Package: xLLiM (via r-universe)

August 22, 2024

Type Package

Title High Dimensional Locally-Linear Mapping

Version 2.3

Author Emeline Perthame (emeline.perthame@inria.fr), Florence Forbes (florence.forbes@inria.fr), Antoine Deleforge (antoine.deleforge@inria.fr), Emilie Devijver (emilie.devijver@kuleuven.be), Melina Gallopin (melina.gallopin@u-psud.fr)

Maintainer Emeline Perthame <emeline.perthame@pasteur.fr>

Description Provides a tool for non linear mapping (non linear regression) using a mixture of regression model and an inverse regression strategy. The methods include the GLLiM model (see Deleforge et al (2015) <DOI:10.1007/s11222-014-9461-5>) based on Gaussian mixtures and a robust version of GLLiM, named SLLiM (see Perthame et al (2016) <DOI:10.1016/j.jmva.2017.09.009>) based on a mixture of Generalized Student distributions. The methods also include BLLiM (see Devijver et al (2017) <arXiv:1701.07899>) which is an extension of GLLiM with a sparse block diagonal structure for large covariance matrices (particularly interesting for transcriptomic data).

License GPL (>= 2)

Imports MASS,abind,corpcor,Matrix,igraph,capushe,glmnet,randomForest,e1071,mda,progress,mixOmics

Suggests shock

biocViews mixOmics

NeedsCompilation no

Repository https://epertham.r-universe.dev

RemoteUrl https://github.com/epertham/xllim

RemoteRef HEAD

RemoteSha dba703acda2955c9462c0afab8b2b9e6ce45827f

Contents

`xLLiM-package` *High Dimensional Locally-Linear Mapping*

Description

Provides a tool for non linear mapping (non linear regression) using a mixture of regression model and an inverse regression strategy. The methods include the GLLiM model (see Deleforge et al (2015) <DOI:10.1007/s11222-014-9461-5>) based on Gaussian mixtures and a robust version of GLLiM, named SLLiM (see Perthame et al (2016) <https://hal.archives-ouvertes.fr/hal-01347455>) based on a mixture of Generalized Student distributions. The methods also include BLLiM (see Devijver et al (2017) <https://arxiv.org/abs/1701.07899>) which is an extension of GLLiM with a sparse block diagonal structure for large covariance matrices (particularly interesting for transcriptomic data).

Details

Package:	xLLiM
Type:	Package
Version:	2.1
Date:	2017-05-23
License:	GPL (>= 2)

The methods implemented in this package adress the following non-linear mapping issue:

$$E(Y|X = x) = g(x),$$

where Y is a L-vector of multivariate responses and X is a large D-vector of covariates' profiles such that $D \gg L$. The methods implemented in this package aims at estimating the non linear regression function g.

First, the methods of this package are based on an inverse regression strategy. The inverse conditional relation $p(X|Y)$ is specified in a way that the forward relation of interest $p(Y|X)$ can be

deduced in closed-from. The large number D of covariates is handled by this inverse regression trick, which acts as a dimension reduction technique. The number of parameters to estimate is therefore drastically reduced.

Second, we propose to approximate the non linear g regression function by a piecewise affine function. Therefore, a hidden discrete variable Z is introduced, in order to separate the space in K regressions such that an affine model holds in each region k between responses Y and variables X:

$$X = \sum_{k=1}^{K} I_{Z=k}(A_k Y + b_k + E_k)$$

where A_k is a $D \times L$ matrix of coeffcients for regression k, b_k is a D-vector of intercepts and E_k is a random noise.

All the models implemented in this package are based on mixture of regression models. The components of the mixture are Gaussian for GLLiM. SLLiM is a robust extension of GLLiM, based on Generalized Student mixtures. Indeed, Generalized Student distributions are heavy-tailed distributions which improve the robustness of the model compared to their Gaussian counterparts. BLLiM is an extension of GLLiM designed to provide an interpretable prediction tool for the analysis of transcriptomic data. It assumes a block diagonal dependence structure between covariates (genes) conditionally to the response. The block structure is automatically chosen among a collection of models using the slope heuristics.

For both GLLiM and SLLiM, this package provides the possibility to add L_w latent variables, when the responses are partially observed. In this situation, the vector $Y = (T, W)$ is split into an observed L_t-vector T and an unobserved L_w-vector W. The total size of the response is therefore $L = L_t + L_w$ where L_w is chosen by the user. See [1] for details but this amounts to consider factors and allows to add structure in the large dimensional covariance matrices. The user must choose the number of mixtures components K and, if needed, the number of latent factors L_w. For small datasets (less than 100 observations), we suggest to select both (K, L_w) by minimizing the BIC criterion. For larger datasets, we suggest to set L_w using BIC while setting K to an arbitrary value large enough to catch non linear relations between responses and covariates and small enough to have several observations (at least 10) in each clusters. Indeed, for large datasets, the number of clusters should not have a strong impact on the results provided it is sufficiently large.

We propose to assess the prediction accuracy of a new response x_{test} by computing the NRMSE (Normalized Root Mean Square Error) which is the RMSE normalized by the RMSE of prediction by the mean of training responses:

$$NRMSE = \frac{||\hat{y} - x_{test}||_2}{||\bar{y} - x_{test}||_2}$$

where $\hat{y}$ is the predicted response, x_{test} is the true testing response and $\bar{y}$ is the mean of training responses.

The functions available in this package are used in this order:

- Step 1 (optional): Initialization of the algorithm using a Multivariate Gaussian mixture model and an EM algorithm implemented in the `emgm` function. Responses and covariates must be concatenated as described in the documentation of `emgm` which corresponds to a joint Gaussian Mixture Model (see Qiao et al, 2009).
- Step 2: Estimation of a regression model using one of the available models (`gllim`, `sllim` or `bllim`). User must specify the following arguments

  - for GLLiM or SLLiM: constraint on the large covariance matrices of covariates named Σ_k. These matrices can be supposed diagonal and homoskedastic (isotropic) by setting `cstr=list(Sigma="i")` which is the default. Other constraints are diagonal and heteroskedastic (`Sigma="d"`), full matrix (`Sigma=""`) or full but equal for each class (`Sigma="*"`). Except for the last constraint, in all previous constraints the matrices have their own parameterization.
  - number of components K in the model.
  - for GLLiM or SLLiM: if needed, number of latent factors L_w
- Step 3: Prediction of responses for a testing dataset using the `gllim_inverse_map` or `sllim_inverse_map` functions.

