Package ‘SFM’

March 5, 2025

Type Package

Title A Package for Analyzing Skew Factor Models

Version 0.2.0

Description
Generates Skew Factor Models data and applies Sparse Online Principal Component (SOPC), Incremental Principal Component (IPC), Projected Principal Component (PPC), Perturbation Principal Component (PPC), Stochastic Approximation Principal Component (SAPC), Sparse Principal Component (SPC) and other PC methods to estimate model parameters. It includes capabilities for calculating mean squared error, relative error, and sparsity of the loading matrix.The philosophy of the package is described in Guo G. (2023) <doi:10.1007/s00180-022-01270-z>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Imports MASS, SOPC, matrixcalc, sn, stats,psych

NeedsCompilation no

Language en-US

Author Guangbao Guo [aut, cre],
Yu Jin [aut]

Maintainer Guangbao Guo <ggb11111111@163.com>

Suggests testthat (>= 3.0.0), ggplot2

Repository CRAN

Date/Publication 2025-03-05 13:50:07 UTC

Contents

---

calculate_errors *calculate_errors Function*

Description

This function calculates the Mean Squared Error (MSE) and relative error for factor loadings and uniqueness estimates obtained from factor analysis.

Usage

```
calculate_errors(data, A, D)
```

Arguments

`data`	Matrix of SFM data.
`A`	Matrix of true factor loadings.
`D`	Matrix of true uniquenesses.

Value

A named vector containing:

`MSEA`	Mean Squared Error for factor loadings.
`MSED`	Mean Squared Error for uniqueness estimates.
`LSA`	Relative error for factor loadings.
`LSD`	Relative error for uniqueness estimates.

Examples

```
set.seed(123) # For reproducibility
# Define dimensions
n <- 10  # Number of samples
p <- 5   # Number of factors

# Generate matrices with compatible dimensions
A <- matrix(runif(p * p, -1, 1), nrow = p)  # Factor loadings matrix (p x p)
D <- diag(runif(p, 1, 2))  # Uniquenesses matrix (p x p)
data <- matrix(runif(n * p), nrow = n)  # Data matrix (n x p)
```

```
# Calculate errors
errors <- calculate_errors(data, A, D)
print(errors)
```

FanPC.SFM *Apply the FanPC method to the Skew 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.SFM(data, m, A, D, p)
```

Arguments

data	A matrix of input data.
m	The number of principal components.
A	The true factor loadings matrix.
D	The true uniquenesses matrix.
p	The number of variables.

Value

A list containing:

AF	Estimated factor loadings.
DF	Estimated uniquenesses.
MSESigmaA	Mean squared error for factor loadings.
MSESigmaD	Mean squared error for uniquenesses.
LSigmaA	Loss metric for factor loadings.
LSigmaD	Loss metric for uniquenesses.

Examples

```
library(SOPC)
library(matrixcalc)
library(MASS)
library(sn)
library(psych)
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)
r <- rsn(n*p,0,1)
epsilon=matrix(r,nrow=n)
D=diag(t(epsilon)%*%epsilon)
data=mu+F%*%t(A)+epsilon
results <- FanPC.SFM(data, m, A, D, p)
print(results)
```

GulPC.SFM *Apply the GulPC method to the Skew factor model*

Description

This function performs General Unilateral Loading Principal Component (GulPC) analysis on a given data set. It calculates the estimated values for the first layer and second layer loadings, specific variances, and the mean squared errors.

Usage

```
GulPC.SFM(data, m, A, D)
```

Arguments

data	A matrix of input data.
m	The number of principal components.
A	The true factor loadings matrix.
D	The true uniquenesses matrix.

Value

A list containing:

AU1	The first layer loading matrix.
AU2	The second layer loading matrix.
DU3	The estimated specific variance matrix.
MSESigmaD	Mean squared error for uniquenesses.
LSigmaD	Loss metric for uniquenesses.

Examples

```
library(SOPC)
library(matrixcalc)
library(MASS)
library(psych)
library(sn)
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)
r <- rsn(n*p,0,1)
epsilon=matrix(r,nrow=n)
D=diag(t(epsilon)%*%epsilon)
data=mu+F%*%t(A)+epsilon
results <- GulPC.SFM(data, m, A, D)
print(results)
```

IPC.SFM *Apply the IPC method to the Skew factor model*

Description

This function performs Incremental Principal Component Analysis (IPC) on the provided data. It updates the estimated factor loadings and uniquenesses as new data points are processed, calculating mean squared errors and loss metrics for comparison with true values.

Usage

