

# Matlab Package: Envelope

May 14, 2012

## Contents

<b>I</b>	<b>tools</b>	<b>4</b>
1	bootstrapse	4
2	bootstrapse_OLS	7
3	modelselectaic	9
4	modelselectbic	11
5	modelselectlrt	13
6	prediction	15
7	testcoefficient	18
<b>II</b>	<b>env</b>	<b>21</b>
8	aic_env	21
9	bic_env	23
10	bstrp_env	25
11	dF4env	27
12	env	28
13	F4env	32
14	lrt_env	33
15	predict_env	35

16 testcoefficient_env	38
<b>III henv</b>	<b>40</b>
17 aic_henv	40
18 bic_henv	42
19 bstrp_henv	44
20 dF4henv	46
21 F4henv	47
22 henv	48
23 lrt_henv	52
24 predict_henv	54
25 testcoefficient_henv	57
<b>IV ienv</b>	<b>59</b>
26 aic_ienv	59
27 bic_ienv	61
28 bstrp_ienv	63
29 dF4ienv	65
30 F4ienv	66
31 ienv	67
32 lrt_ienv	71
33 predict_ienv	73
34 testcoefficient_ienv	76
<b>V penv</b>	<b>78</b>
35 aic_penv	78

36	bic_penv	80
37	bstrp_penv	82
38	lrt_penv	84
39	penv	86
40	predict_penv	90
41	testcoefficient_penv	93
VI senv		95
42	aic_senv	95
43	bic_senv	97
44	bstrp_senv	99
45	dF4senv	101
46	F4senv	102
47	objfun	103
48	predict_senv	104
49	senv	107
VII xenv		111
50	bic_xenv	112
51	bstrp_xenv	113
52	lrt_xenv	115
53	xenv	116

# Part I

## tools

### 1 bootstrapse

Perform bootstrap to estimate actual standard errors for models in the envelope family.

#### Contents

- Syntax
- Input
- Output
- Description
- Example

#### Syntax

```
bootse = bootstrapse(X, Y, u, B, modelType)
bootse = bootstrapse(X, Y, u, B, modelType, Opts)
```

#### Input

**X**: Predictors. The predictors can be univariate or multivariate, discrete or continuous.

For model type for method 'env', 'henv', 'ienv', 'senv', and 'xenv'. X is an n by p matrix, p is the number of predictors.

For model type 'penv', X is A list containing the value of X1 and X2.

- X.X1 (only for 'penv'): 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.
- X.X2 (only for 'penv'): 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 subspace. The legitimate range of u depends on the model specified.

**B**: Number of bootstrap samples. A positive integer.

**modelType:** A string characters indicting the model, choices can be 'env', 'henv', 'ienv', 'penv', 'senv' and 'xenv'.

**Opts:** A list containing the optional input parameter, to control the iterations in sg\_min. If one or several (even all) fields are not defined, the default settings are used.

- Opts.maxIter: Maximum number of iterations. Default value: 300.
- Opts.ftol: Tolerance parameter for F. Default value: 1e-10.
- Opts.gradtol: Tolerance parameter for dF. Default value: 1e-7.
- Opts.verbose: Flag for print out the number of bootstrap samples, logical 0 or 1. Default value: 0.

## Output

bootse: For 'env', 'henv', 'ienv', 'senv' and 'xenv', an r by p matrix containing the standard errors for elements in  $\beta$  computed by bootstrap. For 'penv', an r by p1 matrix containing the standard errors for  $\beta_1$  computed by bootstrap.

## Description

This function computes the bootstrap standard errors for the regression coefficients or for partial envelope model, the main regression coefficients in the specified model by bootstrapping the residuals.

## Example

```
load wheatprotein.txt
X = wheatprotein(:, 8);
Y = wheatprotein(:, 1:6);
alpha = 0.01;
u = lrt_env(X, Y, alpha);
B = 100;
modelType = 'env';
bootse = bootstrapse(X, Y, u, B, modelType)
```

bootse =

```
0.2896
0.4352
0.3189
0.5735
0.2543
0.5840
```

```

load fiberpaper.dat
Y = fiberpaper(:, 1 : 4);
Xtemp = fiberpaper(:, 5 : 7);
X.X1 = Xtemp(:, 3);
X.X2 = Xtemp(:, 1 : 2);
alpha = 0.01;
u = lrt_penv(X, Y, alpha);
B = 100;
modelType = 'penv';
bootse = bootstrapse(X, Y, u, B, modelType)

```

```
bootse =
```

```

0.0027
0.0012
0.0020
0.0009

```

## 2 bootstrapse\_OLS

Compute bootstrap standard error for ordinary least squares.

### Contents

- Syntax
- Input
- Output
- Description
- Example

### Syntax

```
bootse = bootstrapse_OLS(X, Y, B)
```

### Input

**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.

**B**: Number of bootstrap samples. A positive integer.

**Opts**: A list containing the optional input parameter. If not defined, the default setting is used.

- Opts.verbose: Flag for print out the number of bootstrap samples, logical 0 or 1. Default value: 0.

### Output

**bootse**: The standard error for elements in  $\beta$  computed by bootstrap. An  $r$  by  $p$  matrix.

### Description

This function computes the bootstrap standard errors for the regression coefficients in ordinary least squares by bootstrapping the residuals.

## Example

```
load wheatprotein.txt
X = wheatprotein(:, 8);
Y = wheatprotein(:, 1 : 6);
bootse = bootstrapse_OLS(X, Y, 200)

bootse =

    10.2168
     8.3940
     9.0503
     9.9677
    14.5822
     5.5874
```



### 3 modelselectaic

Select the dimension for the envelope family using Akaike information criteria.

#### Contents

- Syntax
- Input
- Output
- Description
- Example

#### Syntax

```
u = modelselectaic(X, Y, modelType)
u = modelselectaic(X, Y, modelType, Opts)
```

#### Input

**X:** Predictors. The predictors can be univariate or multivariate, discrete or continuous.

For model type for method 'env', 'henv', 'ienv', 'senv', and 'xenv'. X is an n by p matrix, p is the number of predictors.

For model type 'penv', X is A list containing the value of X1 and X2.

- X.X1 (only for 'penv'): 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.
- X.X2 (only for 'penv'): 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.

**modelType:** A string characters indicting the model, choices can be 'env', 'henv', 'ienv', 'penv', 'senv' and 'xenv'.

**Opts:** A list containing the optional input parameter, to control the iterations in sg\_min. If one or several (even all) fields are not defined, the default settings are used.

- Opts.maxIter: Maximum number of iterations. Default value: 300.
- Opts.ftol: Tolerance parameter for F. Default value: 1e-10.

- Opts.gradtol: Tolerance parameter for dF. Default value: 1e-7.
- Opts.verbose: Flag for print out dimension selection process, logical 0 or 1. Default value: 0.

## Output

**u:** Dimension of the envelope. An integer between 0 and r.

## Description

This function implements the Akaike information criteria (AIC) to select the dimension of the envelope subspace for method 'env', 'henv', 'ienv', 'penv', 'senv', and 'xenv'.

## Example

```
load wheatprotein.txt
X = wheatprotein(:, 8);
Y = wheatprotein(:, 1 : 6);
modelType = 'env';
u = modelselectaic(X, Y, modelType)
```

u =

1

```
load fiberpaper.dat
Y = fiberpaper(:, 1 : 4);
Xtemp = fiberpaper(:, 5 : 7);
X.X1 = Xtemp(:, 3);
X.X2 = Xtemp(:, 1 : 2);
modelType = 'penv';
u = modelselectaic(X, Y, modelType)
```

u =

3

## 4 modelselectbic

Select the dimension for the envelope family using Bayesian information criteria.

### Contents

- Syntax
- Input
- Output
- Description
- Example

### Syntax

```
u = modelselectbic(X, Y, modelType)
u = modelselectbic(X, Y, modelType, Opts)
```

### Input

**X:** Predictors. The predictors can be univariate or multivariate, discrete or continuous.

For model type for method 'env', 'henv', 'ienv', 'senv', and 'xenv'. X is an n by p matrix, p is the number of predictors.

For model type 'penv', X is A list containing the value of X1 and X2.

- X.X1 (only for 'penv'): 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.
- X.X2 (only for 'penv'): 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.

**modelType:** A string characters indicting the model, choices can be 'env', 'henv', 'ienv', 'penv', 'senv' and 'xenv'.

**Opts:** A list containing the optional input parameter, to control the iterations in sg\_min. If one or several (even all) fields are not defined, the default settings are used.

- Opts.maxIter: Maximum number of iterations. Default value: 300.
- Opts.ftol: Tolerance parameter for F. Default value: 1e-10.

- Opts.gradtol: Tolerance parameter for dF. Default value: 1e-7.
- Opts.verbose: Flag for print out dimension selection process, logical 0 or 1. Default value: 0.

## Output

**u:** Dimension of the envelope. An integer between 0 and r.

## Description

This function implements the Bayesian information criteria (BIC) to select the dimension of the envelope subspace for method 'env', 'henv', 'ienv', 'penv', 'senv', and 'xenv'.

## Example

```
load wheatprotein.txt
X = wheatprotein(:, 8);
Y = wheatprotein(:, 1 : 6);
modelType = 'env';
u = modelselectbic(X, Y, modelType)
```

u =

1

```
load fiberpaper.dat
Y = fiberpaper(:, 1 : 4);
Xtemp = fiberpaper(:, 5 : 7);
X.X1 = Xtemp(:, 3);
X.X2 = Xtemp(:, 1 : 2);
modelType = 'penv';
u = modelselectbic(X, Y, modelType)
```

u =

1

## 5 modelselectlrt

Select the dimension for the envelope family using likelihood ratio testing procedure.

### Contents

- Syntax
- Input
- Output
- Description
- Example

### Syntax

```
u = modelselectlrt(X, Y, alpha, modelType)
u = modelselectlrt(X, Y, alpha, modelType, Opts)
```

### Input

**X:** Predictors. The predictors can be univariate or multivariate, discrete or continuous.

For model type for method 'env', 'henv', 'ienv', and 'xenv'. X is an n by p matrix, p is the number of predictors.

For model type 'penv', X is A list containing the value of X1 and X2.

- X.X1 (only for 'penv'): 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.
- X.X2 (only for 'penv'): 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. A real number between 0 and 1, often taken at 0.05 or 0.01.

**modelType:** A string characters indicting the model, choices can be 'env', 'henv', 'ienv', 'penv' and 'xenv'.

**Opts:** A list containing the optional input parameter, to control the iterations in sg.min. If one or several (even all) fields are not defined, the default settings are used.

- Opts.maxIter: Maximum number of iterations. Default value: 300.
- Opts.ftol: Tolerance parameter for F. Default value: 1e-10.
- Opts.gradtol: Tolerance parameter for dF. Default value: 1e-7.
- Opts.verbose: Flag for print out dimension selection process, logical 0 or 1. Default value: 0.

## Output

**u:** Dimension of the envelope. An integer between 0 and r.

## Description

This function implements the likelihood ratio testing procedure to select the dimension of the envelope subspace for method 'env', 'henv', 'ienv', 'penv', and 'xenv'. The likelihood ratio testing procedure does not support 'senv', because the scaled envelope models are not nested with the standard model.

## Example

```
load wheatprotein.txt
X = wheatprotein(:, 8);
Y = wheatprotein(:, 1 : 6);
alpha = 0.01;
modelType = 'env';
u = modelselectlrt(X, Y, alpha, modelType)
```

u =

1

```
load fiberpaper.dat
Y = fiberpaper(:, 1 : 4);
Xtemp = fiberpaper(:, 5 : 7);
X.X1 = Xtemp(:, 3);
X.X2 = Xtemp(:, 1 : 2);
alpha = 0.01;
modelType = 'penv';
u = modelselectlrt(X, Y, alpha, modelType)
```

u =

1

## 6 prediction

Perform estimation or prediction for models in the envelope family.

### Contents

- Syntax
- Input
- Output
- Description
- Example

### Syntax

```
PredictOutput = prediction(ModelOutput, Xnew, infType, modelType)
```

### Input

**ModelOutput:** A list containing the model outputs from fitting the models.

**Xnew:** The value of X with which to estimate or predict Y.

For 'env', 'henv', 'ienv', 'senv' and 'xenv', it is a p by 1 vector.

For 'penv', it is a list containing the value of X1 and X2.

- \* Xnew.X1 (only for 'penv'): A p1 by 1 vector containing the value of X1.
- \* Xnew.X2 (only for 'penv'): A p2 by 1 vector containing the value of X2.

**infType:** A string of characters indicting the inference type, the choices can be 'estimation' or 'prediction'.

**modelType:** A string characters indicting the model, choices can be 'env', 'henv', 'ienv', 'penv', 'senv' and 'xenv'.

### Output

**PredictOutput:** A list containing the results of the inference.

- PredictOutput.value: The fitted value or the prediction value evaluated at Xnew. An r by 1 vector.
- PredictOutput.covMatrix: The covariance matrix of PredictOutput.value. An r by r matrix.
- PredictOutput.SE: The standard error of elements in PredictOutput.value. An r by 1 vector.

## Description

This function evaluates the user-specified model, could be 'env', 'henv', 'ienv', 'penv', 'senv' or 'xenv', at new value Xnew. It can perform estimation: find the fitted value when  $X = X_{\text{new}}$ , or prediction: predict  $Y$  when  $X = X_{\text{new}}$ . The covariance matrix and the standard errors are also provided.

