Package ‘nda’

February 16, 2025

Type Package

Title Generalized Network-Based Dimensionality Reduction and Analysis

Version 0.2.4

Maintainer Zsolt T. Kosztyan <kosztyan.zsolt@gtk.uni-pannon.hu>

Description Non-parametric dimensionality reduction function. Reduction with and without feature selection. Plot functions. Automated feature selections. Kosztyan et. al. (2024) <doi:10.1016/j.eswa.2023.121779>.

License GPL (>= 2)

Encoding UTF-8

LazyData true

URL https://github.com/kzst/nda

Depends R (>= 4.00)

Imports energy, psych, stats, igraph, Matrix, methods, Rfast, MASS, mco, ppcor, lm.beta, leidenAlg, Metrics, visNetwork

RoxygenNote 7.2.3

NeedsCompilation no

Author Zsolt T. Kosztyan [aut, cre],
Marcell T. Kurbucz [aut],
Attila I. Katona [aut],
Zahid Khan [aut]

Repository CRAN

Date/Publication 2025-02-16 12:00:02 UTC

Contents

`nda-package` *Package of Generalized Network-based Dimensionality Reduction and Analyses*

Description

The package of Generalized Network-based Dimensionality Reduction and Analysis (GNDA).

Author(s)

Zsolt T. Kosztyan*, Marcell T. Kurbucz, Attila I. Katona, Zahid Khan

e-mail*: kosztyan.zsolt@gtk.uni-pannon.hu

References

Kosztyan, Z. T., Kurbucz, M. T., & Katona, A. I. (2022). Network-based dimensionality reduction of high-dimensional, low-sample-size datasets. Knowledge-Based Systems, 109180.

Kosztyán, Z. T., Katona, A. I., Kurbucz, M. T., & Lantos, Z. (2024). Generalized network-based dimensionality analysis. Expert Systems with Applications, 238, 121779. <URL: https://doi.org/10.1016/j.eswa.2023.121779>

See Also

ndr, ndrlm, plot, biplot, summary, dCor.

---

biplot.nda *Biplot function for Generalized Network-based Dimensionality Reduction and Analysis (GNDA)*

Description

Biplot function for Generalized Network-based Dimensionality Reduction and Analysis (GNDA)

Usage

```
## S3 method for class 'nda'
biplot(x, main=NULL,...)
```

Arguments

x	an object of class 'NDA'.
main	main title of biplot.
...	other graphical parameters.

Author(s)

Zsolt T. Kosztyan*, Marcell T. Kurbucz, Attila I. Katona

e-mail*: kosztyan.zsolt@gtk.uni-pannon.hu

References

Kosztyán, Z. T., Katona, A. I., Kurbucz, M. T., & Lantos, Z. (2024). Generalized network-based dimensionality analysis. Expert Systems with Applications, 238, 121779. <URL: https://doi.org/10.1016/j.eswa.2023.121779>

See Also

plot, summary, ndr, data_gen.

Examples

```
# Biplot function without feature selection

# Generate 200 x 50 random block matrix with 3 blocks and lambda=0 parameter

df<-data_gen(200,50,3,0)
p<-ndr(df)
biplot(p)
```

`COVID19_2020` *Covid'19 case datesets of countries (2020), where the data frame has 138 observations of 18 variables.*

Description

Sample datasets for Generalized Network-based Dimensionality Reduction and Analysis (GNDA)

Covid'19 of countries (2020), where the data frame has 138 observations of 18 variables.

Usage

```
data("COVID19_2020")
```

Format

A data frame with 138 observations 18 variables.

Source

Kurbucz, M. T. (2020). A joint dataset of official COVID-19 reports and the governance, trade and competitiveness indicators of World Bank group platforms. Data in brief, 31, 105881.

Examples

```
data(COVID19_2020)
```

`CrimesUSA1990.X` *Crimes in USA cities in 1990. Independent variables (X)*

Description

Sample datasets for Generalized Network-based Dimensionality Reduction and Analysis (GNDA)

Crimes in USA cities in 1990. Independent variables (X)

Usage

```
data("CrimesUSA1990.X")
```

Format

A data frame with 1994 observations 123 variables.

Source

UCI - Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/communities+and+crime

Examples

```
data(CrimesUSA1990.X)
```

`CrimesUSA1990.Y` *Crimes in USA cities in 1990. Dependent variable (Y)*

Description

Sample datasets for Generalized Network-based Dimensionality Reduction and Analysis (GNDA) Crimes in USA cities in 1990. Dependent variable (Y)

Usage

```
data("CrimesUSA1990.Y")
```

Format

A data frame with 1994 observations 1 variables.

Source

UCI - Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/communities+and+crime

Examples

```
data(CrimesUSA1990.Y)
```

`CWTS_2020` *CWTS Leiden's University Ranking 2020 for all scientific fields, within the period of 2016-2019. 1176 observations (i.e., universities), and 42 variables (i.e., indicators).*

Description

Sample datasets for Generalized Network-based Dimensionality Reduction and Analysis (GNDA) CWTS Leiden's 2020 dataset, where the data frame has 1176 observations of 42 variables.

Usage

```
data("CWTS_2020")
```

Format

A data frame with 1176 observations of 42 variables.

Source

CWTS Leiden Ranking 2020: https://www.leidenranking.com/ranking/2020/list

Examples

```
data(CWTS_2020)
```

---

data_gen — *Generate random block matrix for GNDA*

Description

Generate random block matrix for Generalized Network-based Dimensionality Reduction and Analysis (GNDA)

Usage

```
data_gen(n,m,nfactors=2,lambda=1)
```

Arguments

`n`	number of rows
`m`	number of columns
`nfactors`	number of blocks (factors, where the default value is 2)
`lambda`	exponential smoothing, where the default value is 1

Details

`n`, `m`, `nfactors` must beintegers, and they are not less than 1; lambda should be a positive real number.

Value

`M`	a dataframe of a block matrix

Author(s)

Prof. Zsolt T. Kosztyan, Department of Quantitative Methods, Institute of Management, Faculty of Business and Economics, University of Pannonia, Hungary

e-mail: kzst@gtk.uni-pannon.hu

Examples

```
# Specification 30 by 10 random block matrices with 2 blocks/factors
df<-data_gen(30,10)
library(psych)
scree(df)
biplot(ndr(df))
# Specification 40 by 20 random block matrices with 3 blocks/factors
df<-data_gen(40,20,3)
library(psych)
scree(df)
biplot(ndr(df))
plot(ndr(df))

# Specification 50 by 20 random block matrices with 4 blocks/factors
# lambda=0.1
df<-data_gen(50,15,4,0.1)
scree(df)
biplot(ndr(df))
plot(ndr(df))
```

---

dCor — *Calculating distance correlation of two vectors or columns of a matrix*

Description

Calculating distance correlation of two vectors or columns of a matrix for Generalized Network-based Dimensionality Reduction and Analysis (GNDA).

