Package 'Renvlp'

March 16, 2020

Type Package

Title Computing Envelope Estimators

Version 2.8

Date 2020-03-16

Author Minji Lee, Zhihua Su

Maintainer Minji Lee <mlee9@ufl.edu>

Description Provides a general routine, envMU, which allows estimation of the M envelope of span(U) given root n consistent estimators of M and U. The routine envMU does not presume a model. This package implements response envelopes, partial response envelopes, envelopes in the predictor space, heteroscedastic envelopes, simultaneous envelopes, scaled response envelopes, scaled envelopes in the predictor space, groupwise envelopes, weighted envelopes, envelopes in logistic regression and envelopes in Poisson regression. For each of these model-based routines the package provides inference tools including bootstrap, cross validation, estimation and prediction, hypothesis testing on coefficients are included except for weighted envelopes. Tools for selection of dimension include AIC, BIC and likelihood ratio testing. Background is available at Cook, R. D., Forzani, L. and Su, Z. (2016) <doi:10.1016/j.jmva.2016.05.006>. Optimization is based on a clockwise coordinate descent algorithm.

License GPL-2

NeedsCompilation no

Imports Rsolnp, stats

Suggests MASS

Repository CRAN

Date/Publication 2020-03-16 11:00:02 UTC

R topics documented:

`Renvlp-package` *Computing Envelope Estimators*

Description

This package contains functions for estimating envelope models including response envelopes, partial response envelopes, envelopes in the predictor space, heteroscedastic envelopes, simultaneous envelopes, scaled response envelopes, scaled envelopes in the predictor space, groupwise envelopes, weighted envelopes, envelopes in logistic regression and envelopes in poisson regression.

Details

Package:	Renvlp
Type:	Package
Version:	2.8
Date:	2020-03-16
License:	GPL-2
Imports:	Rsolnp
Suggests:	Mass, stats

Author(s)

Minji Lee and Zhihua Su.

Maintainer: Minji Lee <mlee9@ufl.edu>

References

Cook, R. D., Forzani, L. and Su, Z. (2016) A Note on Fast Envelope Estimation. Journal of Multivariate Analysis. 150, 42-54.

Cook, R. D., Li, B. and Chiaromente, F. (2010). Envelope Models for Parsimonious and Efficient Multivariate Linear Regression (with discussion). Statist. Sinica 20, 927- 1010.

Su, Z. and Cook, R.D. (2011). Partial envelopes for efficient estimation in multivariate linear regression. Biometrika 98, 133 - 146.

Cook, R. D., Helland, I. S. and Su, Z. (2013). Envelopes and Partial Least Squares Re- gression. Journal of the Royal Statistical Society: Series B 75, 851 - 877.

Su, Z. and Cook, R.D. (2013). Estimation of Multivariate Means with Heteroscedastic Error Using Envelope Models. Statistica Sinica, 23, 213-230.

Cook, R. D., Zhang, X. (2015). Simultaneous Envelopes for Multivariate Linear Regression. Technometrics 57, 11 - 25.

Cook, R. D., Su, Z. (2013). Scaled Envelopes: scale Invariant and Efficient Estimation in Multivariate Linear Regression. Biometrika 100, 939 - 954.

Cook, R. D., Su, Z. (2016). Scaled Predictor Envelopes and Partial Least Squares Regression. Technometrics 58, 155 - 165.

Park, Y., Su, Z. and Zhu, H. (2017) Groupwise envelope models for Imaging Genetic Analysis. Biometrics, to appear.

Eck, D. J. and Cook, R. D. (2017). Weighted Envelope Estimation to Handle Variability in Model Selection. Biometrika.

Cook, R. D., Zhang, X. (2015). Foundations for Envelope Models and Methods. Journal of the American Statistical Association 110, 599 - 611.

`Berkeley` *Berkeley Guidance Study Data*

Description

Heights of children born in Berkeley

Usage

```
data("Berkeley")
```

Format

A data frame with 93 observations on the following 32 variables.

`V1` Sex.

`V2` Age 1.

`V3` Age 1.25.

`V4` Age 1.5.

`V5` Age 1.75.

`V6` Age 2.

`V7` Age 3.

`V8` Age 4.

`V9` Age 5.

`V10` Age 6.

`V11` Age 7.

`V12` Age 8.

`V13` Age 8.5.

`V14` Age 9.

`V15` Age 9.5.

`V16` Age 10.

`V17` Age 10.5.

`V18` Age 11.

`V19` Age 11.5.

`V20` Age 12.

`V21` Age 12.5.

`V22` Age 13.

`V23` Age 13.5.

`V24` Age 14.

`V25` Age 14.5.

`V26` Age 15.

`V27` Age 15.5.

`V28` Age 16.

`V29` Age 16.5.

`V30` Age 17.

`V31` Age 17.5.

`V32` Age 18.

Details

This data set contains measurements of heights of children born in 1928-29 in Berkeley, CA.

References

Tuddenham, R. D. and Snyder, M. M. (1954). Physical growth of California boys and girls from birth to eighteen years. Publications in child developments. University of California, Berkeley, 1(2), 183-364.

`boot.env` *Bootstrap for env*

Description

Compute bootstrap standard error for the envelope estimator.

Usage

```
boot.env(X, Y, u, B)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`u`	Dimension of the envelope. An integer between 0 and r.
`B`	Number of bootstrap samples. A positive integer.

Details

This function computes the bootstrap standard errors for the regression coefficients in the envelope model by bootstrapping the residuals.

Value

The output is an r by p matrix.

bootse The standard error for elements in beta computed by bootstrap.

Examples

```
data(wheatprotein)
X <- wheatprotein[, 8]
Y <- wheatprotein[, 1:6]

u <- u.env(X, Y)
u

B <- 100
bootse <- boot.env(X, Y, 1, B)
bootse
```

---

boot.genv *Bootstrap for genv*

Description

Compute bootstrap standard error for the groupwise envelope.

Usage

```
boot.genv(X, Y, Z, u, B)
```

Arguments

X	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
Y	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
Z	A group indicator vector of length n, where n denotes the number of observations.
u	Dimension of the groupwise envelope. An integer between 0 and r.
B	Number of bootstrap samples. A positive integer.

Details

This function computes the bootstrap standard errors for the regression coefficients in the groupwise envelope model by bootstrapping the residuals.

Value

The output is an p by r matrix.

`bootse`	The standard error for elements in beta computed by bootstrap.

Examples

```
data(fiberpaper)
X <- fiberpaper[ , c(5, 7)]
Y <- fiberpaper[ , 1:3]
Z <- as.numeric(fiberpaper[ , 6] > mean(fiberpaper[ , 6]))

## Not run: B <- 100
## Not run: res <- boot.genv(X, Y, Z, 2, B)
## Not run: res$bootse[[1]]
## Not run: res$bootse[[2]]
```

`boot.henv` *Bootstrap for henv*

Description

Compute bootstrap standard error for the heteroscedastic envelope.

Usage

```
boot.henv(X, Y, u, B)
```

Arguments

`X`	A group indicator vector of length n, where n denotes the number of observations.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`u`	Dimension of the heteroscedastic envelope. An integer between 0 and r.
`B`	Number of bootstrap samples. A positive integer.

Details

This function computes the bootstrap standard errors for the regression coefficients in the heteroscedastic envelope model by bootstrapping the residuals.

Value

The output is an r by p matrix.

`bootse`	The standard error for elements in beta computed by bootstrap.

Examples

```
data(waterstrider)
X <- waterstrider[ , 1]
Y <- waterstrider[ , 2:5]

B <- 100
## Not run: res <- boot.henv(X, Y, 2, B)
## Not run: res
```

`boot.logit.env` *Bootstrap for logit.env*

Description

Compute bootstrap standard error for the envelope estimator in logistic regression.

Usage

```
boot.logit.env(X, Y, u, B)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.
`Y`	Response. An n by 1 matrix. The univariate response must be binary.
`u`	Dimension of the envelope. An integer between 0 and p.
`B`	Number of bootstrap samples. A positive integer.

Details

This function computes the bootstrap standard errors for the coefficients in the logistic regression envelope by the paired bootstrap.

Value

The output is a p by 1 matrix.

`bootse`	The standard error for elements in beta computed by bootstrap.

Examples

```
data(horseshoecrab)
X1 <- as.numeric(horseshoecrab[ , 1] == 2)
X2 <- as.numeric(horseshoecrab[ , 1] == 3)
X3 <- as.numeric(horseshoecrab[ , 1] == 4)
X4 <- as.numeric(horseshoecrab[ , 2] == 2)
X5 <- as.numeric(horseshoecrab[ , 2] == 3)
X6 <- horseshoecrab[ , 3]
X7 <- horseshoecrab[ , 5]
X <- cbind(X1, X2, X3, X4, X5, X6, X7)
Y <- as.numeric(ifelse(horseshoecrab[ , 4] > 0, 1, 0))

B <- 100
## Not run: bootse <- boot.logit.env(X, Y, 1, B)
## Not run: bootse
```

---

boot.penv *Bootstrap for penv*

Description

Compute bootstrap standard error for the partial envelope estimator.

Usage

```
boot.penv(X1, X2, Y, u, B)
```

Arguments

X1	Predictors of main interest. An n by p1 matrix, n is the number of observations, and p1 is the number of main predictors. The predictors can be univariate or multivariate, discrete or continuous.
X2	Covariates, or predictors not of main interest. An n by p2 matrix, p2 is the number of covariates.
Y	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
u	Dimension of the partial envelope. An integer between 0 and r.
B	Number of bootstrap samples. A positive integer.

Details

This function computes the bootstrap standard errors for the regression coefficients beta1 in the partial envelope model by bootstrapping the residuals.

Value

The output is an r by p1 matrix.

bootse	The standard error for elements in beta1 computed by bootstrap.

Examples

```
data(fiberpaper)
X1 <- fiberpaper[, 7]
X2 <- fiberpaper[, 5:6]
Y <- fiberpaper[, 1:4]

B <- 100
## Not run: bootse <- boot.penv(X1, X2, Y, 1, B)
## Not run: bootse
```

`boot.pois.env` *Bootstrap for pois.env*

Description

Compute bootstrap standard error for the envelope estimator in poisson regression.

Usage

```
boot.pois.env(X, Y, u, B)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.
`Y`	Response. An n by 1 matrix. The univariate response must be counts.
`u`	Dimension of the envelope. An integer between 0 and p.
`B`	Number of bootstrap samples. A positive integer.

Details

This function computes the bootstrap standard errors for the coefficients in the poisson regression envelope by the paired bootstrap.

Value

The output is a p by 1 matrix.

`bootse`	The standard error for elements in beta computed by bootstrap.

Examples

```
data(horseshoecrab)
X1 <- as.numeric(horseshoecrab[ , 1] == 2)
X2 <- as.numeric(horseshoecrab[ , 1] == 3)
X3 <- as.numeric(horseshoecrab[ , 1] == 4)
X4 <- as.numeric(horseshoecrab[ , 2] == 2)
X5 <- as.numeric(horseshoecrab[ , 2] == 3)
X6 <- horseshoecrab[ , 3]
X7 <- horseshoecrab[ , 5]
X <- cbind(X1, X2, X3, X4, X5, X6, X7)
Y <- horseshoecrab[ , 4]

B <- 100
## Not run: bootse <- boot.pois.env(X, Y, 1, B)
## Not run: bootse
```

`boot.senv` *Bootstrap for senv*

Description

Compute bootstrap standard error for the scaled envelope estimator.

Usage

```
boot.senv(X, Y, u, B)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`u`	Dimension of the scaled envelope. An integer between 0 and r.
`B`	Number of bootstrap samples. A positive integer.

Details

This function computes the bootstrap standard errors for the regression coefficients in the scaled envelope model by bootstrapping the residuals.

Value

The output is an r by p matrix.

`bootse`	The standard error for elements in beta computed by bootstrap.

Examples

```
data(sales)
X <- sales[, 1:3]
Y <- sales[, 4:7]

u <- u.senv(X, Y)
u

## Not run: B <- 100
## Not run: bootse <- boot.senv(X, Y, 2, B)
## Not run: bootse
```

`boot.stenv` *Bootstrap for stenv*

Description

Compute bootstrap standard error for the simultaneous envelope estimator.

Usage

```
boot.stenv(X, Y, q, u, B)
```

Arguments

X	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
Y	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
q	Dimension of the X-envelope. An integer between 0 and p.
u	Dimension of the Y-envelope. An integer between 0 and r.
B	Number of bootstrap samples. A positive integer.

Details

This function computes the bootstrap standard errors for the regression coefficients in the envelope model by bootstrapping the residuals.

Value

The output is an p by r matrix.

bootse	The standard error for elements in beta computed by bootstrap.

Examples

```
data(fiberpaper)
X <- fiberpaper[, 5:7]
Y <- fiberpaper[, 1:4]
u <- u.stenv(X, Y)
u

## Not run: B <- 100
## Not run: bootse <- boot.stenv(X, Y, 2, 3, B)
## Not run: bootse
```

`boot.sxenv` *Bootstrap for sxenv*

Description

Compute bootstrap standard error for the scaled envelope in the predictor space estimator.

Usage

```
boot.sxenv(X, Y, u, R, B)
```

Arguments

X	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
Y	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
u	Dimension of the scaled envelope in the predictor space. An integer between 0 and p.
R	The number of replications of the scales. A vector, the sum of all elements of R must be p.
B	Number of bootstrap samples. A positive integer.

Details

This function computes the bootstrap standard errors for the regression coefficients in the scaled envelope model in the predictor space by bootstrapping the residuals.

Value

The output is an p by r matrix.

bootse	The standard error for elements in beta computed by bootstrap.

Examples

```
data(sales)
Y <- sales[, 1:3]
X <- sales[, 4:7]
R <- rep(1, 4)

u <- u.sxenv(X, Y, R)
u

B <- 100
## Not run: bootse <- boot.sxenv(X, Y, 2, R, B)
## Not run: bootse
```

---

boot.xenv *Bootstrap for xenv*

Description

Compute bootstrap standard error for the envelope estimator.

Usage

```
boot.xenv(X, Y, u, B)
```

Arguments

X	Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.
Y	Responses. An n by r matrix, r is the number of responses. The response can be univariate or multivariate and must be continuous variable.
u	Dimension of the envelope. An integer between 0 and p.
B	Number of bootstrap samples. A positive integer.

Details

This function computes the bootstrap standard errors for the regression coefficients in the envelope model in predictor space by bootstrapping the residuals.

Value

The output is a p by r matrix.

bootse	The standard error for elements in beta computed by bootstrap.

Examples

```
data(wheatprotein)
X <- wheatprotein[, 1:6]
Y <- wheatprotein[, 7]

## Not run: B <- 100
## Not run: bootse <- boot.xenv(X, Y, 2, B)
## Not run: bootse
```

---

contr *Contraction matrix*

Description

Generate contraction matrix.

Usage

```
contr(d)
```

Arguments

d	Dimension of the contraction matrix. A positive integer.

Details

The contraction and expansion matrices are links between the "vec" operator and "vech"operator: for an d by d symmetric matrix A, vech(A) = contr(d) * vec(A), and vec(A) = expan(d) * vech(A). The "vec" operator stacks the matrix A into an d ^ 2 dimensional vector columnwise. The "vech" operator stacks the lower triangle or the upper triangle of a symmetric matrix into an d * (d + 1) / 2 vector. For more details of "vec", "vech", contraction and expansion matrix, refer to Henderson and Searle (1979).

Value

The output is a matrix.

contrMatrix	A contraction matrix that has dimension d * (d + 1) / 2 by d ^ 2.

References

Henderson, H. V., and Searle, S. R. (1979). Vec and Vech operators for matrices, with some uses in Jacobians and multivariate statistics. Canadian J. Statist. 7, 65 - 81.

Examples

```
contr(3)
```

cv.env *Cross validation for env*

Description

Compute the prediction error for the envelope estimator using m-fold cross validation.

Usage

```
cv.env(X, Y, u, m, nperm)
```

Arguments

X	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
Y	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
u	Dimension of the envelope. An integer between 0 and r.
m	A positive integer that is used to indicate m-fold cross validation.
nperm	A positive integer indicating number of permutations of the observations, m-fold cross validation is run on each permutation.

Details

This function computes prediction errors using m-fold cross validation. For a fixed dimension u, the data is randomly partitioned into m parts, each part is in turn used for testing for the prediction performance while the rest m-1 parts are used for training. This process is repeated for `nperm` times, and average prediction error is reported. As Y is multivariate, the identity inner product is used for computing the prediction errors.

Value

The output is a real nonnegative number.

cvPE	The prediction error estimated by m-fold cross validation.

Examples

```
data(wheatprotein)
X <- wheatprotein[, 8]
Y <- wheatprotein[, 1:6]
u <- u.env(X, Y)
u

m <- 5
nperm <- 50
cvPE <- cv.env(X, Y, 1, m, nperm)
cvPE
```

cv.genv *Cross validation for genv*

Description

Compute the prediction error for the groupwise envelope estimator using m-fold cross validation.

