Package 'MFSIS'

March 18, 2025

Type Package

Title Model-Free Sure Independent Screening Procedures

Version 0.3.0

Date 2025-03-18

Author Xuewei Cheng [aut, cre],
Hong Wang [aut],
Liping Zhu [aut],
Wei Zhong [aut],
Hanpu Zhou [aut]

Maintainer Xuewei Cheng <xwcheng@hunnu.edu.cn>

Description An implementation of popular screening methods that are commonly employed in ultra-high and high dimensional data.
Through this publicly available package, we provide a unified framework to carry out model-free screening procedures including
SIS (Fan and Lv (2008) <doi:10.1111/j.1467-9868.2008.00674.x>),
SIRS (Zhu et al. (2011)<doi:10.1198/jasa.2011.tm10563>),
DC-SIS (Li et al. (2012) <doi:10.1080/01621459.2012.695654>),
MDC-SIS (Shao and Zhang (2014) <doi:10.1080/01621459.2014.887012>),
Bcor-SIS (Pan et al. (2019) <doi:10.1080/01621459.2018.1462709>),
PC-Screen (Liu et al. (2020) <doi:10.1080/01621459.2020.1783274>),
WLS (Zhong et al.(2021) <doi:10.1080/01621459.2021.1918554>),
Kfilter (Mai and Zou (2015) <doi:10.1214/14-AOS1303>),
MVSIS (Cui et al. (2015) <doi:10.1080/01621459.2014.920256>),
PSIS (Pan et al. (2016) <doi:10.1080/01621459.2014.998760>),
CAS (Xie et al. (2020) <doi:10.1080/01621459.2019.1573734>),
CI-SIS (Cheng and Wang. (2023) <doi:10.1016/j.cmpb.2022.107269>) and
CSIS (Cheng et al. (2023) <doi:10.1007/s00180-023-01399-5>).

License GPL (>= 2)

SystemRequirements Python (>= 3.8.0)

Encoding UTF-8

Imports survival, MASS, Ball, reticulate, stats, crayon, cli, dr, foreach, parallel, doParallel

Suggests knitr, utils, pkgdown

NeedsCompilation no

Repository CRAN

RoxygenNote 7.3.1

Date/Publication 2025-03-18 11:50:07 UTC

Contents

BcorSIS *A Generic Sure Independence Screening Procedure*

Description

A generic nonparametric sure independence screening procedure, called BCor-SIS, on the basis of a recently developed universal dependence measure: Ball correlation. This procedure has strong screening consistency even when the dimensionality is an exponential order of the sample size without imposing sub-exponential moment assumptions on the data.

Usage

```
BcorSIS(X, Y, nsis = (dim(X)[1])/log(dim(X)[1]))
```

Arguments

X	The design matrix of dimensions n * p. Each row is an observation vector.
Y	The response vector of dimension n * 1. For survival models, Y should be an object of class Surv, as provided by the function Surv() in the package survival.
nsis	Number of predictors recruited by BcorSIS. The default is n/log(n).

Value

the labels of first nsis largest active set of all predictors

Author(s)

Xuewei Cheng <xwcheng@hunnu.edu.cn>

References

Pan, W., X. Wang, H. Zhang, H. Zhu, and J. Zhu (2020). Ball covariance: A generic measure of dependence in banach space. Journal of the American Statistical Association 115(529),307–317.

Pan, W., X. Wang, W. Xiao, and H. Zhu (2019). A generic sure independence screening procedure. Journal of the American Statistical Association 114(526), 928–937.

Examples

```
## Scenario 1  generate complete data
n <- 100
p <- 200
rho <- 0.5
data <- GendataLM(n, p, rho, error = "gaussian")
data <- cbind(data[[1]], data[[2]])
colnames(data)[1:ncol(data)] <- c(paste0("X", 1:(ncol(data) - 1)), "Y")
data <- as.matrix(data)
X <- data[, 1:(ncol(data) - 1)]
Y <- data[, ncol(data)]
```

```
A1 <- BcorSIS(X, Y, n / log(n))
A1

## Scenario 2  generate survival data
library(survival)
n <- 100
p <- 200
rho <- 0.5
data <- GendataCox(n, p, rho)
data <- cbind(data[[1]], data[[2]], data[[3]])
colnames(data)[ncol(data)] <- c("status")
colnames(data)[(ncol(data) - 1)] <- c("time")
colnames(data)[(1:(ncol(data) - 2))] <- c(paste0("X", 1:(ncol(data) - 2)))
data <- as.matrix(data)
X <- data[, 1:(ncol(data) - 2)]
Y <- Surv(data[, (ncol(data) - 1)], data[, ncol(data)])
A2 <- BcorSIS(X, Y, n / log(n))
A2
```

CAS *Category-Adaptive Variable Screening for Ultra-High Dimensional Heterogeneous Categorical Data*

Description

A category-adaptive screening procedure with high-dimensional heterogeneous data, which is to detect category-specific important covariates. This proposal is a model-free approach without any specification of a regression model and an adaptive procedure in the sense that the set of active variables is allowed to vary across different categories, thus making it more flexible to accommodate heterogeneity.

Usage

```
CAS(X, Y, nsis)
```

Arguments

X	The design matrix of dimensions n * p. Each row is an observation vector.
Y	The response vector of dimension n * 1.
nsis	Number of predictors recruited by CAS. The default is n/log(n).

Value

the labels of first nsis largest active set of all predictors

Author(s)

Xuewei Cheng <xwcheng@hunnu.edu.cn>

References

Pan, R., Wang, H., and Li, R. (2016). Ultrahigh-dimensional multiclass linear discriminant analysis by pairwise sure independence screening. Journal of the American Statistical Association, 111(513):169–179.

Examples

```
n <- 100
p <- 200
rho <- 0.5
data <- GendataLGM(n, p, rho)
data <- cbind(data[[1]], data[[2]])
colnames(data)[1:ncol(data)] <- c(paste0("X", 1:(ncol(data) - 1)), "Y")
data <- as.matrix(data)
X <- data[, 1:(ncol(data) - 1)]
Y <- data[, ncol(data)]
A <- CAS(X, Y, n / log(n))
A
```

CISIS *Model-Free Feature screening Based on Concordance Index for Ultra-High Dimensional Categorical Data*

Description

The proposed method is based on the concordance index which measures concordance between random vectors. A model-free and robust feature screening method for ultrahigh-dimensional categorical data. The performance is quite robust in the presence of heavy-tailed error distributions, extremely unbalance responses, and category-adaptive data.

