Package 'pnmtrem'

July 2, 2014

Type Package

Title Probit-Normal Marginalized Transition Random Effects Models

Version 1.3

Date 2013-05-19

Author Ozgur Asar, Ozlem Ilk

Maintainer Ozgur Asar <o.asar@lancaster.ac.uk>

Depends MASS

Description An R package for Probit-Normal Marginalized Transition Random Effects Models

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2013-05-19 16:09:03

R topics documented:

pnmtrem *Probit-Normal Marginalized Transition Random Effects Models*

Description

Fits Probit-Normal Marginalized Transition Random Effects Models which is proposed for modeling multivariate longitudinal binary data by

Asar, O., Ilk, O., Sezer, A. D. (2013). A marginalized multilevel model for analyzing multivariate longitudinal binary data. Submitted.

Details

Package:	pnmtrem
Type:	Package
Version:	1.3
Date:	2013-05-19
License:	GPL (>=2)

pnmtrem1 *Function to fit first-order Probit-Normal Marginalized Transition Random Effects Models, PNMTREM(1)*

Description

Fits PNMTREM(1) via maximum likelihood estimation with Fisher-Scoring Algorithm.

Usage

```
pnmtrem1(covmat1, covmat2, respmat1, respmat2, z, nsubj, nresp, param01,
param02, beta0, alpha0, tol1 = 1e-04, tol2 = 1e-04, maxiter1 = 50,
maxiter2 = 50, tun1 = 1,tun2 = 1, x01 = 0, eps1 = 10^-10, x02 = 0,
eps2 = 10^-10, silent = TRUE, delta.print = FALSE, deltastar.print = FALSE)
```

Arguments

covmat1	a $(p_1 + 1) \times N \times k$ matrix or data frame, which has the design matrix form, for the baseline time point $(t = 1)$. Here, p_1 is the number of independent variables included in the baseline model, N is the number of subjects and k is the number of multiple responses.

`covmat2` a $(p_2+1) \times N \times k \times (T-1)$ matrix or data frame, which has the design matrix form, for $t \geq 2$. Here, p_2 is the number of independent variables included in the $t \geq 2$ model, N is the number of subjects, k is the number of multiple responses and T is the number of repeated measurements per subject.

`respmat1` an $(N * k) \times 1$ matrix or data frame for the multiple responses at baseline. The general form of it can be depicted as Respmat$_1$=$(Y_{.11}, \ldots, Y_{.1k})^T$ where $Y_{.1j} = (Y_{11j}, \ldots, Y_{N1j})$.

`respmat2` an $(N * k * T) \times 1$ matrix or data frame for the multiple responses for $t \geq 1$. The general form of it can be illustrated as
Respmat$_2$ $= (Y_{.11}, \ldots, Y_{.1k}, \ldots, Y_{.T1}, \ldots, Y_{.Tk})^T$ where $Y_{.tj} = (Y_{1tj}, \ldots, Y_{Ntj})$.

`z` a $(p_3+1) \times N \times k \times (T-1)$ matrix or data frame to be included in the second level of the $t \geq 2$ model. `z` typically includes a subset of covariates.

`nsubj` an integer which defines the number of subjects in the study.

`nresp` an integer which defines the number of multiple binary responses.

`param01` a length of $[(p_1+1)+(k-1)+1]$ vector where p_1 is the number of covariates included in the baseline model and k is number of multiple responses. `param01` is used to start the Fisher-Scoring (FS) algorithm for the baseline model. The general form of it can be given as param01 $= (\beta^*, \lambda_j^*, c_1)$, where $j = 2, \ldots, k$ and c_1=log(σ_1).