Usage

```
cv.genv(X, Y, Z, u, m, nperm)
```

Arguments

X	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
Y	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
Z	A group indicator vector of length n, where n denotes the number of observations.
u	Dimension of the groupwise envelope. An integer between 0 and r.
m	A positive integer that is used to indicate m-fold cross validation.
nperm	A positive integer indicating number of permutations of the observations, m-fold cross validation is run on each permutation.

Details

This function computes prediction errors using m-fold cross validation. For a fixed dimension u, the data is randomly partitioned into m parts, each part is in turn used for testing for the prediction performance while the rest m-1 parts are used for training. This process is repeated for nperm times, and average prediction error is reported. As Y is multivariate, the identity inner product is used for computing the prediction errors.

Value

The output is a real nonnegative number.

cvPE	The prediction error estimated by m-fold cross validation.

Examples

```
data(fiberpaper)
X <- fiberpaper[ , c(5, 7)]
Y <- fiberpaper[ , 1:3]
Z <- as.numeric(fiberpaper[ , 6] > mean(fiberpaper[ , 6]))

## Not run: m <- 5
## Not run: nperm <- 50
```

```
## Not run: cvPE <- cv.genv(X, Y, Z, 2, m, nperm)
## Not run: cvPE
```

`cv.henv` *Cross validation for henv*

Description

Compute the prediction error for the heteroscedastic envelope estimator using m-fold cross validation.

Usage

```
cv.henv(X, Y, u, m, nperm)
```

Arguments

`X`	A group indicator vector of length n, where n denotes the number of observations.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`u`	Dimension of the heteroscedastic envelope. An integer between 0 and r.
`m`	A positive integer that is used to indicate `m`-fold cross validation.
`nperm`	A positive integer indicating number of permutations of the observations, m-fold cross validation is run on each permutation.

Details

This function computes prediction errors using m-fold cross validation. For a fixed dimension u, the data is randomly partitioned into m parts, each part is in turn used for testing for the prediction performance while the rest m-1 parts are used for training. This process is repeated for `nperm` times, and average prediction error is reported. As Y is multivariate, the identity inner product is used for computing the prediction errors.

Value

The output is a real nonnegative number.

`cvPE`	The prediction error estimated by m-fold cross validation.

Examples

```
data(waterstrider)
X <- waterstrider[ , 1]
Y <- waterstrider[ , 2:5]

m <- 5
nperm <- 50

## Not run: cvPE <- cv.henv(X, Y, 2, m, nperm)
## Not run: cvPE
```

`cv.logit.env` *Cross validation for logit.env*

Description

Compute the prediction error for the envelope estimator in logistic regression using m-fold cross validation.

Usage

```
cv.logit.env(X, Y, u, m, nperm)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.
`Y`	Response. An n by 1 matrix. The univariate response must be binary.
`u`	Dimension of the envelope. An integer between 0 and p.
`m`	A positive integer that is used to indicate m-fold cross validation.
`nperm`	A positive integer indicating number of permutations of the observations, m-fold cross validation is run on each permutation.

Details

This function computes prediction errors using m-fold cross validation. For a fixed dimension u, the data is randomly partitioned into m parts, each part is in turn used for testing for the prediction performance while the rest m-1 parts are used for training. This process is repeated for `nperm` times, and average prediction error is reported.

Value

The output is a real nonnegative number.

`cvPE`	The prediction error estimated by m-fold cross validation.

Examples

```
data(horseshoecrab)
X1 <- as.numeric(horseshoecrab[ , 1] == 2)
X2 <- as.numeric(horseshoecrab[ , 1] == 3)
X3 <- as.numeric(horseshoecrab[ , 1] == 4)
X4 <- as.numeric(horseshoecrab[ , 2] == 2)
X5 <- as.numeric(horseshoecrab[ , 2] == 3)
X6 <- horseshoecrab[ , 3]
X7 <- horseshoecrab[ , 5]
X <- cbind(X1, X2, X3, X4, X5, X6, X7)
Y <- as.numeric(ifelse(horseshoecrab[ , 4] > 0, 1, 0))

m <- 5
nperm <- 50
## Not run: cvPE <- cv.logit.env(X, Y, 1, m, nperm)
## Not run: cvPE
```

`cv.penv` *Cross validation for penv*

Description

Compute the prediction error for the partial envelope estimator using m-fold cross validation.

Usage

```
cv.penv(X1, X2, Y, u, m, nperm)
```

Arguments

X1	Predictors of main interest. An n by p1 matrix, n is the number of observations, and p1 is the number of main predictors. The predictors can be univariate or multivariate, discrete or continuous.
X2	Covariates, or predictors not of main interest. An n by p2 matrix, p2 is the number of covariates.
Y	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
u	Dimension of the envelope. An integer between 0 and r.
m	A positive integer that is used to indicate m-fold cross validation.
nperm	A positive integer indicating number of permutations of the observations, m-fold cross validation is run on each permutation.

Details

This function computes prediction errors using m-fold cross validation. For a fixed dimension u, the data is randomly partitioned into m parts, each part is in turn used for testing for the prediction performance while the rest m-1 parts are used for training. This process is repeated for nperm times, and average prediction error is reported. As Y is multivariate, the identity inner product is used for computing the prediction errors.

Value

The output is a real nonnegative number.

cvPE The prediction error estimated by m-fold cross validation.

Examples

```
data(fiberpaper)
X1 <- fiberpaper[, 7]
X2 <- fiberpaper[, 5:6]
Y <- fiberpaper[, 1:4]

m <- 5
nperm <- 50
## Not run: cvPE <- cv.penv(X1, X2, Y, 1, m, nperm)
## Not run: cvPE
```

---

cv.pois.env *Cross validation for pois.env*

Description

Compute the prediction error for the envelope estimator in poisson regression using m-fold cross validation.

Usage

```
cv.pois.env(X, Y, u, m, nperm)
```

Arguments

X	Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.
Y	Response. An n by 1 matrix. The univariate response must be counts.
u	Dimension of the envelope. An integer between 0 and p.
m	A positive integer that is used to indicate m-fold cross validation.
nperm	A positive integer indicating number of permutations of the observations, m-fold cross validation is run on each permutation.

Details

This function computes prediction errors using m-fold cross validation. For a fixed dimension u, the data is randomly partitioned into m parts, each part is in turn used for testing for the prediction performance while the rest m-1 parts are used for training. This process is repeated for `nperm` times, and average prediction error is reported.

Value

The output is a real nonnegative number.

`cvPE` The prediction error estimated by m-fold cross validation.

Examples

```
data(horseshoecrab)
X1 <- as.numeric(horseshoecrab[ , 1] == 2)
X2 <- as.numeric(horseshoecrab[ , 1] == 3)
X3 <- as.numeric(horseshoecrab[ , 1] == 4)
X4 <- as.numeric(horseshoecrab[ , 2] == 2)
X5 <- as.numeric(horseshoecrab[ , 2] == 3)
X6 <- horseshoecrab[ , 3]
X7 <- horseshoecrab[ , 5]
X <- cbind(X1, X2, X3, X4, X5, X6, X7)
Y <- horseshoecrab[ , 4]

m <- 5
nperm <- 50
## Not run: cvPE <- cv.pois.env(X, Y, 1, m, nperm)
## Not run: cvPE
```

`cv.senv` *Cross validation for senv*

Description

Compute the prediction error for the scaled envelope estimator using m-fold cross validation.

Usage

```
cv.senv(X, Y, u, m, nperm)
```

Arguments

`X` Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.

`Y` Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.

u	Dimension of the scaled envelope. An integer between 0 and r.
m	A positive integer that is used to indicate m-fold cross validation.
nperm	A positive integer indicating number of permutations of the observations, m-fold cross validation is run on each permutation.

Details

This function computes prediction errors using m-fold cross validation. For a fixed dimension u, the data is randomly partitioned into m parts, each part is in turn used for testing for the prediction performance while the rest m-1 parts are used for training. This process is repeated for nperm times, and average prediction error is reported. As Y is multivariate, the identity inner product is used for computing the prediction errors.

Value

The output is a real nonnegative number.

cvPE	The prediction error estimated by m-fold cross validation.

Examples

```
data(sales)
X <- sales[, 1:3]
Y <- sales[, 4:7]

m <- 5
nperm <- 50
## Not run: cvPE <- cv.senv(X, Y, 2, m, nperm)
## Not run: cvPE
```

---

cv.stenv *Cross validation for stenv*

Description

Compute the prediction error for the simultaneous envelope estimator using m-fold cross validation.

Usage

```
cv.stenv(X, Y, q, u, m, nperm)
```

Arguments

X	Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.
Y	Responses. An n by r matrix, r is the number of responses. The response can be univariate or multivariate and must be continuous variable.
q	Dimension of the X-envelope. An integer between 0 and p.

`u`	Dimension of the Y-envelope. An integer between 0 and r.
`m`	A positive integer that is used to indicate m-fold cross validation.
`nperm`	A positive integer indicating number of permutations of the observations, m-fold cross validation is run on each permutation.

Details

This function computes prediction errors using m-fold cross validation. For a fixed dimension (q, u), the data is randomly partitioned into m parts, each part is in turn used for testing for the prediction performance while the rest m-1 parts are used for training. This process is repeated for `nperm` times, and average prediction error is reported. If Y is multivariate, the identity inner product is used for computing the prediction errors.

Value

The output is a real nonnegative number.

`cvPE`	The prediction error estimated by m-fold cross validation.

Examples

```
data(fiberpaper)
X <- fiberpaper[, 5:7]
Y <- fiberpaper[, 1:4]

m <- 5
nperm <- 50
## Not run: cvPE <- cv.stenv(X, Y, 2, 3, m, nperm)
## Not run: cvPE
```

`cv.sxenv` *Cross validation for sxenv*

Description

Compute the prediction error for the scaled envelope in the predictor space estimator using m-fold cross validation.

Usage

```
cv.sxenv(X, Y, u, R, m, nperm)
```

Arguments

X	Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.
Y	Responses. An n by r matrix, r is the number of responses. The response can be univariate or multivariate and must be continuous variable.
u	Dimension of the scaled envelope. An integer between 0 and r.
R	The number of replications of the scales. A vector, the sum of all elements of R must be p.
m	A positive integer that is used to indicate m-fold cross validation.
nperm	A positive integer indicating number of permutations of the observations, m-fold cross validation is run on each permutation.

Details

This function computes prediction errors using m-fold cross validation. For a fixed dimension u, the data is randomly partitioned into m parts, each part is in turn used for testing for the prediction performance while the rest m-1 parts are used for training. This process is repeated for `nperm` times, and average prediction error is reported. As Y is multivariate, the identity inner product is used for computing the prediction errors.

Value

The output is a real nonnegative number.

cvPE	The prediction error estimated by m-fold cross validation.

Examples

```
data(sales)
Y <- sales[, 1:3]
X <- sales[, 4:7]
R <- rep(1, 4)

m <- 5
nperm <- 50
## Not run: cvPE <- cv.sxenv(X, Y, 2, R, m, nperm)
## Not run: cvPE
```

---

cv.xenv *Cross validation for xenv*

Description

Compute the prediction error for the envelope estimator using m-fold cross validation.

Usage

```
cv.xenv(X, Y, u, m, nperm)
```

Arguments

X	Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.
Y	Responses. An n by r matrix, r is the number of responses. The response can be univariate or multivariate and must be continuous variable.
u	Dimension of the envelope. An integer between 0 and p.
m	A positive integer that is used to indicate m-fold cross validation.
nperm	A positive integer indicating number of permutations of the observations, m-fold cross validation is run on each permutation.

Details

This function computes prediction errors using m-fold cross validation. For a fixed dimension u, the data is randomly partitioned into m parts, each part is in turn used for testing for the prediction performance while the rest m-1 parts are used for training. This process is repeated for `nperm` times, and average prediction error is reported. If Y is multivariate, the identity inner product is used for computing the prediction errors.

Value

The output is a real nonnegative number.

cvPE	The prediction error estimated by m-fold cross validation.

Examples

```
data(wheatprotein)
X <- wheatprotein[, 1:6]
Y <- wheatprotein[, 7]

m <- 5
nperm <- 50
## Not run: cvPE <- cv.xenv(X, Y, 2, m, nperm)
## Not run: cvPE
```

---

d.stenv *Select the rank of beta*

Description

This function outputs the rank selected by a chi-squared test developed by Bura and Cook (2003) with specified significance level for the beta.

Usage

```
d.stenv(X, Y, alpha = 0.01)
```

Arguments

X	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
Y	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
alpha	Significance level for testing. The default is 0.01.

Details

This function estimate the rank of beta using a chi-squared test. The test statistic and degrees of freedom are described in Bura and Cook (2003).

Value

rank.beta	Rank of beta guided by the Bura-Cook estimator.

References

Bura, E. and Cook, R. D. (2003). Rank estimation in reduced-rank regression. Journal of Multivariate Analysis, 87, 159 - 176.

---

env — *Fit the envelope model*

Description

Fit the envelope model in multivariate linear regression with dimension u.

Usage

```
env(X, Y, u, asy = TRUE, init = NULL)
```

Arguments

X	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
Y	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
u	Dimension of the envelope. An integer between 0 and r.
asy	Flag for computing the asymptotic variance of the envelope estimator. The default is `TRUE`. When p and r are large, computing the asymptotic variance can take much time and memory. If only the envelope estimators are needed, the flag can be set to `asy = FALSE`.

init	The user-specified value of Gamma for the envelope subspace. An r by u matrix. The default is the one generated by function envMU.

Details

This function fits the envelope model to the responses and predictors,

$$Y = \mu + \Gamma\eta X + \varepsilon, \Sigma = \Gamma\Omega\Gamma' + \Gamma_0\Omega_0\Gamma_0'$$

using the maximum likelihood estimation. When the dimension of the envelope is between 1 and r-1, the starting value and blockwise coordinate descent algorithm in Cook et al. (2016) is implemented. When the dimension is r, then the envelope model degenerates to the standard multivariate linear regression. When the dimension is 0, it means that X and Y are uncorrelated, and the fitting is different.

Value

The output is a list that contains the following components:

beta	The envelope estimator of the regression coefficients.
Sigma	The envelope estimator of the error covariance matrix.
Gamma	An orthonormal basis of the envelope subspace.
Gamma0	An orthonormal basis of the complement of the envelope subspace.
eta	The coordinates of beta with respect to Gamma.
Omega	The coordinates of Sigma with respect to Gamma.
Omega0	The coordinates of Sigma with respect to Gamma0.
mu	The estimated intercept.
loglik	The maximized log likelihood function.
covMatrix	The asymptotic covariance of vec(beta). The covariance matrix returned are asymptotic. For the actual standard errors, multiply by 1 / n.
asySE	The asymptotic standard error for elements in beta under the envelope model. The standard errors returned are asymptotic, for actual standard errors, multiply by 1 / sqrt(n).
ratio	The asymptotic standard error ratio of the standard multivariate linear regression estimator over the envelope estimator, for each element in beta.
n	The number of observations in the data.

References

Cook, R. D., Li, B. and Chiaromente, F. (2010). Envelope Models for Parsimonious and Efficient Multivariate Linear Regression (with discussion). Statist. Sinica 20, 927- 1010.

Cook, R. D., Forzani, L. and Su, Z. (2016) A Note on Fast Envelope Estimation. Journal of Multivariate Analysis. 150, 42-54.

Examples

```
data(wheatprotein)
X <- wheatprotein[, 8]
Y <- wheatprotein[, 1:6]
u <- u.env(X, Y)
u

m <- env(X, Y, 1)
m
m$beta
```

---

envMU — *Estimate the envelope subspace*

Description

Estimate the envelope subspace with specified dimension.

Usage

```
envMU(M, U, u, initial = NULL)
```

Arguments

`M`	M matrix in the envelope objective function. An r by r semi-positive definite matrix.
`U`	U matrix in the envelope objective function. An r by r semi-positive definite matrix.
`u`	Dimension of the envelope. An integer between 0 and r.
`initial`	The user-specified value of Gamma for the envelope subspace.

Details

This function estimate the envelope subspace using an non-Grassmann optimization algorithm. The starting value and optimization algorithm is described in Cook et al. (2016).

Value

`Gammahat`	The orthonormal basis of the envelope subspace.
`Gamma0hat`	The orthonormal basis of the complement of the envelope subspace.
`objfun`	The minimized objective function.

References

Cook, R. D., Forzani, L. and Su, Z. (2016) A Note on Fast Envelope Estimation. Journal of Multivariate Analysis. 150, 42-54.

`expan` *Expansion matrix*

Description

Generate expansion matrix.

Usage