Usage

```
CISIS(X, Y, nsis)
```

Arguments

X	The design matrix of dimensions n * p. Each row is an observation vector.
Y	The response vector of dimension n * 1.
nsis	Number of predictors recruited by CISIS. The default is n/log(n).

Value

the labels of first nsis largest active set of all predictors

Author(s)

Xuewei Cheng <xwcheng@hunnu.edu.cn>

References

Cheng X, Wang H. A generic model-free feature screening procedure for ultra-high dimensional data with categorical response[J]. Computer Methods and Programs in Biomedicine, 2023, 229: 107269.

Examples

```
n <- 100
p <- 200
rho <- 0.5
data <- GendataLGM(n, p, rho)
data <- cbind(data[[1]], data[[2]])
colnames(data)[1:ncol(data)] <- c(paste0("X", 1:(ncol(data) - 1)), "Y")
data <- as.matrix(data)
X <- data[, 1:(ncol(data) - 1)]
Y <- data[, ncol(data)]
A <- CISIS(X, Y, n / log(n))
A
```

Cor *Parallel function This is a parallel function about the projection correlation.*

Description

Parallel function This is a parallel function about the projection correlation.

Usage

```
Cor(Xj, A_y, n)
```

Arguments

Xj	Each column from design matrix of dimensions n * p
A_y	The arccos value about Y
n	The sample size

Value

the projection correlation between Xj and A_y

CSIS *Model-Free Feature screening Based on Concordance Index Statistic*

Description

A model-free and data-adaptive feature screening method for ultrahigh-dimensional data and even survival data. The proposed method is based on the concordance index which measures concordance between random vectors even if one of the vectors is a survival object Surv. This rank correlation based method does not require specifying a regression model, and applies robustly to data in the presence of censoring and heavy tails. It enjoys both sure screening and rank consistency properties under weak assumptions.

Usage

```
CSIS(X, Y, nsis = (dim(X)[1])/log(dim(X)[1]))
```

Arguments

X	The design matrix of dimensions n * p. Each row is an observation vector.
Y	The response vector of dimension n * 1. For survival models, Y should be an object of class Surv, as provided by the function Surv() in the package survival.
nsis	Number of predictors recruited by CSIS. The default is n/log(n).

Value

the labels of first nsis largest active set of all predictors

Author(s)

Xuewei Cheng <xwcheng@hunnu.edu.cn>

References

Cheng X, Li G, Wang H. The concordance filter: an adaptive model-free feature screening procedure[J]. Computational Statistics, 2023: 1-24.

Examples

```
## Scenario 1  generate complete data
n <- 100
p <- 200
rho <- 0.5
data <- GendataLM(n, p, rho, error = "gaussian")
data <- cbind(data[[1]], data[[2]])
colnames(data)[1:ncol(data)] <- c(paste0("X", 1:(ncol(data) - 1)), "Y")
data <- as.matrix(data)
X <- data[, 1:(ncol(data) - 1)]
Y <- data[, ncol(data)]
```

```
A1 <- CSIS(X, Y, n / log(n))
A1

## Scenario 2  generate survival data
library(survival)
n <- 100
p <- 200
rho <- 0.5
data <- GendataCox(n, p, rho)
data <- cbind(data[[1]], data[[2]], data[[3]])
colnames(data)[ncol(data)] <- c("status")
colnames(data)[(ncol(data) - 1)] <- c("time")
colnames(data)[(1:(ncol(data) - 2))] <- c(paste0("X", 1:(ncol(data) - 2)))
data <- as.matrix(data)
X <- data[, 1:(ncol(data) - 2)]
Y <- Surv(data[, (ncol(data) - 1)], data[, ncol(data)])
A2 <- CSIS(X, Y, n / log(n))
A2
```

DCSIS *Feature Screening via Distance Correlation Learning*

Description

A sure independence screening procedure based on the distance correlation (DC-SIS). The DC-SIS can be implemented as easily as the sure independence screening (SIS) procedure based on the Pearson correlation proposed by Fan and Lv(2008). DC-SIS can be used directly to screen grouped predictor variables and multivariate response variables.

Usage

```
DCSIS(X, Y, nsis = (dim(X)[1])/log(dim(X)[1]))
```

Arguments

X	The design matrix of dimensions n * p. Each row is an observation vector.
Y	The response vector of dimension n * 1.
nsis	Number of predictors recruited by DCSIS. The default is n/log(n).

Value

the labels of first nsis largest active set of all predictors

Author(s)

Xuewei Cheng <xwcheng@hunnu.edu.cn>

References

Fan, J. and J. Lv (2008). Sure independence screening for ultrahigh dimensional feature space. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 70(5),849–911.

Li, R., W. Zhong, and L. Zhu (2012). Feature screening via distance correlation learning. Journal of the American Statistical Association 107(499), 1129–1139.

Examples

```
n <- 100
p <- 200
rho <- 0.5
data <- GendataLM(n, p, rho, error = "gaussian")
data <- cbind(data[[1]], data[[2]])
colnames(data)[1:ncol(data)] <- c(paste0("X", 1:(ncol(data) - 1)), "Y")
data <- as.matrix(data)
X <- data[, 1:(ncol(data) - 1)]
Y <- data[, ncol(data)]
A <- DCSIS(X, Y, n / log(n))
A
```

`GendataAFT` *Generate simulation data (Survival data based on the accelerated failure time model)*

Description

This function helps you quickly generate simulation data based on the AFT model. You just need to input the sample and dimension of the data you want to generate and the covariance parameter rho.

