Package: bayesMRM (via r-universe)

October 27, 2026

Type Package

Title Bayesian Multivariate Receptor Modeling

Version 2.4.0

Maintainer Man-Suk Oh <msoh@ewha.ac.kr>

Description Bayesian analysis of multivariate receptor modeling. The package consists of implementations of the methods of Park and Oh (2015) <doi:10.1016/j.chemolab.2015.08.021>.The package uses 'JAGS'(Just Another Gibbs Sampler) to generate Markov chain Monte Carlo samples of parameters.

Date 2022-12-14

Depends R (>= 4.0.0)

License GPL (>= 2)

Encoding UTF-8

Imports rjags,coda,ggplot2,grDevices,graphics,gridExtra,rgl,shiny,shinythemes

Suggests stats

RoxygenNote 7.2.1

NeedsCompilation no

Author Man-Suk Oh [aut, cre], Eun-Kyung Lee [aut], Eun Sug Park [aut]

Repository https://cran.r-universe.dev

Date/Publication 2022-12-14 06:30:05 UTC

RemoteUrl https://github.com/cran/bayesMRM

RemoteRef HEAD

RemoteSha 1282e61a9201cd97e36fbd1151bcfc671f5ad636

Contents

bayesMRMApp 2
bmrm 3
convdiag_bmrm 6
Elpaso 8

idcond_check 9
pcplot 9
plot.bmrm 10
summary.bmrm 11
trace_ACF_plot 11

Index **13**

`bayesMRMApp` *Shiny App for exploring the results of Bayesian multivariate receptor modeling*

Description

Call Shiny to show the results of Bayesian analysis of multivariate receptor modeling in a web-based application. This object contains

- plots of the posterior means and 95% posterior intervals of parameters in an object of class `bmrm`.
- tables of the posterior means of parameters in an object of class `bmrm`.
- tables of the posterior quantiles of parameters in an object of class `bmrm`, for prob=(0.025, 0.05, 0.25, 0.5, 0.75, 0.95, 0.975).
- tables of convergence diagnostics of parameters in an object of class `bmrm`.
- 3-dimensional dynamic principal component plots of data (Y) and source profiles (rows of the estimated source composition matrix P) in an object of class `bmrm`. The plot can be rotated by moving the cursor.
- trace plots and ACF plots of the first 6 elements of a parameter in an object of class `bmrm`.

Usage

```
bayesMRMApp(x)
```

Arguments

`x`	an object of class `bmrm`, the output of the `bmrm` function

Value

shiny App

bmrm *Bayesian Analysis of Multivariate Receptor Modeling*

Description

Generate posterior samples of the source composition matrix P, the source contribution matrix A, and the error variance Σ using 'JAGS', and computes estimates of A,P,Σ.

Usage

```
bmrm(Y, q, muP,errdist="norm", df=4,
            varP.free=100, xi=NULL, Omega=NULL,
              a0=0.01, b0=0.01,
             nAdapt=1000, nBurnIn=5000, nIter=5000, nThin=1,
             P.init=NULL, A.init=NULL, Sigma.init=NULL,...)
```

Arguments

Y	data matrix
q	number of sources. It must be a positive integer.
muP	(q,ncol(Y))-dimensional prior mean matrix for the source composition matrix P, where q is the number of sources. Zeros need to be assigned to prespecified elements of muP to satisfy the identifiability condition C1. For the remaining free elements, any nonnegative numbers (between 0 and 1 preferably) can be assigned. If no or an insufficient number of zeros are preassigned in muP, estimation can still be performed but the resulting estimates may be subject to rotational ambiguities. (default=0.5 for nonzero elements).
errdist	error distribution: either "norm" for normal distribution or "t" for t distribution (default="norm")
df	degrees of freedom of a t-distribution when errdist="t" (default=4)
varP.free	scalar value of the prior variance of the free (nonzero) elements of the source composition matrix P (default=100)
xi	prior mean vector of the q-dimensional source contribution vector at time t (default=vector of 1's)
Omega	diagonal matrix of the prior variance of the q-dimensional source contribution vector at time t (default=identity matrix)
a0	shape parameter of the Inverse Gamma prior of the error variance (default=0.01)
b0	scale parameter of the Inverse Gamma prior of the error variance (default=0.01)
nAdapt	number of iterations for adaptation in 'JAGS' (default=1000)
nBurnIn	number of iterations for the burn-in period in MCMC (default=5000)
nIter	number of iterations for monitoring samples from MCMC (default=5000). nIter samples are saved in each chain of MCMC.
nThin	thinning interval for monitoring samples from MCMC (default=1)