The calculation is very slow for large matrices!

Usage

```
dCor(x,y=NULL)
```

Arguments

x	a numeric vector, matrix or data frame.
y	NULL (default) or a vector, matrix or data frame with compatible dimensions to x. The default is equivalent to y = x (but more efficient).

Details

If `x` is a numeric vector, `y` must be specified. If `x` is a numeric matrix or numeric data frame, `y` will be neglected.

Value

Either a distance correlation coefficient of vectors `x` and `y`, or a distance correlation matrix of `x` if `x` is a matrix or a dataframe.

Author(s)

Prof. Zsolt T. Kosztyan, Department of Quantitative Methods, Institute of Management, Faculty of Business and Economics, University of Pannonia, Hungary

e-mail: kosztyan.zsolt@gtk.uni-pannon.hu

References

Rizzo M, Szekely G (2021). _energy: E-Statistics: Multivariate Inference via the Energy of Data_. R package version 1.7-8, <URL: https://CRAN.R-project.org/package=energy>.

Examples

```
# Specification of distance correlation value of vectors x and y.
x<-rnorm(36)
y<-rnorm(36)
dCor(x,y)
# Specification of distance correlaction matrix.
x<-matrix(rnorm(36),nrow=6)
dCor(x)
```

---

dCov — *Calculating distance covariance of two vectors or columns of a matrix*

Description

Calculating distance covariance of two vectors or columns of a matrix for Generalized Network-based Dimensionality Reduction and Analysis (GNDA).

The calculation is very slow for large matrices!

Usage

```
dCov(x,y=NULL)
```

Arguments

x	a numeric vector, matrix or data frame.
y	NULL (default) or a vector, matrix or data frame with compatible dimensions to x. The default is equivalent to y = x (but more efficient).

Details

If x is a numeric vector, y must be specified. If x is a numeric matrix or numeric data frame, y will be neglected.

Value

Either a distance covariance value of vectors x and y, or a distance covariance matrix of x if x is a matrix or a dataframe.

Author(s)

Prof. Zsolt T. Kosztyan, Department of Quantitative Methods, Institute of Management, Faculty of Business and Economics, University of Pannonia, Hungary

e-mail: kosztyan.zsolt@gtk.uni-pannon.hu

References

Rizzo M, Szekely G (2021). _energy: E-Statistics: Multivariate Inference via the Energy of Data_. R package version 1.7-8, <URL: https://CRAN.R-project.org/package=energy>.

Examples

```
# Specification of distance covariance value of vectors x and y.
x<-rnorm(36)
y<-rnorm(36)
dCov(x,y)
# Specification of distance covariance matrix.
x<-matrix(rnorm(36),nrow=6)
dCov(x)
```

---

fitted.ndrlm — *Calculation of fitted values of Generalized Network-based Dimensionality Reduction and Linear Regression Model (NDRLM)*

Description

Calculation of fitted values of Generalized Network-based Dimensionality Reduction and Linear Regression Model (NDRLM)

Usage

```
## S3 method for class 'ndrlm'
fitted(object, ...)
```

Arguments

`object`	an object of class 'ndrlm'.
`...`	further arguments passed to or from other methods.

Value

Fitted values (data frame)

Author(s)

Zsolt T. Kosztyan*, Marcell T. Kurbucz, Attila I. Katona

e-mail*: kosztyan.zsolt@gtk.uni-pannon.hu

References

Kosztyán, Z. T., Katona, A. I., Kurbucz, M. T., & Lantos, Z. (2024). Generalized network-based dimensionality analysis. Expert Systems with Applications, 238, 121779. <URL: https://doi.org/10.1016/j.eswa.2023.121779>

See Also

plot, print, ndrlm.

Examples

```
# Example of fitted function of NDRLM without optimization of fittings

X<-freeny.x
Y<-freeny.y
NDRLM<-ndrlm(Y,X,optimize=FALSE)

fitted(NDRLM)
```

---

fs.dimred *Feature selection for PCA, FA, and (G)NDA*

Description

This function drops variables that have low communality values and/or are common indicators (i.e., correlates more than one latent variables).

Usage

```
fs.dimred(fn,DF,min_comm=0.25,com_comm=0.25)
```

Arguments

fn	It is a list variable of the output of a principal (PCA), a fa (FA), or an ndr (NDA) function.
DF	Numeric data frame, or a numeric matrix of the data table
min_comm	Scalar between 0 to 1. Minimal communality value, which a variable has to be achieved. The default value is 0.25.
com_comm	Scalar between 0 to 1. The minimal difference value between loadings. The default value is 0.25.

Details

This function only works with principal, and fa, and ndr functions.

This function drops each variable that has a low communality value (under min_comm value). In other words, that variable does not fit enough of any latent variable.

This function also drops so-called common indicators, which correlate highly with more than one latent variable. And the difference in the correlation is either lower than the com_comm value or the greatest absolute factor loading value is not twice greater than the second greatest factor loading.

Value

`dropped_low`	Numeric data frame or numeric matrix. Set of indicators (i.e. variables), which are dropped by their low communalities. This value is NULL if a correlation matrix is used as an input or there is no dropped indicator.
`dropped_com`	Numeric data frame or numeric matrix. Set of dropped common indicators (i.e. common variables). This value is NULL if a correlation matrix is used as an input or there is no dropped indicator.
`remain_DF`	Numeric data frame or numeric matrix. Set of retained indicators
`...`	Other outputs came from

Author(s)

Zsolt T. Kosztyan*, Marcell T. Kurbucz, Attila I. Katona

e-mail*: kosztyan.zsolt@gtk.uni-pannon.hu

References

Abonyi, J., Czvetkó, T., Kosztyán, Z. T., & Héberger, K. (2022). Factor analysis, sparse PCA, and Sum of Ranking Differences-based improvements of the Promethee-GAIA multicriteria decision support technique. Plos one, 17(2), e0264277. doi:10.1371/journal.pone.0264277

See Also

`psych::principal`, `psych::fa`, `ndr`.

