Package 'lnmCluster'

October 13, 2022

Type Package

Title Perform Logistic Normal Multinomial Clustering for Microbiome Compositional Data

Version 0.3.1

Maintainer Wangshu Tu <wangshu.tu@carleton.ca>

Description An implementation of logistic normal multinomial (LNM) clustering. It is an extension of LNM mixture model proposed by Fang and Subedi (2020) <arXiv:2011.06682>, and is designed for clustering compositional data. The package includes 3 extended models: LNM Factor Analyzer (LNM-FA), LNM Bicluster Mixture Model (LNM-BMM) and Penalized LNM Factor Analyzer (LNM-FA). There are several advantages of LNM models: 1. LNM provides more flexible covariance structure; 2. Factor analyzer can reduce the number of parameters to estimate; 3. Bicluster can simultaneously cluster subjects and taxa, and provides significant biological insights; 4. Penalty term allows sparse estimation in the covariance matrix. Details for model assumptions and interpretation can be found in papers: Tu and Subedi (2021) <arXiv:2101.01871> and Tu and Subedi (2022) <doi:10.1002/sam.11555>.

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.1.2

Imports mclust, foreach, MASS, stringr, gtools, pgmm, utils

Suggests knitr, rmarkdown, testthat, mvtnorm

VignetteBuilder knitr

Depends R (>= 3.50)

LinkingTo Rcpp

NeedsCompilation yes

Author Wangshu Tu [aut, cre],
Sanjeena Dang [aut],
Yuan Fang [aut]

Repository CRAN

Date/Publication 2022-07-20 17:50:02 UTC

R topics documented:

`initial_variational_gaussian` *Gives default initial guesses for logistic-normal multinomial biclustering algorithm.*

Description

Gives default initial guesses for logistic-normal multinomial biclustering algorithm.

Usage

```
initial_variational_gaussian(W_count, G, Q_g, cov_str, X)
```

Arguments

`W_count`	The microbiome count matrix that you want to analyze.
`G`	The number of component
`Q_g`	The number of biclusters for each component, a vector.
`cov_str`	The covaraince structure you choose, there are 16 different models belongs to this family:UUU, UUG, UUD, UUC, UGU, UGG, UGD, UGC, GUU, GUG, GUD, GUC, GGU, GGG, GGD, GGC.
`X`	The regression covariates matrix, which generated by model.matrix.

Value

`new_pi_g` Initial guess of proportion

`new_mu_g` Initial guess of mean vector

`new_sig_g` Initial guess of covariance matrix for each component

`new_T_g` Initial guess of covariance of latent variable: u

new_B_g Initial guess of bicluster membership

new_D_g Initial guess of error matrix

new_m Initial guess of variational mean

new_V Initial guess of variational varaince

new_beta_g Initial guess of covariates coefficients.

`initial_variational_lasso` *Gives default initial guesses for penalized logistic-normal multinomial Factor analyzer algorithm.*

Description

Gives default initial guesses for penalized logistic-normal multinomial Factor analyzer algorithm.

Usage

```
initial_variational_lasso(W_count, G, Q_g, cov_str, X)
```

Arguments

W_count	The microbiome count matrix that you want to analyze.
G	The number of component
Q_g	A specific number of latent dimension.
cov_str	The covaraince structure you choose, there are 2 different models belongs to this family:UUU, GUU.
X	The regression covariates matrix, which generated by model.matrix.

Value

new_pi_g Initial guess of proportion

new_mu_g Initial guess of mean vector

new_sig_g Initial guess of covariance matrix for each component

new_B_g Initial guess of loading matrix.

new_T_g The identity matrix of latent variable: u

new_D_g Initial guess of error matrix

new_m Initial guess of variational mean

new_V Initial guess of variational varaince

new_beta_g Initial guess of covariates coefficients.

initial_variational_PGMM *Gives default initial guesses for logistic-normal multinomial Factor analyzer algorithm.*

Description

Gives default initial guesses for logistic-normal multinomial Factor analyzer algorithm.