Author(s)

Emeline Perthame (emeline.perthame@inria.fr), Florence Forbes (florence.forbes@inria.fr), Antoine Deleforge (antoine.deleforge@inria.fr)

References

[1] A. Deleforge, F. Forbes, and R. Horaud. High-dimensional regression with Gaussian mixtures and partially-latent response variables. Statistics and Computing, 25(5):893–911, 2015.

[2] E. Devijver, M. Gallopin, E. Perthame. Nonlinear network-based quantitative trait prediction from transcriptomic data. Submitted, 2017, available at https://arxiv.org/abs/1701.07899.

[3] E. Perthame, F. Forbes, and A. Deleforge. Inverse regression approach to robust nonlinear high-to-low dimensional mapping. Journal of Multivariate Analysis, 163(C):1–14, 2018. https://doi.org/10.1016/j.jmva.2017.09.0

[4] X. Qiao and N. Minematsu. Mixture of probabilistic linear regressions: A unified view of GMM-based mapping techiques. IEEE International Conference on Acoustics, Speech, and Signal Processing, 2009.

The `gllim` and `gllim_inverse_map` functions have been converted to R from the original Matlab code of the GLLiM toolbox available on: https://team.inria.fr/perception/gllim_toolbox/

See Also

shock-package,capushe-package

Examples

```
### Not run

## Example of inverse regression with GLLiM model
# data(data.xllim)
# dim(data.xllim) #  size 52 y 100
# responses = data.xllim[1:2,] # 2 responses in rows and 100 observations in columns
# covariates = data.xllim[3:52,] # 50 covariates in rows and 100 observations in columns

## Set 5 components in the model
# K = 5

## Step 1: initialization of the posterior probabilities (class assignments)
## via standard EM for a joint Gaussian Mixture Model
```

```
# r = emgm(rbind(responses, covariates), init=K);

## Step 2: estimation of the model
## Default Lw=0 and cstr$Sigma="i"
# mod = gllim(responses,covariates,in_K=K,in_r=r)

## Skip Step 1 and go to Step 2: automatic initialization and estimation of the model
# mod = gllim(responses,covariates,in_K=K)

## Alternative: Add Lw=1 latent factor to the model
# mod = gllim(responses,covariates,in_K=K,in_r=r,Lw=1)

## Different constraints on the large covariance matrices can be added:
## see details in the documentation of the GLLiM function
## description
# mod = gllim(responses,covariates,in_K=K,in_r=r,cstr=list(Sigma="i")) #default
# mod = gllim(responses,covariates,in_K=K,in_r=r,cstr=list(Sigma="d"))
# mod = gllim(responses,covariates,in_K=K,in_r=r,cstr=list(Sigma=""))
# mod = gllim(responses,covariates,in_K=K,in_r=r,cstr=list(Sigma="*"))
## End of example of inverse regression with GLLiM model

## Step 3: Prediction on a test dataset
# data(data.xllim.test) size 50 y 20
# pred = gllim_inverse_map(data.xllim.test,mod)
## Predicted responses using the mean of \eqn{p(y | x)}.
# pred$x_exp

## Example of leave-ntest-out (1 fold cross-validation) procedure
# n = ncol(covariates)
# ntest=10
# id.test = sample(1:n,ntest)
# train.responses = responses[,-id.test]
# train.covariates = covariates[,-id.test]
# test.responses = responses[,id.test]
# test.covariates = covariates[,id.test]

## Learn the model on training data
# mod = gllim(train.responses, train.covariates,in_K=K)

## Predict responses on testing data
# pred = gllim_inverse_map(test.covariates,mod)$x_exp

## nrmse : normalized root mean square error to measure prediction performance
## the normalization term is the rmse of the prediction by the mean of training responses
## an nrmse larger than 1 means that the procedure performs worse than prediction by the mean
# norm_term = sqrt(rowMeans(sweep(test.responses,1,rowMeans(train.responses),"-")^2))
## Returns 1 value for each response variable
# nrmse = sqrt(rowMeans((test.responses-pred)^2))/norm_term
```

---

bllim *EM Algorithm for Block diagonal Gaussian Locally Linear Mapping*

Description

EM Algorithm for Block diagonal Gaussian Locally Linear Mapping

Usage

```
bllim(tapp,yapp,in_K,in_r=NULL,ninit=20,maxiter=100,verb=0,in_theta=NULL,plot=TRUE)
```

Arguments

tapp	An L x N matrix of training responses with variables in rows and subjects in columns
yapp	An D x N matrix of training covariates with variables in rows and subjects in columns
in_K	Initial number of components or number of clusters
in_r	Initial assignments (default NULL). If NULL, the model is initialized with the best initialisation among 20, computed by a joint Gaussian mixture model on both response and covariates.
ninit	Number of random initializations. Not used of in_r is specified. Default is 20 and the random initialization which maximizes the likelihood is retained.
maxiter	Maximum number of iterations (default 100). The algorithm stops if the number of iterations exceeds maxiter or if the difference of likelihood between two iterations is smaller than a threshold fixed to $0.001(max(LL) - min(LL))$ where LL is the vector of log-likelihoods at the successive iterations.
verb	Verbosity: print out the progression of the algorithm. If verb=0, there is no print, if verb=1, the progression is printed out. Default is 0.
in_theta	Initial parameters (default NULL), same structure as the output of this function. The EM algorithm can be initialized either with initial assignments or initial parameters values.
plot	Displays plots to allow user to check that the slope heuristics can be applied confidently to select the conditional block structure of predictors, as in the capushe-package package. Default is TRUE.

Details

The BLLiM model implemented in this function adresses the following non-linear mapping issue:

$$E(Y|X = x) = g(x),$$

where Y is a L-vector of multivariate responses and X is a large D-vector of covariates' profiles such that $D \gg L$. As gllim and sllim, the bllim function aims at estimating the non linear regression function g.

First, the methods of this package are based on an inverse regression strategy. The inverse conditional relation $p(X|Y)$ is specified in a way that the forward relation of interest $p(Y|X)$ can be deduced in closed-from. Under some hypothesis on covariance structures, the large number D of covariates is handled by this inverse regression trick, which acts as a dimension reduction technique. The number of parameters to estimate is therefore drastically reduced. Second, we propose

to approximate the non linear g regression function by a piecewise affine function. Therefore, a hidden discrete variable Z is introduced, in order to divide the space into K regions such that an affine model holds between responses Y and variables X in each region k:

$$X = \sum_{k=1}^{K} I_{Z=k}(A_k Y + b_k + E_k)$$

where A_k is a $D \times L$ matrix of coeffcients for regression k, b_k is a D-vector of intercepts and E_k is a Gaussian noise with covariance matrix Σ_k.