Usage

```
GendataAFT(
  n,
  p,
  rho,
  beta = c(rep(1, 5), rep(0, p - 5)),
  lambda = 0.1,
  error = "gaussian"
)
```

Arguments

`n`	Number of subjects in the dataset to be simulated. It will also equal to the number of rows in the dataset to be simulated, because it is assumed that each row represents a different independent and identically distributed subject.

p	Number of predictor variables (covariates) in the simulated dataset. These covariates will be the features screened by model-free procedures.
rho	The correlation between adjacent covariates in the simulated matrix X. The within-subject covariance matrix of X is assumed to has the same form as an AR(1) auto-regressive covariance matrix, although this is not meant to imply that the X covariates for each subject are in fact a time series. Instead, it is just used as an example of a parsimonious but nontrivial covariance structure. If rho is left at the default of zero, the X covariates will be independent and the simulation will run faster.
beta	A vector with length of n, which are the coefficients that you want to generate about Linear model. The default is beta=(1,1,1,1,1,0,...,0)^T;
lambda	This parameter control the censoring rate in survival data. The censored time is generated by exponential distribution with mean 1/lambda. The default is lambda=0.1.
error	The distribution of error term.

Value

the list of your simulation data

Author(s)

Xuewei Cheng <xwcheng@hunnu.edu.cn>

References

Wei LJ (1992). "The accelerated failure time model: a useful alternative to the Cox regression model in survival analysis." Statistics in medicine, 11(14-15), 1871–1879.

Examples

```
n <- 100
p <- 200
rho <- 0.5
data <- GendataAFT(n, p, rho)
```

GendataCox *Generate simulation data (Survival data based on the Cox model)*

Description

This function helps you quickly generate simulation data based on the Cox model. You just need to input the sample and dimension of the data you want to generate and the covariance parameter rho.

Usage

```
GendataCox(n, p, rho, beta = c(rep(1, 5), rep(0, p - 5)), lambda = 0.1)
```

Arguments

n	Number of subjects in the dataset to be simulated. It will also equal to the number of rows in the dataset to be simulated, because it is assumed that each row represents a different independent and identically distributed subject.
p	Number of predictor variables (covariates) in the simulated dataset. These covariates will be the features screened by model-free procedures.
rho	The correlation between adjacent covariates in the simulated matrix X. The within-subject covariance matrix of X is assumed to has the same form as an AR(1) auto-regressive covariance matrix, although this is not meant to imply that the X covariates for each subject are in fact a time series. Instead, it is just used as an example of a parsimonious but nontrivial covariance structure. If rho is left at the default of zero, the X covariates will be independent and the simulation will run faster.
beta	A vector with length of n, which are the coefficients that you want to generate about Linear model. The default is beta=(1,1,1,1,1,0,...,0)^T;
lambda	This parameter control the censoring rate in survival data. The censored time is generated by exponential distribution with mean 1/lambda. The default is lambda=0.1.

Value

the list of your simulation data

Author(s)

Xuewei Cheng <xwcheng@hunnu.edu.cn>

References

Cox DR (1972). "Regression models and life-tables." Journal of the Royal Statistical Society:Series B (Methodological), 34(2), 187–202.

Examples

```
n <- 100
p <- 200
rho <- 0.5
data <- GendataCox(n, p, rho)
```

GendataGP *Generate simulation data (Complete data with group predictors)*

Description

In many regression problems, some predictors may be naturally grouped. The most common example that contains group variables is the multi-factor analysis of variance (ANOVA) problem, where each factor may have several levels and can be expressed through a group of dummy variables. This function helps you quickly generate simulation data with group predictors. You just need to input the sample and dimension of the data you want to generate and the covariance parameter rho. This simulated example comes from Example 2 introduced by Li et al.(2012)

Usage

```
GendataGP(n, p, rho, error = c("gaussian", "t", "cauchy"))
```

Arguments

n	Number of subjects in the dataset to be simulated. It will also equal to the number of rows in the dataset to be simulated, because it is assumed that each row represents a different independent and identically distributed subject.
p	Number of predictor variables (covariates) in the simulated dataset. These covariates will be the features screened by model-free procedures.
rho	The correlation between adjacent covariates in the simulated matrix X. The within-subject covariance matrix of X is assumed to has the same form as an AR(1) auto-regressive covariance matrix, although this is not meant to imply that the X covariates for each subject are in fact a time series. Instead, it is just used as an example of a parsimonious but nontrivial covariance structure. If rho is left at the default of zero, the X covariates will be independent and the simulation will run faster.
error	The distribution of error term.

Value

the list of your simulation data

Author(s)

Xuewei Cheng <xwcheng@hunnu.edu.cn>

References

Li, R., W. Zhong, and L. Zhu (2012). Feature screening via distance correlation learning. Journal of the American Statistical Association 107(499), 1129–1139.

Examples

```
n <- 100
p <- 200
rho <- 0.5
data <- GendataGP(n, p, rho, "gaussian")
```

GendataIM *Generate simulation data (Complete data for intersection variables)*

Description

This function helps you quickly generate simulation data based on transformation model. You just need to input the sample and dimension of the data you want to generate and the covariance parameter rho. This simulated example comes from Section 4.2 introduced by Pan et al.(2019)

Usage

```
GendataIM(n, p, rho, order = 2)
```

Arguments

n	Number of subjects in the dataset to be simulated. It will also equal to the number of rows in the dataset to be simulated, because it is assumed that each row represents a different independent and identically distributed subject.
p	Number of predictor variables (covariates) in the simulated dataset. These covariates will be the features screened by model-free procedures.
rho	The correlation between adjacent covariates in the simulated matrix X. The within-subject covariance matrix of X is assumed to has the same form as an AR(1) auto-regressive covariance matrix, although this is not meant to imply that the X covariates for each subject are in fact a time series. Instead, it is just used as an example of a parsimonious but nontrivial covariance structure. If rho is left at the default of zero, the X covariates will be independent and the simulation will run faster.
order	The number of interactive variables and the default is 2.

Value

the list of your simulation data

Author(s)

Xuewei Cheng <xwcheng@hunnu.edu.cn>

References

Pan, W., X. Wang, W. Xiao, and H. Zhu (2019). A generic sure independence screening procedure. Journal of the American Statistical Association 114(526), 928–937.

