

Package ‘GLMMarp’

October 18, 2009

Version 0.1-1

Date 2009-09-28

Title Generalized Linear Multilevel Model with AR(p) Errors Package

Author Xun Pang <xpang@wustl.edu>

Maintainer Xun Pang <xpang@wustl.edu>

Depends R (>= 2.8.0), coda (>= 0.11-3), MCMCpack, mvtnorm, lattice, msm, MASS, Matrix, accuracy, kinship, panel, bayesSurv

Description This package contains functions to estimate the GLMM-AR(p) model for analyzing discrete time-series cross-sectional data via Markov Chain Monte Carlo simulation. The simulation is done only with the R language. The model returns draws of the parameter posteriors selected by the user in a list format. Each parameter chain is returned as a matrix. The user is responsible to summarize the mcmc output by using the coda package. GLMMarp also contains several useful utility functions, including an independent function for computing the Bayes factor with GLMM-AR(p) output, a function to recover the random coefficients at the individual level, and a function to do prediction by using the posterior distributions. The package also contains a library of supporting functions for the MCMC simulation and Bayes factor estimation. In this version, no tools for visualization are provided. To use the functions in this package, the user needs to load the following two packages by him/herself: `{panel}` and `{bayesSurv}`, because the two packages have no namespace.

License GPL

SystemRequirements gcc (>= 4.0)

URL <http://xpang.wustl.edu/Computing.html>

Repository CRAN

Date/Publication 2009-10-18 09:17:21

R topics documented:

GLMMARp.Binary	2
Marg.Like.Binary	7
Prediction.Binary	9
random.time	11
random.unit	13
Reduced.Binary	14
StateAR2	16
StateAR3	17
StateFailure	17
Index	18

GLMMARp.Binary	<i>Markov Chain Monte Carlo for a Probit Multilevel Model with a pth-order Autoregressive Error Process</i>
----------------	---

Description

This function generates a sample from the posterior distribution of a probit multilevel model with a p th-order autoregressive error process for modeling binary Time-Series Cross-Sectional responses. The model allows observed and unobserved heterogeneities in both the time and spatial dimensions as well as group-level predictors to explain the group-level variations. Serial correlation is corrected with a p th-order autoregressive error process, and the lag order p should be chosen through model comparison with the Bayes factor. The lag order p must be equal or greater than 1, and the algorithm is based on Pang (2009). The user supplies data and priors, and takes the responsibility of specifying different groups of regressors in the structural form of the GLMM-AR(p) model (not the reduced form). The user can choose a subset of parameters to be monitored in the MCMC process. But for the Bayes factor be calculated, all parameters (including the augmented data and the posteriors of the auxiliary variable u) should be monitored. A sample of the posterior distribution is returned as a list containing the individual parameter chain. The output can be easily converted into mcmc output and then analyzed with functions provided in the `coda` package. The function does not provide convergence diagnostics, and it is the user's responsibility to check convergence by using the tools provided in the `coda` package.

Usage

```
GLMMARp.Binary(y, X1, Wi, St, Ai="NULL", Ft="NULL", Unit.Index, Time.Index,
  timeint.add=FALSE, unitint.add=FALSE, m=10000, burnin=5000,
  beta0,B0, D0, d0, E0, e0, nlag, init.phi, init.beta=0,
  init.b=0, init.c=0, pgm=TRUE, tracking=0,
  marginal.likelihood=FALSE, reduced.mcmc=1000, reduced.burnin=100,
  monitor=c("rho", "beta", "bi", "ct", "D", "E", "ystar", "u"),thin=1)
```

