Package 'LDRTools'

July 21, 2025

Type Package

Title Tools for Linear Dimension Reduction

Version 0.2-2

Date 2023-09-17

Author Eero Liski [aut],
Klaus Nordhausen [aut, cre] (ORCID:
<https://orcid.org/0000-0002-3758-8501>),
Hannu Oja [aut] (ORCID: <https://orcid.org/0000-0002-4945-5976>),
Anne Ruiz-Gazen [aut] (ORCID: <https://orcid.org/0000-0001-8970-8061>)

Maintainer Klaus Nordhausen <klausnordhausenR@gmail.com>

Depends R (>= 3.2.2)

Suggests dr

Description Linear dimension reduction subspaces can be uniquely defined using orthogonal projection matrices. This package provides tools to compute distances between such subspaces and to compute the average subspace. For details see Liski, E.Nordhausen K., Oja H., Ruiz-Gazen A. (2016) Combining Linear Dimension Reduction Subspaces <doi:10.1007/978-81-322-3643-6_7>.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2023-09-17 23:12:34 UTC

Contents

LDRTools-package 2
AOP 3
B2P 7
O2P 8
Pdist 9

Index **11**

LDRTools-package *Tools for Linear Dimension Reduction*

Description

Linear dimension reduction subspaces can be uniquely defined using orthogonal projection matrices. This package provides tools to compute distances between such subspaces and to compute the average subspace. For details see Liski, E.Nordhausen K., Oja H., Ruiz-Gazen A. (2016) Combining Linear Dimension Reduction Subspaces <doi:10.1007/978-81-322-3643-6_7>.

Details

Package:	LDRTools
Type:	Package
Title:	Tools for Linear Dimension Reduction
Version:	0.2-2
Date:	2023-09-17
Authors@R:	c(person("Eero", "Liski", role = "aut"), person("Klaus", "Nordhausen", email = "klausnordhausenR@gmail.co
Author:	Eero Liski [aut], Klaus Nordhausen [aut, cre] (<https://orcid.org/0000-0002-3758-8501>), Hannu Oja [aut] (<h
Maintainer:	Klaus Nordhausen <klausnordhausenR@gmail.com>
Depends:	R (>= 3.2.2)
Suggests:	dr
Description:	Linear dimension reduction subspaces can be uniquely defined using orthogonal projection matrices. This pack
License:	GPL (>= 2)

Index of help topics:

AOP	Function to Average Orthogonal Projection Matrices
B2P	Function to Compute an Orthogonal Projection Matrix Based on an Arbitrary Matrix
LDRTools-package	Tools for Linear Dimension Reduction
O2P	Function to Compute an Orthogonal Projection Matrix Based on a Matrix with Orthonormal Columns
Pdist	Function to Compute the Distances Between Orthogonal Projection Matrices

Author(s)

Eero Liski [aut], Klaus Nordhausen [aut, cre] (<https://orcid.org/0000-0002-3758-8501>), Hannu Oja [aut] (<https://orcid.org/0000-0002-4945-5976>), Anne Ruiz-Gazen [aut] (<https://orcid.org/0000-0001-8970-8061>)

Maintainer: Klaus Nordhausen <klausnordhausenR@gmail.com>

References

Liski E., Nordhausen K., Oja H., and Ruiz-Gazen A. (2016), Combining Linear Dimension Reduction Subspaces. In: Agostinelli C., Basu A., Filzmoser P., Mukherjee D. (eds) Recent Advances in Robust Statistics: Theory and Applications. doi:10.1007/9788132236436_7.

`AOP` *Function to Average Orthogonal Projection Matrices*

Description

The function computes the average of orthogonal projection matrices and estimates the average rank.

Usage

```
AOP(x, weights = "constant")
```

Arguments

`x`	List of orthogonal projection matrices, can have different ranks.
`weights`	The weight function used for the individual ranks. Possible inputs are `constant`, `inverse` and `sq.inverse` (see details).