```
expan(d)
```

Arguments

`d`	Dimension of the expansion matrix. A positive integer.

Details

The contraction and expansion matrices are links between the "vec" operator and "vech"operator: for an d by d symmetric matrix A, vech(A) = contr(d) * vec(A), and vec(A) = expan(d) * vech(A). The "vec" operator stacks the matrix A into an d ^ 2 dimensional vector columnwise. The "vech" operator stacks the lower triangle or the upper triangle of a symmetric matrix into an d * (d + 1) / 2 vector. For more details of "vec", "vech", contraction and expansion matrix, refer to Henderson and Searle (1979).

Value

The output is a matrix.

`expanMatrix`	An expansion matrix that has dimension d ^ 2 by d * (d + 1) / 2.

References

Henderson, H. V., and Searle, S. R. (1979). Vec and Vech operators for matrices, with some uses in Jacobians and multivariate statistics. Canadian J. Statist. 7, 65 - 81.

Examples

```
expan(3)
```

`fiberpaper` *Pulp and Paper Data*

Description

Pulp and paper property

Usage

```
data("fiberpaper")
```

Format

A data frame with 62 observations on the following 8 variables.

- `V1` Breaking length.
- `V2` Elastic modulus.
- `V3` Stress at failure.
- `V4` Burst strength.
- `V5` Arithmetic fiber length.
- `V6` Long fiber fraction.
- `V7` Fine fiber fraction.
- `V8` Zero span tensile.

Details

This data set contains measurements of properties of pulp fibers and the paper made from them.

References

Johnson, R.A. and Wichern, D.W. (2007). Applied Multivariate Statistical Analysis, 6th edition.

`GE` *Gaussian elimination*

Description

Gaussian elimination with partial pivoting.

Usage

```
GE(A)
```

Arguments

A	An n by p matrix. n must be greater than or equal to p.

Details

This function performs Gaussian elimination to the input matrix and returns the locations of pivoting elements.

Value

The output is a vector of length n.

idx	A vector of length n. The first p elements are the indices of the pivoting elements, ordered accoridng to columns, and the rest n-p elements are the remaining indices from 1 to n.

---

genv — *Fit the groupwise envelope model*

Description

Fit the groupwise envelope model in multivariate linear regression with dimension u.

Usage

```
genv(X, Y, Z, u, asy = TRUE, fit = TRUE, init = NULL)
```

Arguments

X	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
Y	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
Z	A group indicator vector of length n, where n denotes the number of observations.
u	Dimension of the groupwise envelope. An integer between 0 and r.
asy	Flag for computing the asymptotic variance of the envelope estimator. The default is `TRUE`. When p and r are large, computing the asymptotic variance can take much time and memory. If only the envelope estimators are needed, the flag can be set to `asy = FALSE`.
fit	Flag for computing the fitted response. The default is `TRUE`.
init	The user-specified value of Gamma for the groupwise envelope subspace. An r by u matrix. The default is the one generated by function genvMU.

Details

This function fits the groupwise envelope model to the responses and predictors,

$$Y_{(l)j} = \mu_{(l)} + \Gamma\eta_{(l)j}X_{(l)j} + \varepsilon_{(l)j}, \Sigma_{(l)} = \Gamma\Omega_{(l)}\Gamma' + \Gamma_0\Omega_0\Gamma_0'$$

for l = 1, ..., L, using the maximum likelihood estimation. When the dimension of the groupwise envelope is between 1 and r-1, the starting value and blockwise coordinate descent algorithm in Cook et al. (2016) is implemented. When the dimension is r, then the envelope model degenerates to the standard multivariate linear regression. When the dimension is 0, it means that X and Y are uncorrelated, and the fitting is different. When L is 1, the groupwise envelope model degenerates to the envelope model in Cook et al. (2010).

Value

The output is a list that contains the following components:

`beta`	A list of r by p matrices for the estimator of regression coefficients. `beta`[[i]] indicates the estimator of regression coefficient for the ith group.
`Sigma`	A list of the estimator of error covariance matrix. `Sigma`[[i]] contains the estimated covariance matrix for the ith group.
`Gamma`	An orthonormal basis of the groupwise envelope subspace.
`Gamma0`	An orthonormal basis of the complement of the groupwise envelope subspace.
`eta`	The coordinates of beta with respect to Gamma.
`Omega`	The coordinates of Sigma with respect to Gamma.
`Omega0`	The coordinates of Sigma with respect to Gamma0.
`mu`	The estimator of group mean. A r by L matrix whose ith column contains the mean for the group.
`loglik`	The maximized log likelihood function.
`covMatrix`	A list of the asymptotic covariance of vec(beta). The covariance matrix returned are asymptotic. For the actual standard errors, multiply by 1 / n. `covMatrix`[[i]] contains the asymptotic covariance matrix for the ith group.
`asySE`	A list of the asymptotic standard error for elements in beta under the groupwise envelope model. The standard errors returned are asymptotic, for actual standard errors, multiply by 1 / sqrt(n). `asySE`[[i]] contains the asymptotic standard error for elements in beta[[i]].
`ratio`	A list of the asymptotic standard error ratio of the standard multivariate linear regression estimator over the groupwise envelope estimator, for each element in beta. `ratio`[[i]] contains the asymptotic standard error ratio for the ith group.
`groupInd`	A matrix containing the unique values of group indicators. The matrix has p rows.
`n`	The number of observations in the data.
`ng`	The number of observations in each group.
`Yfit`	Fitted responses.

References

Park, Y., Su, Z. and Zhu, H. (2017) Groupwise envelope models for Imaging Genetic Analysis. Biometrics, to appear.

Cook, R. D., Li, B. and Chiaromente, F. (2010). Envelope Models for Parsimonious and Efficient Multivariate Linear Regression (with discussion). Statist. Sinica 20, 927- 1010.

Cook, R. D., Forzani, L. and Su, Z. (2016) A Note on Fast Envelope Estimation. Journal of Multivariate Analysis. 150, 42-54.

Examples

```
data(fiberpaper)
X <- fiberpaper[ , c(5, 7)]
Y <- fiberpaper[ , 1:3]
Z <- as.numeric(fiberpaper[ , 6] > mean(fiberpaper[ , 6]))
## Not run: u <- u.genv(X, Y, Z)
## Not run: u
## Not run: m <- genv(X, Y, Z, 2)
```

`genvMU` *Estimate the groupwise envelope subspace*

Description

Estimate the groupwise envelope subspace with specified dimension.

Usage

```
genvMU(M, U, MU, u, n, ng, L, initial = NULL)
```

Arguments

`M`	A matrix M for the non-Grassmann manifold optimization problem in Cook et al. (2016)
`U`	A matrix U for the non-Grassmann manifold optimization problem in Cook et al. (2016)
`MU`	Sum of matrix M and U.
`u`	A given dimension of the groupwise envelope space. It should be an interger between 0 and r.
`n`	The number of observations.
`ng`	A L by 1 vector of the number of observations in each group.
`L`	The number of groups.
`initial`	The user-specified value of Gamma for the envelope subspace.

Value

Gamma
: An r by u matrix for the orthonormal basis of the groupwise envelope subspace.

Gamma0
: An r by $(r-u)$ matrix for the orthonormal basis of the complement of the groupwise envelope subspace.

References

Cook, R. D., Forzani, L. and Su, Z. (2016) A Note on Fast Envelope Estimation. Journal of Multivariate Analysis. 150, 42-54.

---

ginv *Generalized Inverse of a Matrix*

Description

Calculates the Moore-Penrose generalized inverse of a matrix X.

Usage

```
ginv(X, tol = sqrt(.Machine$double.eps))
```

Arguments

X
: Matrix for which the Moore-Penrose inverse is required

tol
: A relative tolerance to detect zero sigular values

References

Venables, W.N. and Ripley, B.D. (1999) Modern Applied statistics with S-PLUS. Third Edition. Springer. p.100

---

henv *Fit the heteroscedastic envelope model*

Description

Fit the heteroscedastic envelope model with dimension u.

Usage

```
henv(X, Y, u, asy = TRUE, fit = TRUE, init = NULL)
```

Arguments

`X` A group indicator vector of length n, where n denotes the number of observations.

`Y` Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.

`u` Dimension of the heteroscedastic envelope. An integer between 0 and r.

`asy` Flag for computing the asymptotic variance of the envelope estimator. The default is `TRUE`. When p and r are large, computing the asymptotic variance can take much time and memory. If only the envelope estimators are needed, the flag can be set to `asy = FALSE`.

`fit` Flag for computing the fitted response. The default is `TRUE`.

`init` The user-specified value of Gamma for the heteroscedastic envelope subspace. An r by u matrix. The default is the one generated by function henvMU.

Details

This function fits the heteroscedastic envelope model to the responses,

$$Y_{(i)j} = \mu + \Gamma\eta_{(i)} + \varepsilon_{(i)j}, \Sigma_{(i)} = \Gamma\Omega_{(i)}\Gamma' + \Gamma_0\Omega_0\Gamma_0'$$

for i = 1, ..., p, using the maximum likelihood estimation. When the dimension of the heteroscedastic envelope is between 1 and r-1, the starting value and blockwise coordinate descent algorithm in Cook et al. (2016) is implemented. When the dimension is r, then the envelope model degenerates to the standard multivariate linear regression for comparing group means. When the dimension is 0, it means there is no any group effect, and the fitting is different.

Value

The output is a list that contains the following components:

`beta` The heteroscedastic envelope estimator of the group main effect. An r by p matix, the ith column of the matrix contains the main effect for the ith group.

`Sigma` A list of the heteroscedastic envelope estimator of the error covariance matrix. `Sigma`[[i]] contains the estimated covariance matrix for the ith group.

`Gamma` An orthonormal basis of the heteroscedastic envelope subspace.

`Gamma0` An orthonormal basis of the complement of the heteroscedastic envelope subspace.

`eta` A list of the coordinates of beta with respect to Gamma. `eta` [[i]] indicates the coordinates of the main effect of the ith group with respect to Gamma.

`Omega` A list of the coordinates of Sigma with respect to Gamma. `Omega`[[i]] indicates the coordinates of the covariance matrix of the ith group with respect to Gamma.

`Omega0` The coordinates of Sigma with respect to Gamma0.

`mu` The heteroscedastic envelope estimator of the grand mean. A r by 1 matrix.

`mug` A list of the heteroscedastic envelope estimator of the group mean. An r by p matix, the ith column of the matrix contains the mean for the ith group.

loglik	The maximized log likelihood function.
covMatrix	The asymptotic covariance of (mu, vec(beta)')'. An r(p + 1) by r(p + 1) matrix. The covariance matrix returned are asymptotic. For the actual standard errors, multiply by 1 / n.
asySE	The asymptotic standard error for elements in beta under the heteroscedastic envelope model. An r by p matrix. The standard errors returned are asymptotic, for actual standard errors, multiply by 1 / sqrt(n).
ratio	The asymptotic standard error ratio of the standard multivariate linear regression for comparing group means over the heteroscedastic envelope estimator, for each element in beta. An r by p matrix.
groupInd	A matrix containing the unique values of group indicators. The matrix has p rows.
n	The number of observations in the data.
ng	The number of observations in each group.
Yfit	Fitted responses.

References

Su, Z. and Cook, R. D. (2013) Estimation of Multivariate Means with Heteroscedastic Error Using Envelope Models. Statistica Sinica, 23, 213-230.

Cook, R. D., Li, B. and Chiaromente, F. (2010). Envelope Models for Parsimonious and Efficient Multivariate Linear Regression (with discussion). Statist. Sinica 20, 927- 1010.

Cook, R. D., Forzani, L. and Su, Z. (2016) A Note on Fast Envelope Estimation. Journal of Multivariate Analysis. 150, 42-54.

Examples

```
data(waterstrider)
X <- waterstrider[ , 1]
Y <- waterstrider[ , 2:5]

## Not run: u <- u.henv(X, Y)
## Not run: u

## Not run: m <- henv(X, Y, 2)
```

---

henvMU *Estimate the heteroscedastic envelope subspace*

Description

Estimate the heteroscedastic envelope subspace with specified dimension.

Usage

```
henvMU(M, U, MU, u, n, ng, L, initial = NULL)
```

Arguments

M	A matrix M for the non-Grassmann manifold optimization problem in Cook et al. (2016)
U	A matrix U for the non-Grassmann manifold optimization problem in Cook et al. (2016)
MU	Sum of matrix M and U.
u	A given dimension of the heteroscedastic envelope space. It should be an interger between 0 and r.
n	The number of observations.
ng	A L by 1 vector of the number of observations in each group.
L	The number of groups.
initial	The user-specified value of Gamma for the envelope subspace.

Value

Gamma	An r by u matrix for the orthonormnal basis of the heteroscedastic envelope subspace.
Gamma0	An r by $(r-u)$ matrix for the orthonornal basis of the complement of the heteroscedastic envelope subspace.

References

Cook, R. D., Forzani, L. and Su, Z. (2016) A Note on Fast Envelope Estimation. Journal of Multivariate Analysis. 150, 42-54.

---

horseshoecrab *Horseshoe Crab Data*

Description

The number of satellite male crabs near a female crab upon characteristic of the female horseshoe crabs.

Usage

```
data("horseshoecrab")
```

Format

A data frame with 173 observations on the following 5 variables.

`V1` Color.

`V2` Condition of spine.

`V3` Width of shell.

`V4` Satellite.

`V5` Weight.

Details

This data set contains the number of satellite male crabs and characteristics of the female horseshoe crabs.

References

Agresti, A. (2007). An Introduction to Categorical Data Analysis, 2nd edition.

`logit.env` *Fit the envelope model in logistic regression*

Description

Fit the envelope model in logistic regression with dimension u.

Usage

```
logit.env(X, Y, u, asy = TRUE, init = NULL)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.
`Y`	Response. An n by 1 matrix. The univariate response must be binary.
`u`	Dimension of the envelope. An integer between 0 and p.
`asy`	Flag for computing the asymptotic variance of the envelope estimator. The default is `TRUE`. When p and r are large, computing the asymptotic variance can take much time and memory. If only the envelope estimators are needed, the flag can be set to `asy = FALSE`.
`init`	The user-specified value of Gamma for the envelope subspace in logistic regression. An p by u matrix. The default is the one generated by function logit.envMU.

Details

This function fits the envelope model in logistic regression,

$$Y = exp(\mu + \beta' X)/(1 + exp(\mu + \beta' X)), \Sigma_X = \Gamma\Omega\Gamma' + \Gamma_0\Omega_0\Gamma_0'$$

using the maximum likelihood estimation. When the dimension of the envelope is between 1 and p-1, the starting value and blockwise coordinate descent algorithm in Cook et al. (2016) is implemented. This model works the best when X is multivariate normal.

Value

The output is a list that contains the following components:

`beta`	The envelope estimator of the canonical parameter.
`SigmaX`	The envelope estimator of the covariance matrix of X.
`Gamma`	An orthonormal basis of the envelope subspace.
`Gamma0`	An orthonormal basis of the complement of the envelope subspace.
`eta`	The estimated beta of the canonical parameter with respect to Gamma.
`Omega`	The coordinates of SigmaX with respect to Gamma.
`Omega0`	The coordinates of SigmaX with respect to Gamma0.
`mu`	The estimated intercept of the canonical parameter.
`loglik`	The maximized log likelihood function.
`covMatrix`	The asymptotic covariance of vec(beta). The covariance matrix returned are asymptotic. For the actual standard errors, multiply by 1 / n.
`asySE`	The asymptotic standard error for elements in beta under the envelope model. The standard errors returned are asymptotic, for actual standard errors, multiply by 1 / sqrt(n).
`ratio`	The asymptotic standard error ratio of the standard multivariate linear regression estimator over the envelope estimator, for each element in beta.
`n`	The number of observations in the data.

References

Cook, R. D., Zhang, X. (2015). Foundations for Envelope Models and Methods. Journal of the American Statistical Association 110, 599 - 611.

Cook, R. D., Forzani, L. and Su, Z. (2016) A Note on Fast Envelope Estimation. Journal of Multivariate Analysis. 150, 42-54.

Examples

```
data(horseshoecrab)
X1 <- as.numeric(horseshoecrab[ , 1] == 2)
X2 <- as.numeric(horseshoecrab[ , 1] == 3)
X3 <- as.numeric(horseshoecrab[ , 1] == 4)
X4 <- as.numeric(horseshoecrab[ , 2] == 2)
X5 <- as.numeric(horseshoecrab[ , 2] == 3)
```

```
X6 <- horseshoecrab[ , 3]
X7 <- horseshoecrab[ , 5]
X <- cbind(X1, X2, X3, X4, X5, X6, X7)
Y <- as.numeric(ifelse(horseshoecrab[ , 4] > 0, 1, 0))

## Not run: u <- u.logit.env(X, Y)
## Not run: u

## Not run: m <- logit.env(X, Y, 1)
## Not run: m$beta
```

`logit.envMU` *Estimate the envelope subspace in logistic regression*

Description

Estimate the envelope subspace with specified dimension in logistic regression.