Arguments

<code>y</code>	A vector of the dichotomous responses, taking two unique values of 0 and 1.
<code>X1</code>	A matrix of covariates with fixed effects.
<code>Wi</code>	A matrix of covariates with subject-varying effects.
<code>St</code>	A matrix of covariates with time-varying effects.
<code>Ai</code>	A matrix of group-level predictors explaining the subject-varying effects. The number of rows of this matrix is equal to the length of <code>y</code> . If some or all predictors are time-varying, the function will automatically use the within-subject mean; for time-invariant predictors, they should be repeated over the same times of each subject. The default is "NULL", and there are no group-level covariates.
<code>Ft</code>	A matrix of group-level predictors explaining the time-varying effects. The number of rows of this matrix is equal to the length of <code>y</code> . If some or all predictors are subject-varying, the function will automatically use the within-time-period mean; for subject-invariant predictors, they should be repeated over the same number of subjects in the specific time period. The default is "NULL", and there are no group-level covariates.
<code>Unit.Index</code>	A vector of the subject index, i.e., the <code>i</code> 's. Note: the number of observations of each unit should be larger than the lag order, <code>nlag</code> . Those units which have fewer than or equal to <code>nlag</code> observations should be taken out of the sample in order to use the function.
<code>Time.Index</code>	A vector of the time index, i.e., the <code>t</code> 's. Note: no missing observations in the middle of the sample time periods of a unit are allowed. In other words, unbalanced data structures are allowed, but no broken data structure.
<code>timeint.add</code>	Should a time-specific intercept be added into the model? It takes two values: TRUE or FALSE with default as FALSE. Be sure that there is no time-specific intercept already in the reduced form of the model before adding an intercept here.
<code>unitint.add</code>	Should a subject-specific intercept be added into the model? It takes two values: TRUE or FALSE with default as FALSE. Be sure that there is no subject-specific intercept already in the reduced form of the model before adding an intercept here.
<code>m</code>	The number of iterations after burn-in and to be returned. If the chain is thinned, thinning does not affect the number of iterations to be returned.
<code>burnin</code>	The number of burn-in iterations for the sampler. If the chain is thinned, the number of burn-in iterations will be <code>burnin*thin</code> .
<code>thin</code>	The thinning interval used in the simulation. It does not affect the total number of iterations set by the argument " <code>m</code> ", but will affect the total number of iterations in the simulation.
<code>beta0</code>	The prior mean of β . This should be a vector with dimension equal to the number of fixed-effect parameters in the reduced form. Since the dimension is difficult for the user to compute when the model contains multiple random coefficients and multiple group-level predictors, the function will provide the correct dimension in the error message if the dimension of <code>beta0</code> is incorrectly specified, and the user can respecify <code>beta0</code> with this information and recall the function.

B0	The prior covariance matrix of β . This should be a positive definite matrix with dimension equal to the number of betas in the reduced form of GLMM-AR(p).
d0, D0	The degree of freedom and scale matrix of the Wishart prior on b_i which are the subject-level residuals. D0 should not be too defuse, otherwise it may takes a long time for the chain to converge. Recommended values is $a \cdot I$, where a is between 1.67 and 10. No default is provided.
e0, E0	The degree of freedom and scale matrix of the Wishart prior on c_t which are the time-level residuals. E0 should not be too defuse, otherwise it may takes a long time for the chain to converge. Recommended values is $a \cdot I$, where a is between 1.67 and 10. No default is provided.
nlag	A scalar of the lag order p, which should be an integer equal to or greater than 1. At this stage, the function does not support the GLMM-AR(0) moel, which can be estimated by using BUGS or JAGS.
pgm	Should the parameter expantion method be applied? It takes two values: TRUE or FALSE, and the default is TRUE.
tracking	The tracking interval in the simulation. Every "tracking" iterations, the function will return the information about how many iterations in total have been done. The default is 0 and no tracking information will be given during the simulation.
init.phi	A vector of starting values of the autoregressive coefficients.
init.beta	A vector of starting values of β . The default is 0.
init.b	A vector of starting values of b_i . The default is 0.
init.c	A vector of starting values of c_t . The default is 0.
monitor	A string contains the names of parameters whose MCMC output are to be returned. The string should be a subset of ("rho", "beta", "bi", "ct", "D", "E", "ystar", "u") which is also the default. the meanings of the parameter names are: "rho" – ρ , the autoregressive coefficients; "beta" – β , the fixed-effect coefficients; "bi" – b_i , the subject-level residuals; "ct" – c_t , the time-level errors residuals; "D" – the covariance matrix of b_i ; "E" – the covariance matrix of c_t ; "ystar" – the augmented data; and "u" – the auxiliary variable.
marginal.likelihood	Should the marginal likelihood be calculated. The default is FALSE.
reduced.mcmc	The number of iterations to return in the reduced mcmc simulations.
reduced.burnin	The number of burn-in iterations for the sampler in the reduced mcmc simulations.

Details

GLMMARp.Binary simulates from the posterior distribution of a probit multilevel model with a p th-order autoregressive error process by using the Cholesky-decomposition-and-auxiliary-variable approach. Please consult the coda documentation for a comprehensive list of functions that can be used to analyze the posterior sample.

The model has a structural form and a reduced form. The user fits the data in the function according to the structural form, and the function will estimate the reduced form. Therefore, the posteriors of the fixed-effect coefficients include the fixed-effect coefficients at both the individual and group

levels. The function only returns the group-level residuals, and the random-effect coefficients at the individual level have to be recovered by using the function `random.time` and `random.unit`.