## Example

```
load wheatprotein.txt
X = wheatprotein(:, 8);
Y = wheatprotein(:, 1:6);
modelType = 'env';
u = modelselectbic(X, Y, modelType);
ModelOutput = env(X, Y, u);
Xnew = X(2, :)';
PredictOutput = predict_env(ModelOutput, Xnew, 'estimation')
[PredictOutput.value, Y(2, :)] % Compare the fitted value with
the observed value
```

PredictOutput =

```
value: [6x1 double]
covMatrix: [6x6 double]
SE: [6x1 double]
```

ans =

```
474.7135  458.0000
127.4740  112.0000
251.2044  236.0000
380.8280  368.0000
380.9473  383.0000
-6.3287  -15.0000
```

```
load fiberpaper.dat
Y = fiberpaper(:, 1 : 4);
Xtemp = fiberpaper(:, 5 : 7);
X.X1 = Xtemp(:, 3);
X.X2 = Xtemp(:, 1 : 2);
modelType = 'penv';
```



```

u = modelselectbic(X, Y, modelType);
ModelOutput = penv(X, Y, u);
Xnew.X1 = X.X1(1, :)';
Xnew.X2 = X.X2(1, :)';
PredictOutput = predict_penv(ModelOutput, Xnew, 'estimation')
PredictOutput.SE

```

```
PredictOutput =
```

```

      value: [4x1 double]
covMatrix: [4x4 double]
      SE: [4x1 double]

```

```
ans =
```

```

1.4680
0.4234
0.7145
0.3161

```

## 7 testcoefficient

This function tests the null hypothesis  $L * \beta * R = A$  versus the alternative hypothesis  $L * \beta * R \neq A$ , where  $\beta$  is estimated under the model in the envelope family.

### Contents

- Syntax
- Input
- Output
- Description
- Example

### Syntax

```
TestOutput = testcoefficient(ModelOutput, modelType)
TestOutput = testcoefficient(ModelOutput, modelType, TestInput)
```

### Input

**ModelOutput:** A list containing the model outputs from fitting the models.

**modelType:** A string characters indicting the model, choices can be 'env', 'henv', 'ienv', 'penv', 'senv' and 'xenv'.

**TestInput:** A list that specifies the null hypothesis, including L, R, and A. If not provided by the user, default values will be used.

- TestInput.L: The matrix multiplied to  $\beta$  on the left. According to different model, it has different size requirement. Default value will be set if the user does not specify.
- TestInput.R: The matrix multiplied to  $\beta$  on the right. According to different model, it has different size requirement. Default value will be set if the user does not specify.
- TestInput.A: The matrix on the right hand side of the equation. Default value will be set if the user does not specify.

### Output

**TestOutput:** A list containing test statistics, degrees of freedom for the reference chi-squared distribution, and the p-value. At the same time, a table is printed out.

- TestOutput.chisqStatistic: The test statistics. A real number.
- TestOutput.df: The degrees of freedom of the reference chi-squared distribution. A positive integer.
- TestOutput.pValue: p-value of the test. A real number in [0, 1].

## Description

This function tests for hypothesis  $H_0 : L\beta R = A$ , versus  $H_\alpha : L\beta R \neq A$ . The  $\beta$  is estimated by a model in the envelope model. If the user does not specify the values for L, R and A, then the test is equivalent to the standard F test on if  $\beta = 0$  (for 'env', 'ienv', 'penv', 'senv' and 'xenv'), or if the group main effects are all zeros (for 'henv'). The test statistics used is  $\text{vec}(L\beta R - A) \hat{\Sigma}^{-1} \text{vec}(L\beta R - A)^T$ , and the reference distribution is chi-squared distribution with degrees of freedom the same as the length of  $\text{vec}(A)$ .

## Example

```
load wheatprotein.txt
X = wheatprotein(:, 8);
Y = wheatprotein(:, 1:6);
alpha = 0.01;
u = lrt_env(X, Y, alpha);
ModelOutput = env(X, Y, u);
modelType = 'env';
TestOutout = testcoefficient(ModelOutput, modelType);
```

Test Hypothesis	Chisq Statistic	DF	P-value
L * beta * R = A	116.230	6	0.0000

```
load fiberpaper.dat
Y = fiberpaper(:, 1 : 4);
Xtemp = fiberpaper(:, 5 : 7);
X.X1 = Xtemp(:, 3);
X.X2 = Xtemp(:, 1 : 2);
alpha = 0.01;
u = lrt_penv(X, Y, alpha);
ModelOutput = penv(X, Y, u);
r = size(Y, 2);
p1 = size(X.X1, 2);
TestInput.L = rand(2, r);
TestInput.R = rand(p1, 1);
TestInput.A = zeros(2, 1);
TestOutout = testcoefficient_penv(ModelOutput, TestInput);
```

Test Hypothesis	Chisq Statistic	DF	P-value
L * beta * R = A	12.598	2	0.0018

## Part II

## env

### 8 aic\_env

Select the dimension of the envelope subspace using Akaike information criterion.

#### Contents

- Syntax
- Input
- Output
- Description
- Example

#### Syntax

```
u = aic_env(X, Y)
u = aic_env(X, Y, Opts)
```

#### Input

**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.

**Opts:** A list containing the optional input parameter, to control the iterations in `sg_min`. If one or several (even all) fields are not defined, the default settings are used.

- `Opts.maxIter`: Maximum number of iterations. Default value: 300.
- `Opts.ftol`: Tolerance parameter for F. Default value:  $1e-10$ .
- `Opts.gradtol`: Tolerance parameter for dF. Default value:  $1e-7$ .
- `Opts.verbose`: Flag for print out dimension selection process, logical 0 or 1. Default value: 0.

#### Output

**u:** Dimension of the envelope. An integer between 0 and  $r$ .

## Description

This function implements the Akaike information criteria (AIC) to select the dimension of the envelope subspace.

## Example

```
load wheatprotein.txt
X = wheatprotein(:, 8);
Y = wheatprotein(:, 1:6);
u = aic_env(X, Y)
```

u =

1

## 9 bic\_env

Select the dimension of the envelope subspace using Bayesian information criterion.

### Contents

- Syntax
- Input
- Output
- Description
- Example

### Syntax

```
u = bic_env(X, Y)
u = bic_env(X, Y, Opts)
```

### Input

**X**: Predictors. An  $n$  by  $p$  matrix,  $p$  is the number of predictors and  $n$  is the number of observations. The predictors can be univariate or multivariate, discrete or continuous.

**Y**: Multivariate responses. An  $n$  by  $r$  matrix,  $r$  is the number of responses. The responses must be continuous variables.

**Opts**: A list containing the optional input parameter, to control the iterations in `sg_min`. If one or several (even all) fields are not defined, the default settings are used.

- `Opts.maxIter`: Maximum number of iterations. Default value: 300.
- `Opts.ftol`: Tolerance parameter for F. Default value:  $1e-10$ .
- `Opts.gradtol`: Tolerance parameter for dF. Default value:  $1e-7$ .
- `Opts.verbose`: Flag for print out dimension selection process, logical 0 or 1. Default value: 0.

### Output

**u**: Dimension of the envelope. An integer between 0 and  $r$ .

### Description

This function implements the Bayesian information criteria (BIC) to select the dimension of the envelope subspace.

## Example

```
load wheatprotein.txt
X = wheatprotein(:, 8);
Y = wheatprotein(:, 1:6);
u = bic_env(X, Y)
```

u =

1



## 10 bstrp\_env

Compute bootstrap standard error for the envelope model.

### Contents

- Syntax
- Input
- Output
- Description
- Example

### Syntax

```
bootse = bstrp_env(X, Y, u, B)
bootse = bstrp_env(X, Y, u, B, Opts)
```

### Input

**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 subspace. A positive integer between 0 and  $r$ .

**B**: Number of bootstrap samples. A positive integer.

**Opts**: A list containing the optional input parameter, to control the iterations in `sg_min`. If one or several (even all) fields are not defined, the default settings are used.

- `Opts.maxIter`: Maximum number of iterations. Default value: 300.
- `Opts.ftol`: Tolerance parameter for F. Default value:  $1e-10$ .
- `Opts.gradtol`: Tolerance parameter for dF. Default value:  $1e-7$ .
- `Opts.verbose`: Flag for print out the number of bootstrap samples, logical 0 or 1. Default value: 0.

### Output

**bootse**: The standard error for elements in  $\beta$  computed by bootstrap. An  $r$  by  $p$  matrix.

## Description

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

## Example

```
load wheatprotein.txt
X = wheatprotein(:, 8);
Y = wheatprotein(:, 1:6);
alpha = 0.01;
u = lrt_env(X, Y, alpha)

u =

    1

B = 100;
bootse = bstrp_env(X, Y, u, B)

bootse =

    0.2893
    0.4260
    0.3523
    0.5628
    0.1675
    0.6192
```

## 11 dF4env

The first derivative of the objective function for computing the envelope subspace.

### Contents

- Syntax
- Input
- Output
- Description

### Syntax

```
df = dF4env(R, DataParameter)
```

### Input

**R**: An  $r$  by  $u$  semi orthogonal matrix,  $0 < u \leq r$ .

**DataParameter**: A structure that contains the statistics calculated from the data.

### Output

**df**: An  $r$  by  $u$  matrix containing the value of the derivative function evaluated at  $R$ .

### Description

The objective function is derived in Section 4.3 in Cook et al. (2010) by using maximum likelihood estimation. This function is the derivative of the objective function.

## 12 env

Fit the envelope model.

### Contents

- Syntax
- Input
- Output
- Description
- References
- Example

### Syntax

```
ModelOutput = env(X, Y, u)
ModelOutput = env(X, Y, u, Opts)
```

### Input

**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, and  $r$  should be strictly greater than  $p$ .

**u:** Dimension of the envelope. An integer between 0 and  $r$ .

**Opts:** A list containing the optional input parameter, to control the iterations in `sg_min`. If one or several (even all) fields are not defined, the default settings are used.

- `Opts.maxIter`: Maximum number of iterations. Default value: 300.
- `Opts.ftol`: Tolerance parameter for F. Default value:  $1e-10$ .
- `Opts.gradtol`: Tolerance parameter for dF. Default value:  $1e-7$ .
- `Opts.verbose`: Flag for print out Grassmann manifold optimization process, logical 0 or 1. Default value: 0.

### Output

**ModelOutput:** A list that contains the maximum likelihood estimators and some statistics.

- `ModelOutput.beta`: The envelope estimator of the regression coefficients  $\beta$ . An  $r$  by  $p$  matrix.

- `ModelOutput.Sigma`: The envelope estimator of the error covariance matrix. An  $r$  by  $r$  matrix.
- `ModelOutput.Gamma`: The orthogonal basis of the envelope subspace. An  $r$  by  $u$  semi-orthogonal matrix.
- `ModelOutput.Gamma0`: The orthogonal basis of the complement of the envelope subspace. An  $r$  by  $r-u$  semi-orthogonal matrix.
- `ModelOutput.eta`: The coordinates of  $\beta$  with respect to `Gamma`. An  $u$  by  $p$  matrix.
- `ModelOutput.Omega`: The coordinates of `Sigma` with respect to `Gamma`. An  $u$  by  $u$  matrix.
- `ModelOutput.Omega0`: The coordinates of `Sigma` with respect to `Gamma0`. An  $r-u$  by  $r-u$  matrix.
- `ModelOutput.alpha`: The estimated intercept in the envelope model. An  $r$  by 1 vector.
- `ModelOutput.l`: The maximized log likelihood function. A real number.
- `ModelOutput.covMatrix`: The asymptotic covariance of  $\text{vec}(\beta)$ . An  $rp$  by  $rp$  matrix. The covariance matrix returned are asymptotic. For the actual standard errors, multiply by  $1/n$ .
- `ModelOutput.asyEnv`: The asymptotic standard error for elements in  $\beta$  under the envelope model. An  $r$  by  $p$  matrix. The standard errors returned are asymptotic, for actual standard errors, multiply by  $1/\sqrt{n}$ .
- `ModelOutput.ratio`: The asymptotic standard error ratio of the standard multivariate linear regression estimator over the envelope estimator, for each element in  $\beta$ . An  $r$  by  $p$  matrix.
- `ModelOutput.np`: The number of parameters in the envelope model. A positive integer.
- `ModelOutput.n`: The number of observations in the data. A positive integer.

## Description

This function fits the envelope model to the responses and predictors, using the maximum likelihood estimation. When the dimension of the envelope is between 1 and  $r-1$ , we implemented the algorithm in Cook et al. (2010). 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.

## References

1. The codes is implemented based on the algorithm in Section 4.3 of Cook et al (2010).
2. The Grassmann manifold optimization step calls the package `sg_min` 2.4.1 by Ross Lippert (<http://web.mit.edu/~ripper/www.sgmin.html>).