BLLiM is defined as the following hierarchical Gaussian mixture model for the inverse conditional density $(X|Y)$:

$$p(X|Y = y, Z = k; \theta) = N(X; A_k x + b_k, \Sigma_k)$$
$$p(Y|Z = k; \theta) = N(Y; c_k, \Gamma_k)$$
$$p(Z = k) = \pi_k$$

where Σ_k is a $D \times D$ block diagonal covariance structure automatically learnt from data. θ is the set of parameters $\theta = (\pi_k, c_k, \Gamma_k, A_k, b_k, \Sigma_k)_{k=1}^K$. The forward conditional density of interest $p(Y|X)$ is deduced from these equations and is also a Gaussian mixture of regression model.

For a given number of affine components (or clusters) K and a given block structure, the number of parameters to estimate is:

$$(K-1) + K(DL + D + L + nbpar_\Sigma + L(L+1)/2)$$

where L is the dimension of the response, D is the dimension of covariates and $nbpar_\Sigma$ is the total number of parameters in the large covariance matrix Σ_k in each cluster. This number of parameters depends on the number and size of blocks in each matrices.

Two hyperparameters must be estimated to run BLLiM:

- Number of mixtures components (or clusters) K: we propose to use BIC criterion or slope heuristics as implemented in capushe-package
- For a given number of clusters K, the block structure of large covariance matrices specific of each cluster: the size and the number of blocks of each Σ_k matrix is automatically learnt from data, using an extension of the shock procedure (see shock-package). This procedure is based on a successive thresholding of sample conditional covariance matrix within clusters, building a collection of block structure candidates. The final block structure is retained using slope heuristics.

BLLiM is not only a prediction model but also an interpretable tool. For example, it is useful for the analysis of transcriptomic data. Indeed, if covariates are genes and response is a phenotype, the model provides clusters of individuals based on the relation between gene expression data and the phenotype, and also leads to infer a gene regulatory network specific for each cluster of individuals.

Value

Returns a list with the following elements:

`LLf`	Final log-likelihood
`LL`	Log-likelihood value at each iteration of the EM algorithm

pi	A vector of length K of mixture weights i.e. prior probabilities for each component
c	An (L x K) matrix of means of responses (Y)
Gamma	An (L x L x K) array of K matrices of covariances of responses (Y)
A	An (D x L x K) array of K matrices of linear transformation matrices
b	An (D x K) matrix in which affine transformation vectors are in columns
Sigma	An (D x D x K) array of covariances of X
r	An (N x K) matrix of posterior probabilities
nbpar	The number of parameters estimated in the model

Author(s)

Emeline Perthame (emeline.perthame@pasteur.fr), Emilie Devijver (emilie.devijver@kuleuven.be), Melina Gallopin (melina.gallopin@u-psud.fr)

References

[1] E. Devijver, M. Gallopin, E. Perthame. Nonlinear network-based quantitative trait prediction from transcriptomic data. Submitted, 2017, available at https://arxiv.org/abs/1701.07899.

See Also

xLLiM-package, emgm, gllim_inverse_map,capushe-package,shock-package

Examples

```
data(data.xllim)

## Setting 5 components in the model
K = 5

## the model can be initialized by running an EM algorithm for Gaussian Mixtures (EMGM)
r = emgm(data.xllim, init=K);
## and then the gllim model is estimated
responses = data.xllim[1:2,] # 2 responses in rows and 100 observations in columns
covariates = data.xllim[3:52,] # 50 covariates in rows and 100 observations in columns

## if initialization is not specified, the model is automatically initialized by EMGM
# mod = bllim(responses,covariates,in_K=K)

## Prediction can be performed using prediction function gllim_inverse_map
# pred = gllim_inverse_map(covariates,mod)$x_exp
```

`data.xllim` *Simulated data to run examples of usage of* `gllim` *and* `sllim` *functions*

Description

Matrix of simulated data, generated under a GLLiM model, with K=5 clusters from the true parameters available in object `data.xllim.trueparameters`. The goal is to learn the non linear relation between the responses (Y) and the covariates (X) using `gllim`, `bllim` or `sllim`. Details are given hereafter.

Usage

```
data(data.xllim)
```

Format

A matrix of simulated data with 52 rows and 100 columns (observations). The first 2 rows are responses (Y) and the last 50 rows are covariates (X). The goal is to retrieve Y from X using `gllim` or `sllim`.

Details

This dataset is generated under a GLLiM model with L=2, D=50 and N=100.

First, the responses Y are generated according to a Gaussian Mixture model with K=5 clusters:

$$p(Y = y|Z = k) = N(y; c_k, \Gamma_k)$$

where each $(c_k)_{k=1}^K$ is a L-vector randomly sampled from a standardized Gaussian, $(\Gamma_k)_{k=1}^K$ are LxL random correlation matrix and Z is a multinomial hidden variable which indicates the cluster membership of each observation:

$$p(Z = k) = \pi_k$$

where the probabilities $(\pi_k)_{k=1}^K$ are sampled from a standard uniform distribution and normalized to sum to 1.

Then, the covariates X are generated according to a Gaussian Mixture of regressions. It is recalled that GLLiM models the following inverse relation, which is used to generate X:

$$X = \sum_{k=1}^{K=5} I_{Z=k}(A_k X + b_k + E_k)$$

where Y is the vector of L responses and X is the vector of D covariates and Z is the hidden variable of cluster membership introduced above. Regression coefficients A_k and intercepts b_k are sampled from a standard Gaussian and the covariance matrix of the noise $\Sigma_k = Var(E_k)$ is the identity.

The goal is to retrieve Y from X using `gllim`, `bllim` or `sllim`.

See Also

xLLiM-package, gllim, sllim, data.xllim.test

Examples

```
data(data.xllim)
dim(data.xllim) # 52 100
Y = data.xllim[1:2,] # responses # 2 100
X = data.xllim[3:52,] # covariates # 50 100
```

---

data.xllim.test *Testing data to run examples of usage of* gllim_inverse_map *and* sllim_inverse_map *functions*

Description

data.xllim.test is a matrix of simulated testing data, generated under the same GLLiM model as data.xllim, from the true parameters available in object data.xllim.trueparameters. The goal is to train a GLLiM (resp. SLLiM and BLLiM) model on training data (see data.xllim) and to retrieve the unknown responses from data.xllim.test using gllim_inverse_map (resp. sllim_inverse_map).