Usage

```
logit.envMU(X, Y, u, initial = NULL)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Response. An n by 1 matrix. The univariate response must be binary.
`u`	Dimension of the envelope. An integer between 0 and p.
`initial`	The user-specified value of Gamma for the envelope subspace.

Details

This function estimate the envelope subspace in logistic regression using an non-Grassmann optimization algorithm. The starting value and optimization algorithm is described in Cook et al. (2016).

Value

`Gammahat`	The orthonormal basis of the envelope subspace.
`Gamma0hat`	The orthonormal basis of the complement of the envelope subspace.
`muhat`	The estimated intercept of the canonical parameter.
`etahat`	The estimated beta of the canonical parameter with respect to Gamma.
`weighthat`	The estimated weight defined as C"(theta) / E(C"(theta)) where C(theta) is the conditional log likelihood.
`Vhat`	The estimated V defined as V = theta + (Y - mu (theta) / W).
`avar`	The asympotic covariance of vec(beta).
`objfun`	The minimized objective function.

References

Cook, R. D., Forzani, L. and Su, Z. (2016) A Note on Fast Envelope Estimation. Journal of Multivariate Analysis. 150, 42-54.

`penv` *Fit the partial envelope model*

Description

Fit the partial envelope model in multivariate linear regression with dimension u.

Usage

```
penv(X1, X2, Y, u, asy = TRUE, init = NULL)
```

Arguments

`X1`	Predictors of main interest. An n by p1 matrix, n is the number of observations, and p1 is the number of main predictors. The predictors can be univariate or multivariate, discrete or continuous.
`X2`	Covariates, or predictors not of main interest. An n by p2 matrix, p2 is the number of covariates.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`u`	Dimension of the partial envelope. An integer between 0 and r.
`asy`	Flag for computing the asymptotic variance of the partial envelope estimator. The default is `TRUE`. When p and r are large, computing the asymptotic variance can take much time and memory. If only the partial envelope estimators are needed, the flag can be set to `asy = FALSE`.
`init`	The user-specified value of Gamma for the partial envelope subspace. An r by u matrix. The default is the one generated by function envMU.

Details

This function fits the partial envelope model to the responses Y and predictors X1 and X2,

$$Y = \mu + \Gamma\eta X_1 + \beta_2 X_2 + \varepsilon, \Sigma = \Gamma\Omega\Gamma' + \Gamma_0\Omega_0\Gamma_0'$$

using the maximum likelihood estimation. When the dimension of the envelope is between 1 and r - 1, we implemented the algorithm in Su and Cook (2011), but the partial envelope subspace is estimated using the blockwise coordinate descent algorithm in Cook et al. (2016). When the dimension is r, then the partial envelope model degenerates to the standard multivariate linear regression with Y as the responses and both X1 and X2 as predictors. When the dimension is 0, X1 and Y are uncorrelated, and the fitting is the standard multivariate linear regression with Y as the responses and X2 as the predictors.

Value

The output is a list that contains the following components:

beta1	The partial envelope estimator of beta1, which is the regression coefficients for X1.
beta2	The partial envelope estimator of beta2, which is the regression coefficients for X2.
Sigma	The partial envelope estimator of the error covariance matrix.
Gamma	An orthonormal basis of the partial envelope subspace.
Gamma0	An orthonormal basis of the complement of the partial envelope subspace.
eta	The coordinates of beta1 with respect to Gamma.
Omega	The coordinates of Sigma with respect to Gamma.
Omega0	The coordinates of Sigma with respect to Gamma0.
mu	The estimated intercept in the partial envelope model.
loglik	The maximized log likelihood function.
covMatrix	The asymptotic covariance of vec(beta), while beta = (beta1, beta2). The covariance matrix returned are asymptotic. For the actual standard errors, multiply by 1 / n.
asySE1	The asymptotic standard error for elements in beta1 under the partial envelope model. The standard errors returned are asymptotic, for actual standard errors, multiply by 1 / sqrt(n).
asySE2	The asymptotic standard error for elements in beta2 under the partial envelope model. The standard errors returned are asymptotic, for actual standard errors, multiply by 1 / sqrt(n).
ratio	The asymptotic standard error ratio of the stanard multivariate linear regression estimator over the partial envelope estimator, for each element in beta1.
n	The number of observations in the data.

References

Su, Z. and Cook, R.D. (2011). Partial envelopes for efficient estimation in multivariate linear regression. Biometrika 98, 133 - 146.

Cook, R. D., Forzani, L. and Su, Z. (2016) A Note on Fast Envelope Estimation. Journal of Multivariate Analysis. 150, 42-54.

Examples

```
data(fiberpaper)
X1 <- fiberpaper[, 7]
X2 <- fiberpaper[, 5:6]
Y <- fiberpaper[, 1:4]
u <- u.penv(X1, X2, Y)
u

m <- penv(X1, X2, Y, 1)
m
m$beta1
```

pois.env *Fit the envelope model in poisson regression*

Description

Fit the envelope model in poisson regression with dimension u.

Usage

```
pois.env(X, Y, u, asy = TRUE, init = NULL)
```

Arguments

X	Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.
Y	Response. An n by 1 matrix. The univariate response must be counts.
u	Dimension of the envelope. An integer between 0 and p.
asy	Flag for computing the asymptotic variance of the envelope estimator. The default is TRUE. When p and r are large, computing the asymptotic variance can take much time and memory. If only the envelope estimators are needed, the flag can be set to asy = FALSE.
init	The user-specified value of Gamma for the envelope subspace in poisson regression. An p by u matrix. The default is the one generated by function pois.envMU.

Details

This function fits the envelope model in poisson regression,

$$Y = exp(\mu + \beta' X), \Sigma_X = \Gamma\Omega\Gamma' + \Gamma_0\Omega_0\Gamma_0'$$

using the maximum likelihood estimation. When the dimension of the envelope is between 1 and p-1, the starting value and blockwise coordinate descent algorithm in Cook et al. (2016) is implemented. This model works the best when X is multivariate normal.

Value

The output is a list that contains the following components:

beta	The envelope estimator of the canonical parameter.
SigmaX	The envelope estimator of the covariance matrix of X.
Gamma	An orthonormal basis of the envelope subspace.
Gamma0	An orthonormal basis of the complement of the envelope subspace.
eta	The estimated beta of the canonical parameter with respect to Gamma.
Omega	The coordinates of SigmaX with respect to Gamma.

Omega0	The coordinates of SigmaX with respect to Gamma0.
mu	The estimated intercept of the canonical parameter.
loglik	The maximized log likelihood function.
covMatrix	The asymptotic covariance of vec(beta). The covariance matrix returned are asymptotic. For the actual standard errors, multiply by 1 / n.
asySE	The asymptotic standard error for elements in beta under the envelope model. The standard errors returned are asymptotic, for actual standard errors, multiply by 1 / sqrt(n).
ratio	The asymptotic standard error ratio of the standard multivariate linear regression estimator over the envelope estimator, for each element in beta.
n	The number of observations in the data.

References

Cook, R. D., Zhang, X. (2015). Foundations for Envelope Models and Methods. Journal of the American Statistical Association 110, 599 - 611.

Cook, R. D., Forzani, L. and Su, Z. (2016) A Note on Fast Envelope Estimation. Journal of Multivariate Analysis. 150, 42-54.

Examples

```
data(horseshoecrab)
X1 <- as.numeric(horseshoecrab[ , 1] == 2)
X2 <- as.numeric(horseshoecrab[ , 1] == 3)
X3 <- as.numeric(horseshoecrab[ , 1] == 4)
X4 <- as.numeric(horseshoecrab[ , 2] == 2)
X5 <- as.numeric(horseshoecrab[ , 2] == 3)
X6 <- horseshoecrab[ , 3]
X7 <- horseshoecrab[ , 5]
X <- cbind(X1, X2, X3, X4, X5, X6, X7)
Y <- horseshoecrab[ , 4]

## Not run: u <- u.pois.env(X, Y)
## Not run: u

m <- pois.env(X, Y, 1)
m$beta
```

---

pois.envMU   *Estimate the envelope subspace in poisson regression*

Description

Estimate the envelope subspace with specified dimension in poisson regression.

Usage

```
pois.envMU(X, Y, u, initial = NULL)
```

Arguments

X	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
Y	Response. An n by 1 matrix. The univariate response must be counts.
u	Dimension of the envelope. An integer between 0 and p.
initial	The user-specified value of Gamma for the envelope subspace.

Details

This function estimate the envelope subspace in poisson regression using an non-Grassmann optimization algorithm. The starting value and optimization algorithm is described in Cook et al. (2016).

Value

Gammahat	The orthonormal basis of the envelope subspace.
Gamma0hat	The orthonormal basis of the complement of the envelope subspace.
muhat	The estimated intercept of the canonical parameter.
etahat	The estimated beta of the canonical parameter with respect to Gamma.
weighthat	The estimated weight defined as C"(theta) / E(C"(theta)) where C(theta) is the conditional log likelihood.
Vhat	The estimated V defined as V = theta + (Y - mu(theta) / W).
avar	The asympotic covariance of vec(beta).
objfun	The minimized objective function.

References

Cook, R. D., Forzani, L. and Su, Z. (2016) A Note on Fast Envelope Estimation. Journal of Multivariate Analysis. 150, 42-54.

---

pred.env — *Estimation or prediction for env*

Description

Perform estimation or prediction under the envelope model.

Usage

```
pred.env(m, Xnew)
```

Arguments

m	A list containing estimators and other statistics inherited from env.
Xnew	The value of X with which to estimate or predict Y. A p dimensional vector.

Details

This function evaluates the envelope model at new value Xnew. It can perform estimation: find the fitted value when X = Xnew, or prediction: predict Y when X = Xnew. The covariance matrix and the standard errors are also provided.

Value

The output is a list that contains following components.

value	The fitted value or the predicted value evaluated at Xnew.
covMatrix.estm	The covariance matrix of the fitted value at Xnew.
SE.estm	The standard error of the fitted value at Xnew.
covMatrix.pred	The covariance matrix of the predicted value at Xnew.
SE.pred	The standard error of the predicted value at Xnew.

Examples

```
data(wheatprotein)
X <- wheatprotein[, 8]
Y <- wheatprotein[, 1:6]
u <- u.env(X, Y)
u

m <- env(X, Y, 1)
m

X <- as.matrix(X)
pred.res <- pred.env(m, X[2, ])
pred.res
```

---

pred.genv — *Estimation or prediction for genv*

Description

Perform estimation or prediction under the groupwise envelope model.

Usage

```
pred.genv(m, Xnew, Znew)
```

Arguments

`m`	A list containing estimators and other statistics inherited from env.
`Xnew`	The value of X with which to estimate or predict Y. A p dimensional vector.
`Znew`	A group indicator of X.

Details

This function evaluates the envelope model at new value Xnew. It can perform estimation: find the fitted value when X = Xnew with a group indicator Z = Znew, or prediction: predict Y when X = Xnew with a group indicator Z = Znew. The covariance matrix and the standard errors are also provided.

Value

The output is a list that contains following components.

`value`	The fitted value or the predicted value evaluated at Znew.
`covMatrix.estm`	The covariance matrix of the fitted value at Znew.
`SE.estm`	The standard error of the fitted value at Znew.
`covMatrix.pred`	The covariance matrix of the predicted value at Znew.
`SE.pred`	The standard error of the predicted value at Znew.

Examples

```
data(fiberpaper)
X <- fiberpaper[ , c(5, 7)]
Y <- fiberpaper[ , 1:3]
Z <- as.numeric(fiberpaper[ , 6] > mean(fiberpaper[ , 6]))
u <- u.genv(X, Y, Z)
u

m <- genv(X, Y, Z, 2)
m

X <- as.matrix(X)
pred.res <- pred.genv(m, X[2, ], Z[2])
pred.res
```

`pred.henv` *Estimation or prediction for henv*

Description

Perform estimation or prediction under the heteroscedastic envelope model.

Usage

```
pred.henv(m, Xnew)
```

Arguments

m	A list containing estimators and other statistics inherited from henv.
Xnew	The value of X with which to estimate or predict Y. An r by 1 vector.

Details

This function evaluates the heteroscedastic envelope model at new value Xnew. It can perform estimation: find the fitted value when X = Xnew, or prediction: predict Y when X = Xnew. The covariance matrix and the standard errors are also provided.

Value

The output is a list that contains following components.

value	The fitted value or the predicted value evaluated at Xnew.
covMatrix.estm	The covariance matrix of the fitted value at Xnew.
SE.estm	The standard error of the fitted value at Xnew.
covMatrix.pred	The covariance matrix of the predicted value at Xnew.
SE.pred	The standard error of the predicted value at Xnew.

Examples

```
data(waterstrider)
X <- waterstrider[ , 1]
Y <- waterstrider[ , 2:5]


## Not run: m <- henv(X, Y, 2)

## Not run: pred.res <- pred.henv(m, X[2])
```

---

pred.logit.env *Estimation or prediction for logit.env*

Description

Perform estimation or prediction under the envelope model in logistic regression.

Usage

```
pred.logit.env(m, Xnew)
```

Arguments

`m`	A list containing estimators and other statistics inherited from xenv.
`Xnew`	The value of X with which to estimate or predict Y. A p dimensional vector.

Details

This function evaluates the envelope model at new value Xnew. It can perform estimation: find the fitted value when X = Xnew. The covariance matrix of estimation and the standard errors of estimation are also provided.

Value

The output is a list that contains following components.

`value`	The fitted value or predicted value evaluated at Xnew.
`covMatrix.estm`	The covariance matrix of the fitted value at Xnew.
`SE.estm`	The standard error of the fitted value at Xnew.

Examples

```
data(horseshoecrab)
X1 <- as.numeric(horseshoecrab[ , 1] == 2)
X2 <- as.numeric(horseshoecrab[ , 1] == 3)
X3 <- as.numeric(horseshoecrab[ , 1] == 4)
X4 <- as.numeric(horseshoecrab[ , 2] == 2)
X5 <- as.numeric(horseshoecrab[ , 2] == 3)
X6 <- horseshoecrab[ , 3]
X7 <- horseshoecrab[ , 5]
X <- cbind(X1, X2, X3, X4, X5, X6, X7)
Y <- as.numeric(ifelse(horseshoecrab[ , 4] > 0, 1, 0))

m <- logit.env(X, Y, 1)

pred.res <- pred.logit.env(m, X[1, ])
pred.res
```

`pred.penv` *Estimation or prediction for penv*

Description

Perform estimation or prediction under the partial envelope model.

Usage

```
pred.penv(m, X1new, X2new)
```

Arguments

`m`	A list containing estimators and other statistics inherited from penv.
`X1new`	The value of X1 with which to estimate or predict Y. A p1 dimensional vector.
`X2new`	The value of X2 with which to estimate or predict Y. A p2 dimensional vector.

Details

This function evaluates the partial envelope model at new value Xnew. It can perform estimation: find the fitted value when X = Xnew, or prediction: predict Y when X = Xnew. The covariance matrix and the standard errors are also provided.

Value

The output is a list that contains following components.

`value`	The fitted value or the predicted value evaluated at X1new and X2new.
`covMatrix.estm`	The covariance matrix of the fitted value at X1new and X2new.
`SE.estm`	The standard error of the fitted value at X1new and X2new.
`covMatrix.pred`	The covariance matrix of the predicted value at X1new and X2new.
`SE.pred`	The standard error of the predicted value at X1new and X2new.

Examples

```
data(fiberpaper)
X1 <- fiberpaper[, 7]
X2 <- fiberpaper[, 5:6]
Y <- fiberpaper[, 1:4]
m <- penv(X1, X2, Y, 1)

pred.res <- pred.penv(m, X1[1], X2[1, ])
pred.res
```

`pred.pois.env` *Estimation or prediction for pois.env*

Description

Perform estimation or prediction under the envelope model in poisson regression.

Usage

```
pred.pois.env(m, Xnew)
```

Arguments

`m`	A list containing estimators and other statistics inherited from xenv.
`Xnew`	The value of X with which to estimate or predict Y. A p dimensional vector.

Details

This function evaluates the envelope in poisson regression at new value Xnew. It can perform estimation: find the fitted value when X = Xnew, or prediction: predict Y when X = Xnew. The covariance matrix of estimation and the standard errors of estimation are also provided.

Value

The output is a list that contains following components.

`value`	The fitted value or the predicted value evaluated at Xnew.
`covMatrix.estm`	The covariance matrix of the fitted value at Xnew.
`SE.estm`	The standard error of the fitted value at Xnew.

Examples

```
data(horseshoecrab)
X1 <- as.numeric(horseshoecrab[ , 1] == 2)
X2 <- as.numeric(horseshoecrab[ , 1] == 3)
X3 <- as.numeric(horseshoecrab[ , 1] == 4)
X4 <- as.numeric(horseshoecrab[ , 2] == 2)
X5 <- as.numeric(horseshoecrab[ , 2] == 3)
X6 <- horseshoecrab[ , 3]
X7 <- horseshoecrab[ , 5]
X <- cbind(X1, X2, X3, X4, X5, X6, X7)
Y <- horseshoecrab[ , 4]

m <- pois.env(X, Y, 1)

pred.res <- pred.pois.env(m, X[1, ])
pred.res
```

---

pred.senv   *Estimation or prediction for senv*

Description

Perform estimation or prediction under the scaled envelope model.

Usage