## Example

The following codes will reconstruct the results in the wheat protein data example in Cook et al. (2010).

```
load wheatprotein.txt
X = wheatprotein(:, 8);
Y = wheatprotein(:, 1:6);
alpha = 0.01;
u = lrt_env(X, Y, alpha)

u =

    1

ModelOutput = env(X, Y, u)

ModelOutput =

    beta: [6x1 double]
    Sigma: [6x6 double]
    Gamma: [6x1 double]
    Gamma0: [6x5 double]
    eta: 8.5647
    Omega: 7.8762
    Omega0: [5x5 double]
    alpha: [6x1 double]
    l: -850.7592
    covMatrix: [6x6 double]
    asyEnv: [6x1 double]
    ratio: [6x1 double]
    np: 28
    n: 50

ModelOutput.Omega

ans =

    7.8762
```

```
eig(ModelOutput.Omega0)
```

```
ans =
```

```
1.0e+03 *
```

```
6.5166
```

```
0.2083
```

```
0.0201
```

```
0.0004
```

```
0.0003
```

```
ModelOutput.ratio
```

```
ans =
```

```
28.0945
```

```
18.4326
```

```
23.6384
```

```
16.3211
```

```
65.8245
```

```
6.4668
```

## 13 F4env

Objective function for computing the envelope subspace.

### Contents

- Syntax
- Input
- Output
- Description

### Syntax

```
f = F4env(R, DataParameter)
```

### Input

**R**: An  $r$  by  $u$  semi orthogonal matrix,  $0 < u \leq r$ .

**DataParameter**: A structure that contains the statistics calculated from the data.

### Output

**f**: A scalar containing the value of the objective function evaluated at **R**.

### Description

The objective function is derived in Section 4.3 of Cook et al. (2010) using maximum likelihood estimation. The columns of the semi-orthogonal matrix that minimizes this function span the estimated envelope subspace.



## 14 lrt\_env

Select the dimension of the envelope subspace using likelihood ratio testing.

### Contents

- Syntax
- Input
- Output
- Description
- Example

### Syntax

```
u = lrt_env(X, Y, alpha)
u = lrt_env(X, Y, alpha, Opts)
```

### Input

**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. A real number between 0 and 1, often taken at 0.05 or 0.01.

**Opts**: A list containing the optional input parameter, to control the iterations in `sg_min`. If one or several (even all) fields are not defined, the default settings are used.

- `Opts.maxIter`: Maximum number of iterations. Default value: 300.
- `Opts.ftol`: Tolerance parameter for F. Default value:  $1e-10$ .
- `Opts.gradtol`: Tolerance parameter for dF. Default value:  $1e-7$ .
- `Opts.verbose`: Flag for print out dimension selection process, logical 0 or 1. Default value: 0.

### Output

**u**: Dimension of the envelope. An integer between 0 and  $r$ .

## Description

This function implements the likelihood ratio testing procedure to select the dimension of the envelope subspace, with pre-specified significance level  $\alpha$ .

## Example

```
load wheatprotein.txt
X = wheatprotein(:, 8);
Y = wheatprotein(:, 1:6);
alpha = 0.01;
u = lrt_env(X, Y, alpha)
```

u =

1

## 15 predict\_env

Perform estimation or prediction under the envelope model.

### Contents

- Syntax
- Input
- Output
- Description
- Example

### Syntax

```
PredictOutput = predict_env(ModelOutput, Xnew, infType)
```

### Input

**ModelOutput:** A list containing the maximum likelihood estimators and other statistics inherited from env.

**Xnew:** The value of X with which to estimate or predict Y. A p by 1 vector.

**infType:** A string of characters indicting the inference type, the choices can be 'estimation' or 'prediction'.

### Output

**PredictOutput:** A list containing the results of the inference.

- PredictOutput.value: The fitted value or the prediction value evaluated at Xnew. An r by 1 vector.
- PredictOutput.covMatrix: The covariance matrix of PredictOutput.value. An r by r matrix.
- PredictOutput.SE: The standard error of elements in PredictOutput.value. An r by 1 vector.

### Description

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

## Example

```
load wheatprotein.txt
X = wheatprotein(:, 8);
Y = wheatprotein(:, 1:6);
alpha = 0.01;
u = lrt_env(X, Y, alpha);
ModelOutput = env(X, Y, u);
Xnew = X(2, :)';
PredictOutput = predict_env(ModelOutput, Xnew, 'estimation')
[PredictOutput.value, Y(1, :)] % Compare the fitted value with the data
PredictOutput.SE
```

```
PredictOutput =
```

```
    value: [6x1 double]
 covMatrix: [6x6 double]
         SE: [6x1 double]
```

```
ans =
```

```
474.7135  468.0000
127.4740  123.0000
251.2044  246.0000
380.8280  374.0000
380.9473  386.0000
 -6.3287 -11.0000
```

```
ans =
```

```
4.8892
4.0227
4.3237
4.7470
6.8186
2.6948
```

```
PredictOutput = predict_env(ModelOutput, Xnew, 'prediction')
PredictOutput.SE
```

```
PredictOutput =
```

```
value: [6x1 double]  
covMatrix: [6x6 double]  
SE: [6x1 double]
```

```
ans =
```

```
474.7135  
127.4740  
251.2044  
380.8280  
380.9473  
-6.3287
```

```
ans =
```

```
34.9161  
28.7280  
30.8775  
33.9006  
48.6945  
19.2448
```

## 16 testcoefficient\_env

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

### Contents

- Syntax
- Input
- Output
- Description
- Example

### Syntax

```
TestOutput = testcoefficient_env(ModelOutput)
TestOutput = testcoefficient_env(ModelOutput, TestInput)
```

### Input

**ModelOutput:** A list containing the maximum likelihood estimators and other statistics inherited from env.

**TestInput:** A list that specifies the null hypothesis, including L, R, and A. If not provided by the user, default values will be used.

- TestInput.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. Default value: identity matrix  $I_r$ .
- TestInput.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. Default value: identity matrix  $I_p$ .
- TestInput.A: The matrix on the right hand side of the equation. It is a d1 by d2 matrix. Default value: d1 by d2 zero matrix.

### Output

**TestOutput:** A list containing test statistics, degrees of freedom for the reference chi-squared distribution, and the p-value. At the same time, a table is printed out.

- TestOutput.chisqStatistic: The test statistics. A real number.
- TestOutput.df: The degrees of freedom of the reference chi-squared distribution. A positive integer.
- TestOutput.pValue: p-value of the test. A real number in  $[0, 1]$ .

## Description

This function tests for hypothesis  $H_0 : L\beta R = A$ , versus  $H_\alpha : L\beta R \neq A$ . The  $\beta$  is estimated by the envelope model. If the user does not specify the values for L, R and A, then the test is equivalent to the standard F test on if  $\beta = 0$ . The test statistics used is  $\text{vec}(L\beta R - A) \hat{\Sigma}^{-1} \text{vec}(L\beta R - A)^T$ , and the reference distribution is chi-squared distribution with degrees of freedom  $d1 * d2$ .

## Example

```
load wheatprotein.txt
X = wheatprotein(:, 8);
Y = wheatprotein(:, 1:6);
alpha = 0.01;
u = lrt_env(X, Y, alpha);
ModelOutput = env(X, Y, u);
TestOutout = testcoefficient_env(ModelOutput);
```

Test Hypothesis	Chisq Statistic	DF	P-value
L * beta * R = A	116.230	6	0.0000

```
r = size(Y, 2);
p = size(X, 2);
TestInput.L = rand(2, r);
TestInput.R = rand(p, 1);
TestInput.A = zeros(2, 1);
TestOutout = testcoefficient_senv(ModelOutput, TestInput);
```

Test Hypothesis	Chisq Statistic	DF	P-value
L * beta * R = A	61.344	2	0.0000

## Part III

# henv

## 17 aic\_henv

Select the dimension of the envelope subspace using Akaike information criterion for the heteroscedastic envelope model.

### Contents

- Syntax
- Input
- Output
- Description
- Example

### Syntax

```
u = aic_henv(X, Y)
u = aic_henv(X, Y, Opts)
```

### Input

**X**: Group indicators. A matrix with  $n$  rows.  $X$  can only have  $p$  unique rows, where  $p$  is the number of groups. For example, if there are two groups,  $X$  can only have 2 different kinds of rows, such as  $(0, 1)$  and  $(1, 0)$ , or  $(1, 0, 10)$  and  $(0, 5, 6)$ . The number of columns is not restricted, as long as  $X$  only has  $p$  unique rows.

**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, and  $r$  should be greater than  $p$ .

**Opts**: A list containing the optional input parameter, to control the iterations in `sg_min`. If one or several (even all) fields are not defined, the default settings are used.

- `Opts.maxIter`: Maximum number of iterations. Default value: 300.
- `Opts.ftol`: Tolerance parameter for  $F$ . Default value:  $1e-10$ .
- `Opts.gradtol`: Tolerance parameter for  $dF$ . Default value:  $1e-7$ .
- `Opts.verbose`: Flag for print out dimension selection process, logical 0 or 1. Default value: 0.



## Output

**u**: Dimension of the envelope. An integer between 0 and  $r$ .

## Description

This function implements the Akaike information criteria (AIC) to select the dimension of the envelope subspace for the heteroscedastic envelope model.

## Example

```
load waterstrider.mat  
u = aic_henv(X, Y)
```

u =

6

## 18 bic\_henv

Select the dimension of the envelope subspace using Bayesian information criterion for the heteroscedastic envelope model.

### Contents

- Syntax
- Input
- Output
- Description
- Example

### Syntax

```
u = bic_henv(X, Y)
u = bic_henv(X, Y, Opts)
```

### Input

**X**: Group indicators. A matrix with  $n$  rows.  $X$  can only have  $p$  unique rows, where  $p$  is the number of groups. For example, if there are two groups,  $X$  can only have 2 different kinds of rows, such as  $(0, 1)$  and  $(1, 0)$ , or  $(1, 0, 10)$  and  $(0, 5, 6)$ . The number of columns is not restricted, as long as  $X$  only has  $p$  unique rows.

**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, and  $r$  should be greater than  $p$ .

**Opts**: A list containing the optional input parameter, to control the iterations in `sg_min`. If one or several (even all) fields are not defined, the default settings are used.

- `Opts.maxIter`: Maximum number of iterations. Default value: 300.
- `Opts.ftol`: Tolerance parameter for  $F$ . Default value:  $1e-10$ .
- `Opts.gradtol`: Tolerance parameter for  $dF$ . Default value:  $1e-7$ .
- `Opts.verbose`: Flag for print out dimension selection process, logical 0 or 1. Default value: 0.

### Output

**u**: Dimension of the envelope. An integer between 0 and  $r$ .

## Description

This function implements the Bayesian information criteria (BIC) to select the dimension of the envelope subspace for the heteroscedastic envelope model.

## Example

```
load waterstrider.mat  
u = bic_henv(X, Y)
```

u =

4

## 19 bstrp\_henv

Compute bootstrap standard error for the heteroscedastic envelope model.

### Contents

- Syntax
- Input
- Output
- Description
- Example

### Syntax

```
bootse = bstrp_henv(X, Y, u, B)
bootse = bstrp_henv(X, Y, u, B, Opts)
```

### Input

**X**: Group indicators. A matrix with  $n$  rows.  $X$  can only have  $p$  unique rows, where  $p$  is the number of groups. For example, if there are two groups,  $X$  can only have 2 different kinds of rows, such as  $(0, 1)$  and  $(1, 0)$ , or  $(1, 0, 10)$  and  $(0, 5, 6)$ . The number of columns is not restricted, as long as  $X$  only has  $p$  unique rows.

**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, and  $r$  should be greater than  $p$ .

**u**: Dimension of the envelope subspace. A positive integer between 0 and  $r$ .

**B**: Number of bootstrap samples. A positive integer.

**Opts**: A list containing the optional input parameter, to control the iterations in `sg_min`. If one or several (even all) fields are not defined, the default settings are used.

- `Opts.maxIter`: Maximum number of iterations. Default value: 300.
- `Opts.ftol`: Tolerance parameter for  $F$ . Default value:  $1e-10$ .
- `Opts.gradtol`: Tolerance parameter for  $dF$ . Default value:  $1e-7$ .
- `Opts.verbose`: Flag for print out the number of bootstrap samples, logical 0 or 1. Default value: 0.

## Output

**bootse**: The standard error for elements in  $\beta$  computed by bootstrap. An  $r$  by  $p$  matrix.

## Description

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

## Example

```
load waterstrider.mat

u = lrt_henv(X, Y, 0.01)

u =

     6

B = 100;
bootse = bstrp_henv(X, Y, u, B)

bootse =

    0.0305    0.0466    0.0647
    0.0309    0.0485    0.0682
    0.0305    0.0432    0.0638
    0.0205    0.0289    0.0425
    0.0385    0.0553    0.0799
    0.0295    0.0427    0.0618
    0.0389    0.0567    0.0819
    0.0321    0.0463    0.0665
```

## 20 dF4henv

The first derivative of the objective function for computing the envelope subspace in the heteroscedastic envelope model.

### Contents

- Syntax
- Input
- Output
- Description

### Syntax

```
df = dF4henv(R, DataParameter)
```

### Input

**R:** An  $r$  by  $u$  semi orthogonal matrix,  $0 < u \leq r$ .

**DataParameter:** A structure that contains the statistics calculated from the data.

### Output

**df:** An  $r$  by  $u$  matrix containing the value of the derivative function evaluated at  $R$ .

### Description

The objective function is derived in Section 2.2 in Su and Cook (2012) by using maximum likelihood estimation. This function is the derivative of the objective function.

## 21 F4henv

Objective function for computing the envelope subspace in heteroscedastic envelope model.