Usage

```
data(data.xllim.test)
```

Format

A matrix of simulated testing data with 50 rows (covariates) and 20 columns (observations).

See Also

xLLiM-package, data.xllim, gllim_inverse_map, sllim_inverse_map

Examples

```
data(data.xllim.test)
dim(data.xllim.test) # 50 20
```

data.xllim.trueparameters *True parameters used to simulate the datasets* data.xllim *and* data.xllim.test

Description

data.xllim.trueparameters is a list containing the true parameters of the GLLiM model used to generate the datasets data.xllim and data.xllim.test. We set the number of covariates to D=50, number of responses to L=2 and we simulated a GLLiM model with K=5 components.

Usage

```
data(data.xllim.trueparameters)
```

Format

A list with the following elements

- pi A vector of length K of mixture weights i.e. prior probabilities for each component
- c An (L x K) matrix of means of responses (X)
- Gamma An (L x L x K) array of K matrices of covariances of responses (X)
- A An (D x L x K) array of K matrices of linear transformation matrices
- b An (D x K) matrix in which affine transformation vectors are in columns
- Sigma An (D x D x K) array of covariances of Y

data.xllim.trueparameters has the same that the values returned by gllim function.

See Also

xLLiM-package, data.xllim, gllim_inverse_map, sllim_inverse_map

Examples

```
data(data.xllim.trueparameters)
## data.xllim.trueparameters$pi # A vector with K=5 elements
## data.xllim.trueparameters$c # A matrix with dimension L=2 x K=5
## data.xllim.trueparameters$Gamma # An array with dimension L=2 x L=2 x K=5
## data.xllim.trueparameters$A # An array with dimension D=50 x L=2 x K=5
## data.xllim.trueparameters$b # A matrix with dimension D=50 x K=5
## data.xllim.trueparameters$Sigma # An array with dimension D=50 x D=50 x K=5
```

emgm *Perform EM algorithm for fitting a Gaussian mixture model (GMM)*

Description

Perform EM algorithm for fitting a Gaussian mixture model (GMM). In the GLLiM context, this is done jointly on both responses and covariates

Usage

```
emgm(X, init, maxiter,verb)
```

Arguments

X	An (M x N) matrix with variables in rows and observations in columns. M is D+L in the proposed approach
init	This argument can be a number K of classes (integer), a matrix of posterior probabilities ((N x K) matrix) or a matrix of centers ((M x K) matrix)
maxiter	Maximum number of iterations for estimation of the GMM
verb	Print out the progression of the algorithm. If verb=0, there is no print, if verb=1, the progression is printed out. Default is 0.

Value

Returns a list with the following elements:

label	An N vector of class assignments provided by maximum a posteriori (MAP) on posterior probabilities to belong to each of the K components for each observation
model	A list with the estimated parameters of the GMM
model$mu	An (M x K) matrix of estimations of means in each cluster of the joint GMM
model$Sigma	An (M x M x K) array of estimations of covariance matrix in each cluster of the GMM
model$weight	An K vector of estimated prior probabilities of each cluster
llh	A vector of values of the log-likelihood for each iteration of the algorithm
R	An N x K matrix of estimations of posterior probabilities to belong to each of the K components for each observation

Author(s)

Emeline Perthame (emeline.perthame@inria.fr), Florence Forbes (florence.forbes@inria.fr), Antoine Deleforge (antoine.deleforge@inria.fr)

References

[1] A. Deleforge, F. Forbes, and R. Horaud. High-dimensional regression with Gaussian mixtures and partially-latent response variables. Statistics and Computing,25(5):893–911, 2015.

[2] E. Perthame, F. Forbes, and A. Deleforge. Inverse regression approach to robust nonlinear high-to-low dimensional mapping. Journal of Multivariate Analysis, 163(C):1–14, 2018. https://doi.org/10.1016/j.jmva.2017.09.0

[3] Y. Qiao and N. Minematsu. Mixture of probabilistic linear regressions: A unified view of GMM-based mapping techiques. IEEE International Conference on Acoustics, Speech, and Signal Processing, 2009.

Converted to R from the Matlab code of the GLLiM toolbox available on: https://team.inria.fr/perception/gllim_toolbox/

See Also

xLLiM-package, gllim, sllim

Examples

```
# data(data.xllim)
# K=5
# r = emgm(data.xllim, init=K, verb=0);
# r$R # estimation of posterior probabilities to belong to
## each of the K components for each observation
```

`gllim` *EM Algorithm for Gaussian Locally Linear Mapping*

Description

EM Algorithm for Gaussian Locally Linear Mapping

Usage

```
gllim(tapp,yapp,in_K,in_r=NULL,maxiter=100,Lw=0,cstr=NULL,verb=0,in_theta=NULL,...)
```

Arguments

`tapp`	An `Lt` x `N` matrix of training responses with variables in rows and subjects in columns
`yapp`	An `D` x `N` matrix of training covariates with variables in rows and subjects in columns
`in_K`	Initial number of components
`in_r`	Initial assignments (default NULL)
`maxiter`	Maximum number of iterations (default 100). The algorithm stops if the number of iterations exceeds `maxiter` or if the difference of likelihood between two iterations is smaller than a threshold fixed to $0.001(max(LL) - min(LL))$ where LL is the vector of log-likelihoods at the successive iterations.

`Lw`	Number of hidden components (default 0)
`cstr`	Constraints on error covariance matrices. Must be a list as following `cstr=list(Sigma="i")` constraints Σ_k to be diagonal and isotropic, which is the default. See details section hereafter to see the other available options to constraint the covariance matrices.
`verb`	Verbosity: print out the progression of the algorithm. If `verb=0`, there is no print, if `verb=1`, the progression is printed out. Default is 0.
`in_theta`	The EM algorithm can be initialized either with initial assignments or initial parameters values. In that case, the initial parameters (default NULL) must have the same structure as the output `theta` of this function.
`...`	other arguments to be passed for internal use only

Details

The GLLiM model implemented in this function adresses the following non-linear mapping issue:

$$E(Y|X = x) = g(x),$$

where Y is a L-vector of multivariate responses and X is a large D-vector of covariates' profiles such that $D \gg L$. The methods implemented in this package aims at estimating the non linear regression function g.

