Package 'MixLFA'

October 17, 2024

Type Package

Title Mixture of Longitudinal Factor Analysis Methods

Version 1.0.0

Maintainer Amine Ounajim <amine.ounajim@chu-poitiers.fr>

Description Provides a function for the estimation of mixture of longitudinal factor analysis models using the iterative expectation-maximization algorithm (Ounajim, Slaoui, Louis, Billot, Frasca, Rigoard (2023) <doi:10.1002/sim.9804>) and several tools for visualizing and interpreting the models' parameters.

License GPL-3

Imports Rcpp (>= 1.0.0), ggplot2, pheatmap, GGally, dplyr, GPArotation

LinkingTo Rcpp, RcppEigen

Suggests testthat, rmarkdown

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation yes

Depends R (>= 3.5.0)

Author Amine Ounajim [aut, cre],
   Yousri Slaoui [aut],
   Omar Lahbabi [aut],
   Philippe Rigoard [ctb]

Repository CRAN

Date/Publication 2024-10-17 16:00:05 UTC

Contents

Index **12**

Fixed_coef   *Extract Fixed Coefficients from MLFA Results*

Description

This function extracts the fixed effect coefficients β from the results obtained from a Mixture of Longitudinal Factor Analyzers (MLFA) model for a specified class and factor.

Usage

```
Fixed_coef(res_MLFA, C, d)
```

Arguments

res_MLFA	list containing the MLFA model parameters returned by the MLFA function.
C	an integer giving the number of mixture components.
d	an integer giving the factor index from which to extract the coefficients. This corresponds to the specific latent factor of interest.

Details

The function first determines the number of predictor variables (p) by evaluating the number of columns in the predictor matrix X that was used in the MLFA. It then extracts the relevant coefficients from the estimated fixed effects β vector associated with the specified class C and factor d. The β vector is structured such that coefficients for each factor are stored in contiguous blocks; this function selects the appropriate block corresponding to the factor d within the class C.

Value

A numeric vector of length ncol(X) (number of fixed covariates in the MLFA model) containing the coefficients for the specified class and factor.

Examples

```
# Load the necessary datasets
data(simulated_MLFA)  # Load a simulated dataset based on the MLFA model
# Extract matrices from the list
# Extract matrix Y of outcomes of interest for the factor analysis model
Y <- simulated_MLFA$Y
# Extract matrix X of fixed effect covariates for describing the latent factors
X <- simulated_MLFA$X
# Extract matrix Z of random effect covariates for describing the latent factors
Z <- simulated_MLFA$Z
# Extract matrix id containing subject identifiers.
```

```
id <-simulated_MLFA$id
#' # Run the MLFA (Mixture of Longitudinal Factor Analyzers) function with:
# C: number of classes or clusters in our simulated data was set to 2.
# d: number of latent factors in our simulated data was set to 1.
# max_it: maximum number of iterations is set to 50 for a quick test.
# Estimation of the parameters of the MLFA model using the simulated data.
result_MLFA <- MLFA(C = 2, d = 2, X, Y, Z, id, max_it = 50, fixed_factor =  c(1,6))
# Extract the fixed effect coefficients for the latent factor 1 in cluster 1
coef_vector <- Fixed_coef(result_MLFA, C=1, d=1)
```

---

generate_heatmap   *Generate Heatmap of an MLFA factor loadings*

Description

This function generates a heatmap for visualizing the factor loadings from the MLFA model results.

Usage

```
generate_heatmap(res_MLFA, C)
```

Arguments

res_MLFA   a list containing the MLFA model parameters returned by the MLFA function.

C   an integer Class to display

Value

No return value. This function generates a heatmap plot of the factor loadings matrix.

Examples

```
# Load the necessary datasets
data(simulated_MLFA)  # Load a simulated dataset based on the MLFA model
# Extract matrices from the list
# Extract matrix Y of outcomes of interest for the factor analysis model
Y <- simulated_MLFA$Y
# Extract matrix X of fixed effect covariates for describing the latent factors
X <- simulated_MLFA$X
# Extract matrix Z of random effect covariates for describing the latent factors
Z <- simulated_MLFA$Z
# Extract matrix id containing subject identifiers.
id <-simulated_MLFA$id
#' # Run the MLFA (Mixture of Longitudinal Factor Analyzers) function with:
# C: number of classes or clusters in our simulated data was set to 2.
# d: number of latent factors in our simulated data was set to 1.
# max_it: maximum number of iterations is set to 50 for a quick test.
# Estimation of the parameters of the MLFA model using the simulated data.
```

```
result_MLFA <- MLFA(C = 2, d = 2, X, Y, Z, id, max_it = 50, fixed_factor =  c(1,6))
# Generate a heatmap of the factor loadings of the first cluster
generate_heatmap(result_MLFA,C=1)
```