`P.init`	initial value of the source composition matrix P. If omitted, zeros are assigned to the elements corresponding to zero elements in muP and the nonzero elements of P.init will be randomly generated from a uniform distrbution.
`A.init`	initial value of the source contribution matrix A. If omitted, it will be calculated from Y and P.init.
`Sigma.init`	initial value of the error variance. If omitted, it will be calculated from Y, A.init and P.init.
`...`	arguments to be passed to methods

Details

Model

The basic model for Bayesian multivariate receptor model is as follows:

$Y_t = A_t P + E_t, t = 1, \cdots, T,$

where

- Y_t is a vector of observations of J variables at time t, $t = 1, \cdots, T$.
- P is a $q \times J$ source composition matrix in which the k-th row represents the k-th source composition profiles, $k = 1, \cdots, q$, q is the number of sources.
- A_t is a q dimensional source contribution vector at time t, $t = 1, \cdots, T$.
- $E_t = (E_{t1}, \cdots, E_{tJ})$ is an error term for the t-th observations, following $E_t \sim N(0, \Sigma)$ or $E_t \sim t_{df}(0, \Sigma)$, independently for $j = 1, \cdots, J$, where $\Sigma = diag(\sigma_1^2, ..., \sigma_J^2)$.

Priors

- Prior distribution of A_t is given as a truncated multivariate normal distribution,
  - $A_t \sim N(\xi, \Omega) I(A_t \geq 0)$, independently for $t = 1, \cdots, T$.
- Prior distribution of P_{kj} (the (k, j)-th element of the source composition matrix P) is given as
  - $P_{kj} \sim N(\texttt{muP}_{kj}, \texttt{varP.free}) I(P_{kj} \geq 0)$, for free (nonzero) P_{kj},
  - $P_{kj} \sim N(0, 1e-10) I(P_{kj} \geq 0)$, for zero P_{kj},
    independently for $k = 1, \cdots, q; j = 1, \cdots, J$.
- Prior distribution of σ_j^2 is $IG(a0, b0)$, i.e.,
  - $1/\sigma_j^2 \sim Gamma(a0, b0)$, having mean $a0/b0$, independently for $j = 1, ..., J$.

Notes

- We use the prior $P_{kj} \sim N(0, 1e-10) I(P_{kj} \geq 0)$ that is practically equal to the point mass at 0 to simplify the model building in 'JAGS'.
- The MCMC samples of A and P are post-processed (rescaled) before saving so that $\sum_{j=1}^{J} P_{kj} = 1$ for each $k = 1, ..., q$ (the identifiablity condition C3 of Park and Oh (2015).

Value

in bmrm object

nsource number of sources

nobs number of observations in data Y

nvar number of variables in data Y

Y observed data matrix

muP prior mean of the source composition matrix P

errdist error distribution

df degrees of freedom when errdist="t"

A.hat posterior mean of the source contribution matrix A

P.hat posterior mean of the source composition matrix P

Sigma.hat posterior mean of the error variance Sigma

A.sd posterior standard deviation of the source contribution matrix A

P.sd posterior standard deviation of the source composition matrix P

Sigma.sd posterior standard deviation of the error variance Sigma

A.quantiles posterior quantlies of A for prob=(0.025, 0.05, 0.25, 0.5, 0.75, 0.95, 0.975)

P.quantiles posterior quantiles of P for prob=(0.025, 0.05, 0.25, 0.5, 0.75, 0.95, 0.975)

Sigma.quantiles posterior quantiles of Sigma for prob=(0.025, 0.05, 0.25, 0.5, 0.75, 0.95, 0.975)

Y.hat predicted value of Y computed from A.hat*P.hat

residual Y-Y.hat

codaSamples MCMC posterior samples of A, P, and Σ in class "mcmc.list"

nIter number of MCMC iterations per chain for monitoring samples from MCMC

nBurnIn number of iterations for the burn-in period in MCMC

nThin thinning interval for monitoring samples from MCMC

References

Park, E.S. and Oh, M-S. (2015), Robust Bayesian Multivariate Receptor Modeling, Chemometrics and intelligent laboratory systems, 149, 215-226.

