Package ‘FPCdpca’

May 9, 2025

Type Package

Title The FPCdpca Criterion on Distributed Principal Component Analysis

Date 2025-05-08

Version 0.3.0

Maintainer Guangbao Guo <ggb11111111@163.com>

Description

We consider optimal subset selection in the setting that one needs to use only one data subset to represent the whole data set with minimum information loss, and devise a novel intersection-based criterion on selecting optimal subset, called as the FPC criterion, to handle with the optimal sub-estimator in distributed principal component analysis; That is, the FPCdpca. The philosophy of the package is described in Guo G. (2025) <doi:10.1016/j.physa.2024.130308>.

License Apache License (== 2.0)

Encoding UTF-8

RoxygenNote 7.3.2

Imports matrixcalc,rsvd,stats

LazyData true

Suggests testthat (>= 3.0.0)

NeedsCompilation no

Config/testthat/edition 3

Author Guangbao Guo [aut, cre] (ORCID: <https://orcid.org/0000-0002-4115-6218>), Jiarui Li [aut]

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2025-05-09 02:50:06 UTC

Contents

Depca 2
Dpca 3
Drp 4
Drpca 4
Drsvd 5
Dsvd 6
FPC 6
review 7
riboflavin 8
riboflavinv100 9

Index **11**

Depca *Decentralized PCA*

Description

Decentralized PCA

Usage

```
Depca(data,K, nk, d, eps, nit.max)
```

Arguments

`data`	is a sparse random projection matrix
`K`	is the desired target rank.
`nk`	is the size of subsets.
`d`	is the dimension. p the number of variables.
`eps`	the error of the subsets.
`nit.max`	the maximum of the subsets.

Value

MSEXrp, MSEvrp, MSESrp, kopt

Examples

```
K=20; nk=50; nr=10; p=8;  n=K*nk;d=5
data=matrix(c(rnorm((n-nr)*p,0,1),rpois(nr*p,100)),ncol=p)
set.seed(1234)
eps=10^(-1);nit.max=1000
Depca(data=data,K=K, nk=nk, d=d, eps=eps,nit.max=nit.max)
TXde=TSde=c(rep(0,5))
for (j in 1:5){
 depca=Depca(data,K, nk,d, eps, nit.max)
 TXde[j]=as.numeric(depca)[1]
 TSde[j]=as.numeric(depca)[2]}
mean(TXde)
mean(TSde)
```

Dpca *Distributed Principal Component Analysis (DPCA)*

Description

Performs distributed PCA on a data matrix partitioned into subsets.

Usage

```
Dpca(data, K, nk)
```

Arguments

data	A numeric matrix or data frame containing the data, where rows are observations and columns are variables.
K	Integer, the number of subsets to partition the data into.
nk	Integer, the size of each subset (number of rows per subset).

Details

The function splits the input data matrix into K subsets of size nk each. The parameters n (number of rows) and p (number of columns) are automatically derived from the input data matrix as n = nrow(data) and p = ncol(data).

Value

A list containing:

- MSEXp: Minimum squared reconstruction error.
- MSEvp: MSE of eigenvectors.
- MSESp: MSE of covariance matrix.
- kopt: Optimal subset index.

Examples

```
K <- 20
nk <- 50
nr <- 10
p <- 8
n <- K * nk
d <- 6
data <- matrix(c(rnorm((n - nr) * p, 0, 1), rpois(nr * p, 100)), ncol = p)
Dpca(data = data, K = K, nk = nk)
```

Drp *Distributed random projection*

Description

Distributed random projection

Usage

```
Drp(data,K, nk,d)
```

Arguments

`data`	is sparse random projection matrix
`K`	is the number of distributed nodes.
`nk`	is the size of subsets.
`d`	is the dimension number. n is the sample size. p the number of variables.