The structural form of the GLMM-AR(p) model is as following:

$$y_{i,t_i} = \mathbf{I}(z_{i,t_i} > 0)$$

$$z_{i,t_i} = \mathbf{x}'_{1i,t_i}\beta_1 + \mathbf{w}'_{i,t_i}\beta_{2i} + \mathbf{s}'_{i,t_i}\beta_{3t_i} + \xi_{i,t_i}$$

$$\beta_{2i} = \mathbf{A}_i\beta_2 + \mathbf{b}_i$$

$$\beta_{3t_i} = \mathbf{F}_{t_i}\beta_3 + \mathbf{c}_{t_i}$$

$$\xi_{i,t_i} = \rho_1\xi_{i,t_i-1} + \dots + \rho_p\xi_{i,t_i-p} + \epsilon_{i,t_i}$$

and the reduced form of the model at the latent levels is

$$z_{i,t_i} = \mathbf{x}'_{i,t_i}\beta + \mathbf{w}'_{i,t_i}\mathbf{b}_i + \mathbf{s}'_{i,t_i}\mathbf{c}_{t_i} + \xi_{i,t_i}$$

$$\xi_{i,t_i} = \rho_1\xi_{i,t_i-1} + \dots + \rho_p\xi_{i,t_i-p} + \epsilon_{i,t_i}$$

Value

A list contains the posterior sample of all monitored parameters. This object can be converted into mcmc object by using the `coda` function `as.mcmc`, then be summarized by functions provided by the `coda` package.

References

- Pang, Xun. 2009. "Intertemporal and Contemporaneous Dependence in Binary Time-Series Cross-Sectional Data: Bayesian Hierarchical Model with AR(p) Errors and Non-nested Clustering." <http://xpang.wustl.edu/Dissertation.html>.
- Albert, James A. and Siddhartha Chib. 1993. "Bayesian Analysis of Binary and Polychotomous Response Data." *Journal of the American Statistical Association*. 88: 669-679.
- Chib, Siddhartha. 1995. "Marginal Likelihood from the Gibbs Output." *Journal of the American Statistical Association*. 90: 1313-1321.
- Chib, Siddhartha, and Ivan Jeliazkov. 2001. "Marginal Likelihood from the Metropolis-Hastings Output." *Journal of the American Statistical Association*. 96: 270-281.

Examples

```

## Not run:

## Example 1: GLMM-AR(p) model only with subject-specific and time-specific random intercept
## and serial correlation; no group-level predictors

data(StateFailure)
require(bayesSurv)
require(panel)
y <- StateFailure$failure
Unit <- StateFailure$country
Time <- StateFailure$year
Fixed <- cbind(StateFailure$polddemoc, StateFailure$bnkv123, StateFailure$bnkv117,
               StateFailure$polddurab, StateFailure$faocalry, StateFailure$pwtcgdp,
               StateFailure$macnac, StateFailure$macnciv, StateFailure$wdiinfmt,
               StateFailure$ptsamnes, StateFailure$dis1, StateFailure$bnkv81,
               StateFailure$change.demo)

nphi <- 0.5
nbeta <- 0
nb <- 0
nc <- 0

UnitRandom <- as.matrix(rep( 1, length(y)))
TimeRandom <- as.matrix(rep( 1, length(y)))
UnitPred <- "NULL"
TimePred <- "NULL"

State0AR1 <- GLMMARp.Binary(y=y, X1=Fixed, Wi=UnitRandom, St=TimeRandom, Ai=UnitPred, Ft=TimeP
                        Unit.Index=Unit, Time.Index=Time, timeint.add=0, unitint.add=0, m=10
                        burnin=500, init.phi=nphi, init.b=nb, init.c=nc, init.beta=nbeta,
                        beta0=rep(0,13), B0=diag(10, 13),
                        D0=1.67 ,E0=6, e0=5, d0=5,  pgm=TRUE, nlag=1, tracking=5,
                        monitor=c("rho", "beta", "bi", "ct", "D", "E", "ystar", "u"),
                        marginal.likelihood=TRUE, reduced.mcmc=500, reduced.burnin=100, thin=

## Example 2: GLMM-AR(p) model with a subject-specific coefficient, random intercepts
## and serial correlation; no group-level predictors

UnitRandom <- cbind(log(StateFailure$pwtopen))
TimeRandom <- as.matrix(rep( 1, length(y)))
UnitPred <- "NULL"
TimePred <- "NULL"

StateAR1 <- GLMMARp.Binary(y=y, X1=Fixed, Wi=UnitRandom, St=TimeRandom, Ai=UnitPred, Ft=TimeP
                        Unit.Index=Unit, Time.Index=Time, timeint.add=0, unitint.add=1, m=10
                        burnin=500, init.phi=nphi, init.b=nb,
                        init.c=nc, init.beta=nbeta, beta0=rep(0,14), B0=diag(10, 14),
                        D0=diag(1.67,2) ,E0=6, e0=3, d0=4,  pgm=TRUE, nlag=1, tracking=5,
                        monitor=c("rho", "beta", "bi", "ct", "D", "E", "ystar", "u"),
                        marginal.likelihood=TRUE, reduced.mcmc=500, reduced.burnin=100, thin=

## Example 3: GLMM-AR(p) model with a subject-specific coefficient, random intercepts