Examples

```
data<-I40_2020

library(psych)

# Principal Component Analysis (PCA)

pca<-principal(data,nfactors=2,covar=TRUE)
pca

# Feature selection with default values

PCA<-fs.dimred(pca,data)
```

```
PCA

# List of dropped, low communality value indicators
print(colnames(PCA$dropped_low))

# List of dropped, common communality value indicators
print(colnames(PCA$dropped_com))

# List of retained indicators
print(colnames(PCA$retained_DF))

## Not run:
# Principal Component Analysis (PCA) of correlation matrix

pca<-principal(cor(data,method="spearman"),nfactors=2,covar=TRUE)
pca

# Feature selection
min_comm<-0.25 # Minimal communality value
com_comm<-0.20 # Minimal common communality value

PCA<-fs.dimred(pca,cor(data,method="spearman"),min_comm,com_comm)
PCA

## End(Not run)
```

`fs.KMO` *Feature selection for KMO*

Description

Drop variables if their MSA_i valus is lower than a threshold, in order to increase the overall KMO (MSA) value.

Usage

```
fs.KMO(data,min_MSA=0.5,cor.mtx=FALSE)
```

Arguments

`data`	A numeric data frame
`min_MSA`	A numeric value. Minimal MSA value for variable i
`cor.mtx`	Boolean value. The input is either a correlation matrix (cor.mtx=TRUE), or not (cor.mtx=FALSE)

Details

Low Kaiser-Meyer-Olkin (KMO) measure of sampling adequacy does not suggest using principal component or factor analysis. Therefore, this function drop variables with low KMO/MSA values.

Value

`data` Cleaned data or the cleaned correlation matrix.

Author(s)

Zsolt T. Kosztyan*, Marcell T. Kurbucz, Attila I. Katona

e-mail*: kosztyan.zsolt@gtk.uni-pannon.hu

References

Abonyi, J., Czvetkó, T., Kosztyán, Z. T., & Héberger, K. (2022). Factor analysis, sparse PCA, and Sum of Ranking Differences-based improvements of the Promethee-GAIA multicriteria decision support technique. Plos one, 17(2), e0264277. doi:10.1371/journal.pone.0264277

See Also

`summary`.

Examples

```
library(psych)
data(I40_2020)
data<-I40_2020
KMO(fs.KMO(data,min_MSA=0.7,cor.mtx=FALSE))
```

GOVDB2020 — *Governmental and economic data of countries (2020), where the data frame has 138 observations of 2161 variables.*

Description

Sample datasets for Generalized Network-based Dimensionality Reduction and Analysis (GNDA)

Governmental and economic data of countries (2020), where the data frame has 138 observations of 2161 variables.

Usage

```
data("GOVDB2020")
```

Format

A data frame with 138 observations of 2161 variables.

Source

Kurbucz, M. T. (2020). A joint dataset of official COVID-19 reports and the governance, trade and competitiveness indicators of World Bank group platforms. Data in brief, 31, 105881.

Examples

```
data(GOVDB2020)
```

I40_2020 *NUTS2 regional development data (2020) of I4.0 readiness, where the data frame has 414 observations of 101 variables.*

Description

Sample datasets for Generalized Network-based Dimensionality Reduction and Analysis (GNDA)

NUTS2 regional development data (2020), where the data frame has 414 observations of 101 variables.

Usage

```
data("COVID19_2020")
```

Format

A data frame with 414 observations of 101 variables.

Source

Honti, G., Czvetkó, T., & Abonyi, J. (2020). Data describing the regional Industry 4.0 readiness index. Data in Brief, 33, 106464.

Examples

```
data(I40_2020)
```

---

ndr *Genearlized Network-based Dimensionality Reduction and Analysis (GNDA)*

Description

The main function of Generalized Network-based Dimensionality Reduction and Analysis (GNDA).

Usage

```
ndr(r,covar=FALSE,cor_method=1,cor_type=1,min_R=0,min_comm=2,Gamma=1,null_model_type=4,
mod_mode=6,min_evalue=0,min_communality=0,com_communalities=0,use_rotation=FALSE,
rotation="oblimin",weight=NULL,seed=NULL)
```

Arguments

- `r` — A numeric data frame
- `covar` — If this value is FALSE (default), it finds the correlation matrix from the raw data. If this value is TRUE, it uses the matrix r as a correlation/similarity matrix.
- `cor_method` — Correlation method (optional). '1' Pearson's correlation (default), '2' Spearman's correlation, '3' Kendall's correlation, '4' Distance correlation
- `cor_type` — Correlation type (optional). '1' Bivariate correlation (default), '2' partial correlation, '3' semi-partial correlation
- `min_R` — Minimal square correlation between indicators (default: 0).
- `min_comm` — Minimal number of indicators per community (default: 2).
- `Gamma` — Gamma parameter in multiresolution null modell (default: 1).
- `null_model_type` — '1' Differential Newmann-Grivan's null model, '2' The null model is the mean of square correlations between indicators, '3' The null model is the specified minimal square correlation, '4' Newmann-Grivan's modell (default)
- `mod_mode` — Community-based modularity calculation mode: '1' Louvain modularity, '2' Fast-greedy modularity, '3' Leading Eigen modularity, '4' Infomap modularity, '5' Walktrap modularity, '6' Leiden modularity (default)
- `min_evalue` — Minimal eigenvector centrality value (default: 0)
- `min_communality` — Minimal communality value of indicators (default: 0)
- `com_communalities` — Minimal common communalities (default: 0)
- `use_rotation` — FALSE no rotation (default), TRUE the rotation is used.
- `rotation` — "none", "varimax", "quartimax", "promax", "oblimin", "simplimax", and "cluster" are possible rotations/transformations of the solution. "oblimin" is the default, if use_rotation is TRUE.
- `weight` — The weights of columns. The defalt is NULL (no weights).
- `seed` — default seed value (default=NULL, no seed)

Details

NDA both works on low and high simple size datasets. If min_evalue=min_communality=com_communalities=0 than there is no feature selection.

Value

- `communality` — Communality estimates for each item. These are merely the sum of squared factor loadings for that item. It can be interpreted in correlation matrices.
- `loadings` — A standard loading matrix of class “loadings".
- `uniqueness` — Uniqueness values of indicators.
- `factors` — Number of found factors.
- `EVCs` — The list eigenvector centrality value of indicators.

