Package 'BayesMultMeta'

July 21, 2025

Type Package

Title Bayesian Multivariate Meta-Analysis

Version 0.1.1

Author Olha Bodnar [aut] (ORCID: <https://orcid.org/0000-0003-1359-3311>),
Taras Bodnar [aut] (ORCID: <https://orcid.org/0000-0001-7855-8221>),
Erik Thorsén [aut, cre] (ORCID:
<https://orcid.org/0000-0001-5992-1216>)

Maintainer Erik Thorsén <erik.thorsen@math.su.se>

Description Objective Bayesian inference procedures for the parameters of the
multivariate random effects model with application to multivariate
meta-analysis. The posterior for the model parameters, namely the overall
mean vector and the between-study covariance matrix, are assessed by
constructing Markov chains based on the Metropolis-Hastings algorithms as
developed in Bodnar and Bodnar (2021) (<doi:10.48550/arXiv.2104.02105>). The
Metropolis-Hastings algorithm is designed under the assumption of the
normal distribution and the t-distribution when the Berger and Bernardo
reference prior and the Jeffreys prior are assigned to the model parameters.
Convergence properties of the generated Markov chains are investigated by
the rank plots and the split hat-R estimate based on the rank normalization,
which are proposed in Vehtari et al. (2021) (<DOI:10.1214/20-BA1221>).

License MIT + file LICENSE

Encoding UTF-8

LazyData false

Imports assertthat, Rdpack

Suggests mvmeta, gplots, testthat

RoxygenNote 7.1.2

RdMacros Rdpack

NeedsCompilation no

Repository CRAN

Date/Publication 2022-06-09 08:10:27 UTC

Contents

BayesMultMeta   *Interface for the BayesMultMeta class*

Description

The BayesMultMeta package implements two methods of constructing Markov chains to assess the posterior distribution of the model parameters, namely the overall mean vector μ and the between-study covariance matrix Ψ, of the generalized marginal multivariate random effects models. The Bayesian inference procedures are performed when the model parameters are endowed with the Berger and Bernardo reference prior (Berger and Bernardo 1992) and the Jeffreys prior (Jeffreys 1946). This is achieved by constructing Markov chains using the Metropolis-Hastings algorithms developed in (Bodnar and Bodnar 2021). The convergence properties of the generated Markov chains are investigated by the rank plots and the split-$\hat{R}$ estimate based on the rank normalization, which are proposed in (Vehtari et al. 2021).

Usage

```
BayesMultMeta(X, U, N, burn_in, likelihood, prior, algorithm_version, d = NULL)
```

Arguments

X	A $p \times n$ matrix which contains n observation vectors of dimension p
U	A $pn \times pn$ block-diagonal matrix which contains the covariance matrices of observation vectors.
N	Length of the generated Markov chain.
burn_in	Number of burn-in samples
likelihood	Likelihood to use. It currently supports "normal" and "t".

prior	Prior to use. It currently supports "reference" and "jeffrey".
algorithm_version	
	One of "mu" or "Psi". Both algorithms samples the same quantities.
d	Degrees of freedom for the t-distribution when the "t" option is used for the likelihood.

Value

a BayesMultMeta class which contains simulations from the MCMC inference procedure as well as many of the input parameters. The elements 'psi' and 'mu' in the list contains simulations from the posterior distribution. All other elements are input parameters to the class.

References

Berger JO, Bernardo JM (1992). "On the development of the reference prior method." *Bayesian statistics*, **4**(4), 35–60.

Bodnar O, Bodnar T (2021). "Objective Bayesian meta-analysis based on generalized multivariate random effects model." 2104.02105.

Jeffreys H (1946). "An invariant form for the prior probability in estimation problems." *Proceedings of the Royal Society of London Series A*, **186**(1007), 453-461. doi: 10.1098/rspa.1946.0056.

Vehtari A, Gelman A, Simpson D, Carpenter B, BÃ¼rkner P (2021). "Rank-normalization, folding, and localization: An improved hatR for assessing convergence of MCMC (with Discussion)." *Bayesian analysis*, **16**(2), 667–718.