```

```

##          and serial correlation; only subject-level predictors.

TimeRandom <- as.matrix(rep( 1, length(y)))
UnitPred <- cbind(StateFailure$smacnac, StateFailure$poldemoc)
TimePred <- "NULL"
nphi <- c(0.2, 0.1)

StateAR2 <- GLMMARp.Binary(y=y,X1=Fixed, Wi=UnitRandom, St=TimeRandom, Ai=UnitPred, Ft=TimeP
Unit.Index=Unit, Time.Index=Time, timeint.add=0, unitint.add=1, m=10
burnin=500, init.phi=nphi, init.b=nb,
init.c=nc, init.beta=nbeta,beta0=rep(0,15),B0=diag(10, 15),
D0=diag(1.67,2) ,E0=6,e0=5, d0=6, pgm=TRUE, nlag=2, tracking=5,
monitor=c("rho", "beta", "bi", "ct", "D", "E", "ystar", "u"),
marginal.likelihood=TRUE, reduced.mcmc=500, reduced.burnin=100, thin

## Example 3: GLMM-AR(p) model with a subject-specific coefficient, random intercepts
##          and serial correlation; with both subject- and time-level predictors.

TimePred <- cbind(StateFailure$dis1)
nphi <- c(0.3, 0.1, 0.1)

StateAR3 <- GLMMARp.Binary(y=y,X1=Fixed, Wi=UnitRandom, St=TimeRandom, Ai=UnitPred, Ft=TimeP
Unit.Index=Unit, Time.Index=Time, timeint.add=1, unitint.add=1, m=10
burnin=500, init.phi=nphi, init.b=nb,
init.c=nc, init.beta=nbeta,beta0=rep(0,16),B0=diag(10, 16),
D0=diag(1.67,2) ,E0=diag(1.67,2),e0=5, d0=6, pgm=TRUE, nlag=3, track
monitor=c("rho", "beta", "bi", "ct", "D", "E", "ystar", "u"),
marginal.likelihood=TRUE, reduced.mcmc=500, reduced.burnin=100, thin

## End(Not run)

```

Marg.Like.Binary	<i>Compute the Marginal Likelihood of the GLMM-AR(p) Model by Using the MCMC Output and Reduced Samples</i>
------------------	---

Description

This function estimates the marginal likelihood for model comparison by using the Bayes factor. It is built in the `GLMMARp.Binary` function, but can also be used independently, and the way to fit in the function is similar to how to fit in the `GLMMARp.Binary` function.

Usage

```

Marg.Like.Binary(y,X1, Wi, St, Ai="NULL", Ft="NULL", Unit.Index,
Time.Index, timeint.add=FALSE,unitint.add=FALSE,
mcmc.output, reduced.mcmc, reduced.burnin, nlag,
beta0,B0, D0 , d0, E0, e0, tracking)