```
IPC.SFM(x, m, A, D, p)
```

Arguments

x	The data used in the IPC analysis.
m	The number of common factors.
A	The true factor loadings matrix.
D	The true uniquenesses matrix.
p	The number of variables.

Value

A list of metrics including:

Ai	Estimated factor loadings updated during the IPC analysis, a matrix of estimated factor loadings.

Di	Estimated uniquenesses updated during the IPC analysis, a vector of estimated uniquenesses corresponding to each variable.
MSESigmaA	Mean squared error of the estimated factor loadings (Ai) compared to the true loadings (A).
MSESigmaD	Mean squared error of the estimated uniquenesses (Di) compared to the true uniquenesses (D).
LSigmaA	Loss metric for the estimated factor loadings (Ai), indicating the relative error compared to the true loadings (A).
LSigmaD	Loss metric for the estimated uniquenesses (Di), indicating the relative error compared to the true uniquenesses (D).

Examples

```
library(SOPC)
library(matrixcalc)
library(MASS)
library(psych)
library(sn)
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)
r <- rsn(n*p,0,1)
epsilon=matrix(r,nrow=n)
D=diag(t(epsilon)%*%epsilon)
data=mu+F%*%t(A)+epsilon
result <- IPC.SFM(data, m = m, A = A, D = D, p = p)
print(result)
```

OPC.SFM *Apply the OPC method to the Skew factor model*

Description

This function computes Online Principal Component Analysis (OPC) for the provided input data, estimating factor loadings and uniquenesses. It calculates mean squared errors and sparsity for the estimated values compared to true values.

Usage

```
OPC.SFM(data, m = m, A, D, p)
```

Arguments

data	A matrix of input data.
m	The number of principal components.
A	The true factor loadings matrix.
D	The true uniquenesses matrix.
p	The number of variables.

Value

A list containing:

Ao	Estimated factor loadings.
Do	Estimated uniquenesses.
MSEA	Mean squared error for factor loadings.
MSED	Mean squared error for uniquenesses.
tau	The sparsity.

Examples

```
library(SOPC)
library(matrixcalc)
library(MASS)
library(psych)
library(sn)
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)
r <- rsn(n*p,0,1)
epsilon=matrix(r,nrow=n)
D=diag(t(epsilon)%*%epsilon)
data=mu+F%*%t(A)+epsilon
results <- OPC.SFM(data, m, A, D, p)
print(results)
```

PC1.SFM *Apply the PC method to the Laplace factor model*

Description

This function performs Principal Component Analysis (PCA) on a given data set to reduce dimensionality. It calculates the estimated values for the loadings, specific variances, and the covariance matrix.

Usage

```
PC1.SFM(data, m, A, D)
```

Arguments

data	The total data set to be analyzed.
m	The number of principal components to retain in the analysis.
A	The true factor loadings matrix.
D	The true uniquenesses matrix.

Value

A list containing:

A1	Estimated factor loadings.
D1	Estimated uniquenesses.
MSESigmaA	Mean squared error for factor loadings.
MSESigmaD	Mean squared error for uniquenesses.
LSigmaA	Loss metric for factor loadings.
LSigmaD	Loss metric for uniquenesses.

Examples

```
library(SOPC)
library(matrixcalc)
library(MASS)
library(psych)
library(sn)
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)
r <- rsn(n*p,0,1)
epsilon=matrix(r,nrow=n)
D=diag(t(epsilon)%*%epsilon)
data=mu+F%*%t(A)+epsilon
results <- PC1.SFM(data, m, A, D)
print(results)
```

PC2.SFM *Apply the PC method to the Laplace factor model*

Description

This function performs Principal Component Analysis (PCA) on a given data set to reduce dimensionality. It calculates the estimated values for the loadings, specific variances, and the covariance matrix.

Usage

```
PC2.SFM(data, m, A, D)
```

Arguments

data	The total data set to be analyzed.
m	The number of principal components to retain in the analysis.
A	The true factor loadings matrix.
D	The true uniquenesses matrix.

Value

A list containing:

A2	Estimated factor loadings.
D2	Estimated uniquenesses.
MSESigmaA	Mean squared error for factor loadings.
MSESigmaD	Mean squared error for uniquenesses.
LSigmaA	Loss metric for factor loadings.
LSigmaD	Loss metric for uniquenesses.

Examples

```
library(SOPC)
library(matrixcalc)
library(MASS)
library(psych)
library(sn)
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)
```