Examples

```
dataREM<-mvmeta::hyp
# Observation matrix X
X<-t(cbind(dataREM$sbp,dataREM$dbp))
p<-nrow(X)  # model dimension
n<-ncol(X)  # sample size
# Matrix U
U<-matrix(0,n*p,n*p)
for (i_n in 1:n) {
  Use<-diag(c(dataREM$sbp_se[i_n],dataREM$dbp_se[i_n]))
  Corr_mat<-matrix(c(1,dataREM$rho[i_n],dataREM$rho[i_n],1),p,p)
  U[(p*(i_n-1)+1):(p*i_n),(p*(i_n-1)+1):(p*i_n)]<- Use%*%Corr_mat%*%Use
}

bmgmr_run <- BayesMultMeta(X, U, 1e2, burn_in = 100,
                  likelihood = "normal", prior="jeffrey",
                  algorithm_version = "mu")
summary(bmgmr_run)
plot(bmgmr_run)
```

---

bayes_inference   *Summary statistics from a posterior distribution*

Description

Given a univariate sample drawn from the posterior distribution, this function computes the posterior mean, the posterior median, the posterior standard deviation, and the limits of the $(1 - \alpha)$ probability-symmetric credible interval.

Usage

```
bayes_inference(x, alp)
```

Arguments

x   Univariate sample from the posterior distribution of a parameter.

alp   Significance level used in the computation of the credible interval

Value

a matrix with summary statistics

---

duplication_matrix   *Duplication matrix*

Description

This function creates the duplication matrix of size $p^2 \times p(p + 1)/2$

Usage

```
duplication_matrix(p)
```

Arguments

p   Integer which specifies the dimension of the duplication matrix.

Value

a matrix of size $p^2 \times p(p + 1)/2$

MC_ranks   *Computes the ranks within the pooled draws of Markov chains*

Description

The function computes the ranks within the pooled draws of Markov chains. Average ranks are used for ties.

Usage

```
MC_ranks(MC)
```

Arguments

MC   An $N \times M$ matrix with N draws in each of M constructed Markov chains.

Value

a matrix with the ranks from the MCMC procedure

Examples

```
dataREM<-mvmeta::hyp
# Observation matrix X
X<-t(cbind(dataREM$sbp,dataREM$dbp))
p<-nrow(X) # model dimension
n<-ncol(X) # sample size
# Matrix U
U<-matrix(0,n*p,n*p)
for (i_n in 1:n) {
  Use<-diag(c(dataREM$sbp_se[i_n],dataREM$dbp_se[i_n]))
  Corr_mat<-matrix(c(1,dataREM$rho[i_n],dataREM$rho[i_n],1),p,p)
  U[(p*(i_n-1)+1):(p*i_n),(p*(i_n-1)+1):(p*i_n)]<- Use%*%Corr_mat%*%Use
}
# Generating M Markov chains for mu_1
M<-4 # number of chains
MC <-NULL
for (i in 1:M) {
chain <-  BayesMultMeta(X, U, 1e2, burn_in = 1e2,
                        likelihood = "t", prior="jeffrey",
                        algorithm_version = "mu",d=3)
  MC<- cbind(MC,chain$mu[1,])
}
ranks<-MC_ranks(MC)
id_chain <- 1
hist(ranks[,id_chain],breaks=25,prob=TRUE, labels = FALSE, border = "dark blue",
  col = "light blue", main = expression("Chain 1,"~mu[1]), xlab = expression(),
  ylab = expression(),cex.axis=1.2,cex.main=1.7,font=2)
```

---

plot.BayesMultMeta   *Plot a BayesMultMeta object*

Description

This function produces the trace plots of the constructed Markov chains.

Usage

```
## S3 method for class 'BayesMultMeta'
plot(x, ...)
```

Arguments

x	a BayesMultMeta object
...	additional arguments

Value

No return value, produces trace plots

---

sample_post_nor_jef_marg_mu

Metropolis-Hastings algorithm for the normal distribution and the Jeffreys prior, where μ is generated from the marginal posterior.

Description

This function implements Metropolis-Hastings algorithm for drawing samples from the posterior distribution of μ and $\mathbf{\Psi}$ under the assumption of the normal distribution when the Jeffreys prior is employed. At each step, the algorithm starts with generating a draw from the marginal distribution of μ.

Usage

```
sample_post_nor_jef_marg_mu(X, U, Np)
```

Arguments

X	A $p \times n$ matrix which contains n observation vectors of dimension p.
U	A $pn \times pn$ block-diagonal matrix which contains the covariance matrices of observation vectors.
Np	Length of the generated Markov chain.