Details

The AOP maximizes the function $D(P) = w(k)tr(\bar{P}_w P) - \frac{1}{2}w^2(k)k$, where $\bar{P}_w = \frac{1}{m}\sum_{i=1}^{m} w(k_i)P_i$ is a regular average of weighted orthogonal projection matrices, m is the number of orthogonal projection matrices averaged, $w(k)$is the weight function and k is the rank of P. The possible weights are defined as `constant`: $w(k) = 1$, `inverse`: $w(k) = 1/k$ and `sq.inverse`: $w(k) = 1/\sqrt{k}$. The `constant` weight corresponds to the so called Crone & Crosby distance. Orthogonal projection matrices of zero rank are also possible inputs for the function. In such a case, the function prints a warning giving the number of orthogonal projection matrices with zero rank.

Value

A list containing the following components:

`P`	The estimated average orthogonal projection matrix.
`O`	An orthogonal matrix on which P is based upon.
`k`	The rank of the average orthogonal projection matrix.

Author(s)

Eero Liski and Klaus Nordhausen

References

Crone, L. J., and Crosby, D. S. (1995), Statistical Applications of a Metric on Subspaces to Satellite Meteorology, Technometrics 37, 324-328.

Liski E., Nordhausen K., Oja H., and Ruiz-Gazen A. (2016), Combining Linear Dimension Reduction Subspaces. In: Agostinelli C., Basu A., Filzmoser P., Mukherjee D. (eds) Recent Advances in Robust Statistics: Theory and Applications. doi:10.1007/97881322364436_7.