Examples

```
n <- 100
p <- 200
rho <- 0.5
data <- GendataIM(n, p, rho)
```

GendataLDA *Generate simulation data (Categorial based on linear discriminant analysis model)*

Description

Simulates a dataset that can be used to filter out features for ultrahigh-dimensional discriminant analysis. The simulation is based on the balanced scenarios in Example 3.1 of Cui et al.(2015). The simulated dataset has p numerical X-predictors and a categorical Y-response.

Usage

```
GendataLDA(
  n,
  p,
  R = 3,
  error = c("gaussian", "t", "cauchy"),
  style = c("balanced", "unbalanced")
)
```

Arguments

n	Number of subjects in the dataset to be simulated. It will also equal to the number of rows in the dataset to be simulated, because it is assumed that each row represents a different independent and identically distributed subject.
p	Number of predictor variables (covariates) in the simulated dataset. These covariates will be the features screened by model-free procedures.
R	A positive integer, number of outcome categories for multinomial (categorical) outcome Y.
error	The distribution of error term, you can choose "gaussian" to generate a normal distribution of error or you choose "t" to generate a t distribution of error with degree=2. "cauchy" is represent the error term with cauchy distribution.
style	The balance among categories in categorial data .

Value

the list of your simulation data

Author(s)

Xuewei Cheng <xwcheng@hunnu.edu.cn>

References

Cui, H., Li, R., & Zhong, W. (2015). Model-free feature screening for ultrahigh dimensional discriminant analysis. Journal of the American Statistical Association, 110(510), 630-641.

Examples

```
n <- 100
p <- 200
R <- 3
data <- GendataLDA(n, p, R, error = "gaussian", style = "balanced")
```

`GendataLGM` *Generate simulation data (Binary category data based on logistic model)*

Description

This function helps you quickly generate simulation data based on logistic model. You just need to input the sample and dimension of the data you want to generate and the covariance parameter rho.

Usage

```
GendataLGM(n, p, rho, beta = c(rep(1, 5), rep(0, p - 5)))
```

Arguments

`n`	Number of subjects in the dataset to be simulated. It will also equal to the number of rows in the dataset to be simulated, because it is assumed that each row represents a different independent and identically distributed subject.
`p`	Number of predictor variables (covariates) in the simulated dataset. These covariates will be the features screened by model-free procedures.
`rho`	The correlation between adjacent covariates in the simulated matrix X. The within-subject covariance matrix of X is assumed to has the same form as an AR(1) auto-regressive covariance matrix, although this is not meant to imply that the X covariates for each subject are in fact a time series. Instead, it is just used as an example of a parsimonious but nontrivial covariance structure. If rho is left at the default of zero, the X covariates will be independent and the simulation will run faster.
`beta`	A vector with length of n, which are the coefficients that you want to generate about Linear model. The default is beta=(1,1,1,1,1,0,...,0)^T;

Value

the list of your simulation data

Author(s)

Xuewei Cheng <xwcheng@hunnu.edu.cn>

Examples

```
n <- 100
p <- 200
rho <- 0.5
data <- GendataLGM(n, p, rho)
```

`GendataLM` *Generate simulation data (Complete data based on linear models)*

Description

This function helps you quickly generate simulation data based on linear model. You just need to input the sample and dimension of the data you want to generate and the covariance parameter rho.

Usage

```
GendataLM(
  n,
  p,
  rho,
  beta = c(rep(1, 5), rep(0, p - 5)),
  error = c("gaussian", "t", "cauchy")
)
```

Arguments

`n`	Number of subjects in the dataset to be simulated. It will also equal to the number of rows in the dataset to be simulated, because it is assumed that each row represents a different independent and identically distributed subject.
`p`	Number of predictor variables (covariates) in the simulated dataset. These covariates will be the features screened by model-free procedures.
`rho`	The correlation between adjacent covariates in the simulated matrix X. The within-subject covariance matrix of X is assumed to has the same form as an AR(1) auto-regressive covariance matrix, although this is not meant to imply that the X covariates for each subject are in fact a time series. Instead, it is just used as an example of a parsimonious but nontrivial covariance structure. If rho is left at the default of zero, the X covariates will be independent and the simulation will run faster.

beta	A vector with length of n, which are the coefficients that you want to generate about Linear model. The default is beta=(1,1,1,1,1,0,...,0)^T;
error	The distribution of error term.

Value

the list of your simulation data

Author(s)

Xuewei Cheng <xwcheng@hunnu.edu.cn>

Examples

```
n <- 100
p <- 200
rho <- 0.5
data <- GendataLM(n, p, rho, error = "gaussian")
```

GendataMRM *Generate simulation data (Multivariate response models)*

Description

This function helps you quickly generate simulation data based on transformation model. You just need to input the sample and dimension of the data you want to generate and the covariance parameter rho. This simulated example comes from Example 3 introduced by Li et al.(2020)

Usage

```
GendataMRM(n, p, rho, type = c("a", "b"))
```

Arguments

n	Number of subjects in the dataset to be simulated. It will also equal to the number of rows in the dataset to be simulated, because it is assumed that each row represents a different independent and identically distributed subject.
p	Number of predictor variables (covariates) in the simulated dataset. These covariates will be the features screened by model-free procedures.
rho	The correlation between adjacent covariates in the simulated matrix X. The within-subject covariance matrix of X is assumed to has the same form as an AR(1) auto-regressive covariance matrix, although this is not meant to imply that the X covariates for each subject are in fact a time series. Instead, it is just used as an example of a parsimonious but nontrivial covariance structure. If rho is left at the default of zero, the X covariates will be independent and the simulation will run faster.
type	The type of multivariate response models, which use different mean and covariance structure to generate data. Specially, type="a" is following the Model 3.a and type="b" is following the Model 3.b by Li et al.(2020).

Value

the list of your simulation data

Author(s)

Xuewei Cheng <xwcheng@hunnu.edu.cn>

References

Liu, W., Y. Ke, J. Liu, and R. Li (2020). Model-free feature screening and FDR control with knockoff features. Journal of the American Statistical Association, 1–16.