MLFA   *Estimates a Mixture of Longitudinal Factor Analyzers (MLFA) model*

Description

This function performs a mixture of longitudinal factor analyzers on multivariate longitudinal data as described by Ounajim et al (2023). The MLFA model is written a two equations representing a measurement factor model describing the link between the outcomes / indicators of interest Y and one or several latent factors eta, and a structural mixed effect model describing the link between the latent factors and a set of explanatory variables in the design matrices X and Z for fixed and random effects respectively:

$$y_{itj} = \sum_{c=1}^{C} 1\!\!1_{\{v_i=c\}} \left(\Lambda_{jc}\eta_{i.tc} + \epsilon_{itjc} \right),$$

$$\eta_{iktc} = X_{iktc}\beta_{kc} + Z_{iktc}b_{ikc} + e_{itc},$$

where i is the subject index, t is the measurement time index, j is the outcome index and k is latent factor index. The model parameters are estimated using the Expectation-Maximization (EM) algorithm.

Usage

```
MLFA(C, d, X, Y, Z, id, max_it, fixed_factor = 1:d, seed = 1, scale = TRUE)
```

Arguments

C	an integer giving the number of mixture components.
d	an integer giving the number of latent factors in the factor analysis model.
X	a matrix containing the design matrix for fixed effects covariates for the mixed effect model explaining the latent factors (an unit column can be included to estimate an intercept).
Y	a matrix containing the observed outcomes / indicators of interest for the factor model.
Z	a matrix containing the design matrix for random effects covariates for the mixed effect model explaining the latent factors.
id	a vector containing subject identifiers.
max_it	an integer giving the maximum number of iterations for the expectation-maximization algorithm for parameter estimation. The algorithm might stop before max_it if the mean absolute difference between two successive iterations is smaller than $10^{(-5)}$
fixed_factor	a vector of integers of length d containing the columns in Y with factor loadings fixed to 1.
seed	a seed value for the random initialization of parameters.
scale	an optional Boolean indicating whether the matrix Y needs to be scaled or not.

Value

a list with the following components:

Lam a list of length C containing the estimated factor loading matrix for each cluster.

beta a list of length C containing the estimated fixed effect coefficients numeric vector of length p*d where p is the number of fixed effect and d is the number of latent factors. The function fixed_coef can be used to extract the fixed effect coefficient for a given cluster and for a given latent factor.

S_b a list of length C containing the estimated covariance matrices of the random effects.

S_e a list of length C containing the estimated covariance matrices of the error term in the mixed effect model (both intra and inter-factor covariances).

tau a list of length C containing the vector of estimated variances of the error terms ϵ_{itjc} in the factor 'measurement' model.

pro a numeric vector containing the estimated proportion of each cluster.

BIC a numeric value of the Bayesian Information Criterion for model selection for either the number of clusters or the number of latent factors.

AIC a numeric value of the Akaike Information Criterion for model selection.

ICL a numeric value of the Integrated Completed Likelihood for model selection.

ri a matrix containig the the probability of class membership for each subject.

VerifNan a Boolean indicating whether the model generated Nan values or not.

References

Ounajim, A., Slaoui, Y., Louis, P. Y., Billot, M., Frasca, D., & Rigoard, P. (2023). Mixture of longitudinal factor analyzers and their application to the assessment of chronic pain. Statistics in medicine, 42(18), 3259–3282. https://doi.org/10.1002/sim.9804

Examples

```
# Load the necessary datasets
data(simulated_MLFA)  # Load a simulated dataset based on the MLFA model
# Extract matrices from the list
# Extract matrix Y of outcomes of interest for the factor analysis model
Y <- simulated_MLFA$Y
# Extract matrix X of fixed effect covariates for describing the latent factors
X <- simulated_MLFA$X
# Extract matrix Z of random effect covariates for describing the latent factors
Z <- simulated_MLFA$Z
# Extract matrix id containing subject identifiers.
id <-simulated_MLFA$id
#' # Run the MLFA (Mixture of Longitudinal Factor Analyzers) function with:
# C: number of classes or clusters in our simulated data was set to 2.
# d: number of latent factors in our simulated data was set to 1.
# max_it: maximum number of iterations is set to 50 for a quick test.
# Estimation of the parameters of the MLFA model using the simulated data.
result_MLFA <- MLFA(C = 2, d = 2, X, Y, Z, id, max_it = 50, fixed_factor =  c(1,6))
# Print the resulting factor loading matrices and fixed effect coefficients
```

```
print(result_MLFA$Lam)
print(result_MLFA$beta)
```

Oblimin_Rotation   *Apply Oblimin Rotation to MLFA Factor Loadings*

Description

This function applies an oblimin rotation (from the package GPArotation) <doi:10.1177/0013164404272507>
to the factor loadings from the results. of a Mixture of Longitudinal Factor Analyzers (MLFA)
model. The oblimin default parameters are used.

