Package ‘lmap’

March 8, 2024

Type Package
Title Logistic Mapping
Version 0.1.2
Maintainer Mark de Rooij <rooijm@fsw.leidenuniv.nl>
Description Set of tools for mapping of categorical response variables based on principal component analysis (pca) and multidimensional unfolding (mdu).
Depends R (>= 3.5.0), ggplot2, ggrepel, ggforce, fmdu
Imports nnet, stats, magrittr, dplyr, MASS, Rfast
License BSD_2_clause + file LICENSE
Encoding UTF-8
LazyData TRUE
Author Mark de Rooij [aut, cre, cph],
Frank Busing [aut, cph],
Juan Claramunt Gonzalez [aut]
NeedsCompilation yes
RoxygenNote 7.2.3
Repository CRAN
Date/Publication 2024-03-08 09:10:08 UTC

R topics documented:

clmdu .......................................................... 2
clpca .......................................................... 4
dataExample_clmdu ........................................... 5
dataExample_clpca ............................................ 6
dataExample_lmdu ............................................. 6
dataExample_lpca ............................................. 7
dataExample_mru .............................................. 8
esm ............................................................ 8
fastmbu ....................................................... 10
fastmru ....................................................... 11
clmdu

Cumulative Logistic (Restricted) MDU

Description
Cumulative Logistic (Restricted) MDU

Usage
clmdu(
  Y,
  X = NULL,
  S = 2,
  trace = FALSE,
  start = "svd",
  maxiter = 65536,
  dcrit = 1e-06
)

Arguments
Y An N times R ordinal matrix coded with integers 1,2,...
X An N by P matrix with predictor variables
S Positive number indicating the dimensionality of the solution
trace: boolean to indicate whether the user wants to see the progress of the function (default=TRUE)

start: either starting values (list with (U,V) or (B,V)) or way to compute them (svd, random, ca)

maxiter: maximum number of iterations

dcrit: convergence criterion

**Value**

Y: Matrix Y from input

X: original Matrix X from input

X: Scaled X matrix

mx: Mean values of X

sdX: Standard deviations of X

ynames: Variable names of responses

xnames: Variable names of predictors

probabilities: Estimated values of Y

m: main effects

U: matrix with coordinates for row-objects

B: matrix with regression weight (U = XB)

V: matrix with vectors for items/responses

iter: number of main iterations from the MM algorithm

deviance: value of the deviance at convergence

**Examples**

```r
## Not run:
data(dataExample_clmdu)
Y<-dataExample_clmdu
X<-dataExample_clmdu
output1 = clmdu(Y)
plot(output1)
plot(output1, circles = NULL)
summary(output1)

output2 = clmdu(Y = Y, X = X)
plot(output2, circles = c(1,2))
summary(output2)

## End(Not run)
```
Cumulative Logistic (Restricted) PCA

**Description**
Cumulative Logistic (Restricted) PCA

**Usage**
```r
clpca(
  Y,
  X = NULL,
  S = 2,
  lambda = FALSE,
  trace = FALSE,
  maxiter = 65536,
  dcrit = 1e-06
)
```

**Arguments**
- **Y**: An N times R ordinal matrix.
- **X**: An N by P matrix with predictor variables.
- **S**: Positive number indicating the dimensionality of the solution.
- **lambda**: if TRUE does lambda scaling (see Understanding Biplots, p24).
- **trace**: tracing information during iterations.
- **maxiter**: maximum number of iterations.
- **dcrit**: convergence criterion.