```

Arguments

<code>y</code>	A vector of response variable, dichotomous (0/1).
<code>X1</code>	A matrix of covariates with fixed effects.
<code>Wi</code>	A matrix of covariates with subject-varying effects.
<code>St</code>	A matrix of covariates with time-varying effects.
<code>Ai</code>	A matrix of covariates explaining the subject-varying effects with the same length of <code>y</code> (the values of time-invariant variables have to be repeated over the same times of each subject, for time-varying covariate in <code>Ai</code> , the function will automatically use the within-subject mean). The default is "NULL", no group-level covariates.
<code>Ft</code>	matrix of covariates explaining the time-varying effects with the same length of <code>y</code> (the function will automatically use the within-time-period mean). The default is "NULL", no group-level covariates.
<code>Unit.Index</code>	A vector of subject index, i 's. Note: the number of observations of each unit should be larger than the lag order, <code>nlag</code> . Those units which have fewer than or equal to <code>nlag</code> observations should be taken out of the sample in order to use the function.
<code>Time.Index</code>	A vector of time index, t 's. Note: no missing observations in the middle of the sample time periods of a unit are allowed. In other words, unbalanced data structures are allowed, but no broken data structure.
<code>timeint.add</code>	Should a time-specific intercept be added into the model? It takes two values: TRUE or FALSE with default as FALSE.
<code>unitint.add</code>	Should a subject-specific intercept be added into the model? It takes two values: TRUE or FALSE with default as FALSE.
<code>reduced.mcmc</code>	The number of iterations to return in the reduced mcmc simulations.
<code>reduced.burnin</code>	The number of burn-in iterations for the sampler in the reduced mcmc simulations.
<code>mcmc.output</code>	The mcmc output from the full simulation of GLMM-AR(p) model. The format has to be the same as in the <code>GLMMARp.Binary()</code> function.
<code>beta0</code>	The prior mean of β . This should be a vector with dimension equal to the number of fixed effect parameters in the reduced form. Since the dimension is difficult for the user to compute when the model contains multiple random coefficients and multiple group-level predictors, the function will provide the correct dimension in the error message if the dimension of <code>beta0</code> is specified incorrectly, and the user can respecify <code>beta0</code> with this information and recall the function. No default is provided.
<code>B0</code>	The prior covariance matrix of β . This should be a positive definite matrix with dimensions equal to the number of betas in the reduced form of GLMM-AR(p). No default is provided.
<code>d0, D0</code>	The degree of freedom and scale matrix of the Wishart prior on b_i which is the subject-level residual. <code>D0</code> should not be too defuse, otherwise it may takes a long time for the chain to converge. Recommended values is $a*I$, where a is between 1.67 and 10. No default is provided.

Prediction.Binary *Within-Sample and Out-of-Sample Prediction Based on the Posteriors Returned from GLMMARp.Binary*

Description

This functions conducts Bayes prediction by using the posterior distributions of the parameters simulated with the Markov Chain Monte Carlo method.

Usage

```
Prediction.Binary(y, X1, Wi, St, Ai="NULL", Ft="NULL", Unit.Index, Time.Index,
  timeint.add=FALSE, unitint.add=FALSE, mcmc.output, nlag)
```

Arguments

y	A vector of the dichotomous responses, taking two unique values of 0 and 1.
X1	A matrix of covariates with fixed effects.
Wi	A matrix of covariates with subject-varying effects.
St	A matrix of covariates with time-varying effects.
Ai	A matrix of group-level predictors explaining the subject-varying effects. The number of rows of this matrix is equal to the length of y. If some or all predictors are time-varying, the function will automatically use the within-subject mean; for time-invariant predictors, they should be repeated over the same times of each subject. The default is "NULL", and there are no group-level covariates.
Ft	A matrix of group-level predictors explaining the time-varying effects. The number of rows of this matrix is equal to the length of y. If some or all predictors are subject-varying, the function will automatically use the within-time-period mean; for subject-invariant predictors, they should be repeated over the same number of subjects in the specific time period. The default is "NULL", and there are no group-level covariates.
Unit.Index	A vector of the subject index, i.e., the i's. Note: the number of observations of each unit should be larger than the lag order, nlag. Those units which have fewer than or equal to nlag observations should be taken out of the sample in order to use the function.
Time.Index	A vector of the time index, i.e., the t's. Note: no missing observations in the middle of the sample time periods of a unit are allowed. In other words, unbalanced data structures are allowed, but no broken data structure.
timeint.add	Should a time-specific intercept be added into the model? It takes two values: TRUE or FALSE with default as FALSE. Be sure that there is no time-specific intercept already in the reduced form of the model before adding an intercept here.

unitint.add	Should a subject-specific intercept be added into the model? It takes two values: TRUE or FALSE with default as FALSE. Be sure that there is no subject-specific intercept already in the reduced form of the model before adding an intercept here.
mcmc.output	The MCMC output generated by the full MCMC by using the function of <code>GLMMARp.Binary</code> . takes two values: TRUE or FALSE with default as FALSE.
nlag	A scalar of the lag order p , which should be an integer equal to or greater than 1.

Value

A matrix with the row dimension equal to the number of iterations returned by `GLMMARp.Binary`; with the column dimension equal to the length of the response variable. Therefore, each column forms a predictive distribution of one observation. The user can convert the output into an mcmc output by using the `coda` package, and summarize the predicted distribution with various `coda` functions.