Examples

```
n <- 100
p <- 200
rho <- 0.5
data <- GendataMRM(n, p, rho, type = "a")
```

GendataPM *Generate simulation data (Discrete response data based on poisson model)*

Description

This function helps you quickly generate simulation data based on poisson model. You just need to input the sample and dimension of the data you want to generate and the covariance parameter rho. The simulated examples based on poisson model are significant popular in the screening procedures, such as Model 1.f in Liu et al.(2020).

Usage

```
GendataPM(n, p, rho, beta = c(rep(1, 5), rep(0, p - 5)))
```

Arguments

n	Number of subjects in the dataset to be simulated. It will also equal to the number of rows in the dataset to be simulated, because it is assumed that each row represents a different independent and identically distributed subject.
p	Number of predictor variables (covariates) in the simulated dataset. These covariates will be the features screened by model-free procedures.
rho	The correlation between adjacent covariates in the simulated matrix X. The within-subject covariance matrix of X is assumed to has the same form as an AR(1) auto-regressive covariance matrix, although this is not meant to imply that the X covariates for each subject are in fact a time series. Instead, it is just used as an example of a parsimonious but nontrivial covariance structure. If

rho is left at the default of zero, the X covariates will be independent and the simulation will run faster.

`beta` A vector with length of n, which are the coefficients that you want to generate about Linear model. The default is beta=(1,1,1,1,1,0,...,0)^T;

Value

the list of your simulation data

Author(s)

Xuewei Cheng <xwcheng@hunnu.edu.cn>

References

Liu, W., Y. Ke, J. Liu, and R. Li (2020). Model-free feature screening and FDR control with knockoff features. Journal of the American Statistical Association, 1–16.

Examples

```
n <- 100
p <- 200
rho <- 0.5
data <- GendataPM(n, p, rho)
```

`GendataTM` *Generate simulation data (Complete data based on transformation model)*

Description

This function helps you quickly generate simulation data based on transformation model. You just need to input the sample and dimension of the data you want to generate and the covariance parameter rho. This simulated example comes from Example 3.a introduced by Zhu et al.(2011)

Usage

```
GendataTM(
  n,
  p,
  rho,
  beta = c(rep(1, 5), rep(0, p - 5)),
  error = c("gaussian", "t", "cauchy")
)
```

Arguments

n	Number of subjects in the dataset to be simulated. It will also equal to the number of rows in the dataset to be simulated, because it is assumed that each row represents a different independent and identically distributed subject.
p	Number of predictor variables (covariates) in the simulated dataset. These covariates will be the features screened by model-free procedures.
rho	The correlation between adjacent covariates in the simulated matrix X. The within-subject covariance matrix of X is assumed to has the same form as an AR(1) auto-regressive covariance matrix, although this is not meant to imply that the X covariates for each subject are in fact a time series. Instead, it is just used as an example of a parsimonious but nontrivial covariance structure. If rho is left at the default of zero, the X covariates will be independent and the simulation will run faster.
beta	A vector with length of n, which are the coefficients that you want to generate about Linear model. The default is beta=(1,1,1,1,1,0,...,0)^T;
error	The distribution of error term.

Value

the list of your simulation data

Author(s)

Xuewei Cheng <xwcheng@hunnu.edu.cn>

References

Zhu, L.-P., L. Li, R. Li, and L.-X. Zhu (2011). Model-free feature screening for ultrahigh-dimensional data. Journal of the American Statistical Association 106(496), 1464–1475.

Examples

```
n <- 100
p <- 200
rho <- 0.5
data <- GendataTM(n, p, rho, error = "gaussian")
```

---

get_arccos *Arccos function*

Description

This is a function to get a arccos value based on projection correlation from the Python language.

Usage

```
get_arccos(X)
```

Arguments

X	The design matrix of dimensions n * p. Each row is an observation vector.

Value

the arccos value

`Kfilter` *The Kolmogorov filter for variable screening*

Description

A new model-free screening method called the fused Kolmogorov filter is proposed for high-dimensional data analysis. This new method is fully nonparametric and can work with many types of covariates and response variables, including continuous, discrete and categorical variables.

Usage

```
Kfilter(X, Y, nsis = (dim(X)[1])/log(dim(X)[1]))
```

Arguments

X	The design matrix of dimensions n * p. Each row is an observation vector.
Y	The response vector of dimension n * 1.
nsis	Number of predictors recruited by SIS. The default is n/log(n).

Value

the labels of first nsis largest active set of all predictors

Author(s)

Xuewei Cheng <xwcheng@hunnu.edu.cn>

References

Mai, Q., & Zou, H. (2013). The Kolmogorov filter for variable screening in high-dimensional binary classification. Biometrika, 100(1), 229-234.

Mai, Q., & Zou, H. (2015). The fused Kolmogorov filter: A nonparametric model-free screening method. The Annals of Statistics, 43(4), 1471-1497.

Examples

```
n=100;
p=200;
rho=0.5;
data=GendataLM(n,p,rho,error="gaussian")
data=cbind(data[[1]],data[[2]])
colnames(data)[1:ncol(data)]=c(paste0("X",1:(ncol(data)-1)),"Y")
data=as.matrix(data)
X=data[,1:(ncol(data)-1)];
Y=data[,ncol(data)];
A=Kfilter(X,Y,n/log(n));A
```

`Kfilter_fused` *The fused kolmogorov filter: a nonparametric model-free screening method*

Description

The fused kolmogorov filter: a nonparametric model-free screening method

Usage

```
Kfilter_fused(X, Y, nsis = (dim(X)[1])/log(dim(X)[1]))
```

Arguments

`X`	The design matrix of dimensions n * p. Each row is an observation vector.
`Y`	The response vector of dimension n * 1.
`nsis`	Number of predictors recruited by Kfilter_fused. The default is n/log(n).

Value

the labels of first nsis largest active set of all predictors

References

Mai, Q., & Zou, H. (2015). The fused Kolmogorov filter: A nonparametric model-free screening method. The Annals of Statistics, 43(4), 1471-1497.