```
r <- rsn(n*p,0,1)
epsilon=matrix(r,nrow=n)
D=diag(t(epsilon)%*%epsilon)
data=mu+F%*%t(A)+epsilon
results <- PC2.SFM(data, m, A, D)
print(results)
```

PPC1.SFM *Apply the PPC method to the Skew factor model*

Description

This function computes Perturbation Principal Component Analysis (PPC) for the provided input data, estimating factor loadings and uniquenesses. It calculates mean squared errors and loss metrics for the estimated values compared to true values.

Usage

```
PPC1.SFM(data, m, A, D, p)
```

Arguments

`data`	A matrix of input data.
`m`	The number of principal components.
`A`	The true factor loadings matrix.
`D`	The true uniquenesses matrix.
`p`	The number of variables.

Value

A list containing:

`Ap`	Estimated factor loadings.
`Dp`	Estimated uniquenesses.
`MSESigmaA`	Mean squared error for factor loadings.
`MSESigmaD`	Mean squared error for uniquenesses.
`LSigmaA`	Loss metric for factor loadings.
`LSigmaD`	Loss metric for uniquenesses.

Examples

```
library(SOPC)
library(matrixcalc)
library(MASS)
library(psych)
library(sn)
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)
r <- rsn(n*p,0,1)
epsilon=matrix(r,nrow=n)
D=diag(t(epsilon)%*%epsilon)
data=mu+F%*%t(A)+epsilon
results <- PPC1.SFM(data, m, A, D, p)
print(results)
```

PPC2.SFM *Apply the PPC method to the Skew factor model*

Description

This function performs Projected Principal Component Analysis (PPC) on a given data set to reduce dimensionality. It calculates the estimated values for the loadings, specific variances, and the covariance matrix.

Usage

```
PPC2.SFM(data, m, A, D)
```

Arguments

data	The total data set to be analyzed.
m	The number of principal components.
A	The true factor loadings matrix.
D	The true uniquenesses matrix.

Value

A list containing:

Ap2	Estimated factor loadings.
Dp2	Estimated uniquenesses.

MSESigmaA	Mean squared error for factor loadings.
MSESigmaD	Mean squared error for uniquenesses.
LSigmaA	Loss metric for factor loadings.
LSigmaD	Loss metric for uniquenesses.

Examples

```
library(SOPC)
library(matrixcalc)
library(MASS)
library(psych)
library(sn)
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)
r <- rsn(n*p,0,1)
epsilon=matrix(r,nrow=n)
D=diag(t(epsilon)%*%epsilon)
data=mu+F%*%t(A)+epsilon
results <- PPC2.SFM(data, m, A, D)
print(results)
```

SAPC.SFM *Stochastic Approximation Principal Component Analysis*

Description

This function calculates several metrics for the SAPC method, including the estimated factor loadings and uniquenesses, and various error metrics comparing the estimated matrices with the true matrices.

Usage

```
SAPC.SFM(x, m, A, D, p)
```

Arguments

x	The data used in the SAPC analysis.
m	The number of common factors.
A	The true factor loadings matrix.
D	The true uniquenesses matrix.
p	The number of variables.

Value

A list of metrics including:

- `Asa` — Estimated factor loadings matrix obtained from the SAPC analysis.
- `Dsa` — Estimated uniquenesses vector obtained from the SAPC analysis.
- `MSESigmaA` — Mean squared error of the estimated factor loadings (Asa) compared to the true loadings (A).
- `MSESigmaD` — Mean squared error of the estimated uniquenesses (Dsa) compared to the true uniquenesses (D).
- `LSigmaA` — Loss metric for the estimated factor loadings (Asa), indicating the relative error compared to the true loadings (A).
- `LSigmaD` — Loss metric for the estimated uniquenesses (Dsa), indicating the relative error compared to the true uniquenesses (D).

Examples

```
p = 10
m = 5
n = 2000
mu = t(matrix(rep(runif(p, 0, 100), n), p, n))
mu0 = as.matrix(runif(m, 0))
sigma0 = diag(runif(m, 1))
F = matrix(MASS::mvrnorm(n, mu0, sigma0), nrow = n)
A = matrix(runif(p * m, -1, 1), nrow = p)
xi = 5
omega = 2
alpha = 5
r <- sn::rsn(n * p, omega = omega, alpha = alpha)
D0 = omega * diag(p)
D = diag(D0)
epsilon = matrix(r, nrow = n)
data = mu + F %*% t(A) + epsilon

result <- SAPC.SFM(data, m = m, A = A, D = D, p = p)
print(result)
```

SFM — *The SFM function is to generate Skew Factor Models data.*

Description

The function supports various distribution types for generating the data, including: Skew-Normal Distribution, Skew-Cauchy Distribution, Skew-t Distribution.

Usage