Value

MSEXrp, MSEvrp, MSESrp, kopt

Examples

```
K=20; nk=50; nr=10; p=8; d=5; n=K*nk;
data=matrix(c(rnorm((n-nr)*p,0,1),rpois(nr*p,100)),ncol=p)
data=matrix(rpois((n-nr)*p,1),ncol=p); rexp(nr*p,1); rchisq(10000, df = 5);
Drp(data=data,K=K, nk=nk,d=d)
```

Drpca *Distributed random PCA*

Description

Distributed random PCA

Usage

```
Drpca(data,K, nk,d)
```

Arguments

`data`	is sparse random projection matrix
`K`	is the number of distributed nodes.
`nk`	is the size of subsets.
`d`	is the dimension number. n is the sample size. p the number of variables.

Value

MSEXrp, MSEvrp, kSopt, kxopt

Examples

```
K=20; nk=50; nr=50; p=8;d=5; n=K*nk;
data=matrix(c(rnorm((n-nr)*p,0,1),rpois(nr*p,100)),ncol=p)
```

`Drsvd` *Distributed Random SVD*

Description

Distributed Random SVD

Usage

```
Drsvd(data, K, nk, m, q, k)
```

Arguments

`data`	A numeric matrix or data frame.
`K`	Number of distributed nodes.
`nk`	Size of each subset.
`m`	Target dimension for random projection.
`q`	Number of power iterations.
`k`	Desired rank.

Value

A vector containing MSE values and optimal subset index.

Examples

```
library(rsvd)
library(matrixcalc)
K <- 20
nk <- 50
p <- 8
m <- 5
q <- 5
k <- 4
n <- K * nk
data <- matrix(rexp(n * p, 0.8), ncol = p)
Drsvd(data = data, K = K, nk = nk, m = m, q = q, k = k)
```

Dsvd *Distributed svd*

Description

Distributed svd

Usage

```
Dsvd(data,K, nk,k)
```

Arguments

data	a real input matrix (or data frame) to be decomposed.
K	the number of blocks into which variable X is divided.
nk	The number of each blocks.
k	the desired target rank.

Value

MSE of Xs,vsvd,Ssvd and kopt.

Examples

```
library(matrixcalc)
K=20; nk=50; nr=10; p=8; k=4; n=K*nk;
data=matrix(c(rnorm((n-nr)*p,0,1),rpois(nr*p,100)),ncol=p)
Dsvd(data=data,K=K, nk=nk,k=k)
```

FPC *FPC*

Description

FPC

Usage

```
FPC(data,K,nk)
```

Arguments

data	is a data set
K	is an index subset/sub-vectoris
nk	is an index subset/sub-vectoris for each block

Value

MSEv1,MSEv2,MSEvopt,MSESopt1,MSESopt2,MSESopt,MSEShat,MSESba,MSESw

Examples

```
library(matrixcalc)
K=20; nk=500; p=8; n=10000;m=50
data=matrix(c(rnorm((n-m)*p,0,1),rpois(m*p,100)),ncol=p)
FPC(data=data,K=K,nk=nk)
```

`review` *Review*

Description

This dataset contains travel reviews from TripAdvisor.com, covering destinations in 11 categories across East Asia. Each traveler's rating is mapped to a scale from Terrible (0) to Excellent (4), and the average rating for each category per user is provided.

Usage

```
review
```

Format

A data frame with multiple rows and 12 columns.

- `User_ID`: Unique identifier for each user (Categorical).
- `Art_Galleries`: Average user feedback on art galleries.
- `Dance_Clubs`: Average user feedback on dance clubs.
- `Juice_Bars`: Average user feedback on juice bars.
- `Restaurants`: Average user feedback on restaurants.
- `Museums`: Average user feedback on museums.
- `Resorts`: Average user feedback on resorts.
- `Parks_Picnic_Spots`: Average user feedback on parks and picnic spots.
- `Beaches`: Average user feedback on beaches.
- `Theaters`: Average user feedback on theaters.
- `Religious_Institutions`: Average user feedback on religious institutions.

Details

The dataset is populated by crawling TripAdvisor.com and includes reviews on destinations in 11 categories across East Asia. Each traveler's rating is mapped as follows: Excellent (4), Very Good (3), Average (2), Poor (1), and Terrible (0). The average rating for each category per user is used.

Note

This dataset is licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) license, which allows for sharing and adaptation of the data for any purpose, provided that appropriate credit is given.