Usage

```
initial_variational_PGMM(W_count, G, Q_g, cov_str, X)
```

Arguments

W_count	The microbiome count matrix that you want to analyze.
G	The number of component
Q_g	The number of latent dimensions for each component, a vector.
cov_str	The covaraince structure you choose, there are 8 different models belongs to this family:UUU, UUG, UUD, UUC, GUU, GUG, GUD, GUC.
X	The regression covariates matrix, which generated by model.matrix.

Value

new_pi_g Initial guess of proportion

new_mu_g Initial guess of mean vector

new_sig_g Initial guess of covariance matrix for each component

new_B_g Initial guess of loading matrix.

new_T_g The identity matrix of latent variable: u

new_D_g Initial guess of error matrix

new_m Initial guess of variational mean

new_V Initial guess of variational varaince

new_beta_g Initial guess of covariates coefficients.

`lnmbiclust` *Logistic Normal Multinomial Biclustering algorithm*

Description

Main function that can do LNM biclustering and select the best model based on BIC, AIC or ICL.

Usage

```
lnmbiclust(W_count, range_G, range_Q, model, criteria, iter, permutation, X)
```

Arguments

`W_count`	The microbiome count matrix
`range_G`	All possible number of components. A vector.
`range_Q`	All possible number of bicluster for each component. A vector
`model`	The covaraince structure you choose, there are 16 different models belongs to this family:UUU, UUG, UUD, UUC, UGU, UGG, UGD, UGC, GUU, GUG, GUD, GUC, GGU, GGG, GGD, GGC. You can choose more than 1 covarance structure to do model selection.
`criteria`	one of AIC, BIC or ICL. The best model is depends on the criteria you choose. The default is BIC
`iter`	Max iterations, defaul is 150.
`permutation`	Only has effect when model contains UUU, UUG, UUD or UUC. If TRUE, it assume the number of biclusters could be different for different components. If FALSE, it assume the number of biclusters are the same cross all components. Default is FALSE.
`X`	The regression covariate matrix, which is generated by model.matrix.

Value

z_ig Estimated latent variable z

cluster Component labels

mu_g Estimated component mean

pi_g Estimated component proportion

B_g Estimated bicluster membership

T_g Estimated covariance of latent variable u

D_g Estimated error covariance

COV Estimated sparsity component covariance

beta_g Estimated covariate coefficients

sigma Estimated original component covariance

overall_loglik Complete log likelihood value for each iteration

ICL ICL value

BIC BIC value

AIC AIC value

all_fitted_model display all names of fitted models in a data.frame.

Examples

```
#generate toy data with n=100, K=5,
#set up parameters
n<-100
p<-5
mu1<-c(-2.8,-1.3,-1.6,-3.9,-2.6)
B1<-matrix(c(1,0,1,0,1,0,0,1,0,1),nrow = p, byrow=TRUE)
T1<-diag(c(2.9,0.5))
D1<-diag(c(0.52, 1.53, 0.56, 0.19, 1.32))
cov1<-B1%*%T1%*%t(B1)+D1
mu2<-c(1.5,-2.7,-1.1,-0.4,-1.4)
B2<-matrix(c(1,0,1,0,0,1,0,1,0,1),nrow = p, byrow=TRUE)
T2<-diag(c(0.2,0.003))
D2<-diag(c(0.01, 0.62, 0.45, 0.01, 0.37))
cov2<-B2%*%T2%*%t(B2)+D2

#generate normal distribution
library(mvtnorm)
simp<-rmultinom(n,1,c(0.6,0.4))
lab<-as.factor(apply(t(simp),1,which.max))
df<-matrix(0,nrow=n,ncol=p)
for (i in 1:n) {
 if(lab[i]==1){df[i,]<-rmvnorm(1,mu1,sigma = cov1)}
 else if(lab[i]==2){df[i,]<-rmvnorm(1,mu2,sigma = cov2)}
}
#apply inverse of additive log ratio and transform normal to count data
f_df<-cbind(df,0)
z<-exp(f_df)/rowSums(exp(f_df))
W_count<-matrix(0,nrow=n,ncol=p+1)
for (i in 1:n) {
 W_count[i,]<-rmultinom(1,runif(1,10000,20000),z[i,])
}

#'#if run one model let range_Q be an integer
res<-lnmbiclust(W_count,2,2,model="UUU")

#following will run 2 combinations of Q: 2 2, and 3 3 with G=2.
res<-lnmbiclust(W_count,2,range_Q=c(2:3),model="UUU")

#if run model selection let range_Q and range_G be a vector.
#model selection for all 16 models with G=1 to 3, Q=1 to 3.
res<-lnmbiclust(W_count,c(1:3),c(1:3))
```