### Contents

- Syntax
- Input
- Output
- Description

### Syntax

```
f = F4henv(R, DataParameter)
```

### Input

**R:** An  $r$  by  $u$  semi orthogonal matrix,  $0 < u \leq r$ .

**DataParameter:** A structure that contains the statistics calculated from the data.

### Output

**f:** A scalar containing the value of the objective function evaluated at  $R$ .

### Description

The objective function is derived in Section 2.2 of Su and Cook (2012) using maximum likelihood estimation. The columns of the semi-orthogonal matrix that minimizes this function span the estimated envelope subspace in the heteroscedastic envelope model.

## 22 henv

Fit the heteroscedastic envelope model.

### Contents

- Syntax
- Input
- Output
- Description
- References
- Example

### Syntax

```
ModelOutput = henv(X, Y, u)
ModelOutput = henv(X, Y, u, Opts)
```

### Input

**X**: Group indicators. A matrix with  $n$  rows.  $X$  can only have  $p$  unique rows, where  $p$  is the number of groups. For example, if there are two groups,  $X$  can only have 2 different kinds of rows, such as  $(0, 1)$  and  $(1, 0)$ , or  $(1, 0, 10)$  and  $(0, 5, 6)$ . The number of columns is not restricted, as long as  $X$  only has  $p$  unique rows.

**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, and  $r$  should be greater than  $p$ .

**u**: Dimension of the envelope. An integer between 0 and  $r$ .

**Opts**: A list containing the optional input parameter, to control the iterations in `sg_min`. If one or several (even all) fields are not defined, the default settings are used.

- `Opts.maxIter`: Maximum number of iterations. Default value: 300.
- `Opts.ftol`: Tolerance parameter for  $F$ . Default value:  $1e-10$ .
- `Opts.gradtol`: Tolerance parameter for  $dF$ . Default value:  $1e-7$ .
- `Opts.verbose`: Flag for print out Grassmann manifold optimization process, logical 0 or 1. Default value: 0.



## Output

**ModelOutput:** A list that contains the maximum likelihood estimators and some statistics.

- **ModelOutput.mu:** The heteroscedastic envelope estimator of the grand mean. A  $r$  by 1 vector.
- **ModelOutput.mug:** The heteroscedastic envelope estimator of the group mean. A  $r$  by  $p$  matrix, the  $i$ th column of the matrix contains the mean for the  $i$ th group.
- **ModelOutput.Yfit:** A  $n$  by  $r$  matrix, the  $i$ th row gives the group mean of the group that the  $i$ th observation belongs to. As  $X$  is just a group indicator, and is not ordinal, **ModelOutput.mug** alone does not tell which group corresponds to which group mean.
- **ModelOutput.Gamma:** The orthogonal basis of the envelope subspace. An  $r$  by  $u$  semi-orthogonal matrix.
- **ModelOutput.Gamma0:** The orthogonal basis of the complement of the envelope subspace. An  $r$  by  $r-u$  semi-orthogonal matrix.
- **ModelOutput.beta:** The heteroscedastic envelope estimator of the group main effect. An  $r$  by  $p$  matrix, the  $i$ th column of the matrix contains the main effect for the  $i$ th group.
- **ModelOutput.groupInd:** A matrix containing the unique values of group indicators. The matrix has  $p$  rows. The group mean of the  $i$ th row is stored in the  $i$ th column of **ModelOutput.mug**.
- **ModelOutput.Sigma:** The heteroscedastic envelope estimator of the error covariance matrix. A three dimensional matrix with dimension  $r$ ,  $r$  and  $p$ , **ModelOutput.Sigma**(:, :,  $i$ ) contains the estimated covariance matrix for the  $i$ th group.
- **ModelOutput.eta:** The coordinates of  $\beta$  with respect to **Gamma**. An  $u$  by  $p$  matrix, the  $i$ th column contains the coordinates of the main effect of the  $i$ th group with respect to **Gamma**.
- **ModelOutput.Omega:** The coordinates of **Sigma** with respect to **Gamma**. An  $u$  by  $u$  by  $p$  matrix, **ModelOutput.Omega**(:, :,  $i$ ) contains the coordinates of the covariance matrix of the  $i$ th group with respect to **Gamma**.
- **ModelOutput.Omega0:** The coordinates of **Sigma** with respect to **Gamma0**. An  $r - u$  by  $r - u$  matrix.
- **ModelOutput.l:** The maximized log likelihood function. A real number.
- **ModelOutput.np:** The number of parameters in the heteroscedastic envelope model. A positive integer.
- **ModelOutput.covMatrix:** The asymptotic covariance of  $(\mu', \text{vec}(\beta'))'$ . An  $r(p + 1)$  by  $r(p + 1)$  matrix. The covariance matrix returned are asymptotic. For the actual standard errors, multiply by  $1 / n$ .
- **ModelOutput.asyHenv:** The asymptotic standard errors 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  $1 / \sqrt{n}$ .

- `ModelOutput.ratio`: The asymptotic standard error ratio of the standard multivariate linear regression estimator over the heteroscedastic envelope estimator. An  $r$  by  $p$  matrix, the  $(i, j)$ th element in `ModelOutput.ratio` is the elementwise standard error ratio for the  $i$ th element in the  $j$ th group mean effect.
- `ModelOutput.ng`: The number of observations in each group. A  $p$  by 1 vector.

## Description

This function fits the heteroscedastic envelope model to the responses and predictors, 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 (2012). When the dimension is  $r$ , then the envelope model degenerates to the standard multivariate linear model for comparing group means. When the dimension is 0, it means there is not any group effect, and the fitting is different.

## References

1. The codes is implemented based on the algorithm in Section 2.2 of Su and Cook (2012).
2. The Grassmann manifold optimization step calls the package `sg_min` 2.4.1 by Ross Lippert (<http://web.mit.edu/~ripper/www.sgmin.html>).

## Example

The following codes produce the results of the water strider example in Su and Cook (2011).

```
load waterstrider.mat
u = lrt_henv(X, Y, 0.01)

u =

6

ModelOutput = henv(X, Y, u)
ModelOutput.ratio
```

```
ModelOutput =
```

```
      mu: [8x1 double]
      mug: [8x3 double]
      Yfit: [90x8 double]
      Gamma: [8x6 double]
      Gamma0: [8x2 double]
      beta: [8x3 double]
      groupInd: [3x2 double]
      Sigma: [8x8x3 double]
      eta: [6x3 double]
      Omega: [6x6x3 double]
      Omega0: [2x2 double]
      np: 98
      l: 1.0051e+03
      covMatrix: [32x32 double]
      asyHenv: [8x3 double]
      ratio: [8x3 double]
      ng: [3x1 double]
```

```
ans =
```

```
6.5439    11.2830    6.4954
4.6325     5.3226    4.7242
4.4456     5.0741    4.4198
4.7338     6.2469    5.1937
8.0377    12.5386    9.4823
9.5067    11.5974   11.3444
11.8632   15.6080   12.5611
6.9792    11.1559   10.1002
```

## 23 lrt\_henv

Select the dimension of the envelope subspace using likelihood ratio testing for the heteroscedastic envelope model.

### Contents

- Syntax
- Input
- Output
- Description
- Example

### Syntax

```
u = lrt_henv(X, Y, alpha)
u = lrt_henv(X, Y, alpha, Opts)
```

### Input

**X**: Group indicators. A matrix with  $n$  rows.  $X$  can only have  $p$  unique rows, where  $p$  is the number of groups. For example, if there are two groups,  $X$  can only have 2 different kinds of rows, such as  $(0, 1)$  and  $(1, 0)$ , or  $(1, 0, 10)$  and  $(0, 5, 6)$ . The number of columns is not restricted, as long as  $X$  only has  $p$  unique rows.

**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, and  $r$  should be greater than  $p$ .

**alpha**: Significance level for testing. A real number between 0 and 1, often taken at 0.05 or 0.01.

**Opts**: A list containing the optional input parameter, to control the iterations in `sg_min`. If one or several (even all) fields are not defined, the default settings are used.

- `Opts.maxIter`: Maximum number of iterations. Default value: 300.
- `Opts.ftol`: Tolerance parameter for  $F$ . Default value:  $1e-10$ .
- `Opts.gradtol`: Tolerance parameter for  $dF$ . Default value:  $1e-7$ .
- `Opts.verbose`: Flag for print out dimension selection process, logical 0 or 1. Default value: 0.

### Output

**u**: Dimension of the envelope. An integer between 0 and  $r$ .

## Description

This function implements the likelihood ratio testing procedure to select the dimension of the envelope subspace in heteroscedastic envelope model, with pre-specified significance level  $\alpha$ .

## Example

```
load waterstrider.mat  
u = lrt_henv(X, Y, 0.01)
```

u =

6

## 24 predict\_henv

Perform estimation or prediction under the heteroscedastic envelope model.

### Contents

- Syntax
- Input
- Output
- Description
- Example

### Syntax

```
PredictOutput = predict_henv(ModelOutput, Xnew, infType)
```

### Input

**ModelOutput:** A list containing the maximum likelihood estimators and other statistics inherited from henv.

**Xnew:** A group indicator. It must be a column vector, whose transpose is the same as one of the group indicators from the original data.

**infType:** A string of characters indicating the inference type, the choices can be 'estimation' or 'prediction'.

### Output

**PredictOutput:** A list containing the results of the inference.

- PredictOutput.value: The fitted value or the prediction value evaluated at Xnew. An r by 1 vector.
- PredictOutput.covMatrix: The covariance matrix of PredictOutput.value. An r by r matrix.
- PredictOutput.SE: The standard error of elements in PredictOutput.value. An r by 1 vector.

### Description

This function evaluates the inner envelope model at new value Xnew. It can perform estimation: find the group mean for the group indicated by Xnew, or prediction: predict Y for the group indicated by Xnew. The covariance matrix and the standard errors are also provided.

## Example

```
load waterstrider.mat
u = lrt_henv(X, Y, 0.01);
ModelOutput = henv(X, Y, u);
ModelOutput.groupInd
ModelOutput.mug
Xnew = X(1, :)'
```

```
ans =
```

```
-1    -1
 0     1
 1     0
```

```
ans =
```

```
-1.1417  -1.1267  -1.0845
-1.4063  -1.4067  -1.3132
-1.3314  -1.3336  -1.2152
-0.3113  -0.1839  -0.1736
 0.4003   0.3847   0.3072
 0.4107   0.3753   0.3735
 0.3467   0.3271   0.3179
-0.1954  -0.2100  -0.3488
```

```
Xnew =
```

```
1
0
```

```
PredictOutput = predict_henv(ModelOutput, Xnew, 'estimation')
PredictOutput.value %This is the 3rd group mean
PredictOutput.SE
```

```
PredictOutput =
```

```
value: [8x1 double]
covMatrix: [8x8 double]
SE: [8x1 double]
```

```
ans =
```

```
-1.0845  
-1.3132  
-1.2152  
-0.1736  
0.3072  
0.3735  
0.3179  
-0.3488
```

```
ans =
```

```
0.0682  
0.0695  
0.0651  
0.0436  
0.0832  
0.0636  
0.0847  
0.0698
```

```
PredictOutput = predict_henv(ModelOutput, Xnew, 'prediction')  
PredictOutput$SE
```

```
PredictOutput =
```

```
value: [8x1 double]  
covMatrix: [8x8 double]  
SE: [8x1 double]
```

```
ans =
```

```
0.3720  
0.3812  
0.3581  
0.2398  
0.4612  
0.3519  
0.4710  
0.3854
```



## 25 testcoefficient\_henv

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

### Contents

- Syntax
- Input
- Output
- Description
- Example

### Syntax

```
TestOutput = testcoefficient_henv(ModelOutput)
TestOutput = testcoefficient_henv(ModelOutput, TestInput)
```

### Input

**ModelOutput:** A list containing the maximum likelihood estimators and other statistics inherited from henv.

**TestInput:** A list that specifies the null hypothesis, including L, R, and A. If not provided by the user, default values will be used.

- TestInput.L: The matrix multiplied to  $\beta$  on the left. It is a  $d1$  by  $r$  matrix, while  $d1$  is less than or equal to  $r - 1$ . Default value: identity matrix  $I_r$ .
- TestInput.R: The matrix multiplied to  $\beta$  on the right. It is a  $p$  by  $d2$  matrix, while  $d2$  is less than or equal to  $p$ . Default value: identity matrix  $(I_{p-1}, 0_{(p-1) \times 1})^T$ . This is because the columns of  $\beta$  sum to 0. Then we cannot use  $I_p$  as default.
- TestInput.A: The matrix on the right hand side of the equation. It is a  $d1$  by  $d2$  matrix. Default value:  $d1$  by  $d2$  zero matrix.

### Output

**TestOutput:** A list containing test statistics, degrees of freedom for the reference chi-squared distribution, and the p-value. At the same time, a table is printed out.

- TestOutput.chisqStatistic: The test statistics. A real number.
- TestOutput.df: The degrees of freedom of the reference chi-squared distribution. A positive integer.
- TestOutput.pValue: p-value of the test. A real number in  $[0, 1]$ .

## Description

This function tests for hypothesis  $H_0 : L\beta R = A$ , versus  $H_\alpha : L\beta R \neq A$ . The  $\beta$  is estimated by the heteroscedastic envelope model. If the user does not specify the values for L, R and A, then the test is equivalent to the standard F test on if all the main group effects are 0. The test statistics used is  $\text{vec}(L\beta R - A) \hat{\Sigma}^{-1} \text{vec}(L\beta R - A)^T$ , and the reference distribution is chi-squared distribution with degrees of freedom  $d1 * d2$ .