First, the methods of this package are based on an inverse regression strategy. The inverse conditional relation $p(X|Y)$ is specified in a way that the forward relation of interest $p(Y|X)$ can be deduced in closed-from. Under some hypothesis on covariance structures, the large number D of covariates is handled by this inverse regression trick, which acts as a dimension reduction technique. The number of parameters to estimate is therefore drastically reduced. Second, we propose to approximate the non linear g regression function by a piecewise affine function. Therefore, a hidden discrete variable Z is introduced, in order to divide the space into K regions such that an affine model holds between responses Y and variables X in each region k:

$$X = \sum_{k=1}^{K} I_{Z=k}(A_k Y + b_k + E_k)$$

where A_k is a $D \times L$ matrix of coeffcients for regression k, b_k is a D-vector of intercepts and E_k is a Gaussian noise with covariance matrix Σ_k.

GLLiM is defined as the following hierarchical Gaussian mixture model for the inverse conditional density $(X|Y)$:

$$p(X|Y = y, Z = k; \theta) = N(X; A_k x + b_k, \Sigma_k)$$
$$p(Y|Z = k; \theta) = N(Y; c_k, \Gamma_k)$$
$$p(Z = k) = \pi_k$$

where θ is the set of parameters $\theta = (\pi_k, c_k, \Gamma_k, A_k, b_k, \Sigma_k)_{k=1}^K$. The forward conditional density of interest $p(Y|X)$ is deduced from these equations and is also a Gaussian mixture of regression model.

`gllim` allows the addition of L_w latent variables in order to account for correlation among covariates or if it is supposed that responses are only partially observed. Adding latent factors is known to

improve prediction accuracy, if L_w is not too large with regard to the number of covariates. When latent factors are added, the dimension of the response is $L = L_t + L_w$ and $L = L_t$ otherwise.

For GLLiM, the number of parameters to estimate is:

$$(K-1) + K(DL + D + L_t + nbpar_\Sigma + nbpar_\Gamma)$$

where $L = L_w + L_t$ and $nbpar_\Sigma$ (resp. $nbpar_\Gamma$) is the number of parameters in each of the large (resp. small) covariance matrix Σ_k (resp. Γ_k). For example,

- if the constraint on Σ is `cstr$Sigma="i"`, then $nbpar_\Sigma = 1$,which is the default constraint in the `gllim` function
- if the constraint on Σ is `cstr$Sigma="d"`, then $nbpar_\Sigma = D$,
- if the constraint on Σ is `cstr$Sigma=""`, then $nbpar_\Sigma = D(D+1)/2$,
- if the constraint on Σ is `cstr$Sigma="*"`, then $nbpar_\Sigma = D(D+1)/(2K)$.

The rule to compute the number of parameters of Γ is the same as Σ, replacing D by L_t. Currently the Γ_k matrices are not constrained and $nbpar_\Gamma = L_t(L_t+1)/2$ because for indentifiability reasons the L_w part is set to the identity matrix.

The user must choose the number of mixtures components K and, if needed, the number of latent factors L_w. For small datasets (less than 100 observations), it is suggested to select both (K, L_w) by minimizing the BIC criterion. For larger datasets, it is suggested to save computational time, to set L_w using BIC while setting K to an arbitrary value large enough to catch non linear relations between responses and covariates and small enough to have several observations (at least 10) in each clusters. Indeed, for large datasets, the number of clusters should not have a strong impact on the results while it is sufficiently large.

Value

Returns a list with the following elements:

`LLf`	Final log-likelihood
`LL`	Log-likelihood value at each iteration of the EM algorithm
`pi`	A vector of length `K` of mixture weights i.e. prior probabilities for each component
`c`	An (`L` x `K`) matrix of means of responses (Y) where `L=Lt+Lw`
`Gamma`	An (`L` x `L` x `K`) array of `K` matrices of covariances of responses (Y) where `L=Lt+Lw`
`A`	An (`D` x `L` x `K`) array of `K` matrices of linear transformation matrices where `L=Lt+Lw`
`b`	An (`D` x `K`) matrix in which affine transformation vectors are in columns
`Sigma`	An (`D` x `D` x `K`) array of covariances of X
`r`	An (`N` x `K`) matrix of posterior probabilities
`nbpar`	The number of parameters estimated in the model

Author(s)

Emeline Perthame (emeline.perthame@inria.fr), Florence Forbes (florence.forbes@inria.fr), Antoine Deleforge (antoine.deleforge@inria.fr)

References

[1] A. Deleforge, F. Forbes, and R. Horaud. High-dimensional regression with Gaussian mixtures and partially-latent response variables. Statistics and Computing, 25(5):893–911, 2015.

[2] E. Perthame, F. Forbes, and A. Deleforge. Inverse regression approach to robust nonlinear high-to-low dimensional mapping. Journal of Multivariate Analysis, 163(C):1–14, 2018. https://doi.org/10.1016/j.jmva.2017.09.0(

Converted to R from the Matlab code of the GLLiM toolbox available on: https://team.inria.fr/perception/gllim_toolbox/

See Also

xLLiM-package, emgm, gllim_inverse_map, sllim

Examples

```
data(data.xllim)

## Setting 5 components in the model
K =5

## the model can be initialized by running an EM algorithm for Gaussian Mixtures (EMGM)
r = emgm(data.xllim, init=K);
## and then the gllim model is estimated
responses = data.xllim[1:2,] # 2 responses in rows and 100 observations in columns
covariates = data.xllim[3:52,] # 50 covariates in rows and 100 observations in columns
mod = gllim(responses,covariates,in_K=K,in_r=r);

## if initialization is not specified, the model is automatically initialized by EMGM
## mod = gllim(responses,covariates,in_K=K)

## Adding 1 latent factor
## mod = gllim(responses,covariates,in_K=K,in_r=r,Lw=1)

## Some constraints on the covariance structure of \eqn{X} can be added
## mod = gllim(responses,covariates,in_K=K,in_r=r,cstr=list(Sigma="i"))
# Isotropic covariances
# (same variance among covariates but different in each component)

## mod = gllim(responses,covariates,in_K=K,in_r=r,cstr=list(Sigma="d"))
# Heteroskedastic covariances
# (variances are different among covariates and in each component)

## mod = gllim(responses,covariates,in_K=K,in_r=r,cstr=list(Sigma=""))
# Unconstrained full matrix

## mod = gllim(responses,covariates,in_K=K,in_r=r,cstr=list(Sigma="*"))
# Full matrix but equal between components
```

gllim_inverse_map *Inverse Mapping from gllim or bllim parameters*

Description

This function computes the prediction of a new response from the estimation of the GLLiM model, returned by the function `gllim`. Given an observed X, the prediction of the corresponding Y is obtained by setting Y to the mean of the distribution $p(Y|X)$.