`lnmfa` *Logistic Normal Multinomial factor analyzer algorithm*

Description

Main function that can do LNM factor analyzer and select the best model based on BIC, AIC or ICL.

Usage

```
lnmfa(W_count, range_G, range_Q, model, criteria, iter, X)
```

Arguments

`W_count`	The microbiome count matrix
`range_G`	All possible number of components. A vector.
`range_Q`	All possible number of bicluster for each component. A vector
`model`	The covaraince structure you choose, there are 8 different models belongs to this family:UUU, UUG, UUD, UUC, GUU, GUG, GUD, GUC. You can choose more than 1 covarance structure to do model selection.
`criteria`	one of AIC, BIC or ICL. The best model is depends on the criteria you choose. The default is BIC
`iter`	Max iterations, defaul is 150.
`X`	The regression covariate matrix, which is generated by model.matrix.

Value

z_ig Estimated latent variable z

cluster Component labels

mu_g Estimated component mean

pi_g Estimated component proportion

B_g Estimated bicluster membership

D_g Estimated error covariance

COV Estimated component covariance

beta_g Estimated covariate coefficients

overall_loglik Complete log likelihood value for each iteration

ICL ICL value

BIC BIC value

AIC AIC value

all_fitted_model display all names of fitted models in a data.frame.

Examples

```
#generate toy data with n=100, K=5,
#set up parameters
n<-100
p<-5
mu1<-c(-2.8,-1.3,-1.6,-3.9,-2.6)
B1<-matrix(c(1,0,1,0,1,0,0,1,0,1),nrow = p, byrow=TRUE)
T1<-diag(c(2.9,0.5))
D1<-diag(c(0.52, 1.53, 0.56, 0.19, 1.32))
cov1<-B1%*%T1%*%t(B1)+D1
mu2<-c(1.5,-2.7,-1.1,-0.4,-1.4)
B2<-matrix(c(1,0,1,0,0,1,0,1,0,1),nrow = p, byrow=TRUE)
T2<-diag(c(0.2,0.003))
D2<-diag(c(0.01, 0.62, 0.45, 0.01, 0.37))
cov2<-B2%*%T2%*%t(B2)+D2

#generate normal distribution
library(mvtnorm)
simp<-rmultinom(n,1,c(0.6,0.4))
lab<-as.factor(apply(t(simp),1,which.max))
df<-matrix(0,nrow=n,ncol=p)
for (i in 1:n) {
 if(lab[i]==1){df[i,]<-rmvnorm(1,mu1,sigma = cov1)}
 else if(lab[i]==2){df[i,]<-rmvnorm(1,mu2,sigma = cov2)}
}
#apply inverse of additive log ratio and transform normal to count data
f_df<-cbind(df,0)
z<-exp(f_df)/rowSums(exp(f_df))
W_count<-matrix(0,nrow=n,ncol=p+1)
for (i in 1:n) {
 W_count[i,]<-rmultinom(1,runif(1,10000,20000),z[i,])
}

#'#if run one model let range_Q be an integer
res<-lnmfa(W_count,2,2,model="UUU")

#following will run 2 combinations of Q: 2 2, and 3 3 with G=2.
res<-lnmfa(W_count,2,range_Q=c(2:3),model="UUU")

#if run model selection let range_Q and range_G be a vector.
#model selection for all 16 models with G=1 to 3, Q=1 to 3.
res<-lnmfa(W_count,c(1:3),c(1:3))
```

`Mico_bi_jensens` *run main microbiome bicluster algorithm.*

Description

run main microbiome bicluster algorithm.