**Value**
- **Y**: Matrix Y from input.
- **X**: Original Matrix X from input.
- **X**: Scaled X matrix.
- **mx**: Mean values of X.
- **sdx**: Standard deviations of X.
- **ynames**: Variable names of responses.
- **xnames**: Variable names of predictors.
- **probabilities**: Estimated values of Y.
- **m**: Main effects.
- **U**: Matrix with coordinates for row-objects.
- **B**: Matrix with regression weight (U = XB).
Examples

```r
## Not run:
data(dataExample_clpca)
Y <- as.matrix(dataExample_clpca[,5:8])
X <- as.matrix(dataExample_clpca[,1:4])
out = clpca(Y)
out = clpca(Y, X)
## End(Not run)
```

Description

Dummy data for clmdu example

Usage

dataExample_clmdu

Format

A data frame with 200 observations on the following variables:

- X1 Continuous variable 1.
- X2 Continuous variable 2.
- X3 Continuous variable 3.
- X4 Continuous variable 4.
- Y1 Discrete variable 1.
- Y2 Discrete variable 2.
- Y3 Discrete variable 3.
- Y4 Discrete variable 4.
- Y5 Discrete variable 5.
### dataExample_clpca
*Dummy data for clpca example*

**Description**
Dummy data for clpca example

**Usage**
`dataExample_clpca`

**Format**
A data frame with 200 observations on the following variables:

- `X1` Continuous variable 1.
- `X2` Continuous variable 2.
- `X3` Continuous variable 3.
- `X4` Continuous variable 4.
- `Y1` Discrete variable 1.
- `Y2` Discrete variable 2.
- `Y3` Discrete variable 3.
- `Y4` Discrete variable 4.

---

### dataExample_lmdu
*Dummy data for lmdu example*

**Description**
Dummy data for lmdu example

**Usage**
`dataExample_lmdu`

**Format**
A data frame with 234 observations on the following variables:

- `Y1` Dichotomous variable 1.
- `Y2` Dichotomous variable 2.
- `Y3` Dichotomous variable 3.
- `Y4` Dichotomous variable 4.
Y5 Dichotomous variable 5.
Y6 Dichotomous variable 6.
Y7 Dichotomous variable 7.
Y8 Dichotomous variable 8.
X1 Continuous variable 1.
X2 Continuous variable 2.
X3 Continuous variable 3.
X4 Continuous variable 4.
X5 Continuous variable 5.

---

**Description**

Dummy data for lpca example

**Usage**

dataExample_lpca

**Format**

A data frame with 234 observations on the following variables:

Y1 Dichotomous variable 1.
Y2 Dichotomous variable 2.
Y3 Dichotomous variable 3.
Y4 Dichotomous variable 4.
Y5 Dichotomous variable 5.
Y6 Dichotomous variable 6.
Y7 Dichotomous variable 7.
Y8 Dichotomous variable 8.
X1 Continuous variable 1.
X2 Continuous variable 2.
X3 Continuous variable 3.
X4 Continuous variable 4.
X5 Continuous variable 5.
dataExample_mru  Dummy data for mru example

**Description**

Dummy data for mru example

**Usage**

```r
dataExample_mru
```

**Format**

A data frame with 234 observations on the following variables:

- `y` Categorical variable.
- `X1` Continuous variable 1.
- `X2` Continuous variable 2.
- `X3` Continuous variable 3.
- `X4` Continuous variable 4.
- `X5` Continuous variable 5.

esm  Extended Stereotype Model

**Description**

The function esm performs extended stereotype model analysis for multivariate logistic analysis i.e. a double constrained reduced rank multinomial logistic model

**Usage**

```r
esm(
  X,
  Y,
  S = 2,
  Z = NULL,
  W = NULL,
  ord.z = 1,
  ord.m = R,
  scale.x = FALSE,
  trace = FALSE,
  maxiter = 65536,
  dcrit = 1e-06
)
```
esm

Arguments

- **X**: An N by P matrix with predictor variables
- **Y**: An N times R binary matrix.
- **S**: Positive number indicating the dimensionality of the solution
- **Z**: design matrix for response
- **W**: design matrix for intercepts
- **ord.z**: if Z = NULL, the function creates Z having order ord.z
- **ord.m**: if W = NULL, the function creates W having order ord.m
- **scale.x**: whether X should be scaled to zero mean and standard deviation one
- **trace**: whether progress information should be printed on the screen
- **maxiter**: maximum number of iterations
- **dcrit**: convergence criterion

Value

This function returns an object of the class esm with components:

- **call**: function call
- **Xoriginal**: Matrix X from input
- **X**: Scaled X matrix
- **mx**: Mean values of X
- **sdx**: Standard deviations of X
- **Y**: Matrix Y from input
- **pnames**: Variable names of profiles
- **xnames**: Variable names of predictors
- **znames**: Variable names of responses
- **Z**: Design matrix Z
- **W**: Design matrix W
- **G**: Profile indicator matrix G
- **m**: main effects
- **bm**: regression weights for main effects
- **Bx**: regression weights for X
- **Bz**: regression weights for Z
- **A**: regression weights (Bx Bz’)
- **U**: matrix with coordinates for row-objects
- **V**: matrix with coordinates for column-objects
- **Ghat**: Estimated values of G
- **deviance**: value of the deviance at convergence
- **df**: number of parameters
- **AIC**: Akaike’s information criterion
- **iter**: number of main iterations from the MM algorithm
- **svd**: Singular value decomposition in last iteration
Examples

```r
## Not run:
data(dataExample_lpca)
Y = as.matrix(dataExample_lpca[, 1:5])
X = as.matrix(dataExample_lpca[, 9:13])
#unsupervised
output = esm(X, Y, S = 2, ord.z = 2)
## End(Not run)
```

---

**fastmbu**

*Fast version of mbu. It runs mbu without input checks.*

**Description**

Fast version of mbu. It runs mbu without input checks.

**Usage**

```r
fastmbu(  
  Y = NULL,  
  W = NULL,  
  XU = NULL,  
  BU = NULL,  
  XV = NULL,  
  BV = NULL,  
  mains = TRUE,  
  MAXINNER = 32,  
  FCRIT = 0.001,  
  MAXITER = 65536,  
  DCRIT = 1e-06
)
```

**Arguments**

- **Y** matrix with dichotomous responses
- **W** matrix with weights for each entrance of Y or vector with weights for each row of Y
- **XU** in unsupervised analysis starting values for row coordinates; in supervised analysis matrix with predictor variables for rows
- **BU** for supervised analysis matrix with regression weights for the row coordinates
- **XV** in unsupervised analysis starting values for column coordinates; in supervised analysis matrix with predictor variables for columns
- **BV** for supervised analysis matrix with regression weights for the column coordinates
fastmru

mains whether offsets for the items should be estimated
MAXINNER maximum number of iterations in the inner loop
FCRIT convergence criterion for STRESS in the inner loop
MAXITER maximum number of iterations in the outer loop
DCRIT convergence criterion for the deviance

Value

U estimated coordinate matrix for row objects
BU for supervised analysis the estimated matrix with regression weights for the rows
V estimated coordinate matrix for column objects
BV for supervised analysis the estimated matrix with regression weights for the columns
Mu estimated offsets
Lastinner number of iterations in the last call to STRESS
Lastfdif last difference in STRESS values in the inner loop
lastouter number of iterations in the outer loop
lastddif last difference in deviances in outer loop
deviance obtained deviance

fastmru

Fast version of mru. It runs mru without input checks.

Description

Fast version of mru. It runs mru without input checks.

Usage

```r
fastmru(
  G = NULL,
  X = NULL,
  B = NULL,
  V = NULL,
  MAXINNER = 32,
  FCRIT = 0.001,
  MAXITER = 65536,
  DCRIT = 1e-06,
  error.check = FALSE
)
```
Arguments

G  indicator matrix of the response variable
X  matrix with predictor variables
B  starting values of the regression weights
V  starting values for class locations
MAXINNER maximum number of iterations in the inner loop
FCRIT convergence criterion for STRESS in the inner loop
MAXITER maximum number of iterations in the outer loop
DCRIT convergence criterion for the deviance
error.check  extensive check validity input parameters (default = FALSE).