## Example

```
load waterstrider.mat
u = lrt_henv(X, Y, 0.01);
ModelOutput = henv(X, Y, u);
TestOutout = testcoefficient_henv(ModelOutput);
```

Test Hypothesis	Chisq Statistic	DF	P-value
L * beta * R = A	226.256	16	0.0000

```
r = size(Y, 2);
p = size(ModelOutput.beta, 2);
TestInput.L = rand(2, r);
TestInput.R = rand(p, 1);
TestInput.A = zeros(2, 1);
TestOutout = testcoefficient_henv(ModelOutput, TestInput);
```

Test Hypothesis	Chisq Statistic	DF	P-value
L * beta * R = A	23.429	2	0.0000

## Part IV

# ienv

## 26 aic\_ienv

Select the dimension of the inner envelope subspace using Akaike information criterion.

### Contents

- Syntax
- Input
- Output
- Description
- Example

### Syntax

```
u = aic_ienv(X, Y)
u = aic_ienv(X, Y, Opts)
```

### Input

**X**: Predictors. An  $n$  by  $p$  matrix,  $p$  is the number of predictors and  $n$  is the number of observations. The predictors can be univariate or multivariate, discrete or continuous.

**Y**: Multivariate responses. An  $n$  by  $r$  matrix,  $r$  is the number of responses. The responses must be continuous variables.

**Opts**: A list containing the optional input parameter, to control the iterations in `sg_min`. If one or several (even all) fields are not defined, the default settings are used.

- `Opts.maxIter`: Maximum number of iterations. Default value: 300.
- `Opts.ftol`: Tolerance parameter for F. Default value:  $1e-10$ .
- `Opts.gradtol`: Tolerance parameter for dF. Default value:  $1e-7$ .
- `Opts.verbose`: Flag for print out dimension selection process, logical 0 or 1. Default value: 0.

### Output

**u**: Dimension of the inner envelope. An integer between 0 and  $p$  or equal to  $r$ .

## Description

This function implements the Akaike information criteria (AIC) to select the dimension of the inner envelope subspace.

## Example

```
load irisf.mat  
u = aic_ienv(X, Y)
```

```
u =
```

```
1
```

## 27 bic\_ienv

Select the dimension of the inner envelope subspace using Bayesian information criterion.

### Contents

- Syntax
- Input
- Output
- Description
- Example

### Syntax

```
u = bic_ienv(X, Y)
u = bic_ienv(X, Y, Opts)
```

### Input

**X**: Predictors. An  $n$  by  $p$  matrix,  $p$  is the number of predictors and  $n$  is the number of observations. The predictors can be univariate or multivariate, discrete or continuous.

**Y**: Multivariate responses. An  $n$  by  $r$  matrix,  $r$  is the number of responses. The responses must be continuous variables.

**Opts**: A list containing the optional input parameter, to control the iterations in `sg_min`. If one or several (even all) fields are not defined, the default settings are used.

- `Opts.maxIter`: Maximum number of iterations. Default value: 300.
- `Opts.ftol`: Tolerance parameter for F. Default value:  $1e-10$ .
- `Opts.gradtol`: Tolerance parameter for dF. Default value:  $1e-7$ .
- `Opts.verbose`: Flag for print out dimension selection process, logical 0 or 1. Default value: 0.

### Output

**u**: Dimension of the inner envelope. An integer between 0 and  $p$  or equal to  $r$ .

## Description

This function implements the Bayesian information criteria (BIC) to select the dimension of the inner envelope subspace.

## Example

```
load irisf.mat
u = bic_ienv(X, Y)
```

u =

1

## 28 bstrp\_ienv

Compute bootstrap standard error for the inner envelope model.

### Contents

- Syntax
- Input
- Output
- Description
- Example

### Syntax

```
bootse = bstrp_ienv(X, Y, u, B)
bootse = bstrp_ienv(X, Y, u, B, Opts)
```

### Input

**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 inner envelope. An integer between 0 and  $p$  or equal to  $r$ .

**B**: Number of bootstrap samples. A positive integer.

**Opts**: A list containing the optional input parameter, to control the iterations in `sg_min`. If one or several (even all) fields are not defined, the default settings are used.

- `Opts.maxIter`: Maximum number of iterations. Default value: 300.
- `Opts.ftol`: Tolerance parameter for F. Default value:  $1e-10$ .
- `Opts.gradtol`: Tolerance parameter for dF. Default value:  $1e-7$ .
- `Opts.verbose`: Flag for print out the number of bootstrap samples, logical 0 or 1. Default value: 0.

### Output

**bootse**: The standard error for elements in  $\beta$  computed by bootstrap. An  $r$  by  $p$  matrix.

## Description

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

## Example

```
load irisf.mat

u = bic_ienv(X, Y)

u =

    1

B = 100;
bootse = bstrp_ienv(X, Y, u, B)

bootse =

    13.4695    4.9601
     7.4709    2.7315
    14.9316    5.2913
     8.7597    3.0853
```



## 29 dF4ienv

First derivative of the objective function for computing the inner envelope subspace.

### Contents

- Syntax
- Input
- Output
- Description

### Syntax

```
df = dF4ienv(R, DataParameter)
```

### Input

**R**: An  $r$  by  $u$  semi-orthogonal matrix,  $0 < u \leq p$ .

**DataParameter**: A structure that contains the statistics calculated from the data.

### Output

**df**: The first derivative of the objective function for computing the inner envelope subspace. An  $r$  by  $u$  matrix.

### Description

This first derivative of F4ienv obtained by matrix calculus calculations.

## 30 F4ienv

Objective function for computing the inner envelope subspace.

### Contents

- Syntax
- Input
- Output
- Description

### Syntax

```
f = F4ienv(R, DataParameter)
```

### Input

**R**: An  $r$  by  $u$  semi orthogonal matrix,  $0 < u \leq p$ .

**DataParameter**: A structure that contains the statistics calculated from the data.

### Output

**f**: A scalar containing the value of the objective function evaluated at **R**.

### Description

The objective function is derived in Section 3.3 in Su and Cook (2012) by using maximum likelihood estimation. The columns of the semi-orthogonal matrix that minimizes this function span the estimated inner envelope subspace.

## 31 ienv

Fit the inner envelope model.

### Contents

- Syntax
- Input
- Output
- Description
- References
- Example

### Syntax

```
ModelOutput = ienv(X, Y, u)
ModelOutput = ienv(X, Y, u, Opts)
```

### Input

**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, and  $r$  should be strictly greater than  $p$ .

**u:** Dimension of the inner envelope. An integer between 0 and  $p$  or equal to  $r$ .

**Opts:** A list containing the optional input parameter, to control the iterations in `sg_min`. If one or several (even all) fields are not defined, the default settings are used.

- `Opts.maxIter`: Maximum number of iterations. Default value: 300.
- `Opts.ftol`: Tolerance parameter for  $F$ . Default value:  $1e-10$ .
- `Opts.gradtol`: Tolerance parameter for  $dF$ . Default value:  $1e-7$ .
- `Opts.verbose`: Flag for print out Grassmann manifold optimization process, logical 0 or 1. Default value: 0.

### Output

**ModelOutput:** A list that contains the maximum likelihood estimators and some statistics.

- `ModelOutput.beta`: The envelope estimator of the regression coefficients  $\beta$ . An  $r$  by  $p$  matrix.
- `ModelOutput.Sigma`: The envelope estimator of the error covariance matrix. An  $r$  by  $r$  matrix.
- `ModelOutput.Gamma1`: The orthogonal basis of the inner envelope subspace. An  $r$  by  $u$  semi-orthogonal matrix.
- `ModelOutput.Gamma0`: The orthogonal basis of the complement of the inner envelope subspace. An  $r$  by  $r-u$  semi-orthogonal matrix.
- `ModelOutput.eta1`: The transpose of the coordinates of  $\beta$  with respect to `Gamma1`. An  $p$  by  $u$  matrix.
- `ModelOutput.B`: An  $(r - u)$  by  $(p - u)$  semi-orthogonal matrix, so that  $(\text{Gamma0}, \text{Gamma0} * B)$  spans  $\beta$ .
- `ModelOutput.eta2`: The transpose of the coordinates of  $\beta$  with respect to `Gamma0`. An  $p$  by  $(p-u)$  matrix.
- `ModelOutput.Omega1`: The coordinates of `Sigma` with respect to `Gamma1`. An  $u$  by  $u$  matrix.
- `ModelOutput.Omega0`: The coordinates of `Sigma` with respect to `Gamma0`. An  $r - u$  by  $r - u$  matrix.
- `ModelOutput.alpha`: The estimated intercept in the inner envelope model. An  $r$  by 1 vector.
- `ModelOutput.l`: The maximized log likelihood function. A real number.
- `ModelOutput.covMatrix`: The asymptotic covariance of  $\text{vec}(\beta)$ . An  $rp$  by  $rp$  matrix. The covariance matrix returned are asymptotic. For the actual standard errors, multiply by  $1/n$ .
- `ModelOutput.asyIenv`: Asymptotic standard error for elements in  $\beta$  under the inner envelope model. An  $r$  by  $p$  matrix. The standard errors returned are asymptotic, for actual standard errors, multiply by  $1 / \sqrt{n}$ .
- `ModelOutput.ratio`: The asymptotic standard error ratio of the standard multivariate linear regression estimator over the inner envelope estimator, for each element in  $\beta$ . An  $r$  by  $p$  matrix.
- `ModelOutput.np`: The number of parameters in the inner envelope model. A positive integer.
- `ModelOutput.n`: The number of observations in the data. A positive integer.

## Description

This function fits the inner envelope model to the responses and predictors, using the maximum likelihood estimation. When the dimension of the envelope is between 1 and  $p-1$ , we implemented the algorithm in Su and Cook (2012). When the dimension is  $p$ , then the inner 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.

## References

1. The codes is implemented based on the algorithm in Su and Cook (2012).
2. The Grassmann manifold optimization step calls the package `sg_min` 2.4.1 by Ross Lippert (<http://web.mit.edu/~ripper/www.sgmin.html>).

## Example

The following codes gives the results of the Fisher's iris data example in Su and Cook (2012).

```
load irisf.mat

d = bic_ienv(X, Y)

d =

1

ModelOutput = ienv(X, Y, d)
1 - 1 ./ ModelOutput.ratio

ModelOutput =

    beta: [4x2 double]
    Sigma: [4x4 double]
    Gamma1: [4x1 double]
    Gamma0: [4x3 double]
    B: [3x1 double]
    eta1: [2x1 double]
    eta2: [2x1 double]
    Omega1: 8.3751
    Omega0: [3x3 double]
    alpha: [4x1 double]
    np: 16
    l: -1.4805e+03
    covMatrix: [8x8 double]
    asyIenv: [4x2 double]
    ratio: [4x2 double]
    n: 150

ans =
```

0.0049	0.2122
0.0020	0.0806
0.0033	0.1404
0.0004	0.0138

## 32 lrt\_ienv

Select the dimension of the inner envelope subspace using likelihood ratio testing.

### Contents

- Syntax
- Input
- Output
- Description
- Example

### Syntax

```
u = lrt_ienv(X, Y, alpha)
u = lrt_ienv(X, Y, alpha, Opts)
```

### Input

**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. A real number between 0 and 1, often taken at 0.05 or 0.01.

**Opts**: A list containing the optional input parameter, to control the iterations in `sg_min`. If one or several (even all) fields are not defined, the default settings are used.

- `Opts.maxIter`: Maximum number of iterations. Default value: 300.
- `Opts.ftol`: Tolerance parameter for F. Default value:  $1e-10$ .
- `Opts.gradtol`: Tolerance parameter for dF. Default value:  $1e-7$ .
- `Opts.verbose`: Flag for print out dimension selection process, logical 0 or 1. Default value: 0.

### Output

**u**: Dimension of the inner envelope. An integer between 0 and  $p$  or equal to  $r$ .

## Description

This function implements the likelihood ratio testing procedure to select the dimension of the inner envelope subspace, with pre-specified significance level  $\alpha$ .

## Example

```
load irisf.mat

alpha = 0.01;
u = lrt_ienv(X, Y, alpha)
```

u =

1



## 33 predict\_ienv

Perform estimation or prediction under the inner envelope model.

### Contents

- Syntax
- Input
- Output
- Description
- Example

### Syntax

```
PredictOutput = predict_ienv(ModelOutput, Xnew, infType)
```

### Input

**ModelOutput:** A list containing the maximum likelihood estimators and other statistics inherited from ienv.

**Xnew:** The value of X with which to estimate or predict Y. A p by 1 vector.

**infType:** A string of characters indicting the inference type, the choices can be 'estimation' or 'prediction'.

### Output

**PredictOutput:** A list containing the results of the inference.

- PredictOutput.value: The fitted value or the prediction value evaluated at Xnew. An r by 1 vector.
- PredictOutput.covMatrix: The covariance matrix of PredictOutput.value. An r by r matrix.
- PredictOutput.SE: The standard error of elements in PredictOutput.value. An r by 1 vector.

