Package ‘FPCdpca’

May 27, 2024

Type Package

Title The FPCdpca Criterion on Distributed Principal Component Analysis

Version 0.1.0

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. (2020) <doi:10.1007/s00180-020-00974-4>.

License Apache License (== 2.0)

Encoding UTF-8

Imports matrixcalc,Rdimtools,rsvd,stats

Suggests testthat (>= 3.0.0)

NeedsCompilation no

Config/testthat/edition 3

Author Guangbao Guo [aut, cre, cph],
Jiarui Li [ctb]

Repository CRAN

Date/Publication 2024-05-27 17:30:05 UTC

R topics documented:

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

Index **8**

Depca *Decentralized PCA*

Description

Decentralized PCA is a technology that applies decentralized PCA to distributed computing environments.

Usage

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

Arguments

`data`	is sparse random projection matrix.
`K`	is the desired target rank.
`nk`	is the size of subsets.
`eps`	is the noise.
`nit.max`	is the repeat times.

Value

MSEXrp,MSEvrp, MSESrp, kopt

Examples

```
K=20; nk=50; nr=10; p=8; k=4; n=K*nk;d=6
data=matrix(c(rnorm((n-nr)*p,0,1),rpois(nr*p,100)),ncol=p)
set.seed(1234)
eps=10^(-1);nit.max=1000
TXde=TSde=c(rep(0,5))
for (j in 1:5){
  depca=Depca(data=data,K=K, nk=nk,eps=eps,nit.max=nit.max)
  TXde[j]=as.numeric(depca)[1]
  TSde[j]=as.numeric(depca)[2]
}
mean(TXde)
mean(TSde)
```

Dpca *Distributed PCA*

Description

Distributed PCA is a technology that applies PCA to distributed computing environments.

Usage

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

Arguments

`data` is the n random vectors constitute the data matrix.

`K` is an index subset/sub-vector specifying.

`nk` is the size of subsets.

Value

MSEXp, MSEvp, MSESp, kopt

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,K,nk)
```

Drp *Distributed random projection*

Description

Distributed random projection is a technology that applies random projection to distributed computing environments.

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.

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 is a technology that applies random PCA to distributed computing environments.

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.

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

Drsvd *Distributed random svd*

Description

Distributed random svd is a technology that applies random SVD to distributed computing environments.

Usage

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

Arguments

data	sparse random projection matrix.
K	the number of distributed nodes.
nk	the size of subsets.
m	the dimension of variables.
q	number of additional power iterations.
k	the desired target rank.

Value

MSEXrsvd	The MSE value of Xrsvd
MSEvrsvd	The MSE value of vrsvd
MSESrsvd	The MSE value of Srsvd
kopt	The size of optimal subset

Examples

```
K=20; nk=50; nr=10; p=8; m=5; q=5;k=4;n=K*nk;
data=X=matrix(rexp(n*p,0.8),ncol=p)
#data=matrix(c(rnorm((n-nr)*p,0,1),rpois(nr*p,100)),ncol=p)
#data=X=matrix(rpois((n-nr)*p,1),ncol=p); rexp(nr*p,1); rchisq(10000, df = 5);
#data=X=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 is a technology that applies SVD to distributed computing environments.

Usage

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

Arguments

data	A independent variable.
K	the number of distributed nodes.
nk	the number of each blocks.
k	the desired target rank.

Value

MSEXs	the MSE of Xs
MSEvsvd	the MSE of vsvd
MSESsvd	the MSE of Ssvd
kopt	the size of optimal subset

Examples

```
#install.packages("matrixcalc")
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 is a technology that applies FPC A to distributed computing environments.

Usage

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

Arguments

`data` is a data set matrix.

`K` is the desired target rank.

`nk` is the size of subsets.

Value

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

Examples

```
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)
```

Index

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

FPC, 6