Examples

```
## Not run:
## Example
require(panel)
require(bayesSurv)
data(StateFailure)
y <- StateFailure$failure
Unit <- StateFailure$country
Time <- StateFailure$year
Fixed <- cbind(StateFailure$poldemoc, StateFailure$bnkv123, StateFailure$bnkv117,
               StateFailure$poldurab, StateFailure$faocalry, StateFailure$pwtcgdp,
               StateFailure$macnac, StateFailure$macnciv, StateFailure$wdiinfmt,
               StateFailure$ptsamnes, StateFailure$dis1, StateFailure$bnkv81,
               StateFailure$change.demo)

UnitRandom <- cbind(log(StateFailure$pwtopen))
TimeRandom <- as.matrix(rep(1, length(y)))
UnitPred <- cbind(StateFailure$macnac, StateFailure$poldemoc)
TimePred <- "NULL"

data(StateAR2)
prediction <- Prediction.Binary(y=y, X1=Fixed, Wi=UnitRandom, St=TimeRandom, Ai=UnitPred,
                               Ft=TimePred, Unit.Index=Unit, Time.Index=Time, timeint.add=0,
                               unitint.add=1, mcmc.output=StateAR2, nlag=2)

## End(Not run)
```

random.time

Compute the Time-Specific Coefficients in the Structural Form of GLMM-AR(p)

Description

This functions recovers the time-specific random-effect coefficients from the reduced form estimates. This function is used when the β_{3,t_i} does not only include a random intercept.

Usage

```
random.time(y, X1, Wi, St, Ai, Ft, Unit.Index, Time.Index, timeint, model)
```

Arguments

y	A vector of the dichotomous responses, taking two unique values of 0 and 1.
X1	A matrix of covariates with fixed effects.
Wi	A matrix of covariates with subject-varying effects.
St	A matrix of covariates with time-varying effects.
Ai	A matrix of group-level predictors explaining the subject-varying effects. The number of rows of this matrix is equal to the length of y. If some or all predictors are time-varying, the function will automatically use the within-subject mean; for time-invariant predictors, they should be repeated over the same times of each subject. The default is "NULL", and there are no group-level covariates.
Ft	A matrix of group-level predictors explaining the time-varying effects. The number of rows of this matrix is equal to the length of y. If some or all predictors are subject-varying, the function will automatically use the within-time-period mean; for subject-invariant predictors, they should be repeated over the same number of subjects in the specific time period. The default is "NULL", and there are no group-level covariates.
Unit.Index	A vector of the subject index, i.e., the i's. Note: the number of observations of each unit should be larger than the lag order, nlag. Those units which have fewer than or equal to nlag observations should be taken out of the sample in order to use the function.
Time.Index	A vector of the time index, i.e., the t's. Note: no missing observations in the middle of the sample time periods of a unit are allowed. In other words, unbalanced data structures are allowed, but no broken data structure.
timeint	Is a time-specific intercept added into the model?
model	The MCMC output generated by the full MCMC by using the function of GLMMARp.Binary.

Value

A matrix with the row dimension equal to the number of iterations returned from GLMMARp.Binary; with the column dimension equal to the number of variables, S , times the number of sample time periods, T .

Examples

```
## Not run:
## Example 1: Compute time-specific random effects
data(StateFailure)
```

```

require(bayesSurv)
data(StateFailure)
y <- StateFailure$failure
Unit <- StateFailure$country
Time <- StateFailure$year
Fixed <- cbind(StateFailure$poldemoc, StateFailure$bnkv123, StateFailure$bnkv117,
               StateFailure$poldurab, StateFailure$faocalry, StateFailure$pwtcgdp,
               StateFailure$macnac, StateFailure$macnciv, StateFailure$wdiinfmt,
               StateFailure$ptsamnes, StateFailure$dis1, StateFailure$bnkv81,
               StateFailure$change.demo)

UnitRandom <- cbind(log(StateFailure$pwtopen))
TimeRandom <- as.matrix(rep( 1, length(y)))
UnitPred <- cbind(StateFailure$macnac, StateFailure$poldemoc)
TimePred <- cbind(StateFailure$dis1)
data(StateAR3)
betat <- random.time (y=y,X1=Fixed, Wi=UnitRandom, St=TimeRandom, Ai=UnitPred, Ft=TimePred,
                     Unit.Index=Unit, Time.Index=Time, model=StateAR3,timeint=1)

## End(Not run)

```

random.unit

Compute the Subject-Specific Coefficients in the Structural Form of GLMM-AR(p)

Description

This functions recovers the subject-specific random-effect coefficients from the reduced form estimates. This function is used when the β_{2i} does not only include a random intercept.