Usage

```
Oblimin_Rotation(res_MLFA)
```

Arguments

res_MLFA   a list containing the MLFA model parameters returned by the MLFA function.

Value

A list similar to 'res_MLFA', but with the factor loadings rotated using the oblimin method, which
is a method for obtaining oblique rotations used to transform factor loading vectors or matrices to
simple structure (i.e. a pattern of factor loadings such that each variable loads highly onto one and
only one factor).

Examples

```
# Load the necessary datasets
data(simulated_MLFA)  # Load a simulated dataset based on the MLFA model
# Extract matrices from the list
# Extract matrix Y of outcomes of interest for the factor analysis model
Y <- simulated_MLFA$Y
# Extract matrix X of fixed effect covariates for describing the latent factors
X <- simulated_MLFA$X
# Extract matrix Z of random effect covariates for describing the latent factors
Z <- simulated_MLFA$Z
# Extract matrix id containing subject identifiers.
id <-simulated_MLFA$id
#' # Run the MLFA (Mixture of Longitudinal Factor Analyzers) function with:
# C: number of classes or clusters in our simulated data was set to 2.
# d: number of latent factors in our simulated data was set to 1.
# max_it: maximum number of iterations is set to 50 for a quick test.
# Estimation of the parameters of the MLFA model using the simulated data.
result_MLFA <- MLFA(C = 2, d = 2, X, Y, Z, id, max_it = 50, fixed_factor =  c(1,6))
# Apply the oblimin rotation to the factor loadings
rotated_result <- Oblimin_Rotation(result_MLFA)
```

---

plot_BIC   *Plot BIC Values Across Iterations*

Description

This function plots the Bayesian Information Criterion (BIC) values across iterations for convergence evaluation.

Usage

```
plot_BIC(res_MLFA)
```

Arguments

res_MLFA   a list containing the MLFA model parameters returned by the MLFA function.

Value

No return value. This function generates a plot representing BIC values (y-axis) for different iterations (x-axis).

Examples

```
# Load the necessary datasets
data(simulated_MLFA)  # Load a simulated dataset based on the MLFA model
# Extract matrices from the list
# Extract matrix Y of outcomes of interest for the factor analysis model
Y <- simulated_MLFA$Y
# Extract matrix X of fixed effect covariates for describing the latent factors
X <- simulated_MLFA$X
# Extract matrix Z of random effect covariates for describing the latent factors
Z <- simulated_MLFA$Z
# Extract matrix id containing subject identifiers.
id <-simulated_MLFA$id
#' # Run the MLFA (Mixture of Longitudinal Factor Analyzers) function with:
# C: number of classes or clusters in our simulated data was set to 2.
# d: number of latent factors in our simulated data was set to 1.
# max_it: maximum number of iterations is set to 50 for a quick test.
# Estimation of the parameters of the MLFA model using the simulated data.
result_MLFA <- MLFA(C = 2, d = 2, X, Y, Z, id, max_it = 50, fixed_factor =  c(1,6))
# plot the BIC from iteration 1 to iteration \eqn{max_it}.
plot_BIC(result_MLFA)
```

Predict	*Predict the latent factors using the fixed effect coefficients from MLFA Results*

Description

This function calculates the predicted latent factor conditional expectations given the covariates X for new observations. The predictions are obtained using the estimated fixed effect coefficients of the mixed effect model in the MLFA model.

Usage

```
Predict(res_MLFA, C, d, X = res_MLFA$X)
```

Arguments

res_MLFA   a list containing the MLFA model parameters returned by the MLFA function.

C   an integer giving the number of mixture components.

d   an integer giving the number of latent factors in the factor analysis model.

X   a matrix containing the design matrix for fixed effects covariates for the mixed effect model explaining the latent factors (an unit column can be included to estimate an intercept). Default is the matrix used to estimate the model.

Details

The function extracts the relevant coefficients from the β matrix for the specified class C and factor d. These coefficients are then multiplied by the predictor matrix X to compute the predicted values. The function does not perform any plotting; it simply returns the estimated conditional expectation of the latent factor d.

Value

A matrix containing the predicted latent factor scores given the fixed effects and a cluster indicator.