```
SFM(n, p, m, xi, omega, alpha, distribution_type)
```

Arguments

`n`	Sample size.
`p`	Sample dimensionality.
`m`	Number of factors.
`xi`	A numerical parameter used exclusively in the "Skew-t" distribution, representing the distribution's xi parameter.
`omega`	A numerical parameter representing the omega parameter of the distribution, which affects the degree of skewness in the distribution.
`alpha`	A numerical parameter representing the alpha parameter of the distribution, which influences the shape of the distribution.
`distribution_type`	The type of distribution.

Value

A list containing:

`data`	A matrix of generated data.
`A`	A matrix representing the factor loadings.
`D`	A diagonal matrix representing the unique variances.

Examples

```
library(MASS)
library(SOPC)
library(sn)
library(matrixcalc)
library(psych)
n <- 100
p <- 10
m <- 5
xi <- 5
omega <- 2
alpha <- 5
distribution_type <- "Skew-Normal Distribution"
X <- SFM(n, p, m, xi, omega, alpha, distribution_type)
```

SOPC.SFM *SOPC Estimation Function*

Description

This function processes Skew Factor Model (SFM) data using the Sparse Online Principal Component (SOPC) method.

Usage

```
SOPC.SFM(data, m, p, A, D)
```

Arguments

`data`	A numeric matrix containing the data used in the SOPC analysis.
`m`	An integer specifying the number of subsets or common factors.
`p`	An integer specifying the number of variables in the data.
`A`	A numeric matrix representing the true factor loadings.
`D`	A numeric matrix representing the true uniquenesses.

Value

A list containing the following metrics:

`Aso`	Estimated factor loadings matrix.
`Dso`	Estimated uniquenesses matrix.
`MSEA`	Mean squared error of the estimated factor loadings (Aso) compared to the true loadings (A).
`MSED`	Mean squared error of the estimated uniquenesses (Dso) compared to the true uniquenesses (D).
`LSA`	Loss metric for the estimated factor loadings (Aso), indicating the relative error compared to the true loadings (A).
`LSD`	Loss metric for the estimated uniquenesses (Dso), indicating the relative error compared to the true uniquenesses (D).
`tauA`	Proportion of zero factor loadings in the estimated loadings matrix (Aso), representing the sparsity.

Examples

```
library(SOPC)
library(matrixcalc)
library(MASS)
library(psych)
library(sn)
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)
r <- rsn(n*p,0,1)
epsilon=matrix(r,nrow=n)
D=diag(t(epsilon)%*%epsilon)
data=mu+F%*%t(A)+epsilon
results <- SOPC.SFM(data, m, p, A, D)
print(results)
```

SPC.SFM *Apply the SPC method to the Skew factor model*

Description

This function performs Sparse Principal Component Analysis (SPC) on the input data. It estimates factor loadings and uniquenesses while calculating mean squared errors and loss metrics for comparison with true values.

Usage

```
SPC.SFM(data, A, D, m, p)
```

Arguments

`data`	The data used in the SPC analysis.
`A`	The true factor loadings matrix.
`D`	The true uniquenesses matrix.
`m`	The number of common factors.
`p`	The number of variables.

Value

A list containing:

`As`	Estimated factor loadings, a matrix of estimated factor loadings from the SPC analysis.
`Ds`	Estimated uniquenesses, a vector of estimated uniquenesses corresponding to each variable.
`MSESigmaA`	Mean squared error of the estimated factor loadings (As) compared to the true loadings (A).
`MSESigmaD`	Mean squared error of the estimated uniquenesses (Ds) compared to the true uniquenesses (D).
`LSigmaA`	Loss metric for the estimated factor loadings (As), indicating the relative error compared to the true loadings (A).
`LSigmaD`	Loss metric for the estimated uniquenesses (Ds), indicating the relative error compared to the true uniquenesses (D).
`tau`	Proportion of zero factor loadings in the estimated loadings matrix (As).

Examples

```
library(SOPC)
library(matrixcalc)
library(MASS)
library(psych)
library(sn)
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)
r <- rsn(n*p,0,1)
epsilon=matrix(r,nrow=n)
D=diag(t(epsilon)%*%epsilon)
data=mu+F%*%t(A)+epsilon
results <- SPC.SFM(data, A, D, m, p)
print(results)
```

Index