Usage

```
random.unit(y, X1, Wi, St, Ai, Ft, Unit.Index, Time.Index, unitint, model)
```

Arguments

y	A vector of the dichotomous responses, taking two unique values of 0 and 1.
X1	A matrix of covariates with fixed effects.
Wi	A matrix of covariates with subject-varying effects.
St	A matrix of covariates with time-varying effects.
Ai	A matrix of group-level predictors explaining the subject-varying effects. The number of rows of this matrix is equal to the length of y. If some or all predictors are time-varying, the function will automatically use the within-subject mean; for time-invariant predictors, they should be repeated over the same times of each subject. The default is "NULL", and there are no group-level covariates.

<code>Ft</code>	A matrix of group-level predictors explaining the time-varying effects. The number of rows of this matrix is equal to the length of <code>y</code> . If some or all predictors are subject-varying, the function will automatically use the within-time-period mean; for subject-invariant predictors, they should be repeated over the same number of subjects in the specific time period. The default is "NULL", and there are no group-level covariates.
<code>Unit.Index</code>	A vector of the subject index, i.e., the <code>i</code> 's. Note: the number of observations of each unit should be larger than the lag order, <code>nlag</code> . Those units which have fewer than or equal to <code>nlag</code> observations should be taken out of the sample in order to use the function.
<code>Time.Index</code>	A vector of the time index, i.e., the <code>t</code> 's. Note: no missing observations in the middle of the sample time periods of a unit are allowed. In other words, unbalanced data structures are allowed, but no broken data structure.
<code>unitint</code>	Is a subject-specific intercept added into the model?
<code>model</code>	The MCMC output generated by the full MCMC by using the function of <code>GLMMARp.Binary</code> .

Value

A matrix with the row dimension equal to the number of iterations returned from `GLMMARp.Binary`; with the column dimension equal to the number of variables, W , times the number of sample subjects, N .

Examples

```
## Not run:
## Example 1: Compute uni-specific random effects
data(StateFailure)
require(bayesSurv)
data(StateFailure)
y<-StateFailure$failure
Unit<-StateFailure$country
Time<-StateFailure$year
Fixed<-cbind(StateFailure$poldemoc, StateFailure$bnkv123, StateFailure$bnkv117,
             StateFailure$poldurab, StateFailure$faocalry, StateFailure$pwtcgdp,
             StateFailure$macnac, StateFailure$macnciv, StateFailure$wdiinfmt,
             StateFailure$ptsamnes, StateFailure$dis1, StateFailure$bnkv81,
             StateFailure$change.demo)

UnitRandom<-cbind(log(StateFailure$pwtopen))
TimeRandom<-as.matrix(rep( 1, length(y)))
UnitPred<-cbind(StateFailure$macnac, StateFailure$poldemoc)
TimePred<- "NULL"
data(StateAR2)
betai <- random.unit (y=y,X1=Fixed, Wi=UnitRandom, St=TimeRandom, Ai=UnitPred, Ft=TimePred,
                    Unit.Index=Unit, Time.Index=Time, model=StateAR2, unitint=1)
## End(Not run)
```

Description

This function generates a sample from the reduced Markov Chain Monte Carlo simulation for computing the Bayes factor for model comparison. The sample cannot be used for inferences about the parameters, but only used for the Bayes factor calculation. The function is based on the reduced form of the GLMM-AR(p) model. It is not necessary to check convergence on the sample.

Usage

```
Reduced.Binary(Y,X, W, S, unit.id, time.id, m=1000, burnin=500, mcmc.pos,
              beta0,B0, D0, d0, E0, e0, nlag,tracking,
              monitor=c("rho","q.rho", "beta", "bi", "ct", "D", "E", "ystar", "u")
              fixed="NULL")
```

Arguments

Y	A vector of the dichotomous response variable taking values of 0 and 1.
X	A matrix of covariates with fixed effects at all levels.
W	A matrix of covariates with subject-varying effects.
S	A matrix of covariates with time-varying effects.
unit.id	A vector of subject index, i.e., the i's. Note: the number of observations of each unit should be larger than the lag order, nlag. Those units which have fewer than or equal to nlag observations should be taken out of the sample in order to use the function.
time.id	A vector of time index, i.e., the t's. Note: no missing observations in the middle of the sample time periods of a unit are allowed. In other words, unbalanced data structures are allowed, but no broken data structure.
m	The number of iterations after burn-in and to be returned.
burnin	The number of burn-in iterations for the sampler.
mcmc.pos	The MCMC output generated by the full MCMC by using the function of <code>GLMMARp.Binary</code> .
beta0	The prior mean of β . This should be a vector with dimension equal to the number of fixed-effect parameters in the reduced form. Since the dimension is difficult for the user to compute when the model contains multiple random coefficients and multiple group-level predictors, the function will provide the correct dimension in the error message if the dimension of beta0 is incorrectly specified, and the user can respecify beta0 with this information and recall the function.
B0	The prior covariance matrix of β . This should be a positive definite matrix with dimension equal to the number of betas in the reduced form of GLMM-AR(p).
d0, D0	The degree of freedom and scale matrix of the Wishart prior on b_i which are the subject-level residuals. D0 should not be too defuse, otherwise it may takes a long time for the chain to converge. Recommended values is $a*I$, where a is between 1.67 and 10. No default is provided.