Usage

```
Mico_bi_jensens(
  W_count,
  G,
  Q_g,
  pi_g,
  mu_g,
  sig_g,
  V,
  m,
  B_g,
  T_g,
  D_g,
  cov_str,
  iter,
  const,
  beta_g,
  X
)
```

Arguments

W_count	The microbiome count matrix that you want to analyze.
G	The number of component
Q_g	The number of biclusters for each component, a vector.
pi_g	A vector of initial guesses of component proportion
mu_g	A list of initial guess of mean vector
sig_g	A list of initial guess of covariance matrix for each component
V	A list of initial guess of variational varaince
m	A list of initial guess of variational mean
B_g	A list of initial guess of bicluster membership
T_g	A list of initial guess of covariance of latent variable: u
D_g	A list of initial guess of error matrix
cov_str	The covaraince structure you choose, there are 16 different models belongs to this family:UUU, UUG, UUD, UUC, UGU, UGG, UGD, UGC, GUU, GUG, GUD, GUC, GGU, GGG, GGD, GGC.
iter	Max iterations, default is 150.
const	the permutation constant in multinomial distribution. Calculated before the main algorithm in order to save computation time.
beta_g	initial guess of covariates coefficients.
X	The regression covariates matrix, which generates by model.matrix.

Value

z_ig Estimated latent variable z

cluster Component labels

mu_g Estimated component mean

pi_g Estimated component proportion

B_g Estimated bicluster membership

T_g Estimated covariance of latent variable u

D_g Estimated error covariance

COV Estimated sparsity component covariance

beta_g Estimated covariates coefficients.

sigma Estimated original component covariance

overall_loglik Complete log likelihood value for each iteration

ICL ICL value

BIC BIC value

AIC AIC value

`Mico_bi_lasso` *Penalized Logistic Normal Multinomial factor analyzer main estimation process*

Description

Main function will perform PLNM factor analyzer and return parameters

Usage

```
Mico_bi_lasso(
  W_count,
  G,
  Q_g,
  pi_g,
  mu_g,
  sig_g,
  V,
  m,
  B_K,
  T_K,
  D_K,
  cov_str,
  tuning,
  iter,
  const,
```

```
  beta_g,
  X
)
```

Arguments

W_count	The microbiome count matrix
G	All possible number of components. A vector.
Q_g	A specific number of latent dimension.
pi_g	A vector of initial guesses of component proportion
mu_g	A list of initial guess of mean vector
sig_g	A list of initial guess of covariance matrix for each component
V	A list of initial guess of variational varaince
m	A list of initial guess of variational mean
B_K	A list of initial guess of loading matrix.
T_K	A list of identity matrix with dimension q.
D_K	A list of initial guess of error matrix
cov_str	The covaraince structure you choose, there are 2 different models belongs to this family:UUU and GUU. You can choose more than 1 covarance structure to do model selection.
tuning	length G vector with range 0-1, define the tuning parameter for each component
iter	Max iterations, default is 150.
const	the permutation constant in multinomial distribution. Calculated before the main algorithm in order to save computation time.
beta_g	initial guess of covariates coefficients.
X	The regression covariates matrix, which generates by model.matrix.

Value

z_ig Estimated latent variable z

cluster Component labels

mu_g Estimated component mean

pi_g Estimated component proportion

B_g Estimated sparsity loading matrix

D_g Estimated error covariance

COV Estimated component covariance

beta_g Estimated covariates coefficients.

overall_loglik Complete log likelihood value for each iteration

ICL ICL value

BIC BIC value

AIC AIC value

tuning display the tuning parameter you specified.

Mico_bi_PGMM *run main microbiome Factor Analyzer algorithm.*

Description

run main microbiome Factor Analyzer algorithm.