`param02` a length of $[(p_2+1)+(p_3+1)*(T-1)+(k-1)+(T-1)]$ vector where p_2 is the number of covariates included in the first level of $t \geq 2$ model, p_3 is the number of covariates included in the second level of it, k is the number of multiple responses and T is the number of repeated measurements per subject. `param02` is used to start the FS algorithm for the $t \geq 2$ model. The general form of it can be given as param02 $= (\beta, \alpha_{t,1}, \lambda_j, c_t)$, where $\alpha_{t,1} = (\alpha_{21,1}, \ldots, \alpha_{2p3,1}, \ldots, \alpha_{T1,1}, \ldots, \alpha_{Tp3,1})$ and $j = 2, \ldots, k$ and $t = 2, \ldots, T$ and $c_t = log(\sigma_t)$.

`beta0` a $(p_2+1) \times 1$ matrix for which all the elements are set to 0. It corresponds to the β_0 component of the Implicit Function Theorem (IFT) point, P_0.

`alpha0` a $(p_3+1) \times (T-1)$ matrix for which all the elements are set to 0. It corresponds to the $\alpha_{t,10}$ component of the P_0.

`tol1` the amount of tolerance for the convergence of the FS algorithm for baseline model. The default is set to 0.0001.

`tol2` the amount of tolerance for the convergence of the FS algorithm for $t \geq 2$ model. The default is set to 0.0001.

`maxiter1` the maximum number of iterations expected to be consumed by the FS algorithm for baseline model. The default is set to 50.

`maxiter2` the maximum number of iterations expected to be consumed by the FS algorithm for $t \geq 2$ model. The default is set to 50.

`tun1` the tuning parameter for baseline model need to be chosen preferably as integer to decrease the FS steps in each iteration in cases where the algorithm might miss the convergence of the parameters. The default is set to 1.

`tun2` the tuning parameter for $t \geq 2$ model to decrease the FS steps in each iteration as in the case of `tun1`. The default is set to 1.

`x01` an integer defined for the initial values of the Newton-Raphson (N-R) algorithm to obtain Δ_{i2j0}. The default is set to 0.

`eps1` the amount of tolerance for the convergence of N-R algorithm to obtain Δ_{i2j0}. The default is set to 10^{-10}.

`x02` an integer defined for the initial values of the Newton-Raphson (N-R) algorithm to obtain the empirical Bayesian estimates of the individual characteristics, $\hat{z}_i$. The default is set to 0.

`eps2` the tolerance defined for the convergence of N-R algorithm to obtain $\hat{z}_i$. The default is set to 10^{-10}.

`silent` a logical statement to decide whether the details of the FS algorithm details for both the baseline and t ≥ 2 models to be printed. The default is set to TRUE which means not printing these details.

`delta.print` a logical statement to decide the print of the estimates of Δ_{itj} where $t = 2, \ldots, T$ together with the modeling outputs. The default is set to FALSE which means not printing these estimates.

`deltastar.print` a logical statement to decide the print of the estimates of Δ^*_{itj} where $t = 1, \ldots, T$ together with the modeling outputs. The default is set to FALSE which means not printing these estimates.

Details

The modeling framework assumes two different models: 1) a model for baseline time point (baseline model), a two-level one, and 2) a model for later time points ($t \geq 2$ model), a three-level one. These two models are linked to each other via a marginal constraint equation. Both of them are marginalized models and capture marginal effect of independent variables on the mean responses in their first levels. While the former captures the multivariate response dependence in its second level, the latter captures this dependence in its third level. Furthermore, the $t \geq 2$ model captures the serial dependence in its second level. Implicit function theorem, specifically first-order implicit differentiation was used to explicitly link first and second level of the $t \geq 2$ model. All the integrals are approximated via 20-point Gauss-Hermite Qudratures. Logarithm of the standard deviation parameters of random effects distributions are modeled. A detailed example in terms of data preparation, initial obtaining and setting is provided below.

Value

`pnmtrem1` returns the modeling output of baseline and $t \geq 2$ models and the associated maximized log-likelihood values. Additionally, it automatically prints the empirical Bayesian estimates of the individual characteristics, $\hat{z}_i$. The order of these estimates are in subject order. The estimates of Δ_{itj} (for $t = 2, ..., T$) and Δ^*_{itj} (for $t = 1, ..., T$) are in the same order of the responses and covariates.

Note

Version 1.3.