Value

B estimated regression weights
V estimated class locations
Lastinner number of iterations in the last call to STRESS
Lastfdif last difference in STRESS values in the inner loop
lastouter number of iterations in the outer loop
lastddif last difference in deviances in outer loop
deviance obtained deviance

lmdu

Logistic (Restricted) MDU

Description

This function runs: logistic multidimensional unfolding (if X = NULL) logistic restricted multidimensional unfolding (if X != NULL)

Usage

lmdu(
  Y,
  f = NULL,
  X = NULL,
  S = 2,
  start = "svd",
  maxiter = 65536,
  dcrit = 1e-06
)
**Arguments**

- **Y**: An N times R binary matrix.
- **f**: Vector with frequencies of response patterns in Y (only applicable if (X = NULL)).
- **X**: An N by P matrix with predictor variables.
- **S**: Positive number indicating the dimensionality of the solution.
- **start**: Either user provided starting values (start should be a list with U and V) or a way to compute starting values (choices: random, svd, ca).
- **maxiter**: maximum number of iterations.
- **dcrit**: convergence criterion.

**Value**

- **deviance**: Call to the function.
- **call**: Matrix Y from input.
- **Yoriginal**: Matrix Y from input.
- **Y**: Matrix Y from input.
- **f**: frequencies of rows of Y.
- **Xoriginal**: Matrix X from input.
- **X**: Scaled X matrix.
- **mx**: Mean values of X.
- **sdx**: Standard deviations of X.
- **ynames**: Variable names of responses.
- **xnames**: Variable names of predictors.
- **probabilities**: Estimated values of Y.
- **m**: main effects.
- **U**: matrix with coordinates for row-objects.
- **B**: matrix with regression weight (U = XB).
- **V**: matrix with vectors for items/responses.
- **iter**: number of main iterations from the MM algorithm.
- **deviance**: value of the deviance at convergence.
- **npar**: number of estimated parameters.
- **AIC**: Akaike’s Information Criterion.
- **BIC**: Bayesian Information Criterion.
Examples

```r
## Not run:
data(dataExample_lmdu)
Y = as.matrix(dataExample_lmdu[,1:8])
X = as.matrix(dataExample_lmdu[,9:13])
# unsupervised
output = lmdu(Y = Y, S = 2)
# supervised
output2 = lmdu(Y = Y, X = X, S = 2)
## End(Not run)
```

---

**lpca**  
*Logistic (Restricted) PCA*

## Description

This function runs: logistic principal component analysis (if X = NULL) logistic reduced rank regression (if X != NULL)

## Usage

```r
lpca(
  Y,
  X = NULL,
  S = 2,
  dim.indic = NULL,
  eq = FALSE,
  lambda = FALSE,
  maxiter = 65536,
  dcrit = 1e-06
)
```

## Arguments

- **Y**: An N times R binary matrix.
- **X**: An N by P matrix with predictor variables
- **S**: Positive number indicating the dimensionality of the solution
- **dim.indic**: An R by S matrix indicating which response variable pertains to which dimension
- **eq**: Only applicable when dim.indic not NULL; equality restriction on regression weights per dimension
- **lambda**: if TRUE does lambda scaling (see Understanding Biplots, p24)
- **maxiter**: maximum number of iterations
- **dcrit**: convergence criterion
Value

This function returns an object of the class lpca with components:

- **call**: Call to the function
- **Y**: Matrix Y from input
- **Xoriginal**: Matrix X from input
- **X**: Scaled X matrix
- **mx**: Mean values of X
- **sdx**: Standard deviations of X
- **ynames**: Variable names of responses
- **xnames**: Variable names of predictors
- **probabilities**: Estimated values of Y
- **m**: main effects
- **U**: matrix with coordinates for row-objects
- **B**: matrix with regression weight (U = XB)
- **V**: matrix with vectors for items/responses
- **iter**: number of main iterations from the MM algorithm
- **deviance**: value of the deviance at convergence
- **npar**: number of estimated parameters
- **AIC**: Akaike’s Information Criterion
- **BIC**: Bayesian Information Criterion

Examples

```r
## Not run:
data(dataExample_lpca)
Y = as.matrix(dataExample_lpca[, 1:8])
X = as.matrix(dataExample_lpca[, 9:13])
# unsupervised
output = lpca(Y = Y, S = 2)

## End(Not run)
```
The function `mru` performs multinomial restricted unfolding for a nominal response variable and a set of predictor variables.