`membership`	The membership value of indicators.
`weight`	The weight of indicators.
`scores`	Estimates of the factor scores are reported (if covar=FALSE).
`centers`	Colum mean of unstandardized score values.
`n.obs`	Number of observations specified or found.
`use_rotation`	FALSE no rotation (default), TRUE the rotation is used.
`rotation`	"none", "varimax", "quartimax", "promax", "oblimin", "simplimax", and "cluster" are possible rotations/transformations of the solution. "oblimin" is the default, if use_rotation is TRUE.
`fn`	Factor name: NDA
`seed`	applied seed value (default=NULL, no seed)
`Call`	Callback function

Author(s)

Zsolt T. Kosztyan*, Marcell T. Kurbucz, Attila I. Katona

e-mail*: kosztyan.zsolt@gtk.uni-pannon.hu

References

Kosztyan, Z. T., Kurbucz, M. T., & Katona, A. I. (2022). Network-based dimensionality reduction of high-dimensional, low-sample-size datasets. Knowledge-Based Systems, 109180. doi:10.1016/j.knosys.2022.109180

See Also

`plot`, `biplot`, `summary`.

Examples

```
# Dimension reduction without using any hyperparameters

data(swiss)
df<-swiss
p<-ndr(df)
summary(p)
plot(p)
biplot(p)

# Dimension reduction with using hyperparameters
# min_R=0.1 # The mininal square correlation must be grater than 0.1

p<-ndr(df,min_R = 0.1)
summary(p)
plot(p)

# min_evalue=0.1 # Minimal evector centalities must be greater than 0.1

p<-ndr(df,min_evalue = 0.1)
```

```
summary(p)
plot(p)

# minimal and common communality value must be greater than 0.25

p<-ndr(df,min_communality = 0.25,
 com_communalities = 0.25)

# Print factor matrix
cor(p$scores)
plot(p)

# Use factor rotation

p<-ndr(df,min_communality = 0.25,
 com_communalities = 0.25,use_rotation=TRUE)

# Print factor matrix
cor(p$scores)
biplot(p)

# Data reduction - clustering
# Distance is Euclidean's distance
# covar=TRUE means only the distance matrix is considered.

q<-ndr(1-normalize(as.matrix(dist(df))),covar=TRUE)
summary(q)
plot(q)
```

`ndrlm` *Genearlized Network-based Dimensionality Reduction and Regression (GNDR)*

Description

The main function of Generalized Network-based Dimensionality Reduction and Regression (GNDR) for supervised learning.

Usage

```
ndrlm(Y,X,latents="in",dircon=FALSE,optimize=TRUE,
                target="adj.r.square",rel_weight=FALSE,
                cor_method=1,
                cor_type=1,min_comm=2,Gamma=1,
                null_model_type=4,mod_mode=1,use_rotation=FALSE,
                rotation="oblimin",pareto=FALSE,fit_weights=NULL,
                lower.bounds.x = c(rep(-100,ncol(X))),
                upper.bounds.x = c(rep(100,ncol(X))),
```

```
lower.bounds.latentx = c(0,0,0,0),
upper.bounds.latentx = c(0.6,0.6,0.6,0.3),
lower.bounds.y = c(rep(-100,ncol(Y))),
upper.bounds.y = c(rep(100,ncol(Y))),
lower.bounds.latenty = c(0,0,0,0),
upper.bounds.latenty = c(0.6,0.6,0.6,0.3),
popsize = 20, generations = 30, cprob = 0.7, cdist = 5,
mprob = 0.2, mdist=10, seed=NULL)
```

Arguments

`Y` A numeric data frame of output variables

`X` A numeric data frame of input variables

`latents` The employs of latent variables: "in" employs latent-independent variables (default); "out" employs latent-dependent variables; "both" employs both latent-dependent and latent independent variables; "none" do not employs latent variable (= multiple regression)

`dircon` Wether enable or disable direct connection between input and output variables (default=FALSE)

`optimize` Optimization of fittings (default=TRUE)

`target` Target performance measures. The possible target measure are "adj.r.square" = adjusted R square (default), "r.sqauare" = R square, "MAE" = mean absolute error, "MAPE" = mean absolute percentage error, "MASE" = mean absolute scaled error ,"MSE"= mean square error,"RMSE" = root mean square error

`rel_weight` Use relative weights. In this case, all weights should be non-negative. (default=FALSE)

`cor_method` Correlation method (optional). '1' Pearson's correlation (default), '2' Spearman's correlation, '3' Kendall's correlation, '4' Distance correlation

`cor_type` Correlation type (optional). '1' Bivariate correlation (default), '2' partial correlation, '3' semi-partial correlation

`min_comm` Minimal number of indicators per community (default: 2).

`Gamma` Gamma parameter in multiresolution null modell (default: 1).

`null_model_type`
'1' Differential Newmann-Grivan's null model, '2' The null model is the mean of square correlations between indicators, '3' The null model is the specified minimal square correlation, '4' Newmann-Grivan's modell (default)

`mod_mode` Community-based modularity calculation mode: '1' Louvain modularity (default), '2' Fast-greedy modularity, '3' Leading Eigen modularity, '4' Infomap modularity, '5' Walktrap modularity, '6' Leiden modularity

`use_rotation` FALSE no rotation (default), TRUE the rotation is used.

`rotation` "none", "varimax", "quartimax", "promax", "oblimin", "simplimax", and "cluster" are possible rotations/transformations of the solution. "oblimin" is the default, if use_rotation is TRUE.

`pareto`	in the case of multiple objectives TRUE (default value) provides pareto-optimal solution, while FALSE provides weighted mean of objective functions (see out_weights)
`fit_weights`	weights of fitting the output variables (weights of means of objectives)
`lower.bounds.x`	Lower bounds of weights of independent variables in GNDA
`upper.bounds.x`	Upper bounds of weights of independent variables in GNDA
`lower.bounds.latentx`	Lower bounds of hyper-paremementers of GNDA for independent variables (values must be positive)
`upper.bounds.latentx`	Upper bounds of hyper-paremementers of GNDA for independent variables (value must be lower than one)
`lower.bounds.y`	Lower bounds of weights of dependent variables in GNDA
`upper.bounds.y`	Upper bounds of weights of dependent variables in GNDA
`lower.bounds.latenty`	Lower bounds of hyper-paremementers of GNDA for dependent variables (values must be positive)
`upper.bounds.latenty`	Upper bounds of hyper-paremementers of GNDA for dependent variables (value must be lower than one)
`popsize`	size of population of NSGA-II for fitting betas (default=20)
`generations`	number of generations to breed of NSGA-II for fitting betas (default=30)
`cprob`	crossover probability of NSGA-II for fitting betas (default=0.7)
`cdist`	crossover distribution index of NSGA-II for fitting betas (default=5)
`mprob`	mutation probability of NSGA-II for fitting betas (default=0.2)
`mdist`	mutation distribution index of NSGA-II for fitting betas (default=10)
`seed`	default seed value (default=NULL, no seed)

Details

NDRLM is a variable fitting with feature selection based on the tunes of GNDA method with NSGA-II algorithm for parameter fittings.