Plummer, M. 2003. JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling. Proceedings of the 3rd international workshop on distributed statistical computing, pp. 125. Technische Universit at Wien, Wien, Austria.

Plummer, M. 2015. 'JAGS' Version 4.0.0 user manual.

Examples

```
data(Elpaso); Y=Elpaso$Y ; muP=Elpaso$muP ; q=nrow(muP)
out.Elpaso <- bmrm(Y,q,muP)
summary(out.Elpaso)
plot(out.Elpaso)
```

convdiag_bmrm *Convergence Diagnostics on MCMC samples in* bmrm

Description

Compute convergence diagnostics of Geweke (1992), Heidelberger and Welch (1983), Raftery and Lewis(1992).

Usage

```
convdiag_bmrm(x , var="P", convdiag="geweke",print=TRUE,...)
```

Arguments

x	an object of class bmrm, the output of the bmrm function
var	name of a variable to which convergence disagnostics apply. It should be one of "A" (source contribution matrix), "P" (source composition or profile matrix), "Sigma" (error variance).
convdiag	vector of convergence diagnostic methods. It should be any subvector of ("geweke", "heidel","raftery") (default="geweke").
print	TRUE/FALSE, print convergence diagnostics results (default=TRUE)
...	arguments to be passed to methods

Details

Geweke's convergence diagnostic for Markov chains is based on a test for equality of the means of the first and last part of a Markov chain (by default the first 10% and the last 50%). If the samples are drawn from the stationary distribution of the chain, the two means should be equal and Geweke's statistic has an asymptotically standard normal distribution. We use the function `geweke.diag` in **coda** package (with default option) which provides the test statistics (standard Z-scores) and the upper bound of and p-values.

Heidelberger and Welch's convergence diagnostic tests the null hypothesis that the sampled values come from a stationary distribution. The test is successively applied, firstly to the whole chain, then after discarding the first 10%, 20%, ... of the chain until either the null hypothesis is accepted, or 50% of the chain has been discarded. We use the function `heidel.diag` (with default option) which provides the staionary test results and p-values.

Raftery and Lewis's diagnostic estimates the minimum number of iterations, burn-in, thinning interval for zero autocorrelation, satisfying specified conditions regarding quantile q of parameters of interest. The conditions are specified by a posterior quantile q of parameters, an acceptable tolerance (accuracy) r for q, a probability s of being within the interval $q-r, q+r$. We use the function `raftery.diag` (with default option).

Value

A list of convergence diagnostics results

convdiag selected convergence diagnostic methods

geweke Geweke's z-scores and p-values if `convdiag` includes "geweke", NULL if `convdiag` does not include "geweke"

heidel Heidelberger and Welch's stationary test results and p-values if `convdiag` includes "heidel"; NULL if `convdiag` does not include "heidel"

raftery Raftery and Lewis's estimates of burn-in, minimum number of iterations, and thinning if `convdiag` includes "raftery"; NULL if `convdiag` does not include "raftery"

References

Geweke, J.(1992) Evaluating the accuracy of sampling-based approaches to calculating posterior moments. In Bayesian Statistics 4 (ed JM Bernado, JO Berger, AP Dawid and AFM Smith). Clarendon Press.

Heidelberger P, and Welch PD. (1981) A spectral method for confidence interval generation and run length control in simulations. Comm. ACM. 24, 233-245.

Heidelberger P. and Welch PD.(1983) Simulation run length control in the presence of an initial transient. Opns Res., 31, 1109-44,Oxford, UK.

Plummer, M., Best, N., Cowles, K. and Vines K. (2006) CODA: Convergence Diagnosis and Output Analysis for MCMC, R News, Vol 6, pp. 7-11.

