Package 'SFM'

November 12, 2024

Type Package

Title A Package for Analyzing Skew Factor Models

Version 0.1.0

Description Generates Skew Factor Models data and applies Sparse Online Principal Component (SOPC) method to estimate model parameters. It includes capabilities for calculating mean squared error, relative error, and sparsity of the loading matrix. Additionally, it includes robust regression methods such as adaptive Huber regression.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

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, reshape2

Repository CRAN

Date/Publication 2024-11-12 13:10:02 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)
```

`huber.reg.adaptive.skew` *Adaptive Huber Regression for Skew Factor Models*

Description

Performs adaptive Huber regression tailored for skew factor models, and returns the estimated regression coefficients in a matrix (loading matrix) format.

Usage

```
huber.reg.adaptive.skew(
  X,
  Y,
  tau = 1.35,
  max_iterations = 100,
  tolerance = 1e-06,
  n_factors = 1
)
```

Arguments

`X`	A matrix of predictor variables.
`Y`	A vector of response variables.
`tau`	Initial robustification parameter (default is 1.35).
`max_iterations`	Maximum number of iterations (default is 100).
`tolerance`	Convergence tolerance (default is 1e-6).
`n_factors`	The number of factors (columns) for the loading matrix (default is 1).

Value

A matrix of estimated regression coefficients with dimensions ‘p x n_factors‘.

Examples

```
# Generate some example data for skew factor models
set.seed(123)
n <- 200
d <- 10
beta <- rep(1, d)
skew_factor <- rnorm(n)  # Adding a skew factor
X <- matrix(rnorm(n * d), n, d)
err <- rnorm(n)
Y <- 1 + skew_factor + X %*% beta + err

# Perform adaptive Huber regression for skew factor model
loading_matrix <- huber.reg.adaptive.skew(X, Y, n_factors = 3)
```

```
print(loading_matrix)
```

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)
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_estimation *SOPC Estimation Function*

Description

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

Usage

```
SOPC_estimation(data, gamma, eta)
```

Arguments

data	Matrix of SFM data.
gamma	Tuning parameter for the sparseness of the loadings matrix.
eta	Tuning parameter for the sparseness of the common factors matrix.

Value

A list containing:

Aso	Estimated factor loadings.
Dso	Estimated common factors .
tauA	Sparsity of the loadings matrix, calculated as the proportion of zeros.

Examples

```
set.seed(123) # For reproducibility
data <- matrix(runif(200), nrow = 20) # Skew Factor Model data
sopc_results <- SOPC_estimation(data, 0.1, 0.8)
print(sopc_results)
```

Index