Usage

```
gllim_inverse_map(y,theta,verb=0)
```

Arguments

`y`	An `D x N` matrix of input observations with variables in rows and subjects on columns
`theta`	An object returned by the `gllim` function corresponding to the learned GLLiM model
`verb`	Verbosity: print out the progression of the algorithm. If `verb=0`, there is no print, if `verb=1`, the progression is printed out. Default is 0.

Details

This function computes the prediction of a new response from the estimation of GLLiM or a BLLiM model, returned by functions `gllim` and `bllim`. Indeed, if the inverse conditional density $p(X|Y)$ and the marginal density $p(Y)$ are defined according to a GLLiM model (or BLLiM) (as described on `xLLiM-package` and `gllim`), the forward conditional density $p(Y|X)$ can be deduced.

Under GLLiM and BLLiM model, it is recalled that the inverse conditional $p(X|Y)$ is a mixture of Gaussian regressions with parameters $(\pi_k, c_k, \Gamma_k, A_k, b_k, \Sigma_k)_{k=1}^K$. Interestingly, the forward conditional $p(Y|X)$ is also a mixture of Gaussian regressions with parameters $(\pi_k, c_k^*, \Gamma_k^*, A_k^*, b_k^*, \Sigma_k^*)_{k=1}^K$. These parameters have a closed-form expression depending only on $(\pi_k, c_k, \Gamma_k, A_k, b_k, \Sigma_k)_{k=1}^K$.

Finally, the forward density (of interest) has the following expression:

$$p(Y|X=x) = \sum_{k=1}^{K} \frac{\pi_k N(x; c_k^*, \Gamma_k^*)}{\sum_j \pi_j N(x; c_j^*, \Gamma_j^*)} N(y; A_k^* x + b_k^*, \Sigma_k^*)$$

and a prediction of a new vector of responses is computed as:

$$E(Y|X=x) = \sum_{k=1}^{K} \frac{\pi_k N(x; c_k^*, \Gamma_k^*)}{\sum_j \pi_j N(x; c_j^*, \Gamma_j^*)} (A_k^* x + b_k^*)$$

where x is a new vector of observed covariates.

When applied on a BLLiM model (returned by function `bllim`), the prediction function `gllim_inverse_map` accounts for the block structure of covariance matrices of the model.

Value

Returns a list with the following elements:

x_exp	An L x N matrix of predicted responses by posterior mean. If L_w latent factors are added to the model, the first Lt rows $(1 : Lt)$ are predictions of responses and rows $(L_t+1) : L$ (recall that $L = L_t + L_w$) are estimations of latent factors.
alpha	Weights of the posterior Gaussian mixture model

Author(s)

Emeline Perthame (emeline.perthame@inria.fr), Florence Forbes (florence.forbes@inria.fr), Antoine Deleforge (antoine.deleforge@inria.fr)

References

[1] A. Deleforge, F. Forbes, and R. Horaud. High-dimensional regression with Gaussian mixtures and partially-latent response variables. Statistics and Computing, 25(5):893–911, 2015.

[2] E. Devijver, M. Gallopin, E. Perthame. Nonlinear network-based quantitative trait prediction from transcriptomic data. Submitted, 2017, available at https://arxiv.org/abs/1701.07899.

[3] E. Perthame, F. Forbes, and A. Deleforge. Inverse regression approach to robust nonlinear high-to-low dimensional mapping. Journal of Multivariate Analysis, 163(C):1–14, 2018. https://doi.org/10.1016/j.jmva.2017.09.0(

Converted to R from the Matlab code of the GLLiM toolbox available on: https://team.inria.fr/perception/gllim_toolbox/

See Also

xLLiM-package,gllim

Examples

```
data(data.xllim)

## Setting 5 components in the model
K = 5

## the model can be initialized by running an EM algorithm for Gaussian Mixtures (EMGM)
r = emgm(data.xllim, init=K);
## and then the gllim model is estimated
responses = data.xllim[1:2,] # 2 responses in rows and 100 observations in columns
covariates = data.xllim[3:52,] # 50 covariates in rows and 100 observations in columns
mod = gllim(responses,covariates,in_K=K,in_r=r);

## Charge testing data
data(data.xllim.test)
## Prediction on a test dataset
pred = gllim_inverse_map(data.xllim.test,mod)
## Predicted responses
print(pred$x_exp)

## Can also be applied on an object returned by bllim function
```

```
## Learn the BLLiM model
# mod = bllim(responses,covariates,in_K=K,in_r=r);
## Prediction on a test dataset
# pred = gllim_inverse_map(data.xllim.test,mod)
## Predicted responses
# print(pred$x_exp)
```

`preprocess_data` *A proposition of function to process high dimensional data before running gllim, sllim or bllim*

Description

The goal of `preprocess_data()` is to get relevant clusters for G-, S-, or BLLiM initialization, coupled with a feature selection for high-dimensional datasets. This function is an alternative to the default initialization implemented in `gllim()`, `sllim()` and `bllim()`.

In this function, clusters are initialized with K-means, and variable selection is performed with a LASSO (`glmnet`) within each clusters. Then selected features are merged to get a subset variables before running any prediction method of xLLiM.

Usage

```
preprocess_data(tapp,yapp,in_K,...)
```

Arguments

`tapp`	An `L` x `N` matrix of training responses with variables in rows and subjects in columns
`yapp`	An `D` x `N` matrix of training covariates with variables in rows and subjects in columns
`in_K`	Initial number of components or number of clusters
`...`	Other arguments of glmnet can be passed

Value

`selected.variables`
: Vector of the indexes of selected variables. Selection is made within clusters and merged hereafter.

`clusters`
: Initialization clusters with k-means

Author(s)

Emeline Perthame (emeline.perthame@pasteur.fr), Emilie Devijver (emilie.devijver@kuleuven.be), Melina Gallopin (melina.gallopin@u-psud.fr)

References

[1] E. Devijver, M. Gallopin, E. Perthame. Nonlinear network-based quantitative trait prediction from transcriptomic data. Submitted, 2017, available at https://arxiv.org/abs/1701.07899.