Value

`fval`	Objective function for fitting
`target`	Target performance measures. The possible target measure are "adj.r.square" = adjusted R square (default), "r.sqauare" = R square, "MAE" = mean absolute error, "MAPE" = mean absolute percentage error, "MASE" = mean absolute scaled error ,"MSE"= mean square error,"RMSE" = root mean square error
`hyperparams`	optimized hyperparameters
`pareto`	in the case of multiple objectives TRUE provides pareto-optimal solution, while FALSE (default) provides weighted mean of objective functions (see out_weights)
`Y`	A numeric data frame of output variables

X
: A numeric data frame of input variables

latents
: Latent model: "in", "out", "both", "none"

NDAin
: GNDA object, which is the result of model reduction and features selection in the case of employing latent-independent variables

NDAin_weight
: Weights of input variables (used in ndr)

NDAin_min_evalue
: Optimized minimal eigenvector centrality value (used in ndr)

NDAin_min_communality
: Optimized minimal communality value of indicators (used in ndr)

NDAin_com_communalities
: Optimized minimal common communalities (used in ndr)

NDAin_min_R
: Optimized minimal square correlation between indicators (used in ndr)

NDAout
: GNDA object, which is the result of model reduction and features selection in the case of employing latent-dependent variables

NDAout_weight
: Weights of input variables (used in ndr)

NDAout_min_evalue
: Optimized minimal eigenvector centrality value (used in ndr)

NDAout_min_communality
: Optimized minimal communality value of indicators (used in ndr)

NDAout_com_communalities
: Optimized minimal common communalities (used in ndr)

NDAout_min_R
: Optimized minimal square correlation between indicators (used in ndr)

fits
: List of linear regrassion models

otimized
: Wheter fittings are optimized or not

NSGA
: Outpot structure of NSGA-II optimization (list), if the optimization value is true (see in mco::nsga2)

extra_vars.X
: Logic variable. If direct connection (dircon=TRUE) is allowed not only the latent but the excluded input variables are analyized in the linear models as extra input variables.

extra_vars.Y
: Logic variable. If direct connection (dircon=TRUE) is allowed not only the latent but the excluded output variables are analyized in the linear models as extra input variables.

dircon_X
: The list of input variables which are directly connected to output variables.

dircon_Y
: The list of output variables which are directly connected to output variables.

seed
: applied seed value (default=NULL, no seed)

fn
: Function (regression) name: NDRLM

Call
: Callback function

Author(s)

Zsolt T. Kosztyan*, Marcell T. Kurbucz, Attila I. Katona

e-mail*: kosztyan.zsolt@gtk.uni-pannon.hu

References

Kosztyan, Z. T., Kurbucz, M. T., & Katona, A. I. (2022). Network-based dimensionality reduction of high-dimensional, low-sample-size datasets. Knowledge-Based Systems, 109180. doi:10.1016/j.knosys.2022.109180

See Also

ndr, plot, summary, mco::nsga2.

Examples

```
# Using NDRLM without fitting optimization
X<-freeny.x
Y<-freeny.y
NDRLM<-ndrlm(Y,X,optimize=FALSE)
summary(NDRLM)
plot(NDRLM)

## Not run:
# Using NDRLM with optimized fitting

NDRLM<-ndrlm(Y,X)
summary(NDRLM)

# Using Leiden's modularity for grouping variables

X<-freeny.x
Y<-freeny.y
NDRLM<-ndrlm(Y,X,mod_mode=6)
plot(NDRLM)

# Using relative weights

NDRLM<-ndrlm(Y,X,mod_mode=6,rel_weight=TRUE)
plot(NDRLM)

# Using Spearman's correlation

NDRLM<-ndrlm(Y,X,cor_method=2)
summary(NDRLM)

# Using greater population and generations

NDRLM<-ndrlm(Y,X,popsize=52,generations=40)
summary(NDRLM)

# No latent variables
NDRLM<-ndrlm(Y,X,latents="none")
plot(NDRLM)

# In-out model
library(lavaan)
df<-PoliticalDemocracy # Data of Political Democracy
```

```
dem<-PoliticalDemocracy[,c(1:8)]
ind60<-PoliticalDemocracy[,-c(1:8)]

NBSEM<-ndrlm(dem,ind60,latents = "both",seed = 2)
plot(NBSEM)

## End(Not run)
```

---

normalize *Min-max normalization*

Description

Min-max normalization for data matrices and data frames

Usage

```
normalize(x,type="all")
```

Arguments

x	A data frame or data matrix.
type	The type of normalization. "row" normalization row by row, "col" normalization column by column, and "all" normalization for the entire data frame/matrix (default)

Value

Returns a normalized data.frame/matrix.

Author(s)

Zsolt T. Kosztyan, University of Pannonia

e-mail: kosztyan.zsolt@gtk.uni-pannon.hu

Examples

```
mtx<-matrix(rnorm(20),5,4)
n_mtx<-normalize(mtx) # Fully normalized matrix
r_mtx<-normalize(mtx,type="row") # Normalize row by row
c_mtx<-normalize(mtx,type="col") # Normalize col by col
print(n_mtx) # Print fully normalized matrix
```

pdCor *Calculating partial distance correlation of columns of a matrix*

Description

Calculating partial distance correlation of two columns of a matrix for Generalized Network-based Dimensionality Reduction and Analysis (GNDA).

The calculation is very slow for large matrices!

Usage

```
pdCor(x)
```

Arguments

x	a a numeric matrix, or a numeric data frame

Value

Partial distance correlation matrix of x.

Author(s)

Prof. Zsolt T. Kosztyan, Department of Quantitative Methods, Institute of Management, Faculty of Business and Economics, University of Pannonia, Hungary

e-mail: kosztyan.zsolt@gtk.uni-pannon.hu

References

Rizzo M, Szekely G (2021). _energy: E-Statistics: Multivariate Inference via the Energy of Data_. R package version 1.7-8, <URL: https://CRAN.R-project.org/package=energy>.