Raftery, A.E. and Lewis, S.M. (1992). One long run with diagnostics: Implementation strategies for Markov chain Monte Carlo. Statistical Science, 7, 493-497.

Raftery, A.E. and Lewis, S.M. (1995). The number of iterations, convergence diagnostics and generic Metropolis algorithms. In Practical Markov Chain Monte Carlo (W.R. Gilks, D.J. Spiegelhalter and S. Richardson, eds.). London, U.K.: Chapman and Hall.

Examples

```
data(Elpaso)
Y=Elpaso$Y ; muP=Elpaso$muP
q=nrow(muP)
out.Elpaso <- bmrm(Y,q,muP, nAdapt=1000,nBurnIn=5000,nIter=5000,nThin=1)
conv1<-convdiag_bmrm(out.Elpaso,var="P",convdiag="raftery" )
conv2<-convdiag_bmrm(out.Elpaso,var="A", convdiag="geweke")
conv3<-convdiag_bmrm(out.Elpaso,var="Sigma", convdiag=c("geweke","heidel"))
conv4<-convdiag_bmrm(out.Elpaso,var="Sigma", convdiag=c("geweke","heidel", "raftery"))
```

`Elpaso` *PM2.5 speciation data from El Paso, Texas, USA.*

Description

The data frame has the following components:

- *Y* 224 by 15 matrix of 224 observations on 15 PM2.5 species. PM 2.5 was measured every three days during the time period of 1/2/2006 ~ 4/7/2009 from the Chamizal station in the city of El Paso, USA. Out of the 58 original PM 2.5 species, 15 species were selected. After removing any observations with missig values, the final data consists of 224 complete observatins on the following 15 PM2.5 species:

  Al Aluminum
  Ca Calcium
  Cl2 Chlorine
  EC EC CSN
  Fe Iron
  K_p Potassium ion
  Mg Magnanese
  NV_NO3 Non-volatile nitrate
  NH4_p Ammonium ion
  Na Sodium
  OC OC CSN unadjusted
  SO4 Sulfate
  Si Silicon
  Ti Titanium
  Zn Zinc

- *muP* 4 by 15 matrix of the prior mean of the source composition matrix P for data. Zero values are assigned for some elements of muP to satisfy the identifiability conditions C1-C2 in Park and Oh (2015). The remaining nonzero elements of muP have value 0.5. Note that the number of sources (the number of rows in muP) is presumed to be 4 here.

References

Park, E.S. and Oh, M-S. (2016), Bayesian Quantile Multivariate Receptor Modeling, Chemometrics and intelligent laboratory systems, 159, 174-180.

Examples

```
data(Elpaso)
Y=Elpaso$Y
muP=Elpaso$muP
```

idcond_check *Check the identifiability conditions*

Description

Check the identifiability conditions C1-C2 of Park and Oh (2015).

Usage

```
idCond_check(P)
```

Arguments

P	source composition matrix in multivariate receptor model

Value

idCond TRUE if all the conditions are satisfied, FALSE otherwise

pcplot *Principal component plot*

Description

Draw principal component plots of data (Y) and source profiles (rows) of the estimated source composition matrix P.hat (and P0 if there is another source composition matrix P0 to compare, e.g., P0 could be the true P in simulation or P0 could be another estimate of P)

Usage

```
pcplot(x, P0, G3D=FALSE,...)
```

Arguments

x	an object of class bmrm, the output from the function bmrm
P0	estimated value of P (in simulation it can be the true value of P)
G3D	TRUE/FALSE, dynamic 3D plot (default=FALSE)
...	arguments to be passed to methods

Value

plot

Examples

```
data(Elpaso)
Y=Elpaso$Y ; muP=Elpaso$muP
q=nrow(muP)
out.Elpaso <- bmrm(Y,q,muP, nAdapt=1000,nBurnIn=5000,nIter=5000,nThin=1)
pcplot(out.Elpaso)
pcplot(out.Elpaso,G3D=TRUE)
```

`plot.bmrm` *Produce plots of the parameter estimates*

Description

Produce plots of the estimated posterior mean and 95% posterior intervals of A,P, Sigma based on the MCMC samples in `bmrm`.