**Usage**

```r
mru(y, X, S = 2, start = "da", maxiter = 65536, dcrit = 1e-05)
```

**Arguments**

- **y**: An N vector of the responses (categorical).
- **X**: An N by P matrix with predictor variables
- **S**: Positive number indicating the dimensionality of the solution
- **start**: Type of starting values (da: discriminant analysis, random or list with B and V)
- **maxiter**: Maximum number of iterations
- **dcrit**: Convergence criterion

**Value**

- `Y`: Matrix Y from input
- `X_original`: Matrix X from input
- `X_scaled`: Scaled X matrix
- `G`: Class indicator matrix
- `ynames`: Class names of response variable
- `xnames`: Variable names of the predictors
- `mx`: Means of the predictor variables
- `sdx`: Standard deviations of the predictor variables
- `U`: Coordinate matrix of row objects
- `B`: Matrix with regression coefficients
- `class`: Class coordinate matrix
- `iters`: Number of iterations
- `deviance`: Value of the deviance at convergence
Examples

```r
## Not run:
data(dataExample_mru)
y = as.matrix(dataExample_mru[1:20, 1])
X = as.matrix(dataExample_mru[1:20, 2:6])
output = mru(y = y, X = X, S = 2)
## End(Not run)
```

Description

Plots a Cumulative Logistic MDU model.

Usage

```r
## S3 method for class 'clmdu'
plot(
x,  
dims = c(1, 2),  
circles = seq(1, R),  
ycol = "darkgreen",  
xcol = "lightskyblue",  
ocol = "grey",  
...)
```

Arguments

- `x` an object of type `clmdu`
- `dims` which dimensions to visualize
- `circles` which circles to visualize
- `ycol` colour for representation of response variables
- `xcol` colour for representation of predictor variables
- `ocol` colour for representation of row objects
- `...` additional arguments to be passed.

Value

Plot of the results obtained from `clmdu`
Examples

```r
## Not run:
data(dataExample_clmdu)
Y = as.matrix(dataExample_clmdu[, 1:8])
X = as.matrix(dataExample_clmdu[, 9:13])
# unsupervised
output = clmdu(Y = Y, S = 2)
plot(output)

## End(Not run)
```

---

**plot.clpca**

*Plots a Cumulative Logistic PCA model*

**Description**

Plots a Cumulative Logistic PCA model

**Usage**

```r
## S3 method for class 'clpca'
plot(
x, 
dims = c(1, 2),
ycol = "darkgreen",
xcol = "lightskyblue",
ocol = "grey",
...
)
```

**Arguments**

- `x`: an object of type clpca
- `dims`: which dimensions to visualize
- `ycol`: colour for representation of response variables
- `xcol`: colour for representation of predictor variables
- `ocol`: colour for representation of row objects
- `...`: additional arguments to be passed.

**Value**

Plot of the results obtained from clpca
Examples

```r
## Not run:
data(dataExample_clpca)
Y<-as.matrix(dataExample_clpca[,5:8])
X<-as.matrix(dataExample_clpca[,1:4])
out = clpca(Y, X)
plot(out)

## End(Not run)
```

Description

Plots a Logistic MDU model

Usage

```r
## S3 method for class 'lmdu'
plot(
x, 
dims = c(1, 2),
ycol = "darkgreen",
xcol = "lightskyblue",
ocol = "grey",
..., 
)
```

Arguments

- `x`: an object of type `lmdu`
- `dims`: which dimensions to visualize
- `ycol`: colour for representation of response variables
- `xcol`: colour for representation of predictor variables
- `ocol`: colour for representation of row objects
- `...`: additional arguments to be passed.

