Package: GUEST (via r-universe)

August 31, 2024

Type Package

Title Graphical Models in Ultrahigh-Dimensional and Error-Prone Data via Boosting Algorithm

Version 0.2.0

Description We consider the ultrahigh-dimensional and error-prone data. Our goal aims to estimate the precision matrix and identify the graphical structure of the random variables with measurement error corrected. We further adopt the estimated precision matrix to the linear discriminant function to do classification for multi-label classes.

License GPL-2

Encoding UTF-8

Depends R (>= 3.5.0)

Suggests sna

Imports XICOR, network, GGally

LazyData true

RoxygenNote 7.3.1

NeedsCompilation no

Author Hui-Shan Tsao [aut, cre], Li-Pang Chen [aut]

Maintainer Hui-Shan Tsao <n410412@gmail.com>

Repository CRAN

Date/Publication 2024-07-30 14:30:02 UTC

Contents

boost.graph *Estimation of precision matrix and detection of graphical structure*

Description

This function first applies the regression calibration to deal with measurement error effects. After that, the feature screening technique is employed to screen out independent pairs of random variables and reduce the dimension of random variables. Finally, we adopt the boosting method to detect informative pairs of random variables and estimate the precision matrix. This function can handle various distributions, such as normal, binomial, and Poisson distributions, as well as nonlinear effects among random variables.

Usage

```
boost.graph(data,ite1,ite2,ite3,thre,select = 0.9,inc = 10^(-3),
sigma_e = 0.6,q = 0.8,lambda = 1,pi = 0.5,rep = 100,cor = TRUE)
```

Arguments

data	An n (observations) times p (variables) matrix of random variables, whose distributions can be continuous, discrete, or mixed.
ite1	The number of iterations for continuous variables.
ite2	The number of iterations for binary variables.
ite3	The number of iterations for count variables.
thre	The treshold value for feature screening, whose value should be between 0 and 1.
select	The treshold constant in the boosting algorithm, whose value should be between 0 and 1. The default value is 0.9.
inc	The learning rate of the increment in the boosting algorithm, which shoud be a small value. The default value is 0.001.
sigma_e	The common value in the diagonal covariance matrix of the error for the classical measurement error model when data are continuous. The default value is 0.6.
q	The common value used to characterize misclassification for binary random variables. The default value is 0.8.
lambda	The parameter of the Poisson distribution, which is used to characterize error-prone count random variables. The default value is 1.
pi	The probability in the Binomial distribution, which is used to characterize error-prone count random variables. The default value is 0.5.
rep	The number of bootstrapping iterations. The default value is 100.
cor	Measurement error correction when estimating the precision matrix. The default value is TRUE.

Value

w	The estimator of the precision matrix.
p	The chosen pairs obtained by the feature screening.
xi	The weights sorted with pairs in p.
g	The visualization of the estimated network structure determined by w.

Author(s)

Hui-Shan Tsao and Li-Pang Chen
Maintainer: Hui-Shan Tsao <n410412@gmail.com>

References

Hui-Shan Tsao (2024). *Estimation of Ultrahigh-Dimensional Graphical Models and Its Application to Dsicriminant Analysis.* Master Thesis supervised by Li-Pang Chen, National Chengchi University.

Examples

```
data(MedulloblastomaData)

X <- t(MedulloblastomaData[2:656,]) #covariates
Y <- MedulloblastomaData[1,] #response

X <- matrix(as.numeric(X),nrow=23)

p <- ncol(X)
n <- nrow(X)

#standarization
X_new=data.frame()
for (i in 1:p){
 X_new[1:n,i]=(X[,i]-rep(mean(X[,i]),n))/sd(X[,i])
}
X_new=matrix(unlist(X_new),nrow = n)


#estimate graphical model
result <- boost.graph(data = X_new, thre = 0.2, ite1 = 3, ite2 = 0, ite3 = 0, rep = 1)
theta.hat <- result$w
```

GUEST_package *Graphical Models in Ultrahigh-Dimensional and Error-Prone Data via Boosting Algorithm*

Description

The package GUEST, referred to Graphical models in Ultrahigh-dimensional and Error-prone data via booSTing algorithm, is used to estimate the precision matrix and detect graphical structure for ultrahigh-dimensional, error-prone, and possibly nonlinear random variables. Given the estimated precision matrix, we further apply it to the linear discriminant function to deal with multi-classification. The precision matrix can be estimated by the function `boost.graph`, and the classification can be implemented by the function `LDA.boost`. Finally, we consider the medulloblastoma dataset to demonstrate the implementation of two functions.