Examples

```
##Scenario 1  generate discrete response data
n=100;
p=200;
R=5;
data=GendataLDA(n,p,R,error="gaussian",style="balanced")
data=cbind(data[[1]],data[[2]])
```

```
colnames(data)[1:ncol(data)]=c(paste0("X",1:(ncol(data)-1)),"Y")
data=as.matrix(data)
X=data[,1:(ncol(data)-1)];
Y=data[,ncol(data)];
A1=Kfilter_fused(X,Y,n/log(n));A1

##Scenario 2  generate continuous response data
n=50;
p=200;
rho=0.5;
data=GendataLM(n,p,rho,error="gaussian")
data=cbind(data[[1]],data[[2]])
colnames(data)[1:ncol(data)]=c(paste0("X",1:(ncol(data)-1)),"Y")
data=as.matrix(data)
X=data[,1:(ncol(data)-1)];
Y=data[,ncol(data)];
A2=Kfilter_fused(X,Y,n/log(n));A2
```

`Kfilter_single` *The Kolmogorov filter for variable screening in high-dimensional binary classification*

Description

The Kolmogorov filter for variable screening in high-dimensional binary classification

Usage

```
Kfilter_single(X, Y, nsis = (dim(X)[1])/log(dim(X)[1]))
```

Arguments

`X`	The design matrix of dimensions n * p. Each row is an observation vector.
`Y`	The response vector of dimension n * 1.
`nsis`	Number of predictors recruited by Kfilter_single. The default is n/log(n).

Value

the labels of first nsis largest active set of all predictors

References

#' Mai, Q., & Zou, H. (2013). The Kolmogorov filter for variable screening in high-dimensional binary classification. Biometrika, 100(1), 229-234.

Examples

```
n=100;
p=200;
rho=0.5;
data=GendataLGM(n,p,rho)
data=cbind(data[[1]],data[[2]])
colnames(data)[1:ncol(data)]=c(paste0("X",1:(ncol(data)-1)),"Y")
data=as.matrix(data)
X=data[,1:(ncol(data)-1)];
Y=data[,ncol(data)];
A=Kfilter_single(X,Y,n/log(n));A
```

MDCSIS *Martingale Difference Correlation and Its Use in High-Dimensional Variable Screening*

Description

A new metric, the so-called martingale difference correlation, measure the departure of conditional mean independence between a scalar response variable V and a vector predictor variable U. This metric is a natural extension of distance correlation proposed by Szekely, Rizzo, and Bahirov(2007), which is used to measure the dependence between V and U. The martingale difference correlation and its empirical counterpart inherit a number of desirable features of distance correlation and sample distance correlation, such as algebraic simplicity and elegant theoretical properties.

Usage

```
MDCSIS(X, Y, nsis = (dim(X)[1])/log(dim(X)[1]))
```

Arguments

X	The design matrix of dimensions n * p. Each row is an observation vector.
Y	The response vector of dimension n * 1.
nsis	Number of predictors recruited by MDCSIS. The default is n/log(n).

Value

the labels of first nsis largest active set of all predictors

Author(s)

Xuewei Cheng <xwcheng@hunnu.edu.cn>

References

Szekely, G. J., M. L. Rizzo, and N. K. Bakirov (2007). Measuring and testing dependence by correlation of distances. The annals of statistics 35(6), 2769–2794.

Shao, X. and J. Zhang (2014). Martingale difference correlation and its use in high-dimensional variable screening. Journal of the American Statistical Association 109(507),1302–1318.

Examples

```
n <- 100
p <- 200
rho <- 0.5
data <- GendataLM(n, p, rho, error = "gaussian")
data <- cbind(data[[1]], data[[2]])
colnames(data)[1:ncol(data)] <- c(paste0("X", 1:(ncol(data) - 1)), "Y")
data <- as.matrix(data)
X <- data[, 1:(ncol(data) - 1)]
Y <- data[, ncol(data)]
A <- MDCSIS(X, Y, n / log(n))
A
```

MFSIS *Model-free feature screening procedures*

Description

Through this function, we provide a unified framework to carry out model-free screening procedures including SIS (Fan and Lv (2008) <doi:10.1111/j.1467-9868.2008.00674.x>), SIRS(Zhu et al. (2011)<doi:10.1198/jasa.2011.tm10563>), DC-SIS (Li et al. (2012) <doi:10.1080/01621459.2012.695654>), MDC-SIS(Shao and Zhang (2014) <doi:10.1080/01621459.2014.887012>), Bcor-SIS (Pan et al. (2019) <doi:10.1080/01621459.2018.1462709>), PC-Screen (Liu et al. (2020) <doi:10.1080/01621459.2020.1783274>), WLS (Zhong et al.(2021) <doi:10.1080/01621459.2021.1918554>), Kfilter (Mai and Zou (2015) <doi:10.1214/14-AOS1303>), MVSIS (Cui et al. (2015) <doi:10.1080/01621459.2014.920256>), PSIS (Pan et al. (2016) <doi:10.1080/01621459.2014.998760>), CAS (Xie et al. (2020) <doi:101080/016214592019157373
CI-SIS (Cheng and Wang. (2022) <doi:10.1016/j.cmpb.2022.107269>) and CSIS (Cheng et al. (2023) <doi:10.1007/s00180-023-01399-5>).

Usage

```
MFSIS(
  X,
  Y,
  nsis = (dim(X)[1])/log(dim(X)[1]),
  method = c("SIS", "SIRS", "DCSIS", "MDCSIS", "CSIS", "PCSIS", "BcorSIS", "WLS",
    "MVSIS", "Kfilter")
)
```

Arguments

X	The design matrix of dimensions n * p. Each row is an observation vector.
Y	The response vector of dimension n * 1.
nsis	Number of predictors recruited by the screening method. The default is n/log(n).
method	The method that you choose to perform screening procedure. method=c("SIS", "SIRS", "DCSIS", "MDCSIS", "CSIS", "PCSIS", "BcorSIS", "WLS", "MV-SIS", "Kfilter","PSIS","CAS","CISIS"). If you want to know more information about this method, please use command "help(method)" for detail information.

