Package ‘RRMLRfMC’

June 7, 2021

Type Package

Title Reduced-Rank Multinomial Logistic Regression for Markov Chains

Version 0.4.0

Description Fit the reduced-rank multinomial logistic regression model for Markov chains developed by Wang, Abner, Fardo, Schmitt, Jicha, Eldik and Kryscio (2021)<doi:10.1002/sim.8923> in R. It combines the ideas of multinomial logistic regression in Markov chains and reduced-rank. It is very useful in a study where multi-states model is assumed and each transition among the states is controlled by a series of covariates. The key advantage is to reduce the number of parameters to be estimated. The final coefficients for all the covariates and the p-values for the interested covariates will be reported. The p-values for the whole coefficient matrix can be calculated by two bootstrap methods.

License GPL-2

Encoding UTF-8

LazyData true

Imports nnet

Depends R (>= 3.5.0)

RoxygenNote 7.1.1

Suggests rmarkdown, knitr

NeedsCompilation no

Author Pei Wang [aut, cre],
Richard Kryscio [aut]

Maintainer Pei Wang <wangp33@miamioh.edu>

Repository CRAN

Date/Publication 2021-06-07 07:20:07 UTC

R topics documented:

Aupdate ................................................................. 2
cogdat ................................................................. 3
Description

This function is used to update A matrix.

Usage

Aupdate(Dfix, Gamma, Adata, R, p, q, I, iniA, eps, refA)

Arguments

- **Dfix**: the coefficient matrix for study covariates
- **Gamma**: the G matrix value
- **Adata**: the dataset
- **R**: the rank of reduced rank model
- **p**: the number of covariates in the dimension reduction
- **q**: the number of study covariates
- **I**: a U by U incidence matrix with elements; I(i,j)=1 if state j can be accessed from state i in one step and 0 otherwise
- **iniA**: initial value for the iteration
- **eps**: the tolerance for convergence, default is 10^-5
- **refA**: a vector of reference categories

Value

A list of outputs:

- **NewA**: the updated A matrix
- **loglikeA**: the loglikelihood when updating A
Description
A dataset containing the states and covariates of 649 participants enrolled in the BRAiNS cohort at the University of Kentucky’s Alzheimer’s Disease Research Center.

Usage
cogdat

Format
A data frame with 6240 rows and 14 columns:

- **ID** used to denote the participants; from 1 to 649
- **visitno** used to denote the visit number for each participant
- **prstate** denote the previous state
- **custate** denote the current state
- **bagec** baseline age (centered at age 72)
- **famhx** family history of dementia
- **HBP** self reported high blood pressure
- **apoee** at least one Apolipoprotein-E (APOE) gene $\epsilon_4$ allele
- **smk1** cigarette smoking level (none versus < 10)
- **smk2** cigarette smoking level (11-19)
- **smk2** cigarette smoking level (>= 20 pack years))
- **lowed** low education
- **headinj** self reported head injury

Description
This function is used to calculate the loglikelihood with a given matrix $B=AG$

Usage
derivativeB(B, I, zy, refd)
Arguments

- **B**: a numeric coefficient matrix
- **I**: a U by U incidence matrix with elements; \( I(i,j)=1 \) if state \( j \) can be accessed from state \( i \) in one step and 0 otherwise
- **zy**: the variable values for a given observation
- **refd**: a vector of reference categories

Value

loglikelihood

Description

This function is used calculate the derivative values (first and second derivatives for Newton-Raphson method) and loglikelihood when updating \( A \)

Usage

```r
derivatives(A, Gamma, Dmat, I, zy, refA)
```

Arguments

- **A**: matrix with value from previous iteration
- **Gamma**: G matrix values
- **Dmat**: the coefficient matrix for the fixed variables,
- **I**: a U by U incidence matrix with elements; \( I(i,j)=1 \) if state \( j \) can be accessed from state \( i \) in one step and 0 otherwise
- **zy**: the variable values for a given observation
- **refA**: a vector of reference categories

Value

a list of outputs:

- **fird**: the first derivative value
- **secd**: the second derivative value
- **loglike**: the loglikelihood
### Description

This function is used to expand the Y(category) to a indicator vector.

### Usage

```r
expand(pri, curr, I, refE)
```

### Arguments

- `pri`: the prior state
- `curr`: the current state
- `I`: a U by U incidence matrix with elements; `I(i,j)=1` if state `j` can be accessed from state `i` in one step and `0` otherwise
- `refE`: a vector with the reference categories

### Value

`ry`: a indicator vector

---

### Description

This function is used to update G matrix.

### Usage

```r
Gupdate(A, Gdata, p, q, I, refG)
```

### Arguments

- `A`: numeric matrix
- `Gdata`: the dataset used to update G
- `p`: the number of covariates in the dimension reduction
- `q`: the number of study covariates
- `I`: a U by U incidence matrix with elements; `I(i,j)=1` if state `j` can be accessed from state `i` in one step and `0` otherwise
- `refG`: a vector of reference categories
Value

a list of outputs:

- NewG: the updated G matrix
- loglikeK: the loglikelihood when updating G
- sderr: standard errors for the coefficient matrix

Description

This function is used to normalize a vector to have unit length

Usage

\texttt{norm(x)}

Arguments

\texttt{x} \hspace{1cm} a numeric vector

Value

a normalized vector with length 1

Description

This function is used to fit the reduced rank multinomial logistic regression for markov chain