```
pred.senv(m, Xnew)
```

Arguments

`m`	A list containing estimators and other statistics inherited from scale.env.
`Xnew`	The value of X with which to estimate or predict Y. A p dimensional vector.

Details

This function evaluates the scaled envelope model at new value Xnew. It can perform estimation: find the fitted value when X = Xnew, or prediction: predict Y when X = Xnew. The covariance matrix and the standard errors are also provided.

Value

The output is a list that contains following components.

`value`	The fitted value or the predicted value evaluated at Xnew.
`covMatrix.estm`	The covariance matrix of the fitted value at Xnew.
`SE.estm`	The standard error of the fitted value at Xnew.
`covMatrix.pred`	The covariance matrix of the predicted value at Xnew.
`SE.pred`	The standard error of the predicted value at Xnew.

Examples

```
data(sales)
X <- sales[, 1:3]
Y <- sales[, 4:7]

m <- senv(X, Y, 2)

pred.res <- pred.senv(m, X[2, ])
pred.res
```

---

pred.stenv *Estimation or prediction for stenv*

Description

Perform estimation or prediction under the simultaneous envelope model.

Usage

```
pred.stenv(m, Xnew)
```

Arguments

`m`	A list containing estimators and other statistics inherited from stenv.
`Xnew`	The value of X with which to estimate or predict Y. A p dimensional vector.

Details

This function evaluates the simultaneous envelope model at new value Xnew. It can perform estimation: find the fitted value when X = Xnew, or prediction: predict Y when X = Xnew. The covariance matrix and the standard errors are also provided.

Value

The output is a list that contains following components.

value	The fitted value or the predicted value evaluated at Xnew.
covMatrix.estm	The covariance matrix of the fitted value at Xnew.
SE.estm	The standard error of the fitted value at Xnew.
covMatrix.pred	The covariance matrix of the predicted value at Xnew.
SE.pred	The standard error of the predicted value at Xnew.

Examples

```
data(fiberpaper)
X <- fiberpaper[ , 5:7]
Y <- fiberpaper[ , 1:4]

m <- stenv(X, Y, 2, 3)
m

pred.res <- pred.stenv(m, X[1, ])
pred.res
```

---

pred.sxenv *Estimation or prediction for sxenv*

Description

Perform estimation or prediction under the scaled envelope model in the predictor space.

Usage

```
pred.sxenv(m, Xnew)
```

Arguments

m	A list containing estimators and other statistics inherited from stenv.
Xnew	The value of X with which to estimate or predict Y. A p dimensional vector.

Details

This function evaluates the scaled envelope model in the predictor space at new value Xnew. It can perform estimation: find the fitted value when X = Xnew, or prediction: predict Y when X = Xnew. The covariance matrix and the standard errors are also provided.

Value

The output is a list that contains following components.

value	The fitted value or the predicted value evaluated at Xnew.
covMatrix.estm	The covariance matrix of the fitted value at Xnew.
SE.estm	The standard error of the fitted value at Xnew.
covMatrix.pred	The covariance matrix of the predicted value at Xnew.
SE.pred	The standard error of the predicted value at Xnew.

Examples

```
data(sales)
Y <- sales[, 1:3]
X <- sales[, 4:7]
R <- rep(1, 4)

m <- sxenv(X, Y, 2, R)

pred.res <- pred.sxenv(m, X[1, ])
pred.res
```

---

pred.xenv *Estimation or prediction for xenv*

Description

Perform estimation or prediction under the envelope model in predictor space.

Usage

```
pred.xenv(m, Xnew)
```

Arguments

m	A list containing estimators and other statistics inherited from xenv.
Xnew	The value of X with which to estimate or predict Y. A p dimensional vector.

Details

This function evaluates the envelope model at new value Xnew. It can perform estimation: find the fitted value when X = Xnew, or prediction: predict Y when X = Xnew. The covariance matrix and the standard errors are also provided.

Value

The output is a list that contains following components.

value	The fitted value or the predicted value evaluated at Xnew.
covMatrix.estm	The covariance matrix of the fitted value at Xnew.
SE.estm	The standard error of the fitted value at Xnew.
covMatrix.pred	The covariance matrix of the predicted value at Xnew.
SE.pred	The standard error of the predicted value at Xnew.

Examples

```
data(wheatprotein)
X <- wheatprotein[, 1:6]
Y <- wheatprotein[, 7]

m <- xenv(X, Y, 2)
m

pred.res <- pred.xenv(m, X[1, ])
pred.res
```

---

pred2.env *Estimation or prediction for env*

Description

Perform estimation or prediction under the envelope model through partial envelope model.

Usage

```
pred2.env(X, Y, u, Xnew)
```

Arguments

X	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
Y	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
u	The dimension of the constructed partial envelope model.
Xnew	The value of X with which to estimate or predict Y. A p dimensional vector.

Details

This function evaluates the envelope model at new value Xnew. It can perform estimation: find the fitted value when X = Xnew, or prediction: predict Y when X = Xnew. The covariance matrix and the standard errors are also provided. Compared to predict.env, this function performs prediction through partial envelope model, which can be more accurate if the partial envelope is of smaller dimension and contains less variant material information. The constructed partial envelope model is obtained by the following: Let A0 by a p by p-1 matrix, such that A = (Xnew, A0) has full rank. Let phi1 = beta * Xnew, phi2 = beta * A0, phi = (phi1, phi2) and X = inverse of A * X = (Z1, Z2')'. Then the model Y = alpha + beta * X + epsilon can be reparameterized as Y = alpha + phi1 * Z1 + phi2 * Z2 + epsilon. We then fit a partial envelope model with Z1 as the predictor of interest, and predict at (Z1, Z2')' = inverse of A * Xnew.

Value

The output is a list that contains following components.

`value`	The fitted value or the predicted value evaluated at Xnew.
`covMatrix.estm`	The covariance matrix of the fitted value at Xnew.
`SE.estm`	The standard error of the fitted value at Xnew.
`covMatrix.pred`	The covariance matrix of the predicted value at Xnew.
`SE.pred`	The standard error of the predicted value at Xnew.

Examples

```
data(fiberpaper)
X <- fiberpaper[, 5:7]
Y <- fiberpaper[, 1:4]

u <- u.pred2.env(X, Y, X[10, ])
pred.res <- pred2.env(X, Y, u$u.bic, X[10, ])
pred.res$SE.estm
pred.res$SE.pred
```

`sales` *Sales staff Data*

Description

On the performance of a firm's sales staff

Usage

```
data("sales")
```

Format

A data frame with 50 observations on the following 7 variables.

`V1` Index of sales growth.

`V2` Index of sales profitability.

`V3` Index of new account sales.

`V4` Score on creativity.

`V5` Score on mechanical reasoning test.

`V6` Score on abstract reasoning test.

`V7` Score on Mathematics test.

Details

This data set contains 3 measures of performance and 4 tests scores.

References

Johnson, R.A., Wichern, D.W. (2007). Applied Multivariage Statistical Analysis, 6th edition.

---

senv *Fit the scaled envelope model*

Description

Fit the scaled envelope model in multivariate linear regression with dimension u.

Usage

```
senv(X, Y, u, asy = TRUE, init = NULL)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`u`	Dimension of the scaled envelope. An integer between 0 and r.
`asy`	Flag for computing the asymptotic variance of the envelope estimator. The default is `TRUE`. When p and r are large, computing the asymptotic variance can take much time and memory. If only the scaled envelope estimators are needed, the flag can be set to `asy = FALSE`.
`init`	The user-specified value of Gamma for the scaled envelope subspace. An r by u matrix. The default is the one generated by function senvMU.

Details

This function fits the scaled envelope model to the responses and predictors,

$$Y = \mu + \Lambda\Gamma\eta X + \varepsilon, \Sigma = \Lambda\Gamma\Omega\Gamma'\Lambda + \Lambda\Gamma_0\Omega_0\Gamma_0'\Lambda$$

using the maximum likelihood estimation. When the dimension of the scaled envelope is between 1 and r-1, the starting value and blockwise coordinate descent algorithm in Cook et al. (2016) is implemented. When the dimension is r, then the scaled envelope model degenerates to the standard multivariate linear regression. When the dimension is 0, it means that X and Y are uncorrelated, and the fitting is different.

Value

The output is a list that contains the following components:

`beta`	The scaled envelope estimator of the regression coefficients.
`Sigma`	The scaled envelope estimator of the error covariance matrix.
`Lambda`	The matrix of estimated scale.
`Gamma`	An orthonormal basis of the scaled envelope subspace.
`Gamma0`	An orthonormal basis of the complement of the scaled envelope subspace.
`eta`	The coordinates of beta with respect to Gamma.
`Omega`	The coordinates of Sigma with respect to Gamma.
`Omega0`	The coordinates of Sigma with respect to Gamma0.
`mu`	The estimated intercept.
`loglik`	The maximized log likelihood function.
`covMatrix`	The asymptotic covariance of vec(beta). The covariance matrix returned are asymptotic. For the actual standard errors, multiply by 1 / n.
`asySE`	The asymptotic standard error for elements in beta under the envelope model. The standard errors returned are asymptotic, for actual standard errors, multiply by 1 / sqrt(n).
`ratio`	The asymptotic standard error ratio of the standard multivariate linear regression estimator over the envelope estimator, for each element in beta.
`n`	The number of observations in the data.

References

Cook, R. D., Su, Z. (2013). Scaled Envelopes: scale Invariant and Efficient Estimation in Multivariate Linear Regression. Biometrika 100, 939 - 954.

Cook, R. D., Forzani, L. and Su, Z. (2016) A Note on Fast Envelope Estimation. Journal of Multivariate Analysis. 150, 42-54.

Examples

```
data(sales)
X <- sales[, 1:3]
Y <- sales[, 4:7]

u <- u.senv(X, Y)
u

m <- senv(X, Y, 2)
m$beta
```

---

senvMU *Estimate the scaled envelope subspace*

Description

Estimate the scaled envelope subspace with specified dimension.

Usage

```
senvMU(X, Y, u, initial = NULL)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`u`	Dimension of the envelope. An integer between 0 and r.
`initial`	The user-specified value of Gamma for the envelope subspace.

Details

This function estimate the scaled envelope subspace using an non-Grassmann optimization algorithm and nonlinear optimization using augmented Lagrange method.

Value

`Gammahat`	The orthonormal basis of the scaled envelope subspace.
`Gamma0hat`	The orthonormal basis of the complement of the scaled envelope subspace.
`Lambdahat`	The matrix of estimated scales.
`objfun`	The minimized objective function.

References

Cook, R. D., Forzani, L. and Su, Z. (2016) A Note on Fast Envelope Estimation. Journal of Multivariate Analysis. 150, 42-54.

Ye, Y., Interior algorithms for linear, quadratic, and linearly constrained non linear programming, PhD Thesis, Departments of EES stanford University, Stanford CA.

`stenv` *Fit the simultaneous envelope model*

Description

Fit the simultaneous envelope model in multivariate linear regression with dimension (q, u).

Usage

```
stenv(X, Y, q, u, asy = TRUE, Pinit = NULL, Ginit = NULL)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`q`	Dimension of the X-envelope. An integer between 0 and p.
`u`	Dimension of the Y-envelope. An integer between 0 and r.
`asy`	Flag for computing the asymptotic variance of the envelope estimator. The default is `TRUE`. When p and r are large, computing the asymptotic variance can take much time and memory. If only the envelope estimators are needed, the flag can be set to `asy = FALSE`.
`Pinit`	The user-specified value of Phi for the X-envelope subspace. An p by q matrix. The default is the one generated by function stenvMU.
`Ginit`	The user-specified value of Gamma for the Y-envelope subspace. An r by u matrix. The default is the one generated by function stenvMU.

Details

This function fits the envelope model to the responses and predictors simultaneously,

$$Y = \mu + \Gamma\eta'\Phi'X + \varepsilon, \Sigma_{Y|X} = \Gamma\Omega\Gamma' + \Gamma_0\Omega_0\Gamma_0', \Sigma_X = \Phi\Delta\Phi' + \Phi_0\Delta_0\Phi_0'$$

using the maximum likelihood estimation. When the dimension of the Y-envelope is between 1 and r-1 and the dimension of the X-envelope is between 1 and p-1, the starting value and blockwise coordinate descent algorithm in Cook et al. (2016) is implemented. When the dimension is (p, r), then the envelope model degenerates to the standard multivariate linear regression. When the dimension of the Y-envelope is r, then the envelope model degenerates to the standard envelope model. When the dimension of X-envelope is p, then the envelope model degenerates to the envelope model in the predictor space. When the dimension is 0, it means that X and Y are uncorrelated, and the fitting is different.

Value

The output is a list that contains the following components:

- `beta` The envelope estimator of the regression coefficients.
- `SigmaYcX` The envelope estimator of the error covariance matrix.
- `SigmaX` The envelope estimator of the covariance matrix of X.
- `Gamma` An orthonormal basis of the Y-envelope subspace.
- `Gamma0` An orthonormal basis of the complement of the Y-envelope subspace.
- `eta` The coordinates of beta with respect to Gamma and Phi.
- `Omega` The coordinates of SigmaYcX with respect to Gamma.
- `Omega0` The coordinates of SigmaYcX with respect to Gamma0.
- `mu` The estimated intercept.
- `Phi` An orthonormal basis of the X-envelope subspace.
- `Phi0` An orthonormal basis of the complement of the X-envelope subspace.
- `Delta` The coordinates of SigmaX with respect to Phi.
- `Delta0` The coordinates of SigmaX with respect to Phi0.
- `loglik` The maximized log likelihood function.
- `covMatrix` The asymptotic covariance of vec(beta). The covariance matrix returned are asymptotic. For the actual standard errors, multiply by 1 / n.
- `asySE` The asymptotic standard error for elements in beta under the envelope model. The standard errors returned are asymptotic, for actual standard errors, multiply by 1 / sqrt(n).
- `ratio` The asymptotic standard error ratio of the standard multivariate linear regression estimator over the envelope estimator, for each element in beta.
- `n` The number of observations in the data.

References

Cook, R. D., Zhang, X. (2015). Simultaneous Envelopes for Multivariate Linear Regression. Technometrics 57, 11 - 25.

Cook, R. D., Forzani, L. and Su, Z. (2016) A Note on Fast Envelope Estimation. Journal of Multivariate Analysis. 150, 42-54.

Examples

```
data(fiberpaper)
X <- fiberpaper[, 5:7]
Y <- fiberpaper[, 1:4]
u <- u.stenv(X, Y)
u

m <- stenv(X, Y, 2, 3)
m
m$beta
```

stenvMU	*Estimate the simultaneous envelope subspace*

Description

Estimate the simulatneous envelope subspace with specified dimension.

Usage

```
stenvMU(X, Y, q, u, initial1 = NULL, initial2 = NULL)
```

Arguments

X	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
Y	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
q	Dimension of the X-envelope. An integer between 0 and p.
u	Dimension of the Y-envelope. An integer between 0 and r.
initial1	The user-specified value of Phi for the X-envelope subspace. An p by q matrix.
initial2	The user-specified value of Gamma for the Y-envelope subspace. An r by u matrix.

Details

This function estimate the simultaneous envelope subspace using an non-Grassmann optimization algorithm.

Value

Gammahat	The orthonormal basis of the Y-envelope subspace.
Gamma0hat	The orthonormal basis of the complement of the Y-envelope subspace.
Phihat	The orthonormal basis of the X-envelope subspace.
Phi0hat	The orthonormal basis of the complement of the X-envelope subspace.
objfun	The minimized objective function.

References

Cook, R. D., Forzani, L. and Su, Z. (2016) A Note on Fast Envelope Estimation. Journal of Multivariate Analysis. 150, 42-54.

sxenv *Fit the scaled envelope model in the predictor space*

Description

Fit the scaled envelope model in the predictor space in multivariate linear regression with dimension u.

Usage

```
sxenv(X, Y, u, R, asy = TRUE, init = NULL)
```

Arguments

X	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
Y	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
u	Dimension of the scaled envelope in the predictor space. An integer between 0 and p.
R	The number of replications of the scales. A vector, the sum of all elements of R must be p.
asy	Flag for computing the asymptotic variance of the envelope estimator. The default is `TRUE`. When p and r are large, computing the asymptotic variance can take much time and memory. If only the envelope estimators are needed, the flag can be set to `asy = FALSE`.
init	The user-specified value of Gamma for the scaled envelope subspace in the predictor space. An p by u matrix. The default is the one generated by function sxenvMU.

Details

This function fits the scaled envelope model in the predictor space to the responses and predictors,

$$Y = \mu_Y + \eta'\Gamma'\Lambda^{-1}(X - \mu_X) + \varepsilon, \Sigma_X = \Lambda\Gamma\Omega\Gamma'\Lambda + \Lambda\Gamma_0\Omega_0\Gamma_0'\Lambda$$

using the maximum likelihood estimation. When the dimension of the scaled envelope in the predictor space is between 1 and p-1, the starting value and blockwise coordinate descent algorithm in Cook et al. (2016) is implemented. When the dimension is p, then the scaled envelope model in the predictor space degenerates to the standard multivariate linear regression. When the dimension is 0, it means that X and Y are uncorrelated, and the fitting is different.