Examples

```
# Specification of partial distance correlaction matrix.
x<-matrix(rnorm(36),nrow=6)
pdCor(x)
```

`plot.nda` *Plot function for Generalized Network-based Dimensionality Reduction and Analysis (GNDA)*

Description

Plot variable network graph

Usage

```
## S3 method for class 'nda'
plot(x, cuts=0.3, interactive=TRUE,edgescale=1.0,labeldist=-1.5,show_weights=FALSE,...)
```

Arguments

`x`	an object of class 'NDA'.
`cuts`	minimal square correlation value for an edge in the correlation network graph (default 0.3).
`interactive`	Plot interactive visNetwork graph or non-interactive igraph plot (default TRUE).
`edgescale`	Proportion scale value of edge width.
`labeldist`	Vertex label distance in non-interactive igraph plot (default value =-1.5).
`show_weights`	Show edge weights (default FALSE)).
`...`	other graphical parameters.

Author(s)

Zsolt T. Kosztyan*, Marcell T. Kurbucz, Attila I. Katona

e-mail*: kosztyan.zsolt@gtk.uni-pannon.hu

References

Kosztyán, Z. T., Katona, A. I., Kurbucz, M. T., & Lantos, Z. (2024). Generalized network-based dimensionality analysis. Expert Systems with Applications, 238, 121779. <URL: https://doi.org/10.1016/j.eswa.2023.121779>

See Also

`biplot`, `summary`, `ndr`.

Examples

```
# Plot function with feature selection

data("CrimesUSA1990.X")
df<-CrimesUSA1990.X
p<-ndr(df)
biplot(p,main="Biplot of CrimesUSA1990 without feature selection")
```

```
# Plot function with feature selection
# minimal eigen values (min_evalue) is 0.0065
# minimal communality value (min_communality) is 0.1
# minimal common communality value (com_communalities) is 0.1

p<-ndr(df,min_evalue = 0.0065,min_communality = 0.1,com_communalities = 0.1)

# Plot with default (cuts=0.3)
plot(p)

# Plot with higher cuts
plot(p,cuts=0.6)

# GNDA is used for clustering, where the similarity function is the 1-Euclidean distance
# Data is the swiss data

SIM<-1-normalize(as.matrix(dist(swiss)))
q<-ndr(SIM,covar = TRUE)
plot(q,interactive = FALSE)
```

`plot.ndrlm` *Plot function for Generalized Network-based Dimensionality Reduction and Regression (GNDR)*

Description

Plot the structural equation model, based on the GNDR

Usage

```
## S3 method for class 'ndrlm'
plot(x, sig=0.05, interactive=FALSE,...)
```

Arguments

`x`	An object of class 'NDRLM'.
`sig`	Significance level of relationships
`interactive`	Plot interactive visNetwork graph or non-interactive igraph plot (default FALSE).
`...`	other graphical parameters.

Author(s)

Zsolt T. Kosztyan*, Marcell T. Kurbucz, Attila I. Katona

e-mail*: kosztyan.zsolt@gtk.uni-pannon.hu

References

Kosztyán, Z. T., Katona, A. I., Kurbucz, M. T., & Lantos, Z. (2024). Generalized network-based dimensionality analysis. Expert Systems with Applications, 238, 121779. <URL: https://doi.org/10.1016/j.eswa.2023.121779>

See Also

summary, ndr,ndrlm.

Examples

```
# Plot function for non-optimized SEM

X<-freeny.x
Y<-freeny.y
NDRLM<-ndrlm(Y,X,optimize=FALSE)
plot(NDRLM)
```

---

predict.nda — *Calculation of predicted values of Generalized Network-based Dimensionality Reduction and Analysis (GNDA)*

Description

Calculation of predicted values of Generalized Network-based Dimensionality Reduction and Analysis (GNDA)

Usage

```
## S3 method for class 'nda'
predict(object, newdata, ...)
```

Arguments

object	An object of class 'nda'.
newdata	A required data frame in which to look for variables with which to predict.
...	further arguments passed to or from other methods.

Value

Residual values (data frame)

Author(s)

Zsolt T. Kosztyan*, Marcell T. Kurbucz, Attila I. Katona

e-mail*: kosztyan.zsolt@gtk.uni-pannon.hu

References

Kosztyán, Z. T., Katona, A. I., Kurbucz, M. T., & Lantos, Z. (2024). Generalized network-based dimensionality analysis. Expert Systems with Applications, 238, 121779. <URL: https://doi.org/10.1016/j.eswa.2023.121779>

See Also

plot, print, ndr.

Examples

```
# Example of prediction function of GNDA
set.seed(1) # Fix the random seed
data(swiss) # Use Swiss dataset
resdata<-swiss
sample <- sample(c(TRUE, FALSE), nrow(resdata), replace=TRUE, prob=c(0.9,0.1))
train <- resdata[sample, ] # Split the dataset to train and test
test <- resdata[!sample, ]
p<-ndr(train) # Use GNDA only on the train dataset
P<-ndr(swiss) # USE GNDA on the entire dataset
res<-predict(p,test) # Calculate the prediction to the test dataset
real<-P$scores[!sample, ]
cor(real,res) # The correlation between original and predicted values
```

`predict.ndrlm` *Calculation of predicted values of Generalized Network-based Dimensionality Reduction and Regression with Linear Models (NDRLM)*

Description

Calculation of predicted values of Generalized Network-based Dimensionality Reduction and Regression with Linear Models (NDRLM)

Usage

```
## S3 method for class 'ndrlm'
predict(object, newdata,
         se.fit = FALSE, scale = NULL, df = Inf,
        interval = c("none", "confidence", "prediction"),
        level = 0.95, type = c("response", "terms"),
        terms = NULL, na.action = stats::na.pass,
        pred.var = 1/weights, weights = 1, ...)
```

Arguments

object	An object of class 'ndrlm'.
newdata	An optional data frame in which to look for variables with which to predict. If omitted, the fitted values are used.

`se.fit`	A switch indicating if standard errors are required.
`scale`	Scale parameter for std.err. calculation.
`df`	Degrees of freedom for scale.
`interval`	Type of interval calculation. Can be abbreviated.
`level`	Tolerance/confidence level.
`type`	Type of prediction (response or model term). Can be abbreviated.
`terms`	If type = "terms", which terms (default is all terms), a character vector.
`na.action`	function determining what should be done with missing values in newdata. The default is to predict NA.
`pred.var`	the variance(s) for future observations to be assumed for prediction intervals. See 'Details'.
`weights`	the variance(s) for future observations to be assumed for prediction intervals. See 'Details'.
`...`	further arguments passed to or from other methods.