Author(s)

Ozgur Asar, Ozlem Ilk

References

Asar, O., Ilk, O., Sezer, A. D. (2013). A marginalized multilevel model for analyzing multivariate longitudinal binary data. Submitted.

Examples

```
## Not run:
## loading a simulated bivariate longitudinal binary data with 500 subjects
## and 4 time points
data(pnmtrem1.sim.data1)
data(pnmtrem1.sim.data2)

## number of subjects, multiple responses and time points
nsubj<-500
nresp<-2
ntime<-4

## sepearating the portion of data which pnmtrem1 function will use
covmat1<-as.matrix(pnmtrem1.sim.data1[,5:6])
covmat<-as.matrix(pnmtrem1.sim.data2[,5:7])
mresp1<-as.matrix(pnmtrem1.sim.data1[,4])
mresp<-as.matrix(c(pnmtrem1.sim.data1[,4],pnmtrem1.sim.data2[,4]))

## obtaining initials for \beta^*
glm1<-glm(mresp1~-1+covmat1,family=binomial(link=probit))
bsinit<-glm1$coef;names(bsinit)<-NULL

## initials for parameters in the baseline model, i.e. \beta^*, \lambda^*, c_1
param01<-c(bsinit,1,log(0.5))

## obtaining initials of \beta
# preparing data to be analyzed by mmm2
mresp.mmm<-as.matrix(pnmtrem1.sim.data2[,4])
id<-as.matrix(rep(seq(1:nsubj),((ntime-1)*nresp)))
time<-as.matrix(c(rep(2,nsubj*nresp),rep(3,nsubj*nresp),rep(4,nsubj*nresp)))
data<-cbind(id,time,mresp.mmm,covmat)

# ordering data by subject ID
data2<-NULL
for (i in 1:nsubj){
data.id<-data[data[,1]==i,]
data2<-rbind(data2,data.id)
}
# subsetting data by response type (6th column of data2)
data.resp1<-data2[data2[,6]==1,]
data.resp2<-data2[data2[,6]==0,]
data.mmm<-cbind(data.resp1[,1],data.resp1[,3],data.resp2[,3],data.resp1[,5])
library(mmm2)
```

```
mmm2.fit<-mmm2(data=data.mmm,nresp=2,family=binomial(link=probit),
corstr = "exchangeable")
binit<-coef(mmm2.fit)

## obtaining initials of \alpha
glm3<-glm(mresp[(nsubj*nresp+1):(2*nsubj*nresp),]~-1+mresp1,family=binomial(link=probit))
glm4<-glm(mresp[(2*nsubj*nresp+1):(3*nsubj*nresp),]~-1+
mresp[(nsubj*nresp+1):(2*nsubj*nresp),],family=binomial(link=probit))
glm5<-glm(mresp[(3*nsubj*nresp+1):(4*nsubj*nresp),]~-1+
mresp[(2*nsubj*nresp+1):(3*nsubj*nresp),],family=binomial(link=probit))
alpinit<-c(glm3$coef[1],glm4$coef[1],glm5$coef[1]);names(alpinit)<-NULL

## initials for parameters in the t \geq 2 model, i.e. \beta, \alpha, \lambda, c_2, c_3, c_4
param02<-c(binit,alpinit,1,log(0.5),log(0.5),log(0.5))

## implicit function initials, \beta_0 and \alpha_0
beta0<-matrix(c(0,0,0),ncol=1)
alpha0<-matrix(c(0,0,0),ncol=1)

## covariate set to be interacted with the response history
z<-matrix(rep(1,3*nsubj*nresp),ncol=1)

fit<-pnmtrem1(covmat1=covmat1,covmat2=covmat,respmat1=mresp1,respmat2=mresp,z=z,
nsubj=500,nresp=2,param01=param01,param02=param02,beta0=beta0,alpha0=alpha0,
tol1=0.0001,tol2=0.0001,maxiter1=50,maxiter2=50,tun1=1,tun2=1,x01=0,eps1=10^-10,
x02=0,eps2=10^-10,silent=FALSE,delta.print=TRUE,deltastar.print=TRUE)

## manipulation of the output
fit
fit$output1
fit$maxloglik1
fit$output2
fit$maxloglik2
fit$delta
fit$delstar
fit$empbayes
## End(Not run)
```