Value

The output is a list that contains the following components:

`beta`	The scaled envelope estimator of the regression coefficients.
`Sigma`	The scaled envelope estimator of the error covariance matrix.
`Lambda`	The matrix of estimated scale.
`Gamma`	An orthonormal basis of the scaled envelope subspace.
`Gamma0`	An orthonormal basis of the complement of the scaled envelope subspace.
`eta`	The coordinates of beta with respect to Gamma.
`Omega`	The coordinates of Sigma with respect to Gamma.
`Omega0`	The coordinates of Sigma with respect to Gamma0.
`muY`	The mean of Y.
`muX`	The mean of X.
`loglik`	The maximized log likelihood function.
`covMatrix`	The asymptotic covariance of vec(beta). The covariance matrix returned are asymptotic. For the actual standard errors, multiply by 1 / n.
`asySE`	The asymptotic standard error for elements in beta under the envelope model. The standard errors returned are asymptotic, for actual standard errors, multiply by 1 / sqrt(n).
`ratio`	The asymptotic standard error ratio of the standard multivariate linear regression estimator over the envelope estimator, for each element in beta.
`n`	The number of observations in the data.

References

Cook, R. D., Su, Z. (2016). Scaled Predictor Envelopes and Partial Least Squares Regression. Technometrics 58, 155 - 165.

Cook, R. D., Forzani, L. and Su, Z. (2016) A Note on Fast Envelope Estimation. Journal of Multivariate Analysis. 150, 42-54.

Examples

```
data(sales)
Y <- sales[, 1:3]
X <- sales[, 4:7]
R <- rep(1, 4)

u <- u.sxenv(X, Y, R)
u

m <- sxenv(X, Y, 2, R)
m$beta
```

sxenvMU *Estimate the scaled envelope subspace in the predictor space*

Description

Estimate the scaled envelope subspace in the predictor space with specified dimension.

Usage

```
sxenvMU(X, Y, u, R, initial = NULL)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`u`	Dimension of the scaled envelope in the predictor space. An integer between 0 and p.
`R`	The number of replications of the scales. A vector, the sum of all elements of R must be p.
`initial`	The user-specified value of Gamma for the envelope subspace.

Details

This function estimate the scaled envelope subspace in the predictor space using an non-Grassmann optimization algorithm and nonlinear optimization using augmented Lagrange method.

Value

`Gammahat`	The orthonormal basis of the scaled envelope subspace in the predictor space.
`Gamma0hat`	The orthonormal basis of the complement of the scaled envelope subspace in the predictor space.
`Lambdahat`	The matrix of estimated scales.
`objfun`	The minimized objective function.

References

Cook, R. D., Forzani, L. and Su, Z. (2016) A Note on Fast Envelope Estimation. Journal of Multivariate Analysis. 150, 42-54.

Ye, Y., Interior algorithms for linear, quadratic, and linearly constrained non linear programming, PhD Thesis, Departments of EES stanford University, Stanford CA.

`testcoef.env` *Hypothesis test of the coefficients of the envelope model*

Description

This function tests the null hypothesis L * beta * R = A versus the alternative hypothesis L * beta * R ~= A, where beta is estimated under the envelope model.

Usage

```
testcoef.env(m, L, R, A)
```

Arguments

`m`	A list containing estimators and other statistics inherited from env.
`L`	The matrix multiplied to beta on the left. It is a d1 by r matrix, while d1 is less than or equal to r.
`R`	The matrix multiplied to beta on the right. It is a p by d2 matrix, while d2 is less than or equal to p.
`A`	The matrix on the right hand side of the equation. It is a d1 by d2 matrix. Note that inputs `L`, `R` and `A` must be matrices, if not, use `as.matrix` to convert them.

Details

This function tests for hypothesis H0: L beta R = A, versus Ha: L beta R != A. The beta is estimated by the envelope model. If L = Ir, R = Ip and A = 0, then the test is equivalent to the standard F test on if beta = 0. The test statistic used is vec(L beta R - A) hatSigma^-1 vec(L beta R - A)^T, where beta is the envelope estimator and hatSigma is the estimated asymptotic covariance of vec(L beta R - A). The reference distribution is chi-squared distribution with degrees of freedom d1 * d2.

Value

The output is a list that contains following components.

`chisqStatistic`	The test statistic.
`dof`	The degrees of freedom of the reference chi-squared distribution.
`pValue`	p-value of the test.
`covMatrix`	The covariance matrix of vec(L beta R).

Examples

```
data(wheatprotein)
X <- wheatprotein[, 8]
Y <- wheatprotein[, 1:6]
m <- env(X, Y, 1)
m

L <- diag(6)
R <- as.matrix(1)
A <- matrix(0, 6, 1)

test.res <- testcoef.env(m, L, R, A)
test.res
```

`testcoef.genv` *Hypothesis test of the coefficients of the groupwise envelope model*

Description

This function tests the null hypothesis L * beta * R = A versus the alternative hypothesis L * beta * R ~= A, where beta is estimated under the groupwise envelope model.

Usage

```
testcoef.genv(m, L, R, A)
```

Arguments

`m`	A list containing estimators and other statistics inherited from genv.
`L`	The matrix multiplied to beta on the left. It is a d1 by r matrix, while d1 is less than or equal to r.
`R`	The matrix multiplied to beta on the right. It is a p by d2 matrix, while d2 is less than or equal to p.
`A`	The matrix on the right hand side of the equation. It is a d1 by d2 matrix. Note that inputs `L`, `R` and `A` must be matrices, if not, use `as.matrix` to convert them.

Details

This function tests for hypothesis H0: L beta[[i]] R = A, versus Ha: L beta[[i]] R != A. The beta is estimated by the groupwise envelope model. If L = Ir, R = Ip and A = 0, then the test is equivalent to the standard F test on if beta[[i]] = 0. The test statistic used is vec(L beta R - A) hatSigma^-1 vec(L beta R - A)^T, where beta is the envelope estimator and hatSigma is the estimated asymptotic covariance of vec(L beta R - A). The reference distribution is chi-squared distribution with degrees of freedom d1 * d2.

Value

The output is a list that contains following components.

chisqStatistic	The test statistic.
dof	The degrees of freedom of the reference chi-squared distribution.
pValue	p-value of the test.
covMatrix	The covariance matrix of vec(L beta R).

Examples

```
data(fiberpaper)
X <- fiberpaper[ , c(5, 7)]
Y <- fiberpaper[ , 1:3]
Z <- as.numeric(fiberpaper[ , 6] > mean(fiberpaper[ , 6]))
u <- u.genv(X, Y, Z)
u

m <- genv(X, Y, Z, 2)
m

L <- diag(3)
R <- diag(2)
A <- matrix(0, 3, 2)

test.res <- testcoef.genv(m, L, R, A)
test.res
```

---

testcoef.henv — *Hypothesis test of the coefficients of the heteroscedastic envelope model*

Description

This function tests the null hypothesis L * beta * R = A versus the alternative hypothesis L * beta * R ~= A, where beta is estimated under the heteroscedastic envelope model.

Usage

```
testcoef.henv(m, L, R, A)
```

Arguments

m	A list containing estimators and other statistics inherited from genv.
L	The matrix multiplied to beta on the left. It is a d1 by r matrix, while d1 is less than or equal to r.
R	The matrix multiplied to beta on the right. It is a p by d2 matrix, while d2 is less than or equal to p.

A — The matrix on the right hand side of the equation. It is a d1 by d2 matrix.

Note that inputs L, R and A must be matrices, if not, use `as.matrix` to convert them.

Details

This function tests for hypothesis H0: L beta R = A, versus Ha: L beta R != A. The beta is estimated by the heteroscedastic envelope model. If L = Ir, R = Ip and A = 0, then the test is equivalent to the standard F test on if beta = 0. The test statistic used is vec(L beta R - A) hatSigma^-1 vec(L beta R - A)^T, where beta is the envelope estimator and hatSigma is the estimated asymptotic covariance of vec(L beta R - A). The reference distribution is chi-squared distribution with degrees of freedom d1 * d2.

Value

The output is a list that contains following components.

`chisqStatistic`	The test statistic.
`dof`	The degrees of freedom of the reference chi-squared distribution.
`pValue`	p-value of the test.
`covMatrix`	The covariance matrix of vec(L beta R).

Examples

```
data(waterstrider)
X <- waterstrider[ , 1]
Y <- waterstrider[ , 2:5]

## Not run: m <- henv(X, Y, 2)
## Not run: m

L <- diag(4)
R <- matrix(c(1, -1, 0), 3, 1)
A <- matrix(0, 4, 1)

## Not run: test.res <- testcoef.henv(m, L, R, A)
## Not run: test.res
```

`testcoef.logit.env` *Hypothesis test of the coefficients of the envelope model*

Description

This function tests the null hypothesis L * beta = A versus the alternative hypothesis L * beta ~= A, where beta is estimated under the envelope model in logistic regression.

Usage

```
testcoef.logit.env(m, L, A)
```

Arguments

m	A list containing estimators and other statistics inherited from logit.env.
L	The matrix multiplied to beta on the left. It is a d1 by p matrix, while d1 is less than or equal to p.
A	The matrix on the right hand side of the equation. It is a d1 by 1 matrix. Note that inputs L and A must be matrices, if not, use as.matrix to convert them.

Details

This function tests for hypothesis H0: L beta = A, versus Ha: L beta != A. The beta is estimated by the envelope model in predictor space. If L = Ip and A = 0, then the test is equivalent to the standard F test on if beta = 0. The test statistic used is vec(L beta - A) hatSigma^-1 vec(L beta - A)^T, where beta is the envelope estimator and hatSigma is the estimated asymptotic covariance of vec(L beta - A). The reference distribution is chi-squared distribution with degrees of freedom d1.

Value

The output is a list that contains following components.

chisqStatistic	The test statistic.
dof	The degrees of freedom of the reference chi-squared distribution.
pValue	p-value of the test.
covMatrix	The covariance matrix of vec(L beta).

Examples

```
data(horseshoecrab)
X1 <- as.numeric(horseshoecrab[ , 1] == 2)
X2 <- as.numeric(horseshoecrab[ , 1] == 3)
X3 <- as.numeric(horseshoecrab[ , 1] == 4)
X4 <- as.numeric(horseshoecrab[ , 2] == 2)
X5 <- as.numeric(horseshoecrab[ , 2] == 3)
X6 <- horseshoecrab[ , 3]
X7 <- horseshoecrab[ , 5]
X <- cbind(X1, X2, X3, X4, X5, X6, X7)
Y <- as.numeric(ifelse(horseshoecrab[ , 4] > 0, 1, 0))

m <- logit.env(X, Y, 1)

L <- diag(7)
A <- matrix(0, 7, 1)

test.res <- testcoef.logit.env(m, L, A)
test.res
```

`testcoef.penv` *Hypothesis test of the coefficients of the partial envelope model*

Description

This function tests the null hypothesis L * beta1 * R = A versus the alternative hypothesis L * beta1 * R ~= A, where beta is estimated under the partial envelope model.

Usage

```
testcoef.penv(m, L, R, A)
```

Arguments

`m`	A list containing estimators and other statistics inherited from penv.
`L`	The matrix multiplied to beta on the left. It is a d1 by r matrix, while d1 is less than or equal to r.
`R`	The matrix multiplied to beta on the right. It is a p1 by d2 matrix, while d2 is less than or equal to p1.
`A`	The matrix on the right hand side of the equation. It is a d1 by d2 matrix. Note that inputs `L`, `R` and `A` must be matrices, if not, use `as.matrix` to convert them.

Details

This function tests for hypothesis H0: L beta1 R = A, versus Ha: L beta1 R != A. The beta is estimated by the partial envelope model. If L = Ir, R = Ip1 and A = 0, then the test is equivalent to the standard F test on if beta1 = 0. The test statistics used is vec(L beta1 R - A) hatSigma^-1 vec(L beta1 R - A)^T, where beta is the envelope estimator and hatSigma is the estimated asymptotic covariance of vec(L beta1 R - A). The reference distribution is chi-squared distribution with degrees of freedom d1 * d2.

Value

The output is a list that contains following components.

`chisqStatistic`	The test statistic.
`dof`	The degrees of freedom of the reference chi-squared distribution.
`pValue`	p-value of the test.
`covMatrix`	The covariance matrix of vec(L beta1 R).

Examples

```
data(fiberpaper)
X1 <- fiberpaper[, 7]
X2 <- fiberpaper[, 5:6]
Y <- fiberpaper[, 1:4]
m <- penv(X1, X2, Y, 1)
m

L <- diag(4)
R <- as.matrix(1)
A <- matrix(0, 4, 1)

test.res <- testcoef.penv(m, L, R, A)
test.res
```

`testcoef.pois.env` *Hypothesis test of the coefficients of the envelope model*

Description

This function tests the null hypothesis L * beta = A versus the alternative hypothesis L * beta ~= A, where beta is estimated under the envelope model in poisson regression.

Usage

```
testcoef.pois.env(m, L, A)
```

Arguments

`m`	A list containing estimators and other statistics inherited from pois.env.
`L`	The matrix multiplied to beta on the left. It is a d1 by p matrix, while d1 is less than or equal to p.
`A`	The matrix on the right hand side of the equation. It is a d1 by 1 matrix. Note that inputs `L` and `A` must be matrices, if not, use `as.matrix` to convert them.

Details

This function tests for hypothesis H0: L beta = A, versus Ha: L beta != A. The beta is estimated by the envelope model in predictor space. If L = Ip and A = 0, then the test is equivalent to the standard F test on if beta = 0. The test statistic used is vec(L beta - A) hatSigma^-1 vec(L beta - A)^T, where beta is the envelope estimator and hatSigma is the estimated asymptotic covariance of vec(L beta - A). The reference distribution is chi-squared distribution with degrees of freedom d1.

Value

The output is a list that contains following components.

chisqStatistic	The test statistic.
dof	The degrees of freedom of the reference chi-squared distribution.
pValue	p-value of the test.
covMatrix	The covariance matrix of vec(L beta).

Examples

```
data(horseshoecrab)
X1 <- as.numeric(horseshoecrab[ , 1] == 2)
X2 <- as.numeric(horseshoecrab[ , 1] == 3)
X3 <- as.numeric(horseshoecrab[ , 1] == 4)
X4 <- as.numeric(horseshoecrab[ , 2] == 2)
X5 <- as.numeric(horseshoecrab[ , 2] == 3)
X6 <- horseshoecrab[ , 3]
X7 <- horseshoecrab[ , 5]
X <- cbind(X1, X2, X3, X4, X5, X6, X7)
Y <- horseshoecrab[ , 4]

m <- pois.env(X, Y, 1)

L <- diag(7)
A <- matrix(0, 7, 1)

test.res <- testcoef.pois.env(m, L, A)
test.res
```

---

testcoef.senv *Hypothesis test of the coefficients of the scaled envelope model*

Description

This function tests the null hypothesis L * beta * R = A versus the alternative hypothesis L * beta * R ~= A, where beta is estimated under the scaled envelope model.

Usage

```
testcoef.senv(m, L, R, A)
```

Arguments

m	A list containing estimators and other statistics inherited from scale.env.
L	The matrix multiplied to beta on the left. It is a d1 by r matrix, while d1 is less than or equal to r.

R	The matrix multiplied to beta on the right. It is a p by d2 matrix, while d2 is less than or equal to p.
A	The matrix on the right hand side of the equation. It is a d1 by d2 matrix. Note that inputs `L`, `R` and `A` must be matrices, if not, use `as.matrix` to convert them.

Details

This function tests for hypothesis H0: L beta R = A, versus Ha: L beta R != A. The beta is estimated by the scaled envelope model. If L = Ir, R = Ip and A = 0, then the test is equivalent to the standard F test on if beta = 0. The test statistic used is vec(L beta R - A) hatSigma^-1 vec(L beta R - A)^T, where beta is the envelope estimator and hatSigma is the estimated asymptotic covariance of vec(L beta R - A). The reference distribution is chi-squared distribution with degrees of freedom d1 * d2.

Value

The output is a list that contains following components.

`chisqStatistic`	The test statistic.
`dof`	The degrees of freedom of the reference chi-squared distribution.
`pValue`	p-value of the test.
`covMatrix`	The covariance matrix of vec(L beta R).

Examples

```
data(sales)
X <- sales[, 1:3]
Y <- sales[, 4:7]

m <- senv(X, Y, 2)

L <- diag(4)
R <- as.matrix(c(1, 0, 0))
A <- matrix(0, 4, 1)

test.res <- testcoef.senv(m, L, R, A)
test.res
```

`testcoef.stenv` *Hypothesis test of the coefficients of the simultaneous envelope model*

Description

This function tests the null hypothesis L * beta * R = A versus the alternative hypothesis L * beta * R ~= A, where beta is estimated under the simultaneous envelope model.

Usage