See Also

xLLiM-package, glmnet-package, kmeans

Examples

```
x <- 1
```

---

sllim *EM Algorithm for Student Locally Linear Mapping*

Description

EM Algorithm for Student Locally Linear Mapping

Usage

```
sllim(tapp,yapp,in_K,in_r=NULL,maxiter=100,Lw=0,cstr=NULL,verb=0,in_theta=NULL,
 in_phi=NULL)
```

Arguments

tapp	An Lt x N matrix of training responses with variables in rows and subjects in columns
yapp	An D x N matrix of training covariates with variables in rows and subjects in columns
in_K	Initial number of components
in_r	Initial assignments (default NULL)
maxiter	Maximum number of iterations (default 100). The algorithm stops if the number of iterations exceeds maxiter or if the difference of likelihood between two iterations is smaller than a threshold (fixed to $0.001(max(LL) - min(LL))$ where LL is the vector of successive log-likelihood values at each iteration).
Lw	Number of hidden components (default 0)
cstr	Constraints on X covariance matrices. Must be a list as following cstr=list(Sigma="i") constraints Σ to be diagonal and isotropic, which is the default. See details section hereafter to see the other available options to constraint the covariance matrix.
verb	Verbosity: print out the progression of the algorithm. If verb=0, there is no print, if verb=1, the progression is printed out. Default is 0.
in_theta	Initial parameters (default NULL), same structure as the output of this function
in_phi	Initial parameters (default NULL), same structure as the output of this function

Details

This function implements the robust counterpart of GLLiM model and should be applied when outliers are present in the data.

The SLLiM model implemented in this function addresses the following non-linear mapping issue:

$$E(Y|X = x) = g(x),$$

where Y is a L-vector of multivariate responses and X is a large D-vector of covariates' profiles such that $D \gg L$. The methods implemented in this package aims at estimating the non linear regression function g.

First, the methods of this package are based on an inverse regression strategy. The inverse conditional relation $p(X|Y)$ is specified in a way that the forward relation of interest $p(Y|X)$ can be deduced in closed-from. Under some hypothesis on covariance structures, the large number D of covariates is handled by this inverse regression trick, which acts as a dimension reduction technique. The number of parameters to estimate is therefore drastically reduced. Second, we propose to approximate the non linear g regression function by a piecewise affine function. Therefore, an hidden discrete variable Z is introduced, in order to divide the space in K regions such that an affine model holds between responses Y and variables X, in each region k:

$$X = \sum_{k=1}^{K} I_{Z=k}(A_k Y + b_k + E_k)$$

where A_k is a $D \times L$ matrix of coefficients for regression k, b_k is a D-vector of intercepts and E_k is a noise with covariance matrix proportional to Σ_k.

SLLiM is defined as the following hierarchical generalized Student mixture model for the inverse conditional density $p(X|Y)$:

$$p(X = x|Y = y, Z = k; \theta, \phi) = S(x; A_k x + b_k, \Sigma_k, \alpha_k^x, \gamma_k^x)$$

$$p(Y = y|Z = k; \theta, \phi) = S(y; c_k, \Gamma_k, \alpha_k, 1)$$

$$p(Z = k|\phi) = \pi_k$$

where (θ, ϕ) are the sets of parameters $\theta = (c_k, \Gamma_k, A_k, b_k, \Sigma_k)_{k=1}^K$ and $\phi = (\pi_k, \alpha_k)_{k=1}^K$. In the previous expression, α_k and (α_k^x, γ_k^x) determine the heaviness of the tail of the generalized Student distribution, which gives robustness to the model. Note that $\alpha_k^x = \alpha_k + L/2$ and $\gamma_k^x = 1+1/2\delta(y, c_k, \Gamma_k)$ where δ is the Mahalanobis distance. The forward conditional density of interest can be deduced from these equations and is also a Student mixture of regressions model.

Like `gllim`, `sllim` allows the addition of latent variables in order to account for correlation among covariates or if it is supposed that responses are only partially observed. Adding latent factors is known to improve prediction accuracy, if `Lw` is not too large with regard to the number of covariates. When latent factors are added, the dimension of the response is `L=Lt+Lw` and `L=Lt` otherwise.

For SLLiM, the number of parameters to estimate is:

$$(K - 1) + K(1 + DL + D + L_t + nbpar_\Sigma + nbpar_\Gamma)$$

where $L = L_w + L_t$ and $nbpar_\Sigma$ (resp. $nbpar_\Gamma$) is the number of parameters in each of the large (resp. small) covariance matrix Σ_k (resp. Γ_k). For example,

- if the constraint on Σ_k is `cstr$Sigma="i"`, then $nbpar_\Sigma = 1$,which is the default constraint in the `gllim` function
- if the constraint on Σ_k is `cstr$Sigma="d"`, then $nbpar_\Sigma = D$,
- if the constraint on Σ_k is `cstr$Sigma=""`, then $nbpar_\Sigma = D(D+1)/2$,
- if the constraint on Σ_k is `cstr$Sigma="*"`, then $nbpar_\Sigma = D(D+1)/(2K)$.

The rule to compute the number of parameters of Γ_k is the same as Σ_k, replacing D by L_t. Currently the Γ_k matrices are not constrained and $nbpar_\Gamma = L_t(L_t+1)/2$ because for indentifiability reasons the L_w part is set to the identity matrix.

The user must choose the number of mixtures components K and, if needed, the number of latent factors L_w. For small datasets (less than 100 observations), we suggest to select both (K, L_w) by minimizing the BIC criterion. For larger datasets, to save computation time, we suggest to set L_w using BIC while setting K to an arbitrary value large enough to catch non linear relations between responses and covariates and small enough to have several observations (at least 10) in each clusters. Indeed, for large datasets, the number of clusters should not have a strong impact on the results while it is sufficiently large.

Value

Returns a list with the following elements:

`LLf`	Final log-likelihood
`LL`	Log-likelihood value at each iteration of the EM algorithm
`theta`	A list containing the estimations of parameters as follows:
`c`	An `L x K` matrix of means of responses (Y) where `L=Lt+Lw`
`Gamma`	An `L x L x K` array of `K` matrices of covariances of responses (Y) where `L=Lt+Lw`
`A`	An `D x L x K` array of `K` matrices of affine transformation matrices where `L=Lt+Lw`
`b`	An `D x K` matrix in which affine transformation vectors are in columns
`Sigma`	An `D x D x K` array of X covariances
`nbpar`	The number of parameters estimated in the model
`phi`	A list containing the estimations of parameters as follows:
`r`	An `N x K` matrix of posterior probabilities
`pi`	A vector of length `K` of mixture weights i.e. prior probabilities of all components
`alpha`	A vector of length `K` of degree of freedom parameters (heaviness of the tail) for each Student component

Author(s)

Emeline Perthame (emeline.perthame@inria.fr), Florence Forbes (florence.forbes@inria.fr), Antoine Deleforge (antoine.deleforge@inria.fr)

References

[1] A. Deleforge, F. Forbes, and R. Horaud. High-dimensional regression with Gaussian mixtures and partially-latent response variables. Statistics and Computing, 25(5):893–911, 2015.