Details

To estimate the precision matrix and detect the graphical structure under our scenario, the function `boost.graph` first applies the regression calibration method to deal with measurement error in continuous, binary, or count random variables. After that, the feature screening technique is employed to reduce the dimension of random variable, and we then adopt the boosting algorithm to estimate the precision matrix. The estimated precision matrix also reflects the desired graphical structure. The function `LDA.boost` implements the linear discriminant function to do classification for multi-label classes, where the precision matrix, also known as the inverse of the covariance matrix, in the linear discriminant function can be estimated by the function `boost.graph`.

Value

GUEST_package

`LDA.boost` *Implementation of the linear discriminant function for multi-label classification.*

Description

This function applies the linear discriminant function to do classification for multi-label responses. The precision matrix, or the inverse of the covariance matrix, in the linear discriminant function can be estimated by `w` in the function `boost.graph`. In addition, error-prone covariates in the linear discriminant function are addressed by the regression calibration.

Usage

```
LDA.boost(data, resp, theta, sigma_e = 0.6,q = 0.8,lambda = 1, pi = 0.5)
```

Arguments

`data`	An n (observations) times p (variables) matrix of random variables, whose distributions can be continuous, discrete, or mixed.
`resp`	An n-dimensional vector of categorical random variables, which is the response in the data.
`theta`	The estimator of the precision matrix.

sigma_e	The common value in the diagonal covariance matrix of the error for the classical measurement error model when data are continuous. The default value is 0.6.
q	The common value used to characterize misclassification for binary random variables. The default value is 0.8.
lambda	The parameter of the Poisson distribution, which is used to characterize error-prone count random variables. The default value is 1.
pi	The probability in the Binomial distribution, which is used to characterize error-prone count random variables. The default value is 0.5.

Details

The linear discriminant function used is as follow:

$$\texttt{score}_{i,j} = \log(\pi_i) - 0.5\ \mu_i^\top\ \texttt{theta}\ \mu_i + \texttt{data}_j^\top\ \texttt{theta}\ \mu_i,$$

for the class $i = 1, \cdots, I$ with I being the number of classes in the dataset and subject $j = 1, \cdots, n$, where π_i is the proportion of subjects in the class i, $\texttt{data}_j$ is the vector of covariates for the subject j, theta is the precision matrix of the covariates, and μ_i is the empirical mean vector of the random variables in the class i.

Value

score	The value of the linear discriminant function (see details) with the estimator of the precision matrix accommodated.
class	The result of predicted class for subjects.

Author(s)

Hui-Shan Tsao and Li-Pang Chen
Maintainer: Hui-Shan Tsao <n410412@gmail.com>

References

Hui-Shan Tsao (2024). *Estimation of Ultrahigh-Dimensional Graphical Models and Its Application to Dsicriminant Analysis.* Master Thesis supervised by Li-Pang Chen, National Chengchi University.

Examples

```
data(MedulloblastomaData)

X <- t(MedulloblastomaData[2:655,]) #covariates
Y <- MedulloblastomaData[1,] #response

X <- matrix(as.numeric(X),nrow=23)

p <- ncol(X)
n <- nrow(X)
```

```
#standarization
X_new=data.frame()
for (i in 1:p){
 X_new[1:n,i]=(X[,i]-rep(mean(X[,i]),n))/sd(X[,i])
}
X_new=matrix(unlist(X_new),nrow = n)


#estimate graphical model
result <- boost.graph(data = X_new, thre = 0.2, ite1 = 3, ite2 = 0, ite3 = 0, rep = 1)
theta.hat <- result$w

theta.hat[which(theta.hat<0.8)]=0 #keep the highly dependent pairs

#predict
pre <- LDA.boost(data = X_new, resp = Y, theta = theta.hat)
estimated_Y <- pre$class
```

`MedulloblastomaData` *The medulloblastoma dataset*

Description

The dataset, which is available on https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE468, contains 23 patients with medulloblastoma, and each patient has 2059 gene expression values. The response contains 2 classes: metastatic (M+) or non-metastatic (M0). After removing the missing and duplicate values, the dimension of remaining gene expressions is 655. The dataset is used to illustrate the usage of the `boost.graph` and `LDA.boost` functions.

Usage

```
data(MedulloblastomaData)
```

Format

The dataset has 23 observations and 655 gene expression values.

References

MacDonald, T., Brown, K., LaFleur, B., Peterson K., Lawlor C., Chen Y., Packer RJ., Cogen P., Stephan DA.(2001). *Expression profiling of medulloblastoma: PDGFRA and the RAS/MAPK pathway as therapeutic targets for metastatic disease*. Nat Genet, 29, 143–152.

Examples

```
X <- t(MedulloblastomaData[2:655,]) #covariates
Y <- MedulloblastomaData[1,] #response
```

Index