```
testcoef.stenv(m, L, R, A)
```

Arguments

m	A list containing estimators and other statistics inherited from stenv.
L	The matrix multiplied to beta on the left. It is a d1 by p matrix, while d1 is less than or equal to p.
R	The matrix multiplied to beta on the right. It is an r by d2 matrix, while d2 is less than or equal to r.
A	The matrix on the right hand side of the equation. It is a d1 by d2 matrix. Note that inputs `L`, `R` and `A` must be matrices, if not, use `as.matrix` to convert them.

Details

This function tests for hypothesis H0: L beta R = A, versus Ha: L beta R != A. The beta is estimated by the simultaneous envelope model. If L = Ip, R = Ir and A = 0, then the test is equivalent to the standard F test on if beta = 0. The test statistic used is vec(L beta R - A) hatSigma^-1 vec(L beta R - A)^T, where beta is the envelope estimator and hatSigma is the estimated asymptotic covariance of vec(L beta R - A). The reference distribution is chi-squared distribution with degrees of freedom d1 * d2.

Value

The output is a list that contains following components.

chisqStatistic	The test statistic.
dof	The degrees of freedom of the reference chi-squared distribution.
pValue	p-value of the test.
covMatrix	The covariance matrix of vec(L beta R).

Examples

```
data(fiberpaper)
X <- fiberpaper[ , 5:7]
Y <- fiberpaper[ , 1:4]
m <- stenv(X, Y, 2, 3)

L <- diag(3)
R <- as.matrix(c(1, 0, 0, 0), nrow = 4)
A <- matrix(0, 3, 1)

test.res <- testcoef.stenv(m, L, R, A)
test.res
```

`testcoef.sxenv` *Hypothesis test of the coefficients of the scaled envelope model in the predictor space*

Description

This function tests the null hypothesis L * beta * R = A versus the alternative hypothesis L * beta * R ~= A, where beta is estimated under the scaled envelope model in the predictor space.

Usage

```
testcoef.sxenv(m, L, R, A)
```

Arguments

`m`	A list containing estimators and other statistics inherited from scale.xenv.
`L`	The matrix multiplied to beta on the left. It is a d1 by p matrix, while d1 is less than or equal to p.
`R`	The matrix multiplied to beta on the right. It is an r by d2 matrix, while d2 is less than or equal to r.
`A`	The matrix on the right hand side of the equation. It is a d1 by d2 matrix. Note that inputs `L`, `R` and `A` must be matrices, if not, use `as.matrix` to convert them.

Details

This function tests for hypothesis H0: L beta R = A, versus Ha: L beta R != A. The beta is estimated by the scaled envelope model in the predictor space. If L = Ip, R = Ir and A = 0, then the test is equivalent to the standard F test on if beta = 0. The test statistic used is vec(L beta R - A) hatSigma^-1 vec(L beta R - A)^T, where beta is the envelope estimator and hatSigma is the estimated asymptotic covariance of vec(L beta R - A). The reference distribution is chi-squared distribution with degrees of freedom d1 * d2.

Value

The output is a list that contains following components.

`chisqStatistic`	The test statistic.
`dof`	The degrees of freedom of the reference chi-squared distribution.
`pValue`	p-value of the test.
`covMatrix`	The covariance matrix of vec(L beta R).

Examples

```
data(sales)
Y <- sales[, 1:3]
X <- sales[, 4:7]
R <- rep(1, 4)

u <- u.sxenv(X, Y, R)
u

m <- sxenv(X, Y, 2, R)

L <- diag(4)
R <- as.matrix(c(1, 0, 0))
A <- matrix(0, 4, 1)

test.res <- testcoef.sxenv(m, L, R, A)
test.res
```

`testcoef.xenv` *Hypothesis test of the coefficients of the envelope model*

Description

This function tests the null hypothesis L * beta * R = A versus the alternative hypothesis L * beta * R ~= A, where beta is estimated under the envelope model in predictor space.

Usage

```
testcoef.xenv(m, L, R, A)
```

Arguments

`m`	A list containing estimators and other statistics inherited from xenv.
`L`	The matrix multiplied to beta on the left. It is a d1 by p matrix, while d1 is less than or equal to p.
`R`	The matrix multiplied to beta on the right. It is an r by d2 matrix, while d2 is less than or equal to r.
`A`	The matrix on the right hand side of the equation. It is a d1 by d2 matrix. Note that inputs `L`, `R` and `A` must be matrices, if not, use `as.matrix` to convert them.

Details

This function tests for hypothesis H0: L beta R = A, versus Ha: L beta R != A. The beta is estimated by the envelope model in predictor space. If L = Ip, R = Ir and A = 0, then the test is equivalent to the standard F test on if beta = 0. The test statistic used is vec(L beta R - A) hatSigma^-1 vec(L beta R - A)^T, where beta is the envelope estimator and hatSigma is the estimated asymptotic covariance of vec(L beta R - A). The reference distribution is chi-squared distribution with degrees of freedom d1 * d2.

Value

The output is a list that contains following components.

chisqStatistic	The test statistic.
dof	The degrees of freedom of the reference chi-squared distribution.
pValue	p-value of the test.
covMatrix	The covariance matrix of vec(L beta R).

Examples

```
data(wheatprotein)
X <- wheatprotein[, 1:6]
Y <- wheatprotein[, 7]
m <- xenv(X, Y, 2)
m

L <- diag(6)
R <- as.matrix(1)
A <- matrix(0, 6, 1)

test.res <- testcoef.xenv(m, L, R, A)
test.res
```

---

u.env *Select the dimension of env*

Description

This function outputs dimensions selected by Akaike information criterion (AIC), Bayesian information criterion (BIC) and likelihood ratio testing with specified significance level for the envelope model.

Usage

```
u.env(X, Y, alpha = 0.01)
```

Arguments

X	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
Y	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
alpha	Significance level for testing. The default is 0.01.

Value

u.aic	Dimension of the envelope subspace selected by AIC.
u.bic	Dimension of the envelope subspace selected by BIC.
u.lrt	Dimension of the envelope subspace selected by the likelihood ratio testing procedure.
loglik.seq	Log likelihood for dimension from 0 to r.
aic.seq	AIC value for dimension from 0 to r.
bic.seq	BIC value for dimension from 0 to r.

Examples

```
data(wheatprotein)
X <- wheatprotein[, 8]
Y <- wheatprotein[, 1:6]
u <- u.env(X, Y)
u
```

---

u.genv *Select the dimension of genv*

Description

This function outputs dimensions selected by Akaike information criterion (AIC), Bayesian information criterion (BIC) and likelihood ratio testing with specified significance level for the groupwise envelope model.

Usage

```
u.genv(X, Y, Z, alpha = 0.01)
```

Arguments

X	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
Y	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
Z	A group indicator vector of length n, where n denotes the number of observations.
alpha	Significance level for testing. The default is 0.01.

Value

u.aic	Dimension of the groupwise envelope subspace selected by AIC.
u.bic	Dimension of the groupwise envelope subspace selected by BIC.
u.lrt	Dimension of the groupwise envelope subspace selected by the likelihood ratio testing procedure.
loglik.seq	Log likelihood for dimension from 0 to r.
aic.seq	AIC value for dimension from 0 to r.
bic.seq	BIC value for dimension from 0 to r.

Examples

```
data(fiberpaper)
X <- fiberpaper[ , c(5, 7)]
Y <- fiberpaper[ , 1:3]
Z <- as.numeric(fiberpaper[ , 6] > mean(fiberpaper[ , 6]))

u <- u.genv(X, Y, Z)
u
```

---

u.henv *Select the dimension of henv*

Description

This function outputs dimensions selected by Akaike information criterion (AIC), Bayesian information criterion (BIC) and likelihood ratio testing with specified significance level for the heteroscedastic envelope model.

Usage

```
u.henv(X, Y, alpha = 0.01)
```

Arguments

X	A group indicator vector of length n, where n denotes the number of observations.
Y	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
alpha	Significance level for testing. The default is 0.01.

Value

`u.aic`	Dimension of the heteroscedastic envelope subspace selected by AIC.
`u.bic`	Dimension of the heteroscedastic envelope subspace selected by BIC.
`u.lrt`	Dimension of the heteroscedastic envelope subspace selected by the likelihood ratio testing procedure.
`loglik.seq`	Log likelihood for dimension from 0 to r.
`aic.seq`	AIC value for dimension from 0 to r.
`bic.seq`	BIC value for dimension from 0 to r.

Examples

```
data(waterstrider)
X <- waterstrider[ , 1]
Y <- waterstrider[ , 2:5]

## Not run: u <- u.henv(X, Y)
## Not run: u
```

`u.logit.env` *Select the dimension of logit.env*

Description

This function outputs dimensions selected by Akaike information criterion (AIC), Bayesian information criterion (BIC) and likelihood ratio testing with specified significance level for the envelope model in logistic regression.

Usage

```
u.logit.env(X, Y, alpha = 0.01)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.
`Y`	Response. An n by 1 matrix. The univariate response must be binary.
`alpha`	Significance level for testing. The default is 0.01.

Value

`u.aic`	Dimension of the envelope subspace selected by AIC.
`u.bic`	Dimension of the envelope subspace selected by BIC.
`u.lrt`	Dimension of the envelope subspace selected by the likelihood ratio testing procedure.
`loglik.seq`	Log likelihood for dimension from 0 to p.
`aic.seq`	AIC value for dimension from 0 to p.
`bic.seq`	BIC value for dimension from 0 to p.

Examples

```
data(horseshoecrab)
X1 <- as.numeric(horseshoecrab[ , 1] == 2)
X2 <- as.numeric(horseshoecrab[ , 1] == 3)
X3 <- as.numeric(horseshoecrab[ , 1] == 4)
X4 <- as.numeric(horseshoecrab[ , 2] == 2)
X5 <- as.numeric(horseshoecrab[ , 2] == 3)
X6 <- horseshoecrab[ , 3]
X7 <- horseshoecrab[ , 5]
X <- cbind(X1, X2, X3, X4, X5, X6, X7)
Y <- as.numeric(ifelse(horseshoecrab[ , 4] > 0, 1, 0))

## Not run: u <- u.logit.env(X, Y)
## Not run: u
```

---

u.penv — *Select the dimension of penv*

Description

This function outputs dimensions selected by Akaike information criterion (AIC), Bayesian information criterion (BIC) and likelihood ratio testing with specified significance level for the partial envelope model.

Usage

```
u.penv(X1, X2, Y, alpha = 0.01)
```

Arguments

`X1`	Predictors of main interest. An n by p1 matrix, n is the number of observations, and p1 is the number of main predictors. The predictors can be univariate or multivariate, discrete or continuous.
`X2`	Covariates, or predictors not of main interest. An n by p2 matrix, p2 is the number of covariates.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`alpha`	Significance level for testing. The default is 0.01.

Value

`u.aic`	Dimension of the partial envelope subspace selected by AIC.
`u.bic`	Dimension of the partial envelope subspace selected by BIC.
`u.lrt`	Dimension of the partial envelope subspace selected by the likelihood ratio testing procedure.
`loglik.seq`	Log likelihood for dimension from 0 to r.
`aic.seq`	AIC value for dimension from 0 to r.
`bic.seq`	BIC value for dimension from 0 to r.

Examples

```
data(fiberpaper)
X1 <- fiberpaper[, 7]
X2 <- fiberpaper[, 5:6]
Y <- fiberpaper[, 1:4]

u <- u.penv(X1, X2, Y)
u
```

---

u.pois.env *Select the dimension of pois.env*

Description

This function outputs dimensions selected by Akaike information criterion (AIC), Bayesian information criterion (BIC) and likelihood ratio testing with specified significance level for the envelope model in poisson regression.

Usage

```
u.pois.env(X, Y, alpha = 0.01)
```

Arguments

X	Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.
Y	Response. An n by 1 matrix. The univariate response must be counts.
alpha	Significance level for testing. The default is 0.01.

Value

u.aic	Dimension of the envelope subspace selected by AIC.
u.bic	Dimension of the envelope subspace selected by BIC.
u.lrt	Dimension of the envelope subspace selected by the likelihood ratio testing procedure.
loglik.seq	Log likelihood for dimension from 0 to p.
aic.seq	AIC value for dimension from 0 to p.
bic.seq	BIC value for dimension from 0 to p.

Examples

```
data(horseshoecrab)
X1 <- as.numeric(horseshoecrab[ , 1] == 2)
X2 <- as.numeric(horseshoecrab[ , 1] == 3)
X3 <- as.numeric(horseshoecrab[ , 1] == 4)
X4 <- as.numeric(horseshoecrab[ , 2] == 2)
X5 <- as.numeric(horseshoecrab[ , 2] == 3)
X6 <- horseshoecrab[ , 3]
X7 <- horseshoecrab[ , 5]
X <- cbind(X1, X2, X3, X4, X5, X6, X7)
Y <- horseshoecrab[ , 4]

## Not run: u <- u.pois.env(X, Y)
## Not run: u
```

---

u.pred2.env — *Select the dimension of the constructed partial envelope for prediction based on envelope model*

Description

This function outputs dimensions selected by Akaike information criterion (AIC), Bayesian information criterion (BIC) and likelihood ratio testing with specified significance level for the constructed partial envelope model.

Usage

```
u.pred2.env(X, Y, Xnew, alpha = 0.01)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`Xnew`	The value of X with which to estimate or predict Y. A p dimensional vector.
`alpha`	Significance level for testing. The default is 0.01.

Value

`u.aic`	Dimension of the constructed partial envelope subspace selected by AIC.
`u.bic`	Dimension of the constructed partial envelope subspace selected by BIC.
`u.lrt`	Dimension of the constructed partial envelope subspace selected by the likelihood ratio testing procedure.
`loglik.seq`	Log likelihood for dimension from 0 to r.
`aic.seq`	AIC value for dimension from 0 to r.
`bic.seq`	BIC value for dimension from 0 to r.

Examples

```
data(fiberpaper)
X <- fiberpaper[, 5:7]
Y <- fiberpaper[, 1:4]

u <- u.pred2.env(X, Y, X[10, ])
u
```

`u.senv` *Select the dimension of senv*

Description

This function outputs dimensions selected by Akaike information criterion (AIC), Bayesian information criterion (BIC).

Usage

```
u.senv(X, Y)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.

Value

`u.aic`	Dimension of the scaled envelope subspace selected by AIC.
`u.bic`	Dimension of the scaled envelope subspace selected by BIC.
`aic.seq`	AIC value for dimension from 0 to r.
`bic.seq`	BIC value for dimension from 0 to r.

Examples

```
data(sales)
X <- sales[, 1:3]
Y <- sales[, 4:7]

u <- u.senv(X, Y)
u
```

u.stenv *Select the dimension of stenv*

Description

This function outputs dimensions selected by Akaike information criterion (AIC), Bayesian information criterion (BIC) and likelihood ratio testing with specified significance level for the simultaneous envelope model.

Usage

```
u.stenv(X, Y, alpha = 0.01)
```

Arguments

X	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
Y	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
alpha	Significance level for testing. The default is 0.01.

Value

d	Rank of beta selected by the Bura-Cook estimator.
u.aic	Dimension of the simultaneous envelope subspace selected by AIC.
u.bic	Dimension of the simultaneous envelope subspace selected by BIC.
u.lrt	Dimension of the simultaneous envelope subspace selected by the likelihood ratio testing procedure.
loglik.mat	Log likelihood for dimension from (1, 1) to (r, p).
aic.mat	AIC value for dimension from (1, 1) to (r, p).
bic.mat	BIC value for dimension from (1, 1) to (r, p).

Examples

```
data(fiberpaper)
X <- fiberpaper[, 5:7]
Y <- fiberpaper[, 1:4]

u <- u.stenv(X, Y)
u
```

u.sxenv — *Select the dimension of sxenv*

Description

This function outputs dimensions selected by Akaike information criterion (AIC), Bayesian information criterion (BIC) for the scaled envelope model in the predictor space.

Usage

```
u.sxenv(X, Y, R)
```

Arguments

X	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
Y	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
R	The number of replications of the scales. A vector, the sum of all elements of R must be p.

Value

u.aic	Dimension of the scaled envelope subspace in the predictor space selected by AIC.
u.bic	Dimension of the scaled envelope subspace in the predictor space selected by BIC.
aic.seq	AIC value for dimension from 0 to p.
bic.seq	BIC value for dimension from 0 to p.

Examples

```
data(sales)
Y <- sales[, 1:3]
X <- sales[, 4:7]
R <- rep(1, 4)

u <- u.sxenv(X, Y, R)
u
```

u.xenv *Select the dimension of xenv*

Description

This function outputs dimensions selected by Akaike information criterion (AIC), Bayesian information criterion (BIC) and likelihood ratio testing with specified significance level for the envelope model.

Usage

```
u.xenv(X, Y, alpha = 0.01)
```

Arguments

X	Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.
Y	Responses. An n by r matrix, r is the number of responses. The response can be univariate or multivariate and must be continuous variable.
alpha	Significance level for testing. The default is 0.01.

Value

u.aic	Dimension of the envelope subspace selected by AIC.
u.bic	Dimension of the envelope subspace selected by BIC.
u.lrt	Dimension of the envelope subspace selected by the likelihood ratio testing procedure.
loglik.seq	Log likelihood for dimension from 0 to p.
aic.seq	AIC value for dimension from 0 to p.
bic.seq	BIC value for dimension from 0 to p.