Value

List with the generated samples from the joint posterior distribution of μ and $\boldsymbol{\Psi}$, where the values of $\boldsymbol{\Psi}$ are presented by using the vec operator.

---

sample_post_nor_jef_marg_Psi

> *Metropolis-Hastings algorithm for the normal distribution and the Jeffreys prior, where $\boldsymbol{\Psi}$ is generated from the marginal posterior.*

Description

This function implements Metropolis-Hastings algorithm for drawing samples from the posterior distribution of μ and $\boldsymbol{\Psi}$ under the assumption of the normal distribution when the Jeffreys prior is employed. At each step, the algorithm starts with generating a draw from the marginal distribution of $\boldsymbol{\Psi}$.

Usage

```
sample_post_nor_jef_marg_Psi(X, U, Np)
```

Arguments

X	A $p \times n$ matrix which contains n observation vectors of dimension p.
U	A $pn \times pn$ block-diagonal matrix which contains the covariance matrices of observation vectors.
Np	Length of the generated Markov chain.

Value

List with the generated samples from the joint posterior distribution of μ and $\boldsymbol{\Psi}$, where the values of $\boldsymbol{\Psi}$ are presented by using the vec operator.

---

sample_post_nor_ref_marg_mu

> *Metropolis-Hastings algorithm for the normal distribution and the Berger and Bernardo reference prior, where μ is generated from the marginal posterior.*

Description

This function implements Metropolis-Hastings algorithm for drawing samples from the posterior distribution of μ and $\boldsymbol{\Psi}$ under the assumption of the normal distribution when the Berger and Bernardo reference prior is employed. At each step, the algorithm starts with generating a draw from the marginal distribution of μ.

Usage

```
sample_post_nor_ref_marg_mu(X, U, Np)
```

Arguments

X	A $p \times n$ matrix which contains n observation vectors of dimension p.
U	A $pn \times pn$ block-diagonal matrix which contains the covariance matrices of observation vectors.
Np	Length of the generated Markov chain.

Value

List with the generated samples from the joint posterior distribution of μ and $\mathbf{\Psi}$, where the values of $\mathbf{\Psi}$ are presented by using the vec operator.

---

sample_post_nor_ref_marg_Psi

Metropolis-Hastings algorithm for the normal distribution and the Berger and Bernardo reference prior, where $\mathbf{\Psi}$ is generated from the marginal posterior.

Description

This function implements Metropolis-Hastings algorithm for drawing samples from the posterior distribution of μ and $\mathbf{\Psi}$ under the assumption of the normal distribution when the Berger and Bernardo reference prior is employed. At each step, the algorithm starts with generating a draw from the marginal distribution of $\mathbf{\Psi}$.

Usage

```
sample_post_nor_ref_marg_Psi(X, U, Np)
```

Arguments

X	A $p \times n$ matrix which contains n observation vectors of dimension p.
U	A $pn \times pn$ block-diagonal matrix which contains the covariance matrices of observation vectors.
Np	Length of the generated Markov chain.

Value

List with the generated samples from the joint posterior distribution of μ and $\mathbf{\Psi}$, where the values of $\mathbf{\Psi}$ are presented by using the vec operator.

---

sample_post_t_jef_marg_mu

Metropolis-Hastings algorithm for the t-distribution and the Jeffreys prior, where μ is generated from the marginal posterior.

Description

This function implements Metropolis-Hastings algorithm for drawing samples from the posterior distribution of μ and Ψ under the assumption of the t-distribution when the Jeffreys prior is employed. At each step, the algorithm starts with generating a draw from the marginal distribution of μ.

Usage

```
sample_post_t_jef_marg_mu(X, U, d, Np)
```

Arguments

X	A $p \times n$ matrix which contains n observation vectors of dimension p.
U	A $pn \times pn$ block-diagonal matrix which contains the covariance matrices of observation vectors.
d	Degrees of freedom for the t-distribution
Np	Length of the generated Markov chain.

Value

List with the generated samples from the joint posterior distribution of μ and Ψ, where the values of Ψ are presented by using the vec operator.

---

sample_post_t_jef_marg_Psi

Metropolis-Hastings algorithm for the t-distribution and the Jeffreys prior, where Ψ is generated from the marginal posterior.

Description

This function implements Metropolis-Hastings algorithm for drawing samples from the posterior distribution of μ and Ψ under the assumption of the t-distribution when the Jeffreys prior is employed. At each step, the algorithm starts with generating a draw from the marginal distribution of Ψ.

