samples.
`beta.wam`	The model probability weighted estimate of posterior mean of β (the $p+1$ vector c(intercept, regression coefficients)) based on post burnin samples.
`log.post`	The n.iter vector of log of the unnormalized marginal posterior pmf $P(\gamma\|y)$ evaluated at the samples.

Author(s)

Vivekananda Roy

References

Roy, V.(2024) A geometric approach to informative MCMC sampling https://arxiv.org/abs/2406.09010

Li, D., Dutta, S., Roy, V.(2023) Model Based Screening Embedded Bayesian Variable Selection for Ultra-high Dimensional Settings, Journal of Computational and Graphical Statistics, 32, 61-73

Examples

```
n=50; p=100; nonzero = 3
trueidx <- 1:3
nonzero.value <- 4
TrueBeta <- numeric(p)
TrueBeta[trueidx] <- nonzero.value
rho <- 0.5
xone <- matrix(rnorm(n*p), n, p)
X <- sqrt(1-rho)*xone + sqrt(rho)*rnorm(n)
y <- 0.5 + X %*% TrueBeta + rnorm(n)
result <- geomc.vs(X=X, y=y,model.summary = TRUE)
result$samples # the MCMC samples
result$acceptance.rate #the acceptance.rate
result$mip #marginal inclusion probabilities
result$wmip #weighted marginal inclusion probabilities
result$median.model #the median.model
result$wam #the weighted average model
result$beta.mean #the posterior mean of regression coefficients
result$beta.wam #another estimate of the posterior mean of regression coefficients
result$log.post #the log (unnormalized) posterior probabilities of the MCMC samples.
```

`logp.vs` *The log-unnormalized posterior probability of a model for Bayesian variable selection.*

Description

Calculates the log-unnormalized posterior probability of a model.

Usage

```
logp.vs(model, X, y, lam0 = 0, a0 = 0, b0 = 0, lam, w)
```

Arguments

`model`	The indices of active variables.
`X`	The $n \times p$ covariate matrix without intercept.
`y`	The response vector of length n.
`lam0`	The precision parameter for β_0. Default: 0 (corresponding to improper uniform prior).
`a0`	The shape parameter for prior on σ^2. Default: 0.

`b0`	The scale parameter for prior on σ^2. Default: 0.
`lam`	The slab precision parameter.
`w`	The prior inclusion probability of each variable.

Value

The log-unnormalized posterior probability of the model.

Author(s)

Vivekananda Roy

References

Roy, V.(2024) A geometric approach to informative MCMC sampling https://arxiv.org/abs/2406.09010

Examples

```
n=50; p=100; nonzero = 3
trueidx <- 1:3
nonzero.value <- 4
TrueBeta <- numeric(p)
TrueBeta[trueidx] <- nonzero.value
rho <- 0.5
xone <- matrix(rnorm(n*p), n, p)
X <- sqrt(1-rho)*xone + sqrt(rho)*rnorm(n)
y <- 0.5 + X %*% TrueBeta + rnorm(n)
result <- geomc.vs(X=X, y=y)
logp.vs(result$median.model,X,y,lam = nrow(X)/ncol(X)^2,w = sqrt(nrow(X))/ncol(X))
```

Index