Examples

```
# Load the necessary datasets
data(simulated_MLFA)  # Load a simulated dataset based on the MLFA model
# Extract matrices from the list
# Extract matrix Y of outcomes of interest for the factor analysis model
Y <- simulated_MLFA$Y
# Extract matrix X of fixed effect covariates for describing the latent factors
X <- simulated_MLFA$X
# Extract matrix Z of random effect covariates for describing the latent factors
Z <- simulated_MLFA$Z
# Extract matrix id containing subject identifiers.
id <-simulated_MLFA$id
#' # Run the MLFA (Mixture of Longitudinal Factor Analyzers) function with:
```

```
# C: number of classes or clusters in our simulated data was set to 2.
# d: number of latent factors in our simulated data was set to 1.
# max_it: maximum number of iterations is set to 50 for a quick test.
# Estimation of the parameters of the MLFA model using the simulated data.
result_MLFA <- MLFA(C = 2, d = 2, X, Y, Z, id, max_it = 50, fixed_factor =  c(1,6))
# predict the latent factor scores from the resulting MLFA model.
Predict(result_MLFA, 1, 1)
```

---

simulated_MLFA *simulated_MLFA: Simulated data from the MLFA model*

Description

This dataset contains a list of matrices, each with a specific purpose or structure. The list includes four matrices: Y, X, Z, and id. simulated using the following parameters (described in the simulation study in (Ounajim et al., 2023)):

Usage

```
data(simulated_MLFA)
```

Format

A list with four elements:

Y the observed outcomes matrix with 10 columns and 500 rows (100 subjects with 5 observations each).

X the fixed effects design matrix with two columns (two explanatory variables for explaining the factor loadings variation) and 500 rows.

Z the random effect design matrix similar to X.

id a vector of length 500 containing subject identifiers

References

Ounajim, A., Slaoui, Y., Louis, P. Y., Billot, M., Frasca, D., & Rigoard, P. (2023). Mixture of longitudinal factor analyzers and their application to the assessment of chronic pain. Statistics in medicine, 42(18), 3259–3282. https://doi.org/10.1002/sim.9804

Examples

```
# Load the necessary datasets
data(simulated_MLFA)  # Load a simulated dataset based on the MLFA model
# Extract matrices from the list
# Extract matrix Y of outcomes of interest for the factor analysis model
Y <- simulated_MLFA$Y
# Extract matrix X of fixed effect covariates for describing the latent factors
X <- simulated_MLFA$X
# Extract matrix Z of random effect covariates for describing the latent factors
```

```
Z <- simulated_MLFA$Z
# Extract matrix id containing subject identifiers.
id <-simulated_MLFA$id
#' # Run the MLFA (Mixture of Longitudinal Factor Analyzers) function with:
# C: number of classes or clusters in our simulated data was set to 2.
# d: number of latent factors in our simulated data was set to 1.
# max_it: maximum number of iterations is set to 50 for a quick test.
# Estimation of the parameters of the MLFA model using the simulated data.
result_MLFA <- MLFA(C = 2, d = 2, X, Y, Z, id, max_it = 50, fixed_factor =  c(1,6))
```

---

stdUnique *Generate Standardized Uniqueness from MLFA Results*

Description

This function generates uniqueness plots (proportion of variance in the outcome variables in Y that is not explained by the factor analysis model) based on the estimated error variance.

Usage

```
stdUnique(res_MLFA, C)
```

Arguments

res_MLFA   a list containing the MLFA model parameters returned by the MLFA function.

C   an integer giving the number of mixture components.

Value

No return value. This function generates a plot of the uniqueness of the factor analysis model for a given cluster C (variance that is 'unique' to the outcome variable and not explained by the common factors).

Examples

```
# Load the necessary datasets
data(simulated_MLFA)  # Load a simulated dataset based on the MLFA model
# Extract matrices from the list
# Extract matrix Y of outcomes of interest for the factor analysis model
Y <- simulated_MLFA$Y
# Extract matrix X of fixed effect covariates for describing the latent factors
X <- simulated_MLFA$X
# Extract matrix Z of random effect covariates for describing the latent factors
Z <- simulated_MLFA$Z
# Extract matrix id containing subject identifiers.
id <-simulated_MLFA$id
#' # Run the MLFA (Mixture of Longitudinal Factor Analyzers) function with:
# C: number of classes or clusters in our simulated data was set to 2.
# d: number of latent factors in our simulated data was set to 1.
```

```
# max_it: maximum number of iterations is set to 50 for a quick test.
# Estimation of the parameters of the MLFA model using the simulated data.
result_MLFA <- MLFA(C = 2, d = 2, X, Y, Z, id, max_it = 50, fixed_factor =  c(1,6))
# Generate the uniqueness plots for the first cluster
stdUnique(result_MLFA, C=1)
```

Index