Usage

```
sample_post_t_jef_marg_Psi(X, U, d, Np)
```

Arguments

X	A $p \times n$ matrix which contains n observation vectors of dimension p.
U	A $pn \times pn$ block-diagonal matrix which contains the covariance matrices of observation vectors.
d	Degrees of freedom for the t-distribution
Np	Length of the generated Markov chain.

Value

List with the generated samples from the joint posterior distribution of μ and Ψ, where the values of Ψ are presented by using the vec operator.

---

sample_post_t_ref_marg_mu

> *Metropolis-Hastings algorithm for the t-distribution and Berger and Bernardo reference prior, where μ is generated from the marginal posterior.*

Description

This function implements Metropolis-Hastings algorithm for drawing samples from the posterior distribution of μ and Ψ under the assumption of the t-distribution when the Berger and Bernardo prior is employed. At each step, the algorithm starts with generating a draw from the marginal distribution of μ.

Usage

```
sample_post_t_ref_marg_mu(X, U, d, Np)
```

Arguments

X	A $p \times n$ matrix which contains n observation vectors of dimension p.
U	A $pn \times pn$ block-diagonal matrix which contains the covariance matrices of observation vectors.
d	Degrees of freedom for the t-distribution
Np	Length of the generated Markov chain.

Value

List with the generated samples from the joint posterior distribution of μ and Ψ, where the values of Ψ are presented by using the vec operator.

---

sample_post_t_ref_marg_Psi

> *Metropolis-Hastings algorithm for the t-distribution and Berger and Bernardo reference prior, where $\mathbf{\Psi}$ is generated from the marginal posterior.*

Description

This function implements Metropolis-Hastings algorithm for drawing samples from the posterior distribution of μ and $\mathbf{\Psi}$ under the assumption of the t-distribution when the Berger and Bernardo prior is employed. At each step, the algorithm starts with generating a draw from the marginal distribution of $\mathbf{\Psi}$.

Usage

```
sample_post_t_ref_marg_Psi(X, U, d, Np)
```

Arguments

X	A $p \times n$ matrix which contains n observation vectors of dimension p.
U	A $pn \times pn$ block-diagonal matrix which contains the covariance matrices of observation vectors.
d	Degrees of freedom for the t-distribution
Np	Length of the generated Markov chain.

Value

List with the generated samples from the joint posterior distribution of μ and $\mathbf{\Psi}$, where the values of $\mathbf{\Psi}$ are presented by using the vec operator.

---

split_rank_hatR *Computes the split-$\hat{R}$ estimate based on the rank normalization*

Description

The function computes the split-$\hat{R}$ estimate based on the rank normalization.

Usage

```
split_rank_hatR(MC)
```

Arguments

MC	An $N \times M$ matrix with N draws in each of M constructed Markov chains.

Value

a value with the the split-$\hat{R}$ estimate based on the rank normalization

Examples

```
dataREM<-mvmeta::hyp
# Observation matrix X
X<-t(cbind(dataREM$sbp,dataREM$dbp))
p<-nrow(X) # model dimension
n<-ncol(X) # sample size
# Matrix U
U<-matrix(0,n*p,n*p)
for (i_n in 1:n) {
  Use<-diag(c(dataREM$sbp_se[i_n],dataREM$dbp_se[i_n]))
  Corr_mat<-matrix(c(1,dataREM$rho[i_n],dataREM$rho[i_n],1),p,p)
  U[(p*(i_n-1)+1):(p*i_n),(p*(i_n-1)+1):(p*i_n)]<- Use%*%Corr_mat%*%Use
}
# Generating M Markov chains for mu_1
M<-4 # number of chains
MC <-NULL
for (i in 1:M) {
  chain <-  BayesMultMeta(X, U, 1e2, burn_in = 1e2,
                          likelihood = "t", prior="jeffrey",
                          algorithm_version = "mu",d=3)
  MC<- cbind(MC,chain$mu[1,])
}
split_rank_hatR(MC)
```

---

summary.BayesMultMeta   *Summary statistics from the posterior of a BayesMultMeta class*

Description

Summary statistics from the posterior of a BayesMultMeta class

Usage

```
## S3 method for class 'BayesMultMeta'
summary(object, alpha = 0.95, ...)
```

Arguments

object	BayesMultMeta class
alpha	Significance level used in the computation of the credible interval.
...	not used

Value

a list with summary statistics

Index