Usage

\texttt{rrmultinom(I, z1 = NULL, z2 = NULL, T, R, eps = 1e-05, ref = NULL)}
**Arguments**

- **I**
  - a U by U incidence matrix with elements; U is number of states; I(i,j)=1 if state j can be accessed from state i in one step and 0 otherwise

- **z1**
  - a n by p matrix with covariates involved in the dimension reduction(DR), n is the number of subjects, p is the number of covariates involved in DR

- **z2**
  - a n by q matrix with study covariates (not in dimension reduction), q is the number of study covariates

- **T**
  - a M by 3 state matrix,
    - the first column is a subject number between 1,...,n;
    - the second column is time;
    - the third column is the state occupied by subject in column 1 at time indicated in column 2

- **R**
  - the rank

- **eps**
  - the tolerance for convergence; the default is 10^-5

- **ref**
  - a vector of reference categories; the default is NULL and if NULL is used, the function will use the first category as the reference category for each row

**Value**

- a list of outputs:
  - **Alpha**: the final A matrix
  - **Gamma**: the final G matrix
  - **Beta**: the coefficient matrix for variables involved in reduced rank
  - **Dcoe**: the coefficient matrix for the fixed variables
  - **Dsderr**: the standard error matrix for the fixed variables
  - **Dpval**: the p-value matrix for the fixed variables
  - **coemat**: the overall coefficient matrix
  - **niter**: the iteration number to get converged
  - **df**: the degrees of freedom
  - **loglik**: the final loglikelihood
  - **converge**: three possible values with 0 means fail to converge, 1 means converges, and 2 means the maximum iteration is achieved

**Examples**

```r
# generate the Markov chain
U=7
I1=I2=I3=rep(1,7)
I4=c(0,0,0,1,1,1,1)
I5=I6=I7=rep(0,7)
I=rbind(I1,I2,I3,I4,I5,I6,I7)
# prepare the data
data=cogdat
```
n=length(unique(data[,1]))
M=nrow(data)+1
Mc=0
z=matrix(0,n,9)
colnames(z)=colnames(data)[5:13]
T=matrix(0,M,3)
for(i in 1:n){
  subdat=data[which(data[,1]==i),,drop=FALSE]
  z[i,]=subdat[1,5:13]
  mc=nrow(subdat)
  T[(Mc+1):(Mc+mc+1),1]=i
  T[(Mc+1):(Mc+mc+1),2]=0:mc
  T[(Mc+1):(Mc+mc+1),3]=c(subdat[1,3],subdat[,4])
  Mc=Mc+mc+1
}
#z1=z[,c(1:3),drop=FALSE]
# fit the model with rank 1
rrmultinom(I,z1=1,z2=2,T,1,eps=9,ref=c(1,1,1,4))

sdfun

Description

This function is used to get the standard error matrix from bootstrap method. It returns the matrices of standard error and p-value for the coefficient matrix.

Usage

sdfun(I, z1 = NULL, z2 = NULL, T, R, eps = 1e-05, B, tpoint = NULL, ref)

Arguments

I
  a U by U incidence matrix with elements; U is the number of states; I(i,j)=1 if state j can be accessed from state i in one step and 0 otherwise

z1
  a n by p matrix with covariates involved in the dimension reduction (DR), n is the number of subjects, p is the number of covariates involved in DR

z2
  a n by q matrix with study covariates (not in dimension reduction), q is the number of study covariates

T
  a M by 3 state matrix,
  • the first column is a subject number between 1,...,n;
  • the second column is time;
  • the third column is the state occupied by subject in column 1 at time indicated in column 2

R
  the rank
eps            the tolerance for convergence; the default is $10^{-5}$
B              the bootstrap number
Tpoint         a matrix has two columns with the participants’ visit information about timeline
Ref            a vector of reference categories

Value

- coe: the coefficient matrix of the original data
- sd: the standard error matrix
- pvalue: the p-value matrix

Examples

```r
# generate the Markov chain
U=7
I1=I2=I3=rep(1,7)
I4=c(0,0,0,1,1,1,1)
I5=I6=I7=rep(0,7)
I=rbind(I1,I2,I3,I4,I5,I6,I7)
# prepare the data
data=cogdat
n=length(unique(data[,1]))
M=nrow(data)+n
Mc=0
z=matrix(0,n,9)
colnames(z)=colnames(data)[5:13]
T=matrix(0,M,3)
for(i in 1:n){
  subdat=data[which(data[,1]==i),,drop=FALSE]
  z[i,]=subdat[1,5:13]
  mc=nrow(subdat)
  T[(Mc+1):(Mc+mc+1),1]=i
  T[(Mc+1):(Mc+mc+1),2]=0:mc
  T[(Mc+1):(Mc+mc+1),3]=c(subdat[1,3],subdat[,4])
  Mc=Mc+mc+1
}
#z1=z[,c(1:3),drop=FALSE]
z2=z[,4,drop=FALSE]
# find the standard deviation matrix for the model with rank 1
sdfun(I,z1=NULL,z2,T,1,eps = 9.2,ref=c(1,1,1,4))
```
Index

* datasets
  cogdat, 3
Aupdate, 2
cogdat, 3
derivativeB, 3
derivatives, 4
expand, 5
Gupdate, 5
norm, 6
rrmultinom, 6
sdfun, 8