Value

Plot of the results obtained from `lmdu`
Examples

```r
## Not run:
data(dataExample_lmdu)
Y = as.matrix(dataExample_lmdu[, 1:8])
X = as.matrix(dataExample_lmdu[, 9:13])
# unsupervised
output = lmdu(Y = Y, S = 2)
plot(output)

## End(Not run)
```

---

**plot.lpca**

*Plots a Logistic PCA Model*

### Description

Plots a Logistic PCA Model

### Usage

```r
## S3 method for class 'lpca'
plot(
  x, 
  dims = c(1, 2),
  type = "H",
  ycol = "darkgreen",
  xcol = "lightskyblue",
  ocol = "grey",
  ...
)
```

### Arguments

- `x` an object of type lpca
- `dims` which dimensions to visualize
- `type` either H (hybrid), I (inner product/pca), or D (distance/melodic)
- `ycol` colour for representation of response variables
- `xcol` colour for representation of predictor variables
- `ocol` colour for representation of row objects
- `...` additional arguments to be passed.

### Value

Plot of the results obtained from lpca
plot.mru

Examples

## Not run:
data(dataExample_lpca)
Y = as.matrix(dataExample_lpca[, 1:8])
X = as.matrix(dataExample_lpca[, 9:13])
# unsupervised
output = lpca(Y = Y, S = 2)
plot(output)

## End(Not run)

plot.mru  

Plots a Multinomial Restricted MDU model

Description

Plots a Multinomial Restricted MDU model

Usage

## S3 method for class 'mru'
plot(
  x,  
  dims = c(1, 2),
  class.regions = FALSE,
  ycol = "darkgreen",
  xcol = "lightskyblue",
  ocol = "grey",
  ...
)

Arguments

x  
an object of type mru

dims  
which dimensions to visualize

class.regions  
whether a voronoi diagram with classification regions should be included

ycol  
colour for representation of response variables

xcol  
colour for representation of predictor variables

ocol  
colour for representation of row objects

...  
additional arguments to be passed.

Value

Plot of the results obtained from mru
Examples

```r
## Not run:
data(dataExample_mru)
y = as.matrix(dataExample_mru[, 1])
X = as.matrix(dataExample_mru[, 2:6])
output = mru(y = y, X = X, S = 2)
plot(output)

## End(Not run)
```

**predict.clmdu**  
*The function `predict.clmdu` makes predictions for a test/validation set based on a fitted cl restricted multidimensional unfolding model (clmdu with X)*

**Description**

The function `predict.clmdu` makes predictions for a test/validation set based on a fitted cl restricted multidimensional unfolding model (clmdu with X)

**Usage**

```r
## S3 method for class 'clmdu'
predict(object, newX, newY = NULL, ...)
```

**Arguments**

- `object`  
  An clmdu object

- `newX`  
  An N by P matrix with predictor variables for a test/validation set

- `newY`  
  An N by R matrix with response variables for a test/validation set

- `...`  
  Additional arguments to be passed.

**Value**

This function returns an object of the class `predclpca` with components:

- `Yhat`  
  Predicted values for the test set

- `devr`  
  Estimated prediction deviance for separate responses

- `devtot`  
  Estimated prediction deviance for all responses
**Examples**

```r
## Not run:
data(dataExample_clpca)
Y = as.matrix(dataExample_clmdu[, 1:8])
X = as.matrix(dataExample_clmdu[, 9:13])
newY = as.matrix(dataExample_clmdu[1:20, 1:8])
newX = as.matrix(dataExample_clmdu[1:20, 9:13])
# supervised
output = clmdu(Y = Y, X = X, S = 2)
preds = predict(output, newX = newX, newY = newY)
## End(Not run)
```

**Description**

The function `predict.clpca` makes predictions for a test/validation set based on a fitted `clrrr` model (`clpca` with `X`).

**Usage**

```r
## S3 method for class 'clpca'
predict(object, newX, newY = NULL, ...)
```

**Arguments**

- **object**: An `clpca` object
- **newX**: An N by P matrix with predictor variables for a test/validation set
- **newY**: An N by R matrix with response variables for a test/validation set
- ... additional arguments to be passed.