### Description

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

## Example

```
load irisf.mat
d = bic_ienv(X, Y);
ModelOutput = ienv(X, Y, d);
Xnew = X(1, :)';
PredictOutput = predict_ienv(ModelOutput, Xnew, 'estimation')
[PredictOutput.value, Y(1, :)] % Compare the fitted value with the data
PredictOutput.SE
```

```
PredictOutput =
```

```
    value: [4x1 double]
 covMatrix: [4x4 double]
        SE: [4x1 double]
```

```
ans =
```

```
49.9458    51.0000
34.2592    35.0000
14.5771    14.0000
 2.4513     2.0000
```

```
ans =
```

```
1.0978
0.7146
0.9265
0.4357
```

```
PredictOutput = predict_ienv(ModelOutput, Xnew, 'prediction')
PredictOutput.SE
```

```
PredictOutput =
```

```
    value: [4x1 double]
 covMatrix: [4x4 double]
        SE: [4x1 double]
```

```
ans =
```

```
5.2197
```

3.3897  
4.3996  
2.0642

## 34 testcoefficient\_ienv

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

### Contents

- Syntax
- Input
- Output
- Description
- Example

### Syntax

```
TestOutput = testcoefficient_ienv(ModelOutput)
TestOutput = testcoefficient_ienv(ModelOutput, TestInput)
```

### Input

**ModelOutput:** A list containing the maximum likelihood estimators and other statistics inherited from ienv.

**TestInput:** A list that specifies the null hypothesis, including L, R, and A. If not provided by the user, default values will be used.

- TestInput.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. Default value: identity matrix  $I_r$ .
- TestInput.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. Default value: identity matrix  $I_p$ .
- TestInput.A: The matrix on the right hand side of the equation. It is a d1 by d2 matrix. Default value: d1 by d2 zero matrix.

### Output

**TestOutput:** A list containing test statistics, degrees of freedom for the reference chi-squared distribution, and the p-value. At the same time, a table is printed out.

- TestOutput.chisqStatistic: The test statistics. A real number.
- TestOutput.df: The degrees of freedom of the reference chi-squared distribution. A positive integer.
- TestOutput.pValue: p-value of the test. A real number in  $[0, 1]$ .

## Description

This function tests for hypothesis  $H_0 : L\beta R = A$ , versus  $H_\alpha : L\beta R \neq A$ . The  $\beta$  is estimated by the inner envelope model. If the user does not specify the values for L, R and A, then the test is equivalent to the standard F test on if  $\beta = 0$ . The test statistics used is  $\text{vec}(L\beta R - A) \hat{\Sigma}^{-1} \text{vec}(L\beta R - A)^T$ , and the reference distribution is chi-squared distribution with degrees of freedom  $d1 * d2$ .

## Example

```
load irisf.mat
d = bic_ienv(X,Y);
ModelOutput = ienv(X,Y,d);
TestOutout = testcoefficient_ienv(ModelOutput);
```

Test Hypothesis	Chisq Statistic	DF	P-value
L * beta * R = A	4642.913	8	0.0000

```
TestInput.L = rand(2, 4);
TestInput.R = rand(2, 1);
TestInput.A = zeros(2, 1);
TestOutout = testcoefficient_ienv(ModelOutput, TestInput);
```

Test Hypothesis	Chisq Statistic	DF	P-value
L * beta * R = A	1834.229	2	0.0000

## Part V

# penv

### 35 aic\_penv

Select the dimension of the partial envelope subspace using Akaike information criterion.

#### Contents

- Syntax
- Input
- Output
- Description
- Example

#### Syntax

```
u = aic_penv(X, Y)
u = aic_penv(X, Y, Opts)
```

#### Input

**X:** A list containing the value of X1 and X2.

- X.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.
- X.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.

**Opts:** A list containing the optional input parameter, to control the iterations in sg\_min. If one or several (even all) fields are not defined, the default settings are used.

- Opts.maxIter: Maximum number of iterations. Default value: 300.
- Opts.ftol: Tolerance parameter for F. Default value: 1e-10.
- Opts.gradtol: Tolerance parameter for dF. Default value: 1e-7.
- Opts.verbose: Flag for print out dimension selection process, logical 0 or 1. Default value: 0.

## Output

**u**: Dimension of the envelope. An integer between 0 and  $r$ .

## Description

This function implements the Akaike information criteria (AIC) to select the dimension of the partial envelope subspace.

## Example

```
load fiberpaper.dat
Y = fiberpaper(:, 1 : 4);
Xtemp = fiberpaper(:, 5 : 7);
X.X1 = Xtemp(:, 3);
X.X2 = Xtemp(:, 1 : 2);
u = aic_penv(X, Y)
```

u =

3

## 36 bic\_penv

Select the dimension of the partial envelope subspace using Bayesian information criterion.

### Contents

- Syntax
- Input
- Output
- Description
- Example

### Syntax

```
u = bic_penv(X, Y)
u = bic_penv(X, Y, Opts)
```

### Input

**X:** A list containing the value of X1 and X2.

- X.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.
- X.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.

**Opts:** A list containing the optional input parameter, to control the iterations in sg\_min. If one or several (even all) fields are not defined, the default settings are used.

- Opts.maxIter: Maximum number of iterations. Default value: 300.
- Opts.ftol: Tolerance parameter for F. Default value: 1e-10.
- Opts.gradtol: Tolerance parameter for dF. Default value: 1e-7.
- Opts.verbose: Flag for print out dimension selection process, logical 0 or 1. Default value: 0.

### Output

**u:** Dimension of the envelope. An integer between 0 and r.



## Description

This function implements the Bayesian information criteria (BIC) to select the dimension of the partial envelope subspace.

## Example

```
load fiberpaper.dat
Y = fiberpaper(:, 1 : 4);
Xtemp = fiberpaper(:, 5 : 7);
X.X1 = Xtemp(:, 3);
X.X2 = Xtemp(:, 1 : 2);
u = bic_penv(X, Y)
```

u =

1

## 37 bstrp\_penv

Compute bootstrap standard error for the partial envelope model.

### Contents

- Syntax
- Input
- Output
- Description
- Example

### Syntax

```
bootse = bstrp_penv(X, Y, u, B)
bootse = bstrp_penv(X, Y, u, B, Opts)
```

### Input

**X**: A list containing the value of X1 and X2.

- X.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.
- X.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 subspace. A positive integer between 0 and r.

**B**: Number of bootstrap samples. A positive integer.

**Opts**: A list containing the optional input parameter, to control the iterations in sg\_min. If one or several (even all) fields are not defined, the default settings are used.

- Opts.maxIter: Maximum number of iterations. Default value: 300.
- Opts.ftol: Tolerance parameter for F. Default value: 1e-10.
- Opts.gradtol: Tolerance parameter for dF. Default value: 1e-7.
- Opts.verbose: Flag for print out the number of bootstrap samples, logical 0 or 1. Default value: 0.

## Output

**bootse:** The standard error for elements in  $\beta_1$  computed by bootstrap. An  $r$  by  $p1$  matrix.

## Description

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

## Example

```
load fiberpaper.dat
Y = fiberpaper(:, 1 : 4);
Xtemp = fiberpaper(:, 5 : 7);
X.X1 = Xtemp(:, 3);
X.X2 = Xtemp(:, 1 : 2);
alpha = 0.01;
u = lrt_penv(X, Y, alpha)
```

u =

```
1
```

```
B = 100;
bootse = bstrp_penv(X, Y, u, B)
```

bootse =

```
0.0074
0.0021
0.0043
0.0019
```

## 38 lrt\_penv

Select the dimension of the partial envelope subspace using likelihood ratio testing.

### Contents

- Syntax
- Input
- Output
- Description
- Example

### Syntax

```
u = lrt_penv(X, Y, alpha)
u = lrt_penv(X, Y, alpha, Opts)
```

### Input

**X**: A list containing the value of X1 and X2.

- X.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.
- X.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. A real number between 0 and 1, often taken at 0.05 or 0.01.

**Opts**: A list containing the optional input parameter, to control the iterations in sg\_min. If one or several (even all) fields are not defined, the default settings are used.

- Opts.maxIter: Maximum number of iterations. Default value: 300.
- Opts.ftol: Tolerance parameter for F. Default value: 1e-10.
- Opts.gradtol: Tolerance parameter for dF. Default value: 1e-7.
- Opts.verbose: Flag for print out dimension selection process, logical 0 or 1. Default value: 0.

## Output

**u**: Dimension of the partial envelope subspace. An integer between 0 and  $r$ .

## Description

This function implements the likelihood ratio testing procedure to select the dimension of the partial envelope subspace, with pre-specified significance level  $\alpha$ .

## Example

```
load fiberpaper.dat
Y = fiberpaper(:, 1 : 4);
Xtemp = fiberpaper(:, 5 : 7);
X.X1 = Xtemp(:, 3);
X.X2 = Xtemp(:, 1 : 2);
alpha = 0.01;
u = lrt_penv(X, Y, alpha)
```

u =

1

## 39 penv

Fit the partial envelope model.

### Contents

- Syntax
- Input
- Output
- Description
- References
- Example

### Syntax

```
ModelOutput = penv(X, Y, u)
ModelOutput = penv(X, Y, u, Opts)
```

### Input

**X**: A list containing the value of X1 and X2.

- X.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.
- X.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, and r should be strictly greater than p1.

**u**: Dimension of the partial envelope. An integer between 0 and r.

**Opts**: A list containing the optional input parameter, to control the iterations in sg\_min. If one or several (even all) fields are not defined, the default settings are used.

- Opts.maxIter: Maximum number of iterations. Default value: 300.
- Opts.ftol: Tolerance parameter for F. Default value: 1e-10.
- Opts.gradtol: Tolerance parameter for dF. Default value: 1e-7.
- Opts.verbose: Flag for print out Grassmann manifold optimization process, logical 0 or 1. Default value: 0.

## Output

**ModelOutput:** A list that contains the maximum likelihood estimators and some statistics.

- `ModelOutput.beta1`: The partial envelope estimator of  $\beta_1$ , which is the regression coefficients for X1. An r by p1 matrix.
- `ModelOutput.beta2`: The partial envelope estimator of  $\beta_2$ , which is the regression coefficients for X2. An r by p2 matrix.
- `ModelOutput.Sigma`: The partial envelope estimator of the error covariance matrix. An r by r matrix.
- `ModelOutput.Gamma`: The orthogonal basis of the partial envelope subspace. An r by u semi-orthogonal matrix.
- `ModelOutput.Gamma0`: The orthogonal basis of the complement of the partial envelope subspace. An r by r - u semi-orthogonal matrix.
- `ModelOutput.eta`: The coordinates of  $\beta_1$  with respect to Gamma. An u by p1 matrix.
- `ModelOutput.Omega`: The coordinates of Sigma with respect to Gamma. An u by u matrix.
- `ModelOutput.Omega0`: The coordinates of Sigma with respect to Gamma0. An r - u by r - u matrix.
- `ModelOutput.alpha`: The estimated intercept in the partial envelope model. An r by 1 vector.
- `ModelOutput.l`: The maximized log likelihood function. A real number.
- `ModelOutput.covMatrix`: The asymptotic covariance of  $(\text{vec}(\beta_2)', \text{vec}(\beta_1)')'$ . An rp by rp matrix. The covariance matrix returned are asymptotic. For the actual standard errors, multiply by 1/n.
- `ModelOutput.asyPenv`: Asymptotic standard error for elements in  $\beta_1$  under the partial envelope model. An r by p1 matrix. The standard errors returned are asymptotic, for actual standard errors, multiply by 1/sqrt(n).
- `ModelOutput.ratio`: The asymptotic standard error ratio of the standard multivariate linear regression estimator over the partial envelope estimator, for each element in  $\beta_1$ . An r by p1 matrix.
- `ModelOutput.np`: The number of parameters in the envelope model. A positive integer.
- `ModelOutput.n`: The number of observations in the data. A positive integer.

## Description

This function fits the partial envelope model to the responses Y and predictors X1 and X2, 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). 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  $X_2$  as the predictors.

## References

1. The codes is implemented based on the algorithm in Section 3.2 of Su and Cook (2012).
2. The Grassmann manifold optimization step calls the package `sg_min` 2.4.1 by Ross Lippert (<http://web.mit.edu/~ripper/www.sgmin.html>).

## Example

The following codes reconstruct the results of the paper and fiber example in Su and Cook (2012).

```
load fiberpaper.dat
Y = fiberpaper(:, 1 : 4);
Xtemp = fiberpaper(:, 5 : 7);
X.X1 = Xtemp(:, 3);
X.X2 = Xtemp(:, 1 : 2);
alpha = 0.01;
u = lrt_penv(X, Y, alpha)

u =

1

ModelOutput = penv(X, Y, u)
ModelOutput.Omega
eig(ModelOutput.Omega0)
ModelOutput.ratio

ModelOutput =

    beta1: [4x1 double]
    beta2: [4x2 double]
    alpha: [4x1 double]
    Gamma: [4x1 double]
        eta: 0.0047
    Omega: 0.0149
    Omega0: [3x3 double]
    Sigma: [4x4 double]
         l: -35.6323
```



```
np: 23
covMatrix: [12x12 double]
asyPenv: [4x1 double]
ratio: [4x1 double]
n: 62
```

```
ans =
```

```
0.0149
```

```
ans =
```

```
4.9819
0.0999
0.0050
```

```
ans =
```

```
66.0742
6.9326
10.5048
9.6279
```

## 40 `predict_penv`

Perform estimation or prediction under the partial envelope model.