Usage

```
Mico_bi_PGMM(
  W_count,
  G,
  Q_g,
  pi_g,
  mu_g,
  sig_g,
  V,
  m,
  B_K,
  T_K,
  D_K,
  cov_str,
  iter,
  const,
  beta_g,
  X
)
```

Arguments

W_count	The microbiome count matrix that you want to analyze.
G	The number of component
Q_g	The number of latent dimensions for each component, a vector.
pi_g	A vector of initial guesses of component proportion
mu_g	A list of initial guess of mean vector
sig_g	A list of initial guess of covariance matrix for each component
V	A list of initial guess of variational varaince
m	A list of initial guess of variational mean
B_K	A list of initial guess of loading matrix.
T_K	A list of identity matrix with dimension q.
D_K	A list of initial guess of error matrix
cov_str	The covaraince structure you choose, there are 8 different models belongs to this family:UUU, UUG, UUD, UUC, GUU, GUG, GUD, GUC.

iter	Max iterations, default is 150.
const	the permutation constant in multinomial distribution. Calculated before the main algorithm in order to save computation time.
beta_g	initial guess of covariates coefficients.
X	The regression covariates matrix, which generates by model.matrix.

Value

z_ig Estimated latent variable z

cluster Component labels

mu_g Estimated component mean

pi_g Estimated component proportion

B_g Estimated loading matix.

D_g Estimated error covariance

COV Estimated component covariance

beta_g Estimated covariates coefficients.

overall_loglik Complete log likelihood value for each iteration

ICL ICL value

BIC BIC value

AIC AIC value

---

model_selection *Model selections for* lnmbicluster

Description

fit several models for lnmbicluster along with 3 criteria values: AIC BIC and ICL

Usage

```
model_selection(W_count, range_G, range_Q, model, permutation, iter, const, X)
```

Arguments

W_count	The microbiome count matrix that you want to analyze.
range_G	All possible number of component groups, a vector.
range_Q	All possible number of bicluster groups Q, a vector.
model	A vector of string that contain cov_str you want to select. Default is all 16 models.
permutation	Only has effect when model contains UUU, UUG, UUD or UUC. If TRUE, it assume the number of biclusters could be different for different components. If FALSE, it assume the number of biclusters are the same cross all components.

iter	Max iterations, defaul is 150.
const	Constant permutation term in multinomial distribution.
X	The regression covariates matrix, which generates from model.matrix.

Value

A dataframe that contain the cov_str, K, Q, AIC, BIC, ICL values for model. There may be a lot rows if large K and Q, because of lots of combinations: it is a sum of a geometric series with multiplier max(Q) from 1 to max(K).

`model_selection_lasso` *Model selections for* `plnmfa`

Description

fit several models for plnmfa along with 3 criteria values: AIC BIC and ICL

Usage

```
model_selection_lasso(W_count, K, Q_K, model, range_tuning, iter, const, X)
```

Arguments

W_count	The microbiome count matrix that you want to analyze.
K	A specific number of component
Q_K	A specific number of latent dimension.
model	A specific model name, UUU or GUU
range_tuning	A range of tuning parameters specified, ranged from 0-1.
iter	Max iterations, defaul is 150.
const	Constant permutation term in multinomial distribution.
X	The regression covariates matrix, which generates from model.matrix.

Value

A dataframe that contain the cov_str, K, Q, AIC, BIC, ICL values for model. There may be a lot rows if long range of tuning parameters.

`model_selection_PGMM` *Model selections for* `lnmfa`

Description

fit several models for lnmfa along with 3 criteria values: AIC BIC and ICL

Usage

```
model_selection_PGMM(
  W_count,
  range_G,
  range_Q,
  model,
  permutation,
  iter,
  const,
  X
)
```

Arguments

`W_count`	The microbiome count matrix that you want to analyze.
`range_G`	All possible number of component groups, a vector.
`range_Q`	All possible number of bicluster groups Q, a vector.
`model`	A vector of string that contain cov_str you want to select. Default is all 8 models.
`permutation`	Only has effect when model contains UUU, UUG, UUD or UUC. If TRUE, it assume the number of latent dimension could be different for different components. If FALSE, it assume the number of latent dimension are the same cross all components.
`iter`	Max iterations, defaul is 150.
`const`	Constant permutation term in multinomial distribution.
`X`	The regression covariates matrix, which generates from model.matrix.