`pnmtrem1.sim.data1` *A portion of a simulated dataset, for the baseline time point (t=1), from a first-order Probit-Normal Marginalized Transition Random Effects Models for 500 subjects with 4 follow-ups*

Description

The dataset includes randomly generated bivariate longitudinal binary responses and an associated covariate which has a standard uniform distribution, U(0,1). The assumed parameters to generate the data are: $\beta^* = (\beta_0^*, \beta_1^*) = (-1, 1.9)$, $\lambda_j = (\lambda_1^*, \lambda_2^*) = (1, 1.07)$ and $b_{i1} \sim N(0, \sigma_1^2)$, $\sigma_1 = 0.7$. It is assumed that there are 500 subjects. The data include no missing value.

Usage

```
data(pnmtrem1.sim.data1)
```

Format

A data frame with 1000 observations on the following 6 variables.

`time` a numeric vector for the time information at which data is available

`response` a numeric vector with the response information for which data is available

`subject` a numeric vector for subject id

`y` a numeric vector for bivariate longitudinal binary responses

`ones` a numeric vector for which all the elements are 1

`x` a numeric vector for the covariate

Details

When one carefully investigates the `time`, `response` and `subject` orders, s/he can easily understand the data structure which the model accepts. Baseline and later time points of the data may include different number of independent variables. Therefore, datasets for $t = 1$ and $t \geq 2$ are presented in different data objects, `pnmtrem1.sim.data1` and `pnmtrem1.sim.data2`, respectively.

Examples

```
data(pnmtrem1.sim.data1)
head(pnmtrem1.sim.data1)
str(pnmtrem1.sim.data1)
```

---

pnmtrem1.sim.data2 *A portion of a simulated dataset, for $t \geq 2$ period, from a first-order Probit-Normal Marginalized Transition Random Effects Models for 500 subjects with 4 follow-ups*

Description

The dataset includes bivariate longitudinal binary responses and two associated covariates. The first covariate, X1 is a time-independent one which means it takes same values at t=1, 2, 3, 4. For the details of X1, see `pnmtrem1.sim.data1`. The second covariate, X2 is a response type indicator variable which takes 1 for the first response, and takes 0 for the second one. The assumed parameters to generate the data are: $\beta = (\beta_0, \beta_1, \beta_2) = (-1, 2, 0.2)$, $\alpha_{t,1} = (\alpha_{21,1}, \alpha_{31,1}, \alpha_{41,1}) = (0.5, 0.7, 0.9)$, $\lambda_j = (\lambda_1, \lambda_2) = (1, 1.05)$ and $b_{it} \sim N(0, \sigma_t^2)$, $\sigma_t = (\sigma_2, \sigma_3, \sigma_4) = (0.66, 0.63, 0.60)$. It is assumed that there 500 subjects. The dataset has no missing value.

Usage

```
data(pnmtrem1.sim.data2)
```

Format

A data frame with 3000 observations on the following 7 variables.

`time` a numeric vector for the time information at which data is available

`response` a numeric vector with the response information for which data is available

`subject` a numeric vector for subject id

`y` a numeric vector for bivariate longitudinal binary responses

`ones` a numeric vector for which all the elements are 1

`x1` a numeric vector for the first covariate, X1

`x2` a numeric vector for the second covariate, X2

Details

When one carefully investigates the `time`, `response` and `subject` orders, s/he can easily understand the data structure which the model accepts. Baseline and later time points of the data may include different number of independent variables. Therefore, datasets for $t = 1$ and $t \geq 2$ are presented in different data objects, `pnmtrem1.sim.data1` and `pnmtrem1.sim.data2`, respectively.

Examples

```
data(pnmtrem1.sim.data2)
head(pnmtrem1.sim.data2)
str(pnmtrem1.sim.data2)
```

Index