### Contents

- Syntax
- Input
- Output
- Description
- Example

### Syntax

```
PredictOutput = predict_penv(ModelOutput, Xnew, infType)
```

### Input

**ModelOutput:** A list containing the maximum likelihood estimators and other statistics inherited from `penv`.

**Xnew:** A list containing the value of X1 and X2 with which to estimate or predict Y.

- \* **Xnew.X1:** A p1 by 1 vector containing the value of X1.
- \* **Xnew.X2:** A p2 by 1 vector containing the value of X2.

**infType:** A string of characters indicting the inference type, the choices can be 'estimation' or 'prediction'.

### Output

**PredictOutput:** A list containing the results of the inference.

- **PredictOutput.value:** The fitted value or the prediction value evaluated at Xnew. An r by 1 vector.
- **PredictOutput.covMatrix:** The covariance matrix of PredictOutput.value. An r by r matrix.
- **PredictOutput.SE:** The standard error of elements in PredictOutput.value. An r by 1 vector.

## Description

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

## Example

```
load fiberpaper.dat
Y = fiberpaper(:, 1 : 4);
Xtemp = fiberpaper(:, 5 : 7);
X.X1 = Xtemp(:, 3);
X.X2 = Xtemp(:, 1 : 2);
alpha = 0.01;
u = lrt_penv(X, Y, alpha);
ModelOutput = penv(X, Y, u);
Xnew.X1 = X.X1(1, :)';
Xnew.X2 = X.X2(1, :)';
PredictOutput = predict_penv(ModelOutput, Xnew, 'estimation')
[PredictOutput.value, Y(1, :)] % Compare the fitted value with the data
PredictOutput.SE
```

PredictOutput =

```
value: [4x1 double]
covMatrix: [4x4 double]
SE: [4x1 double]
```

ans =

```
21.1169    21.3120
 7.1173     7.0390
 5.3637     5.3260
 0.8737     0.9320
```

ans =

```
1.4680
0.4234
0.7145
0.3161
```

```

PredictOutput = predict_penv(ModelOutput, Xnew, 'prediction')
PredictOutput.SE

PredictOutput =

      value: [4x1 double]
 covMatrix: [4x4 double]
        SE: [4x1 double]

ans =

2.4277
0.6982
1.1802
0.5220

```

## 41 testcoefficient\_penv

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

### Contents

- Syntax
- Input
- Output
- Description
- Example

### Syntax

```
TestOutput = testcoefficient_penv(ModelOutput)
TestOutput = testcoefficient_penv(ModelOutput, TestInput)
```

### Input

**ModelOutput:** A list containing the maximum likelihood estimators and other statistics inherited from `penv`.

**TestInput:** A list that specifies the null hypothesis, including  $L$ ,  $R$ , and  $A$ . If not provided by the user, default values will be used.

- **TestInput.L:** The matrix multiplied to  $\beta$  on the left. It is a  $d_1$  by  $r$  matrix, while  $d_1$  is less than or equal to  $r$ . Default value: identity matrix  $I_r$ .
- **TestInput.R:** The matrix multiplied to  $\beta$  on the right. It is a  $p_1$  by  $d_2$  matrix, while  $d_2$  is less than or equal to  $p_1$ . Default value: identity matrix  $I_{p_1}$ .
- **TestInput.A:** The matrix on the right hand side of the equation. It is a  $d_1$  by  $d_2$  matrix. Default value:  $d_1$  by  $d_2$  zero matrix.

### Output

**TestOutput:** A list containing test statistics, degrees of freedom for the reference chi-squared distribution, and the p-value. At the same time, a table is printed out.

- **TestOutput.chisqStatistic:** The test statistics. A real number.
- **TestOutput.df:** The degrees of freedom of the reference chi-squared distribution. A positive integer.
- **TestOutput.pValue:** p-value of the test. A real number in  $[0, 1]$ .

## Description

This function tests for hypothesis  $H_0 : L\beta R = A$ , versus  $H_\alpha : L\beta R \neq A$ . The  $\beta$  is estimated by the partial envelope model. If the user does not specify the values for L, R and A, then the test is equivalent to the standard F test on if  $\beta = 0$ . The test statistics used is  $\text{vec}(L\beta R - A) \hat{\Sigma}^{-1} \text{vec}(L\beta R - A)^T$ , and the reference distribution is chi-squared distribution with degrees of freedom  $d1 * d2$ .

## Example

```
load fiberpaper.dat
Y = fiberpaper(:, 1 : 4);
Xtemp = fiberpaper(:, 5 : 7);
X.X1 = Xtemp(:, 3);
X.X2 = Xtemp(:, 1 : 2);
alpha = 0.01;
u = lrt_penv(X, Y, alpha);
ModelOutput = penv(X, Y, u);
TestOutout = testcoefficient_penv(ModelOutput);
```

Test Hypothesis	Chisq Statistic	DF	P-value
L * beta * R = A	12.604	4	0.0134

```
r = size(Y, 2);
p1 = size(X.X1, 2);
TestInput.L = rand(2, r);
TestInput.R = rand(p1, 1);
TestInput.A = zeros(2, 1);
TestOutout = testcoefficient_penv(ModelOutput, TestInput);
```

Test Hypothesis	Chisq Statistic	DF	P-value
L * beta * R = A	11.452	2	0.0033

## Part VI

### seiv

#### 42 aic\_seiv

Select the dimension of the scaled envelope subspace using Akaike information criterion.

#### Contents

- Syntax
- Input
- Output
- Description
- Example

#### Syntax

```
u = aic_seiv(X, Y)
u = aic_seiv(X, Y, Opts)
```

#### Input

**X**: Predictors. An  $n$  by  $p$  matrix,  $p$  is the number of predictors and  $n$  is the number of observations. The predictors can be univariate or multivariate, discrete or continuous.

**Y**: Multivariate responses. An  $n$  by  $r$  matrix,  $r$  is the number of responses. The responses must be continuous variables.

**Opts**: A list containing the optional input parameter, to control the iterations in `sg_min`. If one or several (even all) fields are not defined, the default settings are used.

- `Opts.maxIter`: Maximum number of iterations. Default value: 300.
- `Opts.ftol`: Tolerance parameter for F. Default value:  $1e-10$ .
- `Opts.gradtol`: Tolerance parameter for dF. Default value:  $1e-7$ .
- `Opts.verbose`: Flag for print out dimension selection process, logical 0 or 1. Default value: 0.

#### Output

**u**: Dimension of the inner envelope. An integer between 0 and  $r$ .

## Description

This function implements the Akaike information criteria (AIC) to select the dimension of the scaled envelope subspace.

## Example

```
load('sales.txt')
Y = sales(:, 4 : 7);
X = sales(:, 1 : 3);
u = aic_senv(X, Y)
```

u =

4



## 43 `bic_senv`

Select the dimension of the scaled envelope subspace using Bayesian information criterion.

### Contents

- Syntax
- Input
- Output
- Description
- Example

### Syntax

```
u = bic_senv(X, Y)
u = bic_senv(X, Y, Opts)
```

### Input

**X**: Predictors. An  $n$  by  $p$  matrix,  $p$  is the number of predictors and  $n$  is the number of observations. The predictors can be univariate or multivariate, discrete or continuous.

**Y**: Multivariate responses. An  $n$  by  $r$  matrix,  $r$  is the number of responses. The responses must be continuous variables.

**Opts**: A list containing the optional input parameter, to control the iterations in `sg_min`. If one or several (even all) fields are not defined, the default settings are used.

- `Opts.maxIter`: Maximum number of iterations. Default value: 300.
- `Opts.ftol`: Tolerance parameter for F. Default value:  $1e-10$ .
- `Opts.gradtol`: Tolerance parameter for dF. Default value:  $1e-7$ .
- `Opts.verbose`: Flag for print out dimension selection process, logical 0 or 1. Default value: 0.

### Output

**u**: Dimension of the inner envelope. An integer between 0 and  $r$ .

### Description

This function implements the Bayesian information criteria (BIC) to select the dimension of the scaled envelope subspace.

### Example

```
load('sales.txt')
Y = sales(:, 4 : 7);
X = sales(:, 1 : 3);
u = bic_senv(X, Y)
```

u =

2

## 44 bstrp\_senv

Compute bootstrap standard error for the scaled envelope model.

### Contents

- Syntax
- Input
- Output
- Description
- Example

### Syntax

```
bootse = bstrp_senv(X, Y, u, B)
bootse = bstrp_senv(X, Y, u, B, Opts)
```

### Input

**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 subspace. A positive integer between 0 and  $r$ .

**B**: Number of bootstrap samples. A positive integer.

**Opts**: A list containing the optional input parameter, to control the iterations in `sg_min`. If one or several (even all) fields are not defined, the default settings are used.

- `Opts.maxIter`: Maximum number of iterations. Default value: 300.
- `Opts.ftol`: Tolerance parameter for F. Default value:  $1e-10$ .
- `Opts.gradtol`: Tolerance parameter for dF. Default value:  $1e-7$ .
- `Opts.verbose`: Flag for print out the number of bootstrap samples, logical 0 or 1. Default value: 0.

### Output

**bootse**: The standard error for elements in  $\beta$  computed by bootstrap. An  $r$  by  $p$  matrix.

## Description

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

## Example

```
load('sales.txt')
Y = sales(:, 4 : 7);
X = sales(:, 1 : 3);

u = bic_senv(X, Y)

u =

    2

B = 20;
bootse = bstrp_senv(X, Y, u, B)

bootse =

    0.0539    0.0472    0.0554
    0.0675    0.0912    0.1178
    0.0567    0.0781    0.0791
    0.0986    0.0944    0.1283
```

## 45 dF4senv

First derivative of the objective function for computing the envelope subspace in the scaled envelope model.

### Contents

- Syntax
- Input
- Output
- Description

### Syntax

```
df = dF4senv(R, DataParameter)
```

### Input

**R:** An  $r$  by  $u$  semi-orthogonal matrix,  $0 < u \leq r$ .

**DataParameter:** A structure that contains the statistics calculated from the data.

### Output

**df:** The first derivative of the objective function for computing the envelope subspace. An  $r$  by  $u$  matrix.

### Description

This first derivative of F4senv obtained by matrix calculus calculations.

## 46 F4senv

Objective function for computing the envelope subspace in scaled envelope model.

### Contents

- Syntax
- Input
- Output
- Description

### Syntax

```
f = F4senv(R, DataParameter)
```

### Input

**R**: An  $r$  by  $u$  semi orthogonal matrix,  $0 < u \leq r$ .

**DataParameter**: A structure that contains the statistics calculated from the data.

### Output

**f**: A scalar containing the value of the objective function evaluated at **R**.

### Description

The objective function is derived in Section 4.1 in Cook and Su (2012) using maximum likelihood estimation. The columns of the semi-orthogonal matrix that minimizes this function span the estimated envelope subspace.

## 47 objfun

Objective function for computing the scales in the scaled envelope model.

### Contents

- Syntax
- Input
- Output
- Description

### Syntax

```
f = objfun(d, Gamma, DataParameter)
```

### Input

**d:** An  $r - 1$  dimensional column vector containing the scales for the 2nd to the  $r$ th responses. All the entries in  $d$  are positive.

**Gamma:** A  $r$  by  $u$  semi-orthogonal matrix that spans the envelope subspace or the estimated envelope subspace.

**DataParameter:** A structure that contains the statistics calculated from the data.

### Output

**f:** A scalar containing the value of the objective function evaluated at  $d$ .

### Description

The objective function is derived in Section 4.1 of Su and Cook (2012) using maximum likelihood estimation.

## 48 `predict_senv`

Perform estimation or prediction under the scaled envelope model.

### Contents

- Syntax
- Input
- Output
- Description
- Example

### Syntax

```
PredictOutput = predict_senv(ModelOutput, Xnew, infType)
```

### Input

**ModelOutput:** A list containing the maximum likelihood estimators and other statistics inherited from `senv`.

**Xnew:** The value of  $X$  with which to estimate or predict  $Y$ . A  $p$  by 1 vector.

**infType:** A string of characters indicting the inference type, the choices can be 'estimation' or 'prediction'.

### Output

**PredictOutput:** A list containing the results of the inference.

- `PredictOutput.value`: The fitted value or the prediction value evaluated at  $X_{\text{new}}$ . An  $r$  by 1 vector.
- `PredictOutput.covMatrix`: The covariance matrix of `PredictOutput.value`. An  $r$  by  $r$  matrix.
- `PredictOutput.SE`: The standard error of elements in `PredictOutput.value`. An  $r$  by 1 vector.

### Description

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



## Example

```
load('sales.txt')
Y = sales(:, 4 : 7);
X = sales(:, 1 : 3);
u = bic_senv(X, Y);
ModelOutput = senv(X, Y, u);
Xnew = X(1, :)';
PredictOutput = predict_senv(ModelOutput, Xnew, 'estimation')
[PredictOutput.value, Y(1, :)] % Compare the fitted value with the data
PredictOutput.SE

PredictOutput =

    value: [4x1 double]
 covMatrix: [4x4 double]
        SE: [4x1 double]

ans =

    8.9109    9.0000
   11.5096   12.0000
    9.6063    9.0000
   19.5119   20.0000

ans =

    7.7627
    5.8952
    4.2205
    8.0134

PredictOutput = predict_senv(ModelOutput, Xnew, 'prediction')
PredictOutput.SE

PredictOutput =

    value: [4x1 double]
 covMatrix: [4x4 double]
        SE: [4x1 double]
```

ans =

8.3024  
6.3052  
4.4254  
8.5956

## 49 `senv`

Fit the scaled envelope model.