Examples

```
data(wheatprotein)
X <- wheatprotein[, 1:6]
Y <- wheatprotein[, 7]

u <- u.xenv(X, Y)
u
```

`waterstrider` *Water strider data*

Description

Measures of characteristics of the water striders

Usage

```
data("waterstrider")
```

Format

A data frame with 90 observations on the following 9 variables.

- `V1` Index of water strider species.
- `V2` Logarithm of length of the first antennal segment.
- `V3` Logarithm of length of the second antennal segment.
- `V4` Logarithm of length of the third antennal segment.
- `V5` Logarithm of length of the fourth antennal segment.
- `V6` Logarithm of length of fomora of middle leg.
- `V7` Logarithm of length of tibiae of middle leg.
- `V8` Logarithm of length of fomora of hind leg.
- `V9` Logarithm of length of tibiae of hind leg.

Details

This data set contains 8 measures of water striders and an indicator of the species of water striders.

References

Klingenberg, C. R. and Spence, J. R. (1993). Heterochrony and Allometry Lessons from the Water Strider Genus Limnoporus. Evolution 47, 1834-1853

`weighted.env` *Weighted envelope estimator*

Description

Compute the weighted envelope estimator with weights computed from BIC.

Usage

```
weighted.env(X, Y, bstrpNum = 0, min.u = 1,
max.u = ncol(as.matrix(Y)), boot.resi = "full")
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
`Y`	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
`bstrpNum`	Number of bootstrap samples. A positive integer.
`min.u`	Lower bound of the range of u to compute bootstrap error. A postive integer between 1 and p. This argument is relevant only when `bstrpNum`>0.
`max.u`	Upper bound of the range of u to compute bootstrap error. A postive integer between 1 and p. This argument is relevant only when `bstrpNum`>0.
`boot.resi`	A string that can be "full" or "weighted" indicating the model from which the residuals are calculated. If the input is "full", then the residuals are obtained using the standard estimators; and if the input is "weighted", then the residuals are obtained using the weighted envelope estimators. This argument is for computing residuals in residual bootstrap, and it is relevant only when `bstrpNum`>0.

Details

This function computes the weighted envelope estimator in a standard multivariate linear regression. And the weighted envelope estimator takes the form

$$\hat{\beta}_w = \sum_{j=1}^{r} w_j \hat{\beta}_j,$$

where $\hat{\beta}_j$ is the envelope estimator of β with $u = j$ and w_j's are the weights computed from BIC values

$$w_j = \frac{\exp(-b_j)}{\sum_{k=1}^{r} \exp(-b_k)},$$

where b_j is the BIC criterion evaluated at the envelope estimator $\hat{\beta}_j$. For details, see Eck and Cook (2017).

The variation of the weighted envelope estimator is estimated by residual bootstrap. The user can specify the range for bootstrap $u =$(`min.u`, `max.u`), if the weights outside of the range are small.

Value

The output is a list that contains the following components:

`beta`	The weighted envelope estimator of the regression coefficients.
`mu`	The weighted estimated intercept.
`Sigma`	The weighted envelope estimator of the error covariance matrix.
`w`	Weights computed based on BIC.
`loglik`	The log likelihood function computed with weighted envelope estimator.
`n`	The number of observations in the data.
`bootse`	The standard error for elements in beta computed by residual bootstrap. This output is available only when `bstrpNum`>0.
`ratios`	The boostrap standard error ratio of the standard multivariate linear regression estimator over the weighted envelope estimator for each element in beta. This output is available only when `bstrpNum`>0.
`bic_select`	A table that lists how many times BIC selected each candidate dimension. If BIC never selects a dimension, this dimension does not appear on the table. This output is available only when `bstrpNum`>0.

References

Eck, D. J. and Cook, R. D. (2017). Weighted Envelope Estimation to Handle Variability in Model Selection. Biometrika. To appear.

Examples

```
data(wheatprotein)
X <- wheatprotein[, 8]
Y <- wheatprotein[, 1:6]
m <- weighted.env(X, Y)
m$w
m$beta

## Not run: m2 <- weighted.env(X, Y, bstrpNum = 100, min.u = 1, max.u = 6, boot.resi = "full")
## Not run: m2$bic_select
## Not run: m2$bootse
```

`weighted.penv` *Weighted partial envelope estimator*

Description

Compute the weighted partial envelope estimator with weights computed from BIC.

Usage

```
weighted.penv(X1, X2, Y, bstrpNum = 0, min.u = 1,
max.u = ncol(as.matrix(Y)), boot.resi = "full")
```

Arguments

X1	Predictors of main interest. An n by p1 matrix, n is the number of observations, and p1 is the number of main predictors. The predictors can be univariate or multivariate, discrete or continuous.
X2	Covariates, or predictors not of main interest. An n by p2 matrix, p2 is the number of covariates.
Y	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
bstrpNum	Number of bootstrap samples. A positive integer.
min.u	Lower bound of the range of u to compute bootstrap error. A postive integer between 1 and p. This argument is relevant only when bstrpNum>0.
max.u	Upper bound of the range of u to compute bootstrap error. A postive integer between 1 and p. This argument is relevant only when bstrpNum>0.
boot.resi	A string that can be "full" or "weighted" indicating the model from which the residuals are calculated. If the input is "full", then the residuals are obtained using the standard estimators; and if the input is "weighted", then the residuals are obtained using the weighted envelope estimators. This argument is for computing residuals in residual bootstrap, and it is relevant only when bstrpNum>0.

Details

This function computes the weighted partial envelope estimator in a standard multivariate linear regression. And the weighted partial envelope estimator takes the form

$$\hat{\beta}_w = \sum_{j=1}^{r} w_j \hat{\beta}_j,$$

where $\hat{\beta}_j$ is the partial envelope estimator of β with $u = j$ and w_j's are the weights computed from BIC values

$$w_j = \frac{\exp(-b_j)}{\sum_{k=1}^{r} \exp(-b_k)},$$

where b_j is the BIC criterion evaluated at the partial envelope estimator $\hat{\beta}_j$. For details, see Eck and Cook (2017).

The variation of the weighted partial envelope estimator is estimated by residual bootstrap. The user can specify the range for bootstrap $u =$(min.u, max.u), if the weights outside of the range are small.

Value

The output is a list that contains the following components:

beta	The weighted partial envelope estimator of the regression coefficients.
mu	The weighted estimated intercept.
Sigma	The weighted partial envelope estimator of the error covariance matrix.
w	Weights computed based on BIC.

loglik	The log likelihood function computed with weighted partial envelope estimator.
n	The number of observations in the data.
bootse	The standard error for elements in beta1 computed by residual bootstrap. This output is available only when bstrpNum>0.
ratios	The boostrap standard error ratio of the standard multivariate linear regression estimator over the weighted partial envelope estimator for each element in beta1. This output is available only when bstrpNum>0.
bic_select	A table that lists how many times BIC selected each candidate dimension. If BIC never selects a dimension, this dimension does not appear on the table. This output is available only when bstrpNum>0.

References

Eck, D. J. and Cook, R. D. (2017). Weighted Envelope Estimation to Handle Variability in Model Selection. Biometrika. To appear.

Examples

```
data(fiberpaper)
X1 <- fiberpaper[, 7]
X2 <- fiberpaper[, 5:6]
Y <- fiberpaper[, 1:4]
m <- weighted.penv(X1, X2, Y)
m$w
m$beta1

m2 <- penv(X1, X2, Y, 2)
m2$beta1


## Not run: m3 <- weighted.penv(X1, X2, Y, bstrpNum = 100, boot.resi = "full")
## Not run: m3$w
## Not run: m3$bic_select
## Not run: m3$bootse

## Not run: boot.penv(X1, X2, Y, 2, 100)
```

---

weighted.pred.env *Estimation or prediction using weighted partial envelope*

Description

Perform estimation or prediction through weighted partial envelope model.

Usage

```
weighted.pred.env(X, Y, Xnew)
```

Arguments

X	Predictors. An n by p matrix, p is the number of predictors. The predictors can be univariate or multivariate, discrete or continuous.
Y	Multivariate responses. An n by r matrix, r is the number of responses and n is number of observations. The responses must be continuous variables.
Xnew	The value of X with which to estimate or predict Y. A p dimensional vector.

Details

This function evaluates the envelope model at new value Xnew. It can perform estimation: find the fitted value when X = Xnew, or prediction: predict Y when X = Xnew. But it does not provide the estimation or prediction error. This function performs prediction using the same procedure as in pred2.env, except that the partial envelope estimator with dimension u is replaced by a weighted partial envelope estimator. The weights are decided based on BIC values.

Value

value	The fitted value or the predicted value evaluated at Xnew.

Examples

```
data(fiberpaper)
X <- fiberpaper[, 5:7]
Y <- fiberpaper[, 1:4]

## Not run: pred.res <- weighted.pred.env(X, Y, X[10, ])
```

---

weighted.xenv *Weighted predictor envelope estimator*

Description

Compute the weighted predictor envelope estimator with weights computed from BIC.

Usage

```
weighted.xenv(X, Y, bstrpNum = 0, min.u = 1,
max.u = ncol(as.matrix(X)), boot.resi = "full")
```

Arguments

X	Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.
Y	Responses. An n by r matrix, r is the number of responses. The response can be univariate or multivariate and must be continuous variable.

`bstrpNum`	Number of bootstrap samples. A positive integer.
`min.u`	Lower bound of the range of u to compute bootstrap error. A postive integer between 1 and p. This argument is relevant only when `bstrpNum`>0.
`max.u`	Upper bound of the range of u to compute bootstrap error. A postive integer between 1 and p. This argument is relevant only when `bstrpNum`>0.
`boot.resi`	A string that can be "full" or "weighted" indicating the model from which the residuals are calculated. If the input is "full", then the residuals are obtained using the standard estimators; and if the input is "weighted", then the residuals are obtained using the weighted predictor envelope estimators. This argument is for computing residuals in residual bootstrap, and it is relevant only when `bstrpNum`>0.

Details

This function computes the weighted predictor envelope estimator in a standard multivariate linear regression. And the weighted predictor envelope estimator takes the form

$$\hat{\beta}_w = \sum_{j=1}^{p} w_j \hat{\beta}_j,$$

where $\hat{\beta}_j$ is the predictor envelope estimator of β with $u = j$ and w_j's are the weights computed from BIC values

$$w_j = \frac{\exp(-b_j)}{\sum_{k=1}^{p} \exp(-b_k)},$$

where b_j is the BIC criterion evaluated at the predictor envelope estimator $\hat{\beta}_j$. For details, see Eck and Cook (2017).

The variation of the weighted predictor envelope estimator is estimated by residual bootstrap. The user can specify the range for bootstrap $u =$(`min.u`, `max.u`), if the weights outside of the range are small.

Value

The output is a list that contains the following components:

`beta`	The weighted predictor envelope estimator of the regression coefficients.
`mu`	The weighted estimated intercept.
`SigmaX`	The weighted predictor envelope estimator of the covariance matrix of X.
`SigmaYcX`	The weighted predictor envelope estimator of the error covariance matrix.
`w`	Weights computed based on BIC.
`loglik`	The log likelihood function computed with weighted predictor envelope estimator.
`n`	The number of observations in the data.
`bootse`	The standard error for elements in beta computed by residual bootstrap. This output is available only when `bstrpNum`>0.

`ratios` The boostrap standard error ratio of the standard multivariate linear regression estimator over the weighted predictor envelope estimator for each element in beta. This output is available only when `bstrpNum`>0.

`bic_select` A table that lists how many times BIC selected each candidate dimension. If BIC never selects a dimension, this dimension does not appear on the table. This output is available only when `bstrpNum`>0.

References

Eck, D. J. and Cook, R. D. (2017). Weighted Envelope Estimation to Handle Variability in Model Selection. Biometrika. To appear.

Examples

```
data(wheatprotein)
X <- wheatprotein[, 1:6]
Y <- wheatprotein[, 7]
m <- weighted.xenv(X, Y)
m$w
m$beta

## Not run: m2 <- weighted.xenv(X, Y, bstrpNum = 100, min.u = 2, max.u = 4, boot.resi = "full")
## Not run: m2$w
## Not run: m2$bootse
```

`wheatprotein` *Wheat Protein Data*

Description

The protein content of ground wheat samples.

Usage

```
data(wheatprotein)
```

Format

A data frame with 50 observations on the following 8 variables.

- `V1` Measurements of the reflectance of NIR radiation by the wheat samples at 1680nm. The measurements were made on the log(1/reflectance) scale.
- `V2` Measurements of the reflectance of NIR radiation by the wheat samples at 1806nm.
- `V3` Measurements of the reflectance of NIR radiation by the wheat samples at 1932nm.
- `V4` Measurements of the reflectance of NIR radiation by the wheat samples at 2058nm.
- `V5` Measurements of the reflectance of NIR radiation by the wheat samples at 2184nm.
- `V6` Measurements of the reflectance of NIR radiation by the wheat samples at 2310nm.

`V7` The protein content of each sample (in percent).

`V8` Binary indicator, 0 for high protein content and 1 for low protein content. The cut off point is if the protein content is smaller than 9.75.

Details

The data are the result of an experiment to calibrate a near infrared reflectance (NIR) instrument for measuring the protein content of ground wheat samples. The protein content of each sample (in percent) was measured by the standard Kjeldahl method. In Fearn (1983), the problem is to find a linear combination of the measurements that predicts protein content. The estimated coefficients can then be entered into the instrument allowing the protein content of future samples to be read directly. The first 24 cases were used for calibration and the last 26 samples were used for prediction.

References

Fearn, T. (1983). A misuse of ridge regression in the calibration of a near infrared reflectance instrument.

`xenv` *Fit the envelope model in the predictor space*

Description

Fit the envelope model in the predictor space with dimension u under linear regression.

Usage

```
xenv(X, Y, u, asy = TRUE, init = NULL)
```

Arguments

`X`	Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.
`Y`	Responses. An n by r matrix, r is the number of responses. The response can be univariate or multivariate and must be continuous variable.
`u`	Dimension of the envelope. An integer between 0 and p.
`asy`	Flag for computing the asymptotic variance of the envelope estimator. The default is `TRUE`. When p and r are large, computing the asymptotic variance can take much time and memory. If only the envelope estimators are needed, the flag can be set to `asy = FALSE`.
`init`	The user-specified value of Gamma for the envelope subspace in the predictor space. An p by u matrix. The default is the one generated by function envMU.

Details

This function fits the envelope model in the predictor space,

$$Y = \mu + \eta' \Omega^{-1} \Gamma' X + \varepsilon, \Sigma_X = \Gamma \Omega \Gamma' + \Gamma_0 \Omega_0 \Gamma_0'$$

using the maximum likelihood estimation. When the dimension of the envelope is between 1 and p-1, the starting value and blockwise coordinate descent algorithm in Cook et al. (2016) is implemented. When the dimension is p, then the envelope model degenerates to the standard multivariate linear regression. When the dimension is 0, it means that X and Y are uncorrelated, and the fitting is different.

Value

The output is a list that contains the following components:

`beta`	The envelope estimator of the regression coefficients.
`SigmaX`	The envelope estimator of the covariance matrix of X.
`Gamma`	An orthonormal basis of the envelope subspace.
`Gamma0`	An orthonormal basis of the complement of the envelope subspace.
`eta`	The estimated eta. According to the envelope parameterization, beta = Gamma * Omega^-1 * eta.
`Omega`	The coordinates of SigmaX with respect to Gamma.
`Omega0`	The coordinates of SigmaX with respect to Gamma0.
`mu`	The estimated intercept.
`SigmaYcX`	The estimated conditional covariance matrix of Y given X.
`loglik`	The maximized log likelihood function.
`covMatrix`	The asymptotic covariance of vec(beta). The covariance matrix returned are asymptotic. For the actual standard errors, multiply by 1 / n.
`asySE`	The asymptotic standard error for elements in beta under the envelope model. The standard errors returned are asymptotic, for actual standard errors, multiply by 1 / sqrt(n).
`ratio`	The asymptotic standard error ratio of the standard multivariate linear regression estimator over the envelope estimator, for each element in beta.
`n`	The number of observations in the data.

References

Cook, R. D., Helland, I. S. and Su, Z. (2013). Envelopes and Partial Least Squares Re- gression. Journal of the Royal Statistical Society: Series B 75, 851 - 877.

Cook, R. D., Forzani, L. and Su, Z. (2016) A Note on Fast Envelope Estimation. Journal of Multivariate Analysis. 150, 42-54.

See Also

`simpls.fit` for partial least squares (PLS).

Examples

```
## Fit the envelope in the predictor space
data(wheatprotein)
X <- wheatprotein[, 1:6]
Y <- wheatprotein[, 7]
u <- u.xenv(X, Y)
u

m <- xenv(X, Y, 4)
m
m$beta

## Fit the partial least squares
## Not run: m1 <- pls::simpls.fit(X, Y, 4)
## Not run: m1$coefficients
```

Index