Package 'DLFM'

October 30, 2025

Type Package

Version 0.1.0

Title Distributed Laplace Factor Model

Description Distributed estimation method is based on a Laplace factor model to solve the estimates of load and specific variance. The philosophy of the package is described in Guangbao Guo. (2022). <doi:10.1007/s00180-022-01270-z>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Imports stats, FarmTest, MASS, LaplacesDemon, matrixcalc, relliptical, LFM

NeedsCompilation no

Language en-US

Author Guangbao Guo [aut, cre], Siqi Liu [aut]

Depends R (>= 3.5.0)

BuildManual yes

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Maintainer Guangbao Guo <ggb11111111@163.com>

Repository CRAN

Date/Publication 2025-10-30 20:10:08 UTC

Contents

DGulPC *Distributed general unilateral loading principal component*

Description

Distributed general unilateral loading principal component

Usage

```
DGulPC(data, m, n1, K)
```

Arguments

data	is a total data set
m	is the number of principal component
n1	is the length of each data subset
K	is the number of nodes

Value

AU1,AU2,DU3,Shat

Examples

```
library(LFM)
data_from_package <- Wine
data_a <- Wine
DGulPC(data_a,m=3,n1=128,K=2)
```

DPC *Distributed principal component*

Description

Distributed principal component

Usage

```
DPC(data, m, n1, K)
```

Arguments

data	is a total data set
m	is the number of principal component
n1	is the length of each data subset
K	is the number of nodes

Value

Ahat,Dhat,Sigmahathat

Examples

```
library(LFM)
data_from_package <- Wine
data_a <- Wine
DPC(data_a,m=3,n1=128,K=2)
```

DPPC *Distributed projection principal component*

Description

Distributed projection principal component

Usage

```
DPPC(data, m, n1, K)
```

Arguments

data	is a total data set
m	is the number of principal component
n1	is the length of each data subset
K	is the number of nodes

Value

Apro,pro,Sigmahathatpro

Examples

```
library(LFM)
data_from_package <- Wine
data_a <- Wine
DPPC(data_a,m=3,n1=128,K=2)
```

FanPC *Apply the FanPC method to the Laplace factor model*

Description

This function performs Factor Analysis via Principal Component (FanPC) on a given data set. It calculates the estimated factor loading matrix (AF), specific variance matrix (DF), and the mean squared errors.

Usage

```
FanPC(data, m)
```

Arguments

data	A matrix of input data.
m	is the number of principal component

Value

AF,DF,SigmahatF

Examples

```
library(LaplacesDemon)
library(MASS)
n=1000
p=10
m=5
mu=t(matrix(rep(runif(p,0,1000),n),p,n))
mu0=as.matrix(runif(m,0))
sigma0=diag(runif(m,1))
F=matrix(mvrnorm(n,mu0,sigma0),nrow=n)
A=matrix(runif(p*m,-1,1),nrow=p)
lanor <- rlaplace(n*p,0,1)
epsilon=matrix(lanor,nrow=n)
D=diag(t(epsilon)%*%epsilon)
data=mu+F%*%t(A)+epsilon
results <- FanPC(data, m)
print(results)
```

Ftest *Apply the Farmtest method to the Laplace factor model*

Description

This function simulates data from a Lapalce factor model and applies the FarmTest for multiple hypothesis testing. It calculates the false discovery rate (FDR) and power of the test.

Usage

```
Ftest(data, p1)
```

Arguments

data	A matrix or data frame of simulated or observed data from a Laplace factor model.
p1	The proportion of non-zero hypotheses.

Value

A list containing the following elements:

FDR	The false discovery rate, which is the proportion of false positives among all discoveries (rejected hypotheses).
Power	The statistical power of the test, which is the probability of correctly rejecting a false null hypothesis.
PValues	A vector of p-values associated with each hypothesis test.
RejectedHypotheses	The total number of hypotheses that were rejected by the FarmTest.

Examples

```
library(LaplacesDemon)
library(MASS)
n=1000
p=10
m=5
mu=t(matrix(rep(runif(p,0,1000),n),p,n))
mu0=as.matrix(runif(m,0))
sigma0=diag(runif(m,1))
F=matrix(mvrnorm(n,mu0,sigma0),nrow=n)
A=matrix(runif(p*m,-1,1),nrow=p)
lanor <- rlaplace(n*p,0,1)
epsilon=matrix(lanor,nrow=n)
D=diag(t(epsilon)%*%epsilon)
data=mu+F%*%t(A)+epsilon
p1=40
results <- Ftest(data, p1)
```

```
print(results$FDR)
print(results$Power)
```

GulPC *General unilateral loading principal component*

Description

General unilateral loading principal component

Usage

```
GulPC(data, m)
```

Arguments

data	is a total data set
m	is the number of first layer principal component

Value

AU1,AU2,DU3,SigmaUhat

Examples

```
library(LFM)
data_from_package <- Wine
data_a <- Wine
GulPC(data=data_a,m=5)
```

LFM *Generate Laplace factor models*

Description

The function is to generate Laplace factor model data. The function supports various distribution types for generating the data, including: - 'truncated_laplace': Truncated Laplace distribution - 'log_laplace': Univariate Symmetric Log-Laplace distribution - 'Asymmetric Log_Laplace': Log-Laplace distribution - 'Skew-Laplace': Skew-Laplace distribution

Usage

```
LFM(n, p, m, distribution_type)
```

Arguments

n	An integer specifying the sample size.
p	An integer specifying the sample dimensionality or the number of variables.
m	An integer specifying the number of factors in the model.
distribution_type	A character string indicating the type of distribution to use for generating the data.

Value

A list containing the following elements:

data	A numeric matrix of the generated data.
A	A numeric matrix representing the factor loadings.
D	A numeric matrix representing the uniquenesses, which is a diagonal matrix.

Examples

```
library(MASS)
library(matrixcalc)
library(relliptical)
n <- 1000
p <- 10
m <- 5
sigma1 <- 1
sigma2 <- matrix(c(1,0.7,0.7,1), 2, 2)
distribution_type <- "truncated_laplace"
results <- LFM(n, p, m, distribution_type)
print(results)
```

PC *Principal component*

Description

Principal component

Usage

```
PC(data, m)
```

Arguments

data	is a total data set
m	is the number of principal component

Value

Ahat, Dhat, Sigmahat

Examples

```
library(LFM)
data_from_package <- Wine
data_a <- Wine
PC(data_a,m=5)
```

PPC *Projection principal component*

Description

Projection principal component

Usage

```
PPC(data, m)
```

Arguments

data	is a total data set
m	is the number of principal component

Value

Apro, Dpro, Sigmahatpro

Examples

```
library(LFM)
data_from_package <- Wine
data_a <- Wine
PPC(data=data_a,m=5)
```

Index