<code>e0, E0</code>	The degree of freedom and scale matrix of the Wishart prior on c_t which are the time-level residuals. E0 should not be too defuse, otherwise it may takes a long time for the chain to converge. Recommended values is $a \cdot I$, where a is between 1.67 and 10. No default is provided. All the priors put into this function should be the same as those used in the full MCMC.
<code>nlag</code>	A scalar of the lag order p , which should be an integer equal to or greater than 1. At this stage, the function does not support the GLMM-AR(0) model, which can be estimated by using BUGs or JAGs.
<code>tracking</code>	The tracking interval in the simulation. Every "tracking" iterations, the function will return the information about how many iterations in total have been done. The default is 0 and no tracking information will be given during the simulation.
<code>monitor</code>	A string contains the names of parameters whose MCMC output are to be returned. The string has to be a subset of ("rho", "beta", "bi", "ct", "D", "E", "ystar", "u") which is also the default.
<code>fixed</code>	A string specifying which parameters are fixed in the reduced simulation.

Details

`ReducedMCMC.Binary` is a function for conducting reduced runs for empirically integration in the process of computing the Bayes factor. The partition of the parameters is set by the function, and the user is not given the option of re-partitioning those parameters.

Value

A list that contains the posterior sample. The same format as the sample returned by `GLM-MARp.Binary()`.

References

- Albert, James A. and Siddhartha Chib. 1993. "Bayesian Analysis of Binary and Polychotomous Response Data." *Journal of the American Statistical Association*. 88: 669-679.
- Chib, Siddhartha. 1995. "Marginal Likelihood from the Gibbs Output." *Journal of the American Statistical Association*. 90: 1313-1321.
- Chib, Siddhartha, and Ivan Jeliazkov. 2001. "Marginal Likelihood from the Metropolis-Hastings Output." *Journal of the American Statistical Association*. 96: 270-281.

Examples

```
## Not run:

## End(Not run)
```

`StateAR2`*A List of the MCMC Output Based on The GLMM-AR(p) Model*

Description

This list contains the mcmc output from the R function of `GLMMARp.Binary()`.

Usage

```
data(StateAR2)
```

Format

The mcmc output as a list contains the posterior distributions of all the parameters in the GLMM-AR(p) model, produced by the function of `GLMMARp.Binary()`.

Source

Original data provided by the author of the GLMMarp Package.

`StateAR3`*A List of the MCMC Output Based on The GLMM-AR(p) Model*

Description

This list contains the mcmc output from the R function of `GLMMARp.Binary()`.

Usage

```
data(StateAR2)
```

Format

The mcmc output as a list contains the posterior distributions of all the parameters in the GLMM-AR(p) model, produced by the function of `GLMMARp.Binary()`.

Source

Original data provided by the author of the GLMMarp Package.

`StateFailure`*Sub-Saharan State Failure Study Data*

Description

This dataframe contains a matrix in which is a TSCS data set about state failure events in the sub-Saharan countries

Usage

```
data(StateFailure)
```

Format

The data set is based on the Political Instability Task Force (PITF) 2001 data base. The data used here only contain the observations in the sub-Saharan countries. There are many missing data in the original data base, but missingness in this used data set has already been handled with the multiple imputation method.

Source

Political Instability Task Force. *PITF Phase III Findings*. <http://globalpolicy.gmu.edu/pitf/>.

Index

*Topic **datasets**

StateAR2, [16](#)

StateAR3, [16](#)

StateFailure, [17](#)

*Topic **manip**

Prediction.Binary, [9](#)

random.time, [11](#)

random.unit, [12](#)

*Topic **models**

GLMMARp.Binary, [2](#)

Marg.Like.Binary, [7](#)

Reduced.Binary, [14](#)

GLMMARp.Binary, [2, 8](#)

Marg.Like.Binary, [7](#)

Prediction.Binary, [9](#)

random.time, [11](#)

random.unit, [12](#)

Reduced.Binary, [14](#)

StateAR2, [16](#)

StateAR3, [16](#)

StateFailure, [17](#)