**Value**

This function returns an object of the class `predclpca` with components:

- **Yhat**: Predicted values for the test set
- **devr**: Estimated prediction deviance for separate responses
- **devtot**: Estimated prediction deviance for all responses
Examples

```r
## Not run:
data(dataExample_clpca)
Y = as.matrix(dataExample_clpca[, 1:8])
X = as.matrix(dataExample_clpca[, 9:13])
newY = as.matrix(dataExample_clpca[1:20, 1:8])
newX = as.matrix(dataExample_clpca[1:20, 9:13])
# supervised
output = clpca(Y = Y, X = X, S = 2)
preds = predict(output, newX = newX, newY = newY)

## End(Not run)
```

predict.lmdu

The function predict.lmdu makes predictions for a test/validation set based on a fitted lmdu model (lmdu with X)

Description

The function predict.lmdu makes predictions for a test/validation set based on a fitted lmdu model (lmdu with X)

Usage

```r
## S3 method for class 'lmdu'
predict(object, newX, newY = NULL, ...)
```

Arguments

- `object`: An lmdu object
- `newX`: An N by P matrix with predictor variables for a test/validation set
- `newY`: An N by R matrix with response variables for a test/validation set
- `...`: additional arguments to be passed.

Value

This function returns an object of the class lpca with components:

- `Yhat`: Predicted values for the test set
- `devr`: Estimated prediction deviance for separate responses
- `devtot`: Estimated prediction deviance for all responses
- `Brier.r`: Estimated Brier score for separate responses
- `Brier`: Estimated Brier score for all responses
Examples

```r
## Not run:
data(dataExample_lpca)
Y = as.matrix(dataExample_lmdu[-c(1:20), 1:8])
X = as.matrix(dataExample_lmdu[-c(1:20), 9:13])
newY = as.matrix(dataExample_lmdu[1:20, 1:8])
ewX = as.matrix(dataExample_lmdu[1:20, 9:13])
# supervised
output = lmdu(Y = Y, X = X, S = 2)
preds = predict(output, newX = newX, newY = newY)
## End(Not run)
```

predict.lpca

The function `predict.lpca` makes predictions for a test/validation set based on a fitted lrrr model (lpca with X)

Description

The function `predict.lpca` makes predictions for a test/validation set based on a fitted lrrr model (lpca with X)

Usage

```r
## S3 method for class 'lpca'
predict(object, newX, newY = NULL, ...)
```

Arguments

- **object**: An lpca object
- **newX**: An N by P matrix with predictor variables for a test/validation set
- **newY**: An N by R matrix with response variables for a test/validation set
- **...**: additional arguments to be passed.

Value

This function returns an object of the class lpca with components:

- **Yhat**: Predicted values for the test set
- **devr**: Estimated prediction deviance for separate responses
- **devtot**: Estimated prediction deviance for all responses
- **Brier.r**: Estimated Brier score for separate responses
- **Brier**: Estimated Brier score for all responses
Examples

```r
## Not run:
data(dataExample_lpca)
Y = as.matrix(dataExample_lpca[-c(1:20) , 1:8])
X = as.matrix(dataExample_lpca[-c(1:20) , 9:13])
newY = as.matrix(dataExample_lpca[1:20 , 1:8])
newX = as.matrix(dataExample_lpca[1:20 , 9:13])
# supervised
output = lpca(Y = Y, X = X, S = 2)
preds = predict(output, newX = newX, newY = newY)

## End(Not run)
```

---

**predict.mru**  
*The function predict.mru makes predictions for a test/validation set based on a fitted mru model*

**Description**  
The function predict.mru makes predictions for a test/validation set based on a fitted mru model.

**Usage**  
```r
## S3 method for class 'mru'
predict(object, newX, newG = NULL, ...)
```

**Arguments**  
- **object**: An lmdu object
- **newX**: An N by P matrix with predictor variables for a test/validation set
- **newG**: An N by R matrix with response variables for a test/validation set
- **...**: additional arguments to be passed.