Details

predict.ndrlm produces predicted values, obtained by evaluating the multiple regression function and model reduction by GNDA in the frame newdata (which defaults to model.frame(object)). If the logical se.fit is TRUE, standard errors of the predictions are calculated. If the numeric argument scale is set (with optional df), it is used as the residual standard deviation in the computation of the standard errors, otherwise this is extracted from the model fit. Setting intervals specifies computation of confidence or prediction (tolerance) intervals at the specified level, sometimes referred to as narrow vs. wide intervals.

If the fit is rank-deficient, some of the columns of the design matrix will have been dropped. Prediction from such a fit only makes sense if newdata is contained in the same subspace as the original data. That cannot be checked accurately, so a warning is issued.

If newdata is omitted the predictions are based on the data used for the fit. In that case how cases with missing values in the original fit are handled is determined by the na.action argument of that fit. If na.action = na.omit omitted cases will not appear in the predictions, whereas if na.action = na.exclude they will appear (in predictions, standard errors or interval limits), with value NA. See also napredict.

The prediction intervals are for a single observation at each case in newdata (or by default, the data used for the fit) with error variance(s) pred.var. This can be a multiple of res.var, the estimated value of standard deviation: the default is to assume that future observations have the same error variance as those used for fitting. If weights is supplied, the inverse of this is used as a scale factor. For a weighted fit, if the prediction is for the original data frame, weights defaults to the weights used for the model fit, with a warning since it might not be the intended result. If the fit was weighted and newdata is given, the default is to assume constant prediction variance, with a warning.

Value

predict.ndrlm produces list of a vector of predictions or a matrix of predictions and bounds with column names fit, lwr, and upr if interval is set. For type = "terms" this is a matrix with a column per term and may have an attribute "constant".

The 'prediction' list contains the following element:

fit	vector or matrix as above
se.fit	residual standard deviations
residual.scale	residual standard deviations
df	degrees of freedom for residual

Note

Variables are first looked for in newdata and then searched for in the usual way (which will include the environment of the formula used in the fit). A warning will be given if the variables found are not of the same length as those in newdata if it was supplied.

Notice that prediction variances and prediction intervals always refer to future observations, possibly corresponding to the same predictors as used for the fit. The variance of the residuals will be smaller.

Strictly speaking, the formula used for prediction limits assumes that the degrees of freedom for the fit are the same as those for the residual variance. This may not be the case if res.var is not obtained from the fit.

Author(s)

Zsolt T. Kosztyan*, Marcell T. Kurbucz, Attila I. Katona

e-mail*: kosztyan.zsolt@gtk.uni-pannon.hu

References

Kosztyán, Z. T., Katona, A. I., Kurbucz, M. T., & Lantos, Z. (2024). Generalized network-based dimensionality analysis. Expert Systems with Applications, 238, 121779. <URL: https://doi.org/10.1016/j.eswa.2023.121779>

See Also

plot, print, ndr.

Examples

```
# Example of prediction function of NDRLM without optimization of fittings

set.seed(1)
X<-as.data.frame(freeny.x)
Y<-as.data.frame(freeny.y)
sample <- sample(c(TRUE, FALSE), nrow(X), replace=TRUE, prob=c(0.9,0.1))
train.X <- X[sample, ] # Split the dataset X to train and test
test.X <- X[!sample, ]
train.Y <- as.data.frame(Y[sample,]) # Split the dataset Y to train and test
colnames(train.Y)<-colnames(Y)
test.Y <- as.data.frame(Y[!sample,])
colnames(test.Y)<-colnames(Y)
train<-cbind(train.Y,train.X)
test<-cbind(test.Y,test.X)
res<-predict(lm(x~.,train),test)
cor(test.Y,res) # The correlation between original and predicted values
```

```
# Use NDRLM without optimization
NDRLM<-ndrlm(train.Y,train.X,optimize=FALSE)

# Calculate the prediction to the test dataset
res<-predict(NDRLM,test)
cor(test.Y,res[[1]]) # The correlation between original and predicted values
```

`print.nda` *Print function of Generalized Network-based Dimensionality Reduction and Analysis (GNDA)*

Description

Print summary of Generalized Network-based Dimensionality Reduction and Analysis (GNDA)

Usage

```
## S3 method for class 'nda'
print(x, digits = getOption("digits"), ...)
```

Arguments

`x`	an object of class 'nda'.
`digits`	the number of significant digits to use when `add.stats` = `TRUE`.
`...`	additional arguments affecting the summary produced.

Author(s)

Zsolt T. Kosztyan*, Marcell T. Kurbucz, Attila I. Katona

e-mail*: kzst@gtk.uni-pannon.hu

References

Kosztyán, Z. T., Katona, A. I., Kurbucz, M. T., & Lantos, Z. (2024). Generalized network-based dimensionality analysis. Expert Systems with Applications, 238, 121779. <URL: https://doi.org/10.1016/j.eswa.2023.121779>

See Also

biplot, plot, summary, ndr.

Examples

```
# Example of summary function of NDA without feature selection

data("CrimesUSA1990.X")
df<-CrimesUSA1990.X
p<-ndr(df)
summary(p)

# Example of summary function of NDA with feature selection
# minimal eigen values (min_evalue) is 0.0065
# minimal communality value (min_communality) is 0.1
# minimal common communality value (com_communalities) is 0.1

p<-ndr(df,min_evalue = 0.0065,min_communality = 0.1,com_communalities = 0.1)
print(p)
```

`print.ndrlm` *Print summary of Generalized Network-based Dimensionality Reduction and Linear Regression Model (NDRLM)*

Description

Print summary of Generalized Network-based Dimensionality Reduction and Linear Regression Model (NDRLM)

Usage

```
## S3 method for class 'ndrlm'
print(x, digits = getOption("digits"), ...)
```

Arguments

`x`	an object of class 'ndrlm'.
`digits`	the number of significant digits to use when `add.stats = TRUE`.
`...`	additional arguments affecting the summary produced.

Author(s)

Zsolt T. Kosztyan*, Marcell T. Kurbucz, Attila I. Katona

e-mail*: kzst@gtk.uni-pannon.hu

References

Kosztyán, Z. T., Katona, A. I., Kurbucz, M. T., & Lantos, Z. (2024). Generalized network-based dimensionality analysis. Expert Systems with Applications, 238, 121779. <URL: https://doi.org/10.1016/j.eswa.2023.121779>

See Also

biplot, plot, summary, ndrlm.