Usage

```
## S3 method for class 'bmrm'
plot(x, type = "both", ...)
```

Arguments

`x`	an object of class `bmrm`, the output of the function `bmrm`
`type`	name of a variable (default="P"). It should be one of "P"(source composition or profile matrix P), "A"(source contribution matrix A), "both" (both P and A), "Sigma" (error variance).
`...`	arguments to be passed to methods

Details

The following types of plots are drawn depending on the selected parameters:

- P: bar plots of the posterior means with 95% posterior intervals of elements for each row of P
- A: time series plots of posterior means with 95% posterior intervals elements for each column of A
- Sigma: posterior means with error bars for 95% posterior intervals of elements of Sigma

Value

plot

summary.bmrm *Summarize the output of the* bmrm *function*

Description

An S3 method that summarizes the output of the bmrm function in an object of class bmrm. This object contains the posterior mean, the posterior standard deviation, and (0.025, 0.05,0.25, 0.5, 0.75,0.95, 0.975) posterior quantiles of A, P, Σ. It also contains other relevant information about the MCMC procedure such as the burn-in iterations, the number of MCMC chains, etc.

Usage

```
## S3 method for class 'bmrm'
summary(object, digits = 3, ...)
```

Arguments

object	an object of class bmrm, the output of the bmrm function
digits	integer indicating the humber of signifiant digits
...	arguments to be passed to methods

Value

summary of bmrm class (output of bmrm function)

trace_ACF_plot *Trace and/or ACF plots of elements of a variable in* bmrm *object*

Description

Produce trace and Auto-Correlation Function plots (along with Effective sample size) of MCMC samples of elements of A, nonzero elements of P, elements of Sigma.

Usage

```
trace_ACF_plot(x,var="P", ACF=FALSE, nplot=0,irow=1, icol=1, saveFile=FALSE,...)
```

Arguments

x	an object of class bmrm, the output of the bmrm function
var	name of a variable to which the plots apply. It should be one of "A" (source contribution matrix), "P" (source composition matrix), "Sigma" (error variance).
ACF	TRUE/FALSE if TRUE ACF plots will be provided along with effective sample sizes(dafault: FALSE)

nplot	number of elements of 'var' for trace and/or ACF plots. If 'nplot' is smaller than the total number of elements of 'var' then plots of 'nplot' selected elements will be drawn. Otherwise, trace and/or ACF plots of all elements will be drawn. (default=0 implies that all elements will be selected if var="P" or "Sigma", and the first 12 elements will be selected if var="A")
irow	row index of A/P matrix or index of element of Sigma vector. Plots of 'nplot' elements starting from (irow, icol) element of A/P or elements starting from irow element of Sigma will be drawn (default=1).
icol	column number of A/P matrix. Plots of 'nplot' elements starting from (irow, icol) element of A/P will be drawn (default=1).
saveFile	TRUE/FALSE, save the plots in file *'var'-trace.pdf* (default=FALSE)
...	arguments to be passed to methods

Value

plot

Examples

```
data(Elpaso); Y=Elpaso$Y ; muP=Elpaso$muP ; q=nrow(muP)
out.Elpaso <- bmrm(Y,q,muP, nAdapt=1000,nBurnIn=5000,nIter=5000,nThin=1)
trace_ACF_plot(out.Elpaso,"Sigma", ACF=TRUE)
trace_ACF_plot(out.Elpaso,"P", ACF=TRUE)
trace_ACF_plot(out.Elpaso,"A",ACF=TRUE, nplot=12, irow=2, icol=3)
```

Index

∗ **datasets**
Elpaso, 8

bayesMRMApp, 2
bmrm, 3

convdiag_bmrm, 6

Elpaso, 8

idCond_check *(*idcond_check*)*, 9
idcond_check, 9

pcplot, 9
plot *(*plot.bmrm*)*, 10
plot.bmrm, 10

summary *(*summary.bmrm*)*, 11
summary.bmrm, 11

trace_ACF_plot, 11