**Value**  
This function returns an object of the class p.mru with components:

- **Yhat**: Predicted values for the test set
- **dev**: Estimated prediction deviance
Examples

```r
## Not run:
data(dataExample_lpca)
Y = as.matrix(dataExample_mru[-c(1:20) , 1:8])
X = as.matrix(dataExample_mru[-c(1:20) , 9:13])
newY = as.matrix(dataExample_mru[1:20 , 1:8])
newX = as.matrix(dataExample_mru[1:20 , 9:13])
# supervised
output = mru(Y = Y, X = X, S = 2)
preds = predict(output, newX = newX, newY = newY)
```

## End(Not run)

summary.clmdu

### Summary of the results obtained from clmdu

The function `summary.lmdu` gives a summary from an object from `clmdu()

Description

Summarizing Cumulative Logistic MDU models

The function `summary.lmdu` gives a summary from an object from `clmdu()

Usage

```
## S3 method for class 'clmdu'
summary(object, ...)
```

Arguments

- **object**: An object resulting from `clmdu`
- **...**: additional arguments to be passed.

Value

Summary of the results obtained from `clmdu`
summary.clpca  Summarizing Cumulative Logistic PCA models

Description
The function summary.clpca gives a summary from an object from clpca()

Usage
## S3 method for class 'clpca'
summary(object, ...)

Arguments
object  An object resulting from clpca
...    additional arguments to be passed.

Value
Summary of the results obtained from clpca

summary.esm  Summarizing an Extended Stereotype Model

Description
The function summary.esm gives a summary from an object from esm()

Usage
## S3 method for class 'esm'
summary(object, ...)

Arguments
object  An object resulting from esm
...    additional arguments to be passed.

Value
Summary of the results obtained from esm
Summary Function: `summary.lmdu`

**Description**

The function `summary.lmdu` gives a summary from an object from `lmdu()`. The summary function provides a concise overview of the logistic MDU models.

**Usage**

```r
## S3 method for class 'lmdu'
summary(object, ...)  
```

**Arguments**

- `object`: An object resulting from `lmdu`
- `...`: additional arguments to be passed.

**Value**

A summary of the results obtained from `lmdu`.

---

Summary Function: `summary.lpca`

**Description**

The function `summary.lpca` gives a summary from an object from `lpca()`. The summary function provides a concise overview of the logistic PCA models.

**Usage**

```r
## S3 method for class 'lpca'
summary(object, ...)  
```

**Arguments**

- `object`: An object resulting from `lpca`
- `...`: additional arguments to be passed.

**Value**

A summary of the results obtained from `lpca`.

---
summary.mru

*Summarizing Multinomial Logistic Unfolding model* The function summary.mru gives a summary from an object from mru()

**Description**

Summarizing Multinomial Logistic Unfolding model

The function summary.mru gives a summary from an object from mru()

**Usage**

```r
## S3 method for class 'mru'
summary(object, ...)
```

**Arguments**

- `object` An object resulting from mru
- `...` additional arguments to be passed.

**Value**

Summary of the results obtained from mru

---

twomodedistance

*The function twomodedistance computes the two mode (unfolding) distance*

**Description**

The function twomodedistance computes the two mode (unfolding) distance

**Usage**

```r
twomodedistance(U, V)
```

**Arguments**

- `U` An N times S matrix with coordinates in S dimensional Euclidean space.
- `V` An R times S matrix with coordinates in S dimensional Euclidean space.

**Value**

- `D` a N by R matrix with Euclidean distances
Index

* datasets
  dataExample_clmdu, 5
  dataExample_clpca, 6
  dataExample_lmdu, 6
  dataExample_lpca, 7
  dataExample_mru, 8

clmdu, 2
clpca, 4

dataExample_clmdu, 5
dataExample_clpca, 6
dataExample_lmdu, 6
dataExample_lpca, 7
dataExample_mru, 8

esm, 8

fastmbu, 10
fastmru, 11

lmdu, 12
lpca, 14

mru, 16

plot.clmdu, 17
plot.clpca, 18
plot.lmdu, 19
plot.lpca, 20
plot.mru, 21
predict.clmdu, 22
predict.clpca, 23
predict.lmdu, 24
predict.lpca, 25
predict.mru, 26

summary.clmdu, 27
summary.clpca, 28
summary.esm, 28
summary.lmdu, 29

summary.lpca, 29
summary.mru, 30

twomodedistance, 30