Source

UCI Machine Learning Repositor

Examples

```
# Load the dataset
data(review)

# Print the first few rows of the dataset
print(head(review))

# Summary statistics for the ratings
summary(review[, 2:11])
```

`riboflavin` *Riboflavin Production Data*

Description

This dataset contains measurements of riboflavin (vitamin B2) production by Bacillus subtilis, a Gram-positive bacterium commonly used in industrial fermentation processes. The dataset includes $n = 71$ observations with $p = 4088$ predictors, representing the logarithm of the expression levels of 4088 genes. The response variable is the log-transformed riboflavin production rate.

Usage

```
data(riboflavin)
```

Format

- **y** Log-transformed riboflavin production rate (original name: `q_RIBFLV`). This is a continuous variable indicating the efficiency of riboflavin production by the bacterial strain.
- **x** A matrix of dimension 71×4088 containing the logarithm of the expression levels of 4088 genes. Each column corresponds to a gene, and each row corresponds to an observation (experimental condition or time point).

Details

The riboflavin dataset is a high-dimensional dataset commonly used in statistical research, particularly in the fields of bioinformatics and systems biology. It was originally collected to study the genetic regulation of riboflavin biosynthesis in Bacillus subtilis. The data were generated using DNA microarray technology to measure gene expression levels under various experimental conditions.

Note

The dataset is provided by DSM Nutritional Products Ltd., a leading company in the field of nutritional ingredients. The data have been preprocessed and normalized to account for technical variations in the microarray measurements.

Source

DSM Nutritional Products Ltd., Basel, Switzerland.

References

- Bühlmann, P., Kalisch, M., & Meier, L. (2014). 'High-dimensional statistics with a view towards applications in biology.' *Annual Review of Statistics and its Applications*, **1**, 255–278.
- DSM Nutritional Products Ltd. (2005). 'Genome-scale analysis of Bacillus subtilis riboflavin production.' *Internal Report*.

Examples

```
# Load the riboflavin dataset
data(riboflavin)

# Display the dimensions of the dataset
print(dim(riboflavin$x))
print(length(riboflavin$y))

# Summary statistics for the response variable
summary(riboflavin$y)
```

---

riboflavinv100 *Riboflavin Production Data (Top 100 Genes)*

Description

This dataset is a subset of the riboflavin production data by Bacillus subtilis, containing $n = 71$ observations. It includes the response variable (log-transformed riboflavin production rate) and the 100 genes with the largest empirical variances from the original dataset.

Usage

```
data(riboflavinv100)
```

Format

y Log-transformed riboflavin production rate (original name: `q_RIBFLV`). This is a continuous variable indicating the efficiency of riboflavin production by the bacterial strain.

x A matrix of dimension 71×100 containing the logarithm of the expression levels of the 100 genes with the largest empirical variances.

Details

This dataset is derived from the original riboflavin dataset, which contains 4088 gene expressions. The riboflavinV100 dataset is created for ease of reproduction in examples and contains only the 100 genes with the largest empirical variances. It is commonly used in statistical research for high-dimensional data analysis.

Note

The dataset is provided by DSM Nutritional Products Ltd., a leading company in the field of nutritional ingredients. The data have been preprocessed and normalized.

Source

DSM Nutritional Products Ltd., Basel, Switzerland.

References

- Bühlmann, P., Kalisch, M., & Meier, L. (2014). 'High-dimensional statistics with a view towards applications in biology.' *Annual Review of Statistics and its Applications*, **1**, 255–278.
- DSM Nutritional Products Ltd. (2005). 'Genome-scale analysis of Bacillus subtilis riboflavin production.' *Internal Report*.

Examples

```
# Load the riboflavinv100 dataset
data(riboflavinv100)

# Display the dimensions of the dataset
print(dim(riboflavinv100$x))
print(length(riboflavinv100$y))

# Summary statistics for the response variable
summary(riboflavinv100$y)
```

Index

∗ **datasets**
review, 7
riboflavin, 8
riboflavinv100, 9

Depca, 2
Dpca, 3
Drp, 4
Drpca, 4
Drsvd, 5
Dsvd, 6

FPC, 6

review, 7
riboflavin, 8
riboflavinv100, 9