Value

the labels of first nsis largest active set of all predictors

Author(s)

Xuewei Cheng <xwcheng@hunnu.edu.cn>

Examples

```
n <- 100
p <- 200
rho <- 0.5
data <- GendataLM(n, p, rho, error = "gaussian")
data <- cbind(data[[1]], data[[2]])
colnames(data)[1:ncol(data)] <- c(paste0("X", 1:(ncol(data) - 1)), "Y")
data <- as.matrix(data)
X <- data[, 1:(ncol(data) - 1)]
Y <- data[, ncol(data)]
A <- MFSIS(X, Y, n / log(n), method = "CSIS")
A
```

MVSIS — *Model-Free Feature Screening for Ultrahigh Dimensional Discriminant Analysis*

Description

A marginal feature screening procedure based on empirical conditional distribution function. The response variable is categorical in discriminant analysis. This method uses the conditional distribution function to construct a new index for feature screening.

Usage

```
MVSIS(X, Y, nsis)
```

Arguments

X	The design matrix of dimensions n * p. Each row is an observation vector.
Y	The response vector of dimension n * 1.
nsis	Number of predictors recruited by MVSIS. The default is n/log(n).

Value

the labels of first nsis largest active set of all predictors

Author(s)

Xuewei Cheng <xwcheng@hunnu.edu.cn>

References

Cui, H., Li, R., & Zhong, W. (2015). Model-free feature screening for ultrahigh dimensional discriminant analysis. Journal of the American Statistical Association, 110(510), 630-641.

Examples

```
n <- 100
p <- 200
rho <- 0.5
data <- GendataLGM(n, p, rho)
data <- cbind(data[[1]], data[[2]])
colnames(data)[1:ncol(data)] <- c(paste0("X", 1:(ncol(data) - 1)), "Y")
data <- as.matrix(data)
X <- data[, 1:(ncol(data) - 1)]
Y <- data[, ncol(data)]
A <- MVSIS(X, Y, n / log(n))
A
```

PCSIS *Model-Free Feature Screening Based on the Projection Correlation*

Description

A model-free screening method is based on the projection correlation which measures the dependence between two random vectors. This projection correlation based method does not require specifying a regression model, and applies to data in the presence of heavy tails and multivariate responses. It enjoys both sure screening and rank consistency properties under weak assumptions.

Usage

```
PCSIS(X, Y, nsis = (dim(X)[1])/log(dim(X)[1]))
```

Arguments

X	The design matrix of dimensions n * p. Each row is an observation vector.
Y	The response vector of dimension n * 1.
nsis	Number of predictors recruited by PCSIS. The default is n/log(n).

Value

the labels of first nsis largest active set of all predictors

Author(s)

Xuewei Cheng <xwcheng@hunnu.edu.cn>

References

Zhu, L., K. Xu, R. Li, and W. Zhong (2017). Projection correlation between two random vectors. Biometrika 104(4), 829–843.

Liu, W., Y. Ke, J. Liu, and R. Li (2020). Model-free feature screening and FDR control with knockoff features. Journal of the American Statistical Association, 1–16.

Examples

```
# have_numpy=reticulate::py_module_available("numpy")
# if (have_numpy){
# req_py()
n=100;
p=200;
rho=0.5;
data=GendataLM(n,p,rho,error="gaussian")
data=cbind(data[[1]],data[[2]])
colnames(data)[1:ncol(data)]=c(paste0("X",1:(ncol(data)-1)),"Y")
data=as.matrix(data)
X=data[,1:(ncol(data)-1)];
Y=data[,ncol(data)];
# A=PCSIS(X,Y,n/log(n));A
# }else{
#    print('You should have the Python testing environment!')
#}
```

---

projection_corr *Projection correlation function*

Description

Projection correlation between X[,j] and Y from the Python language

Usage

```
projection_corr(A_x, A_y, n)
```

Arguments

A_x	The arccos value about X
A_y	The arccos value about Y
n	The sample size

Value

the projection correlation

PSIS *Ultrahigh-Dimensional Multiclass Linear Discriminant Analysis by Pairwise Sure Independence Screening*

Description

A novel pairwise sure independence screening method for linear discriminant analysis with an ultrahigh-dimensional predictor. This procedure is directly applicable to the situation with many classes.

Usage

```
PSIS(X, Y, nsis)
```

Arguments

X	The design matrix of dimensions n * p. Each row is an observation vector.
Y	The response vector of dimension n * 1.
nsis	Number of predictors recruited by PSIS. The default is n/log(n).

Value

the labels of first nsis largest active set of all predictors

Author(s)

Xuewei Cheng <xwcheng@hunnu.edu.cn>

References

Pan, R., Wang, H., and Li, R. (2016). Ultrahigh-dimensional multiclass linear discriminant analysis by pairwise sure independence screening. Journal of the American Statistical Association, 111(513):169–179.

Examples

```
n <- 100
p <- 200
rho <- 0.5
data <- GendataLGM(n, p, rho)
data <- cbind(data[[1]], data[[2]])
colnames(data)[1:ncol(data)] <- c(paste0("X", 1:(ncol(data) - 1)), "Y")
data <- as.matrix(data)
X <- data[, 1:(ncol(data) - 1)]
Y <- data[, ncol(data)]
A <- PSIS(X, Y, n / log(n))
A
```

`req_py` *Detect Python Module*

Description

A function to detect Python module.

Usage

```
req_py()
```

Author(s)

Xuewei Cheng <xwcheng@hunnu.edu.cn>

`Simdata` *Generate simulation data (The unified class framework to generate simulation data)*

Description

This function is a unified framework that helps you quickly generate simulation data. You just need to input the sample and dimension of the data you want to generate and the covariance parameter rho.