See Also

Pdist

Examples

```
## Ex.1
##
library(dr)
# Australian athletes data with 202 observations
data(ais)
# 10 explanatory variables
X <- as.matrix(ais[,c(2:3,5:12)])
colnames(X) <- names(ais[,c(2:3,5:12)])
p <- dim(X)[2]
# Response variable lean body mass (LBM)
y <- ais$LBM
# Significance level
alpha <- 0.05


# SIR
s0.sir <- dr(y ~ X, method="sir")
# Estimate of k
k.sir <- sum(dr.test(s0.sir, numdir=4)[,3] < alpha)
# List of transformation matrices corresponding to
# k.sir and fixed k=1, respectively
B.sir.list <- list(B1=s0.sir$evectors[,1:k.sir], B2=s0.sir$evectors[,1:1])
# List of orthogonal projectors corresponding to
# k.sir, fixed k=1 and fixed k=0, respectively
P.sir.list <- list(P1=O2P(B.sir.list$B1), P2=O2P(B.sir.list$B2),
P3=diag(0,p))


# SAVE
s0.save <- dr(y ~ X, method="save")
# Estimate of k
k.save <- sum(dr.test(s0.save, numdir=4)[,3] < alpha)
# List of transformation matrices corresponding to
# k.save and fixed k=1, respectively
B.save.list <- list(B1=s0.save$evectors[,1:k.save],
B2=s0.save$evectors[,1:1])
# List of orthogonal projectors corresponding to
# k.save, fixed k=1 and fixed k=0, respectively
```

```
P.save.list <- list(P1=O2P(B.save.list$B1), P2=O2P(B.save.list$B2),
P3=diag(0,p))


# DR k-estimates
dr.k <- c(k.sir, k.save)
names(dr.k) <- c("SIR","SAVE")
dr.k


# List of individually estimated projectors
proj.list.a <- list(P.sir.list$P1, P.save.list$P1)
# List of fixed projectors
proj.list.b <- list(P.sir.list$P2, P.save.list$P2)
# List of zero projectors
proj.list.c <- list(P.sir.list$P3, P.save.list$P3)
# List of zero-rank SIR-projector and
# other individually estimated projectors
proj.list.d <- list(P.sir.list$P3, P.save.list$P1)


# AOP (constant) object corresponding to the first projector list
AOP.const.a <- AOP(proj.list.a, weights="constant")

# AOP (inverse) objects corresponding to three projector lists
AOP.inv.a <- AOP(proj.list.a, weights="inverse")
AOP.inv.b <- AOP(proj.list.b, weights="inverse")
AOP.inv.c <- AOP(proj.list.c, weights="inverse")

# AOP (sq.inverse) objects corresponding to three projector lists
AOP.sqinv.a <- AOP(proj.list.a, weights="sq.inverse")
AOP.sqinv.c <- AOP(proj.list.c, weights="sq.inverse")
AOP.sqinv.d <- AOP(proj.list.d, weights="sq.inverse")


# k-estimates of the AOP's
AOP.a <- c(AOP.const.a$k, AOP.inv.a$k, AOP.sqinv.a$k)
names(AOP.a) <- c("const","inv","sqinv")
AOP.a

AOP.c <- AOP.inv.c$k
names(AOP.c) <- c("inv")
AOP.c

AOP.d <- AOP.sqinv.d$k
names(AOP.d) <- c("sqinv")
AOP.d


# Scatter plots between the response and the transformed data
# corresponding to the different AOP transformation matrices

# AOP.inverse
```

```
newdata.inv.AOPa <- cbind(y,X %*% AOP.inv.a$O)
pairs(newdata.inv.AOPa)

newdata.inv.AOPb <- cbind(y,X %*% AOP.inv.b$O)
pairs(newdata.inv.AOPb)


# AOP.sq.inverse
newdata.sqinv.AOPc <- cbind(y,X %*% AOP.sqinv.c$O)
pairs(newdata.sqinv.AOPc)

newdata.sqinv.AOPd <- cbind(y,X %*% AOP.sqinv.d$O)
pairs(newdata.sqinv.AOPd)




###################################
## Ex.2
##
a <- c(1,1,rep(0,8))
A <- diag(a)
B <- diag(0,10)
B[3,1] <- 1
P.A <- O2P(A[,1:2])
P.B <- O2P(B[,1])
zero.mat <- diag(0,10)
# True projector, k=3
P.C <- P.A + P.B

# Average P.A and P.B
proj.list <- list(P.A, P.B)
AOP.const <- AOP(proj.list, weights="constant")
AOP.inv <- AOP(proj.list, weights="inverse")
AOP.sqinv <- AOP(proj.list, weights="sq.inverse")
k.list <- c(AOP.const$k, AOP.inv$k, AOP.sqinv$k)
names(k.list) <- c("const","inv","sqinv")
k.list

# Average P.A, P.B and three zero rank matrices
proj.list <- list(P.A, P.B, zero.mat, zero.mat, zero.mat)
AOP.const <- AOP(proj.list, weights="constant")
AOP.inv <- AOP(proj.list, weights="inverse")
AOP.sqinv <- AOP(proj.list, weights="sq.inverse")
k.list <- c(AOP.const$k, AOP.inv$k, AOP.sqinv$k)
names(k.list) <- c("const","inv","sqinv")
k.list
```

B2P *Function to Compute an Orthogonal Projection Matrix Based on an Arbitrary Matrix*

Description

Function to compute an orthogonal projection matrix based on an arbitrary matrix.

Usage

```
B2P(x)
```

Arguments

x	A matrix with p rows and k columns.

Details

The orthogonal projection matrix P corresponding to matrix x is defined as $P = x(x^T x)^{-1} x^T$.

Value

The resulting orthogonal projection matrix.

Author(s)

Klaus Nordhausen

See Also

O2P

Examples

```
set.seed(1)
X <- matrix(rnorm(30),ncol=3)
P <- B2P(X)
```

O2P *Function to Compute an Orthogonal Projection Matrix Based on a Matrix with Orthonormal Columns*

Description

Function to compute an orthogonal projection matrix based on a matrix with orthonormal columns.

Usage

```
O2P(x)
```

Arguments

x	a matrix with k orthonormal columns of length p.