Value

A dataframe that contain the cov_str, K, Q, AIC, BIC, ICL values for model. There may be a lot rows if large K and Q, because of lots of combinations: it is a sum of a geometric series with multiplier max(Q) from 1 to max(K).

plnmfa *Penalized Logistic Normal Multinomial factor analyzer algorithm*

Description

Main function that can do PLNM factor analyzer and select the best model based on BIC, AIC or ICL.

Usage

```
plnmfa(W_count, range_G, range_Q, model, criteria, range_tuning, iter, X)
```

Arguments

W_count	The microbiome count matrix
range_G	All possible number of components. A vector.
range_Q	A specific number of latent dimension.
model	The covaraince structure you choose, there are 2 different models belongs to this family:UUU and GUU. You can choose more than 1 covarance structure to do model selection.
criteria	one of AIC, BIC or ICL. The best model is depends on the criteria you choose. The default is BIC
range_tuning	A range of tuning parameters specified, ranged from 0-1.
iter	Max iterations, default is 150.
X	The regression covariate matrix, which is generated by model.matrix.

Value

z_ig Estimated latent variable z

cluster Component labels

mu_g Estimated component mean

pi_g Estimated component proportion

B_g Estimated bicluster membership

D_g Estimated error covariance

COV Estimated component covariance

beta_g Estimated covariate coefficients

overall_loglik Complete log likelihood value for each iteration

ICL ICL value

BIC BIC value

AIC AIC value

all_fitted_model display all names of fitted models in a data.frame.

Examples

```
#'#generate toy data with n=100, K=5,
#set up parameters
n<-100
p<-5
mu1<-c(-2.8,-1.3,-1.6,-3.9,-2.6)
B1<-matrix(c(1,0,1,0,1,0,0,1,0,1),nrow = p, byrow=TRUE)
T1<-diag(c(2.9,0.5))
D1<-diag(c(0.52, 1.53, 0.56, 0.19, 1.32))
cov1<-B1%*%T1%*%t(B1)+D1
mu2<-c(1.5,-2.7,-1.1,-0.4,-1.4)
B2<-matrix(c(1,0,1,0,0,1,0,1,0,1),nrow = p, byrow=TRUE)
T2<-diag(c(0.2,0.003))
D2<-diag(c(0.01, 0.62, 0.45, 0.01, 0.37))
cov2<-B2%*%T2%*%t(B2)+D2

#generate normal distribution
library(mvtnorm)
simp<-rmultinom(n,1,c(0.6,0.4))
lab<-as.factor(apply(t(simp),1,which.max))
df<-matrix(0,nrow=n,ncol=p)
for (i in 1:n) {
 if(lab[i]==1){df[i,]<-rmvnorm(1,mu1,sigma = cov1)}
 else if(lab[i]==2){df[i,]<-rmvnorm(1,mu2,sigma = cov2)}
}
#apply inverse of additive log ratio and transform normal to count data
f_df<-cbind(df,0)
z<-exp(f_df)/rowSums(exp(f_df))
W_count<-matrix(0,nrow=n,ncol=p+1)
for (i in 1:n) {
 W_count[i,]<-rmultinom(1,runif(1,10000,20000),z[i,])
}

#if run one model let range_G, and range_tuning be an integer
#remember you can always overspecify Q, so we don't suggest to run models with a range of Q.
res<-plnmfa(W_count,2,2,model="UUU",range_tuning=0.6)

#if run model selection let any \code{range_} parameters be a vector.
res<-plnmfa(W_count,c(2:3),3,range_tuning=seq(0.5,0.8,by=0.1))
```

Index