Usage

```
Simdata(
  n,
  p,
  rho,
  beta = c(rep(1, 5), rep(0, p - 5)),
  error = c("gaussian", "t", "cauchy"),
  R = 3,
  style = c("balanced", "unbalanced"),
  lambda = 0.1,
  order = 2,
  type = c("a", "b"),
 model = c("linear", "nonlinear", "binomial", "poisson", "classification", "Cox",
    "interaction", "group", "multivariate", "AFT")
)
```

Arguments

n	Number of subjects in the dataset to be simulated. It will also equal to the number of rows in the dataset to be simulated, because it is assumed that each row represents a different independent and identically distributed subject.
p	Number of predictor variables (covariates) in the simulated dataset. These covariates will be the features screened by model-free procedures.
rho	The correlation between adjacent covariates in the simulated matrix X. The within-subject covariance matrix of X is assumed to has the same form as an AR(1) auto-regressive covariance matrix, although this is not meant to imply that the X covariates for each subject are in fact a time series. Instead, it is just used as an example of a parsimonious but nontrivial covariance structure. If rho is left at the default of zero, the X covariates will be independent and the simulation will run faster.
beta	A vector with length of n, which are the coefficients that you want to generate about chosen model. The default is beta=(1,1,1,1,1,0,...,0)^T.
error	The distribution of error term.
R	A positive integer, number of outcome categories for multinomial (categorical) outcome Y.
style	Whether categories in categorial data are balanced or not.
lambda	This parameter control the censoring rate in survival data. The censored time is generated by exponential distribution with mean 1/lambda. The default is lambda=0.1.
order	The number of interactive variables and the default is 2.
type	The type of multivariate response models, which use different mean and covariance structure to generate data. Specially, type="a" is following the Model 3.a and type="b" is following the Model 3.b by Liu et al.(2020).
model	The model that you choose to generate simulation data.

Value

the list of your simulation data

Author(s)

Xuewei Cheng <xwcheng@hunnu.edu.cn>

References

Liu, W., Y. Ke, J. Liu, and R. Li (2020). Model-free feature screening and FDR control with knockoff features. Journal of the American Statistical Association, 1–16.

Examples

```
n <- 100
p <- 200
rho <- 0.5
data <- Simdata(n, p, rho, error = "gaussian", model = "linear")
```

SIRS *Model-Free Feature Screening for Ultrahigh Dimensional Data*

Description

A novel feature screening procedure under a unified model framework, which covers a wide variety of commonly used parametric and semi-parametric models. This method does not require imposing a specific model structure on regression functions, and thus is particularly appealing to ultrahigh-dimensional regressions, where there are a huge number of candidate predictors but little information about the actual model forms.

Usage

```
SIRS(X, Y, nsis = (dim(X)[1])/log(dim(X)[1]))
```

Arguments

X	The design matrix of dimensions n * p. Each row is an observation vector.
Y	The response vector of dimension n * 1.
nsis	Number of predictors recruited by SIRS. The default is n/log(n).

Value

the labels of first nsis largest active set of all predictors

Author(s)

Xuewei Cheng <xwcheng@hunnu.edu.cn>

References

Zhu, L.-P., L. Li, R. Li, and L.-X. Zhu (2011). Model-free feature screening for ultrahigh-dimensional data. Journal of the American Statistical Association 106(496), 1464–1475.

Examples

```
n <- 100
p <- 200
rho <- 0.5
data <- GendataLM(n, p, rho, error = "gaussian")
data <- cbind(data[[1]], data[[2]])
colnames(data)[1:ncol(data)] <- c(paste0("X", 1:(ncol(data) - 1)), "Y")
data <- as.matrix(data)
X <- data[, 1:(ncol(data) - 1)]
Y <- data[, ncol(data)]
A <- SIRS(X, Y, n / log(n))
A
```

SIS *Sure Independent Screening*

Description

To overcome challenges caused by ultra-high dimensionality, Fan and Lv (2008) proposed a sure independence screening (SIS) method, which aims to screen out the redundant features by ranking their marginal Pearson correlations. The SIS method is named after the SIS property, which states the selected subset of features contains all the active ones with probability approaching one.

Usage

```
SIS(X, Y, nsis = (dim(X)[1])/log(dim(X)[1]))
```

Arguments

X	The design matrix of dimensions n * p. Each row is an observation vector.
Y	The response vector of dimension n * 1.
nsis	Number of predictors recruited by SIS. The default is n/log(n).

Value

the labels of first nsis largest active set of all predictors

Author(s)

Xuewei Cheng <xwcheng@hunnu.edu.cn>

References

Fan, J. and J. Lv (2008). Sure independence screening for ultrahigh dimensional feature space. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 70(5),849–911.

Examples

```
n <- 100
p <- 200
rho <- 0.5
data <- GendataLM(n, p, rho, error = "gaussian")
data <- cbind(data[[1]], data[[2]])
colnames(data)[1:ncol(data)] <- c(paste0("X", 1:(ncol(data) - 1)), "Y")
data <- as.matrix(data)
X <- data[, 1:(ncol(data) - 1)]
Y <- data[, ncol(data)]
A <- SIS(X, Y, n / log(n))
A
```

WLS *A Model-free Variable Screening Method Based on Leverage Score*

Description

An innovative and effective sampling scheme based on leverage scores via singular value decompositions has been proposed to select rows of a design matrix as a surrogate of the full data in linear regression. Analogously, variable screening can be viewed as selecting rows of the design matrix. However, effective variable selection along this line of thinking remains elusive. This method propose a weighted leverage variable screening method by using both the left and right singular vectors of the design matrix.

Usage

```
WLS(X, Y, nsis = (dim(X)[1])/log(dim(X)[1]))
```

Arguments

X	The design matrix of dimensions n * p. Each row is an observation vector.
Y	The response vector of dimension n * 1.
nsis	Number of predictors recruited by WLS. The default is n/log(n).

Value

the labels of first nsis largest active set of all predictors.

Author(s)

Xuewei Cheng <xwcheng@hunnu.edu.cn>

References

Zhong, W., Liu, Y., & Zeng, P. (2021). A Model-free Variable Screening Method Based on Leverage Score. Journal of the American Statistical Association, (just-accepted), 1-36.

Examples

```
n <- 100
p <- 200
rho <- 0.5
data <- GendataLM(n, p, rho, error = "gaussian")
data <- cbind(data[[1]], data[[2]])
colnames(data)[1:ncol(data)] <- c(paste0("X", 1:(ncol(data) - 1)), "Y")
data <- as.matrix(data)
X <- data[, 1:(ncol(data) - 1)]
Y <- data[, ncol(data)]
A <- WLS(X, Y, n / log(n))
A
```

Index