Examples

```
# Example of print function of NDRLM without optimization of fittings

X<-freeny.x
Y<-freeny.y
NDRLM<-ndrlm(Y,X,optimize=FALSE)
print(NDRLM)
```

---

residuals.ndrlm — *Calculation of residual values of Generalized Network-based Dimensionality Reduction and Linear Regression Model (NDRLM)*

Description

Calculation of residual values of Generalized Network-based Dimensionality Reduction and Linear Regression Model (NDRLM)

Usage

```
## S3 method for class 'ndrlm'
residuals(object, ...)
```

Arguments

`object`	an object of class 'ndrlm'.
`...`	further arguments passed to or from other methods.

Value

Residual values (data frame)

Author(s)

Zsolt T. Kosztyan*, Marcell T. Kurbucz, Attila I. Katona

e-mail*: kosztyan.zsolt@gtk.uni-pannon.hu

References

Kosztyán, Z. T., Katona, A. I., Kurbucz, M. T., & Lantos, Z. (2024). Generalized network-based dimensionality analysis. Expert Systems with Applications, 238, 121779. <URL: https://doi.org/10.1016/j.eswa.2023.121779>

See Also

plot, print, ndrlm.

Examples

```
# Example of residual function of NDRLM without optimization of fittings

X<-freeny.x
Y<-freeny.y
NDRLM<-ndrlm(Y,X,optimize=FALSE)

# Normality test for residuals
shapiro.test(residuals(NDRLM))
```

---

spdCor *Calculating semi-partial distance correlation of columns of a matrix*

Description

Calculating semi-partial distance correlation of two columns of a matrix for Generalized Network-based Dimensionality Reduction and Analysis (GNDA).

The calculation is very slow for large matrices!

Usage

```
spdCor(x)
```

Arguments

x	a a numeric matrix, or a numeric data frame

Value

Semi-partial distance correlation matrix of x.

Author(s)

Prof. Zsolt T. Kosztyan, Department of Quantitative Methods, Institute of Management, Faculty of Business and Economics, University of Pannonia, Hungary

e-mail: kosztyan.zsolt@gtk.uni-pannon.hu

References

Rizzo M, Szekely G (2021). _energy: E-Statistics: Multivariate Inference via the Energy of Data_. R package version 1.7-8, <URL: https://CRAN.R-project.org/package=energy>.

Examples

```
# Specification of semi-partial distance correlaction matrix.
x<-matrix(rnorm(36),nrow=6)
spdCor(x)
```

---

summary.nda — *Summary function of Generalized Network-based Dimensionality Reduction and Analysis (GNDA)*

Description

Print summary of Generalized Network-based Dimensionality Reduction and Analysis (GNDA)

Usage

```
## S3 method for class 'nda'
summary(object, digits = getOption("digits"), ...)
```

Arguments

`object`	an object of class 'nda'.
`digits`	the number of significant digits to use when `add.stats = TRUE`.
`...`	additional arguments affecting the summary produced.

Value

`communality`	Communality estimates for each item. These are merely the sum of squared factor loadings for that item. It can be interpreted in correlation matrices.
`loadings`	A standard loading matrix of class "loadings".
`uniqueness`	Uniqueness values of indicators.
`factors`	Number of found factors.
`scores`	Estimates of the factor scores are reported (if covar=FALSE).
`n.obs`	Number of observations specified or found.

Author(s)

Zsolt T. Kosztyan*, Marcell T. Kurbucz, Attila I. Katona

e-mail*: kosztyan.zsolt@gtk.uni-pannon.hu

References

Kosztyán, Z. T., Katona, A. I., Kurbucz, M. T., & Lantos, Z. (2024). Generalized network-based dimensionality analysis. Expert Systems with Applications, 238, 121779. <URL: https://doi.org/10.1016/j.eswa.2023.121779>

See Also

biplot, plot, print, ndr.

Examples

```
# Example of summary function of NDA without feature selection

data("CrimesUSA1990.X")
df<-CrimesUSA1990.X
p<-ndr(df)
summary(p)

# Example of summary function of NDA with feature selection
# minimal eigen values (min_evalue) is 0.0065
# minimal communality value (min_communality) is 0.1
# minimal common communality value (com_communalities) is 0.1

p<-ndr(df,min_evalue = 0.0065,min_communality = 0.1,com_communalities = 0.1)
summary(p)
```

`summary.ndrlm` *Summary function of Generalized Network-based Dimensionality Reduction and Linear Regression Model (NDRLM)*

Description

Print summary of Generalized Network-based Dimensionality Reduction and Linear Regression Model (NDRLM)

Usage

```
## S3 method for class 'ndrlm'
summary(object, digits = getOption("digits"), ...)
```

Arguments

`object`	an object of class 'ndrlm'.
`digits`	the number of significant digits to use when `add.stats` = `TRUE`.
`...`	additional arguments affecting the summary produced.

Value

`Call`	Callback function
`fval`	Objective function for fitting
`pareto`	in the case of multiple objectives TRUE (default value) provides pareto-optimal solution, while FALSE provides weighted mean of objective functions (see out_weights)

X	A numeric data frame of input variables
Y	A numeric data frame of output variables
NDA	GNDA object, which is the result of model reduction and features selection
fits	List of linear regrassion models
NDA_weight	Weights of input variables (used in ndr)
NDA_min_evalue	Optimized minimal eigenvector centrality value (used in ndr)
NDA_min_communality	Optimized minimal communality value of indicators (used in ndr)
NDA_com_communalities	Optimized minimal common communalities (used in ndr)
NDA_min_R	Optimized minimal square correlation between indicators (used in ndr)
NSGA	Outpot structure of NSGA-II optimization (list), if the optimization value is true (see in mco::nsga2)
fn	Function (regression) name: NDLM

Author(s)

Zsolt T. Kosztyan*, Marcell T. Kurbucz, Attila I. Katona

e-mail*: kosztyan.zsolt@gtk.uni-pannon.hu

References

Kosztyán, Z. T., Katona, A. I., Kurbucz, M. T., & Lantos, Z. (2024). Generalized network-based dimensionality analysis. Expert Systems with Applications, 238, 121779. <URL: https://doi.org/10.1016/j.eswa.2023.121779>

See Also

biplot, plot, print, ndrlm.

Examples

```
# Example of summary function of NDRLM without optimization of fittings

X<-freeny.x
Y<-freeny.y
NDRLM<-ndrlm(Y,X,optimize=FALSE)
summary(NDRLM)
```

Index