Details

The orthogonal projection matrix P corresponding to matrix x is defined as $P = xx^T$.

Value

The resulting orthogonal projection matrix.

Author(s)

Klaus Nordhausen

See Also

B2P

Examples

```
X <- tcrossprod(matrix(rnorm(100),ncol=10))
# Orthogonal projector based on the first three eigenvectors of X
P <- O2P(eigen(X)$vectors[,1:3])
```

`Pdist` *Function to Compute the Distances Between Orthogonal Projection Matrices*

Description

The function computes distances between orthogonal projection matrices that might have different ranks. Different weight functions for the ranks are available.

Usage

```
Pdist(x, weights = "constant")
```

Arguments

`x`	List of othogonal projection matrices (can have different ranks).
`weights`	The weight function used for the individual ranks. Possible inputs are `constant`, `inverse` and `sq.inverse` (see details).

Details

A weighted distance between subspaces P_1 and P_2 with ranks k_1 and k_2 is given by $D_w^2(P_1, P_2) = \frac{1}{2}||w(k_1)P_1 - w(k_2)P_2||^2$, where w denotes the weight function. The possible weights are defined as `constant`: $w(k) = 1$, `inverse`: $w(k) = 1/k$ and `sq.inverse`: $w(k) = 1/\sqrt{k}$. The `constant` weight corresponds to the so called Crone & Crosby distance. Orthogonal projection matrices of zero rank are also possible inputs for the function.

Value

an object of class `dist` having the attributes:

`Size`	number of orthogonal projection matrices.
`Labels`	names of orthogonal projection matrices if available.
`Diag`	FALSE.
`Upper`	FALSE.
`methods`	The name of the weights used.

Author(s)

Eero Liski and Klaus Nordhausen

References

Crone, L. J., and Crosby, D. S. (1995), Statistical Applications of a Metric on Subspaces to Satellite Meteorology, Technometrics 37, 324-328.

Liski E., Nordhausen K., Oja H., and Ruiz-Gazen A. (2016), Combining Linear Dimension Reduction Subspaces. In: Agostinelli C., Basu A., Filzmoser P., Mukherjee D. (eds) Recent Advances in Robust Statistics: Theory and Applications. doi:10.1007/9788132236436_7.

See Also

AOP

Examples

```
# Ex.1
X.1 <- tcrossprod(matrix(rnorm(16),ncol=4))
X.2 <- tcrossprod(matrix(rnorm(16),ncol=4))
X.3 <- tcrossprod(matrix(rnorm(16),ncol=4))
U1 <- eigen(X.1)$vectors
U2 <- eigen(X.2)$vectors
U3 <- eigen(X.3)$vectors

PRO <- list(P1=O2P(U1),P2=O2P(U2),P3=O2P(U3))

DIST.MAT<-Pdist(PRO)
str(DIST.MAT)
as.matrix(DIST.MAT)
print(DIST.MAT, diag=TRUE)
print(DIST.MAT, diag=TRUE, upper=TRUE)

PRO2 <- list(O2P(U1),O2P(U2),O2P(U3))
Pdist(PRO2, weights="inverse")

############################
# Ex.2
a <- c(1,1,rep(0,8))
A <- diag(a)
b <- c(1,1,1,1,rep(0,6))
B <- diag(b)
P.A <- O2P(A[,1:2])
P.B <- O2P(B[,1:4])

proj.list <- list(P.A,P.B)
Pdist(proj.list, weights="constant")
Pdist(proj.list, weights="inverse")
Pdist(proj.list, weights="sq.inverse")
```

Index

∗ **multivariate**
AOP, 3
B2P, 7
O2P, 8
Pdist, 9
∗ **package**
LDRTools-package, 2

AOP, 3, *10*

B2P, 7, *8*

LDRTools *(*LDRTools-package*)*, 2
LDRTools-package, 2

O2P, *7*, 8

Pdist, *4*, 9