### Contents

- Syntax
- Input
- Output
- Description
- References
- Example

### Syntax

```
ModelOutput = senv(X, Y, u)  
ModelOutput = senv(X, Y, u, Opts)
```

### Input

**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, and  $r$  should be strictly greater than  $p$ .

**u:** Dimension of the envelope. An integer between 0 and  $r$ .

**Opts:** A list containing the optional input parameter, to control the iterations in `sg_min`. If one or several (even all) fields are not defined, the default settings are used.

- `Opts.maxIter`: Maximum number of iterations. Default value: 300.
- `Opts.ftol`: Tolerance parameter for F. Default value:  $1e-10$ .
- `Opts.gradtol`: Tolerance parameter for dF. Default value:  $1e-7$ .
- `Opts.verbose`: Flag for print out number of iterations, logical 0 or 1. Default value: 0.

### Output

**ModelOutput:** A list that contains the maximum likelihood estimators and some statistics.

- `ModelOutput.beta`: The scaled envelope estimator of the regression coefficients  $\beta$ . An  $r$  by  $p$  matrix.

- `ModelOutput.Sigma`: The scaled envelope estimator of the error covariance matrix. An  $r$  by  $r$  matrix.
- `ModelOutput.Lambda`: The matrix of estimated scales. An  $r$  by  $r$  diagonal matrix with the first diagonal element equal to 1 and other diagonal elements being positive.
- `ModelOutput.Gamma`: The orthogonal basis of the envelope subspace. An  $r$  by  $u$  semi-orthogonal matrix.
- `ModelOutput.Gamma0`: The orthogonal basis of the complement of the envelope subspace. An  $r$  by  $r - u$  semi-orthogonal matrix.
- `ModelOutput.eta`: The coordinates of  $\beta$  with respect to `Gamma`. An  $u$  by  $p$  matrix.
- `ModelOutput.Omega`: The coordinates of `Sigma` with respect to `Gamma`. An  $u$  by  $u$  matrix.
- `ModelOutput.Omega0`: The coordinates of `Sigma` with respect to `Gamma0`. An  $r - u$  by  $r - u$  matrix.
- `ModelOutput.alpha`: The estimated intercept in the scaled envelope model. An  $r$  by 1 vector.
- `ModelOutput.l`: The maximized log likelihood function. A real number.
- `ModelOutput.covMatrix`: The asymptotic covariance of  $\text{vec}(\beta)$ . An  $rp$  by  $rp$  matrix. The covariance matrix returned are asymptotic. For the actual standard errors, multiply by  $1 / n$ .
- `ModelOutput.asySenv`: Asymptotic standard error for elements in  $\beta$  under the scaled envelope model. An  $r$  by  $p$  matrix. The standard errors returned are asymptotic, for actual standard errors, multiply by  $1 / \sqrt{n}$ .
- `ModelOutput.ratio`: The asymptotic standard error ratio of the standard multivariate linear regression estimator over the scaled envelope estimator, for each element in  $\beta$ . An  $r$  by  $p$  matrix.
- `ModelOutput.np`: The number of parameters in the scaled envelope model. A positive integer.
- `ModelOutput.n`: The number of observations in the data. A positive integer.

## Description

This function fits the scaled envelope model to the responses and predictors, using the maximum likelihood estimation. When the dimension of the envelope is between 1 and  $r - 1$ , we implemented the algorithm in Cook and Su (2012). 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.

## References

1. The codes is implemented based on the algorithm in Section 4.1 of Cook and Su (2012).

2. The Grassmann manifold optimization step calls the package `sg_min` 2.4.1 by Ross Lippert (<http://web.mit.edu/~ripper/www.sgmin.html>).

## Example

The following codes produce the results of the test and performance example in Cook and Su (2012).

```
load('sales.txt')
Y = sales(:, 4 : 7);
X = sales(:, 1 : 3);
u = bic_senv(X, Y)

u =

    2

ModelOutput = senv(X, Y, u)
ModelOutput.Lambda
1 - 1 ./ ModelOutput.ratio

ModelOutput =

    beta: [4x3 double]
    Sigma: [4x4 double]
    Lambda: [4x4 double]
    Gamma: [4x2 double]
    Gamma0: [4x2 double]
    eta: [2x3 double]
    Omega: [2x2 double]
    Omega0: [2x2 double]
    alpha: [4x1 double]
    np: 23
    l: -386.1900
    covMatrix: [12x12 double]
    asySenv: [4x3 double]
    ratio: [4x3 double]
    n: 50

ans =

    1.0000         0         0         0
```

0	0.9729	0	0
0	0	0.8067	0
0	0	0	1.7016

ans =

0.7020	0.5218	0.6517
0.6137	0.3219	0.5614
0.4439	0.1858	0.3924
0.5375	0.4639	0.5228

## 50 testcoefficient\_senv

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

### Contents

- Syntax
- Input
- Output
- Description
- Example

### Syntax

```
TestOutput = testcoefficient_senv(ModelOutput)
TestOutput = testcoefficient_senv(ModelOutput, TestInput)
```

### Input

**ModelOutput:** A list containing the maximum likelihood estimators and other statistics inherited from `senv`.

**TestInput:** A list that specifies the null hypothesis, including  $L$ ,  $R$ , and  $A$ . If not provided by the user, default values will be used.

- TestInput.L: The matrix multiplied to  $\beta$  on the left. It is a  $d1$  by  $r$  matrix, while  $d1$  is less than or equal to  $r$ . Default value: identity matrix  $I_r$ .
- TestInput.R: The matrix multiplied to  $\beta$  on the right. It is a  $p$  by  $d2$  matrix, while  $d2$  is less than or equal to  $p$ . Default value: identity matrix  $I_p$ .
- TestInput.A: The matrix on the right hand side of the equation. It is a  $d1$  by  $d2$  matrix. Default value:  $d1$  by  $d2$  zero matrix.

### Output

**TestOutput:** A list containing test statistics, degrees of freedom for the reference chi-squared distribution, and the p-value. At the same time, a table is printed out.

- TestOutput.chisqStatistic: The test statistics. A real number.
- TestOutput.df: The degrees of freedom of the reference chi-squared distribution. A positive integer.
- TestOutput.pValue: p-value of the test. A real number in  $[0, 1]$ .

## Description

This function tests for hypothesis  $H_0 : L\beta R = A$ , versus  $H_\alpha : L\beta R \neq A$ . The  $\beta$  is estimated by the scaled envelope model. If the user does not specify the values for L, R and A, then the test is equivalent to the standard F test on if  $\beta = 0$ . The test statistics used is  $\text{vec}(L\beta R - A) \hat{\Sigma}^{-1} \text{vec}(L\beta R - A)^T$ , and the reference distribution is chi-squared distribution with degrees of freedom  $d1 * d2$ .

## Example

```
load('sales.txt')
Y = sales(:,4:7);
X = sales(:,1:3);
u = bic_senv(X,Y)
ModelOutput = senv(X,Y,u);
TestOutout = testcoefficient_senv(ModelOutput);
```

Test Hypothesis	Chisq Statistic	DF	P-value
L * beta * R = A	4827.816	12	0.0000

```
r = size(Y, 2);
p = size(X, 2);
TestInput.L = rand(2, r);
TestInput.R = rand(p, 1);
TestInput.A = zeros(2, 1);
TestOutout = testcoefficient_senv(ModelOutput, TestInput);
```

Test Hypothesis	Chisq Statistic	DF	P-value
L * beta * R = A	1025.948	2	0.0000



## Part VII

### xenv

#### aic\_xenv

Use Akaike information criterion to select the dimension of the envelope subspace for the reduction on X.

#### Contents

- Usage
- Description
- Example

#### Usage

```
u=aic_xenv(X,Y)
```

Input

- 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.

Output

- u: Dimension of the envelope. An integer between 0 and p.

#### Description

This function implements the Akaike information criteria (AIC) to select the dimension of the envelope subspace for the reduction on X.

#### Example

```
load wheatprotein.txt
X=wheatprotein(:,1:6);
Y=wheatprotein(:,7);
u=aic_xenv(X,Y)
```

## 51 bic\_xenv

Use Bayesian information criterion to select the dimension of the envelope subspace for the reduction on X.

### Contents

- Usage
- Description
- Example

### Usage

```
u=bic_xenv(X,Y)
```

Input

- 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.

Output

- u: Dimension of the envelope. An integer between 0 and p.

### Description

This function implements the Bayesian information criteria (BIC) to select the dimension of the envelope subspace for the reduction on X.

### Example

```
load wheatprotein.txt
X=wheatprotein(:,1:6);
Y=wheatprotein(:,7);
u=bic_xenv(X,Y)
```

## 52 bstrp\_xenv

Compute bootstrap standard error of the envelope model for the reduction on X.

### Contents

- Usage
- Description
- Example

### Usage

```
bootse=bstrp_xenv(X,Y,B,u)
```

#### Input

- 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.
- B: Number of bootstrap samples. A positive integer.
- u: Dimension of the envelope subspace. A positive integer between 0 and p.

#### Output

- bootse: The standard error for elements in  $\beta$  computed by bootstrap. An p by r matrix.

### Description

This function computes the bootstrap standard errors for the regression coefficients in the envelope model by bootstrapping the residuals. The envelope model here is for the reduction on X.

### Example

```
load wheatprotein.txt
X=wheatprotein(:,1:6);
Y=wheatprotein(:,7);
alpha=0.01;
u=lrt_xenv(Y,X,alpha)
```

```
B=100;  
bootse=bstrp_xenv(X,Y,B,u)
```

## 53 lrt\_xenv

Use likelihood ratio testing to select the dimension of the envelope subspace for the reduction on X.

### Contents

- Usage
- Description
- Example

### Usage

```
u=lrt_xenv(X,Y,alpha)
```

#### Input

- 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. A real number between 0 and 1, often taken at 0.05 or 0.01.

#### Output

- u: Dimension of the envelope. An integer between 0 and p.

### Description

This function implements the likelihood ratio testing procedure to select the dimension of the envelope subspace for the reduction on X, with pre-specified significance level  $\alpha$ .

### Example

```
load wheatprotein.txt
X=wheatprotein(:,1:6);
Y=wheatprotein(:,7);
u=lrt_xenv(X,Y,0.01)
```

## 54 xenv

Fit the envelope model for the reduction on X.

### Contents

- Usage
- Description
- References
- Example

### Usage

`stat=xenv(X,Y,u)`

#### Input

- 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, and r should be strictly greater than p.
- u: Dimension of the envelope. An integer between 0 and p.

#### Output

stat: A list that contains the maximum likelihood estimators and some statistics.

- stat.beta: The envelope estimator of the regression coefficients  $\beta$ . An p by r matrix.
- stat.SigX: The envelope estimator of the covariance matrix of X,  $\Sigma_X$ . A p by p matrix.
- stat.Gamma: The orthogonal basis of the envelope subspace. An p by u semi-orthogonal matrix.
- stat.Gamma0: The orthogonal basis of the complement of the envelope subspace. An p by p-u semi-orthogonal matrix.
- stat.eta: The coordinates of  $\beta$  with respect to Gamma. An u by r matrix.
- stat.Omega: The coordinates of  $\Sigma_X$  with respect to Gamma. An u by u matrix.
- stat.Omega0: The coordinates of  $\Sigma_X$  with respect to Gamma0. An p-u by p-u matrix.
- stat.mu: The estimated intercept. An r by 1 vector.
- stat.sigYcX: The estimated conditional covariance matrix of Y given X. An r by r matrix.
- stat.l: The maximized log likelihood function. A real number.

- `stat.asyEnv`: Asymptotic standard error for elements in  $\beta$  under the envelope model. An  $r$  by  $p$  matrix. The standard errors returned are asymptotic, for actual standard errors, multiply by  $1/\sqrt{n}$ .
- `stat.ratio`: The asymptotic standard error ratio of the standard multivariate linear regression estimator over the envelope estimator, for each element in  $\beta$ . An  $p$  by  $r$  matrix.
- `stat.np`: The number of parameters in the envelope model. A positive integer.

## Description

This function fits the envelope model to the responses and predictors, using the maximum likelihood estimation. When the dimension of the envelope is between 1 and  $r-1$ , we implemented the algorithm in Cook et al. (2010). 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.

## References

- The codes is implemented based on the algorithm in Section 4.5.1 of Cook et al (2012).
- The Grassmann manifold optimization step calls the package `sg_min` 2.4.1 by Ross Lippert (

<http://web.mit.edu/~ripper/www.sgmin.html>)

## Example

```
load wheatprotein.txt
X=wheatprotein(:,1:6);
Y=wheatprotein(:,7);
stat=xenv(X,Y,0);
```

```
p=size(X,2);
stat=xenv(X,Y,p);
```

```
% When u=p, the envelope model reduces to the ordinary least squares %
regression
```

```
temp=fit_OLS(X,Y);
temp.SigmaOLS
```

```
stat.sigYcX  
temp.betaOLS'  
stat.beta
```

```
stat=xenv(X,Y,5);
```

```
% To compare with the results obtained by Partial Least Squares, use the com-  
mand
```

```
[XL,YL,XS,YS,BETA,PCTVAR,MSE,stats] = plsregress(X,Y,5);  
stat.beta  
stat.mu  
BETA
```