[2] E. Perthame, F. Forbes, and A. Deleforge. Inverse regression approach to robust nonlinear high-to-low dimensional mapping. Journal of Multivariate Analysis, 163(C):1–14, 2018. https://doi.org/10.1016/j.jmva.2017.09.0

See Also

xLLiM-package, emgm, sllim_inverse_map, gllim

Examples

```
data(data.xllim)
responses = data.xllim[1:2,] # 2 responses in rows and 100 observations in columns
covariates = data.xllim[3:52,] # 50 covariates in rows and 100 observations in columns

## Setting 5 components in the model
K = 5

## the model can be initialized by running an EM algorithm for Gaussian Mixtures (EMGM)
r = emgm(rbind(responses, covariates), init=K);
## and then the sllim model is estimated
mod = sllim(responses,covariates,in_K=K,in_r=r);

## if initialization is not specified, the model is automatically initialized by EMGM
## mod = sllim(responses,covariates,in_K=K)

## Adding 1 latent factor
## mod = sllim(responses,covariates,in_K=K,in_r=r,Lw=1)

## Some constraints on the covariance structure of \eqn{X} can be added
## mod = sllim(responses,covariates,in_K=K,in_r=r,cstr=list(Sigma="i"))
# Isotropic covariance matrices
# (same variance among covariates but different in each component)

## mod = sllim(responses,covariates,in_K=K,in_r=r,cstr=list(Sigma="d"))
# Heteroskedastic covariance matrices
# (variances are different among covariates and in each component)

## mod = sllim(responses,covariates,in_K=K,in_r=r,cstr=list(Sigma=""))
# Unconstrained full covariance matrices

## mod = sllim(responses,covariates,in_K=K,in_r=r,cstr=list(Sigma="*"))
# Full covariance matrices but equal for all components
```

---

sllim_inverse_map *Inverse Mapping from sllim parameters*

Description

This function computes the prediction of a new response from the estimation of the SLLiM model, returned by the function `sllim`.

Usage

```
sllim_inverse_map(y,theta,verb=0)
```

Arguments

y	An D x N matrix of input observations with variables in rows and subjects on columns
theta	An object returned by the sllim function
verb	Verbosity: print out the progression of the algorithm. If verb=0, there is no print, if verb=1, the progression is printed out. Default is 0.

Details

This function computes the prediction of a new response from the estimation of a SLLiM model, returned by the function sllim. Indeed, if the inverse conditional density $p(X|Y)$ and the marginal density $p(Y)$ are defined according to a SLLiM model (as described in xLLiM-package and sllim), the forward conditional density $p(Y|X)$ can be deduced.

Under SLLiM model, it is recalled that the inverse conditional $p(X|Y)$ is a mixture of Student regressions with parameters $(c_k, \Gamma_k, A_k, b_k, \Sigma_k)_{k=1}^K$ and $(\pi_k, \alpha_k)_{k=1}^K$. Interestingly, the forward conditional $p(Y|X)$ is also a mixture of Student regressions with parameters $(c_k^*, \Gamma_k^*, A_k^*, b_k^*, \Sigma_k^*)_{k=1}^K$ and $(\pi_k, \alpha_k)_{k=1}^K$. These parameters have a closed-form expression depending only on $(c_k, \Gamma_k, A_k, b_k, \Sigma_k)_{k=1}^K$ and $(\pi_k, \alpha_k)_{k=1}^K$.

Finally, the forward density (of interest) has the following expression:

$$p(Y|X=x) = \sum_k \frac{\pi_k S(x; c_k^*, \Gamma_k^*, \alpha_k, 1)}{\sum_j \pi_j S(x; c_j^*, \Gamma_j^*, \alpha_j, 1)} S(y; A_k^* x + b_k^*, \Sigma_k^*, \alpha_k^y, \gamma_k^y)$$

where (α_k^y, γ_k^y) determine the heaviness of the tail of the Generalized Student distribution. Note that $\alpha_k^y = \alpha_k + D/2$ and $\gamma_k^y = 1 + 1/2\delta(x, c_k^*, \Gamma_k^*)$ where δ is the Mahalanobis distance. A prediction of a new vector of responses is computed by:

$$E(Y|X=x) = \sum_k \frac{\pi_k S(x; c_k^*, \Gamma_k^*, \alpha_k, 1)}{\sum_j \pi_j S(x; c_j^*, \Gamma_j^*, \alpha_j, 1)} (A_k^* x + b_k^*)$$

where x is a new vector of observed covariates.

Value

Returns a list with the following elements:

x_exp	An L x N matrix of predicted responses by posterior mean. If L_w latent factors are added to the model, the first Lt rows $(1 : Lt)$ are predictions of responses and rows $(L_t+1) : L$ (recall that $L = L_t + L_w$) are estimations of latent factors.
alpha	Weights of the posterior Gaussian mixture model

Author(s)

Emeline Perthame (emeline.perthame@inria.fr), Florence Forbes (florence.forbes@inria.fr), Antoine Deleforge (antoine.deleforge@inria.fr)

References

[1] A. Deleforge, F. Forbes, and R. Horaud. High-dimensional regression with Gaussian mixtures and partially-latent response variables. Statistics and Computing, 25(5):893–911, 2015.

[2] E. Perthame, F. Forbes, and A. Deleforge. Inverse regression approach to robust nonlinear high-to-low dimensional mapping. Journal of Multivariate Analysis, 163(C):1–14, 2018. https://doi.org/10.1016/j.jmva.2017.09.0

See Also

xLLiM-package,sllim

Examples

```
data(data.xllim)

## Setting 5 components in the model
K = 5

## the model can be initialized by running an EM algorithm for Gaussian Mixtures (EMGM)
r = emgm(data.xllim, init=K);
## and then the sllim model is estimated
responses = data.xllim[1:2,] # 2 responses in rows and 100 observations in columns
covariates = data.xllim[3:52,] # 50 covariates in rows and 100 observations in columns
mod = sllim(responses,covariates,in_K=K,in_r=r);

# Prediction on a test dataset
data(data.xllim.test)
pred = sllim_inverse_map(data.xllim.test,mod)
## Predicted responses
print(pred$x_exp)
```

Index