

MATLAB Toolbox **envlp**: Reference Manual

March 23, 2015

Introduction

The toolbox “envlp” provides MATLAB functions that fit a variety of envelope models for multivariate analysis, especially multivariate linear regression. This document contains the complete description of this toolbox. It includes information on syntax and semantics, description and example for the functions in the toolbox. This manual is intended for users who need detailed information on the structure of the toolbox and its functions. It is also helpful to advanced users who want to write extensions for the toolbox.

The content of each module is as follows:

tools Functions for dimension selection and inference.

env Functions that implement the envelope model.

envseq Functions that implement the envelope model using sequential algorithm.

henv Functions that implement the heteroscedastic envelope model.

ienv Functions that implement the inner envelope model.

penv Functions that implement the partial envelope model.

senv Functions that implement the scaled envelope model.

xenv Functions that implement the envelope model in the predictor space.

xenvpls Functions that implement the envelope model in the predictor space using partial least squares algorithm.

envmean Functions that implement the envelope estimator of the multivariate mean.

auxiliary Auxiliary functions used internally in the toolbox.

Contents	2
1 tools	5
1.1 bootstrapse	5
1.2 bootstrapse_OLS	8
1.3 mfoldcv	10
1.4 modelselectaic	12
1.5 modelselectbic	14
1.6 modelselectlrt	16
1.7 prediction	18
1.8 testcoefficient	21
2 env	23
2.1 aic_env	23
2.2 bic_env	25
2.3 bstrp_env	27
2.4 dF4env	29
2.5 env	30
2.6 F4env	34
2.7 lrt_env	35
2.8 mfoldcv_env	37
2.9 predict_env	40
2.10 predict2_env	43
2.11 testcoefficient_env	46
3 envmean	48
3.1 aic_envmean	48
3.2 bic_envmean	50
3.3 bstrp_envmean	51
3.4 dF4envmean	53
3.5 envmean	54
3.6 F4envmean	57
3.7 lrt_envmean	58
3.8 mfoldcv_envmean	60
3.9 predict_envmean	63
3.10 testcoefficient_envmean	65
4 envseq	67
4.1 bstrp_envseq	67
4.2 envseq	69
4.3 mfoldcv_envseq	71
5 henv	74
5.1 aic_henv	74
5.2 bic_henv	76
5.3 bstrp_henv	78
5.4 dF4henv	80
5.5 F4henv	81
5.6 henv	82
5.7 lrt_henv	85
5.8 mfoldcv_henv	87
5.9 predict_henv	90

5.10	testcoefficient_henv	93
6	ienv	95
6.1	aic_ienv	95
6.2	bic_ienv	97
6.3	bstrp_ienv	99
6.4	dF4ienv	101
6.5	F4ienv	102
6.6	ienv	103
6.7	lrt_ienv	106
6.8	mfoldcv_ienv	108
6.9	predict_ienv	111
6.10	testcoefficient_ienv	113
7	penv	115
7.1	aic_penv	115
7.2	bic_penv	117
7.3	bstrp_penv	119
7.4	lrt_penv	121
7.5	mfoldcv_penv	123
7.6	penv	126
7.7	predict_penv	129
7.8	testcoefficient_penv	132
8	senv	134
8.1	aic_senv	134
8.2	bic_senv	136
8.3	bstrp_senv	138
8.4	dF4senv	140
8.5	F4senv	141
8.6	mfoldcv_senv	142
8.7	objfun	145
8.8	predict_senv	146
8.9	senv	148
8.10	testcoefficient_senv	152
9	sxenv	154
9.1	aic_sxenv	154
9.2	bic_sxenv	156
9.3	bstrp_spls	158
9.4	bstrp_sxenv	160
9.5	dF4sxenv	162
9.6	F4sxenv	163
9.7	mfoldcv_spls	164
9.8	mfoldcv_sxenv	166
9.9	objfun_spls	168
9.10	objfun_sxenv	169
9.11	predict_sxenv	170
9.12	spls	172
9.13	sxenv	175
9.14	testcoefficient_sxenv	178

10	xenv	180
10.1	aic_xenv	180
10.2	bic_xenv	182
10.3	bstrp_xenv	184
10.4	dF4xenv	186
10.5	F4xenv	187
10.6	lrt_xenv	188
10.7	mfoldcv_xenv	190
10.8	predict_xenv	193
10.9	testcoefficient_xenv	195
10.10	xenv	197
11	xenvpls	201
11.1	bstrp_xenvpls	201
11.2	mfoldcv_xenvpls	203
11.3	xenvpls	206
12	auxiliary	208
12.1	center	208
12.2	Contr	209
12.3	Expan	210
12.4	fit_OLS	211
12.5	get_envelope	213
12.6	get_Init	214
12.7	get_Init4envmean	215
12.8	get_Init4henv	216
12.9	Kpd	217
12.10	Lmatrix	218
12.11	make_dF	219
12.12	make_F	220
12.13	make_opts	221
12.14	make_parameter	222
12.15	mtest	224

1.1 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 methods 'env', 'envpls', 'henv', 'ienv', 'senv', 'spl', 'sxenv', 'xenv' and 'xenvpls'. 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 of characters indicating the model, choices can be 'env', 'envpls', 'henv', 'ienv', 'penv', 'senv', 'spl', 'sxenv', 'xenv' and 'xenvpls'.

Opts: A list containing the optional input parameters, 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 to print out the number of bootstrap samples, logical 0 or 1. Default value: 0.

Output

`bootse`: For `'env'`, `'envpls'`, `'henv'`, `'ienv'`, and `'senv'`, an r by p matrix containing the standard errors for elements in β computed by bootstrap. For `'penv'`, an r by p_1 matrix containing the standard errors for β_1 computed by bootstrap. For `'spls'`, `'sxenv'`, `'xenv'` and `'xenvpls'`, a p by r matrix containing the standard errors for elements in β 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 = modelselectlrt(X, Y, alpha, 'env');
B = 100;
modelType = 'env';
bootse = bootstrapse(X, Y, u, B, modelType)
```

```
bootse =
```

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

```
modelType = 'envpls';
bootse = bootstrapse(X, Y, u, B, modelType)
```

```
bootse =
```

```
9.3899  
7.7384  
8.3303  
9.0875  
13.4416  
5.1477
```

```
load fiberpaper.dat  
Y = fiberpaper(:, 1 : 4);  
X.X1 = fiberpaper(:, 7);  
X.X2 = fiberpaper(:, 5 : 6);  
alpha = 0.01;  
u = modelselectlrt(X, Y, alpha, 'penv');  
B = 100;  
modelType = 'penv';  
bootse = bootstrapse(X, Y, u, B, modelType)
```

```
bootse =
```

```
0.0027  
0.0012  
0.0020  
0.0009
```

1.2 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 parameters. If not defined, the default setting is used.

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

Output

bootse: The standard error for elements in β 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

1.3 mfoldcv

Select the dimension for the envelope family using m-fold cross validation.

Contents

- Syntax
- Input
- Output
- Description
- Example

Syntax

```
SelectOutput = mfoldcv(X, Y, m, modelType)
SelectOutput = mfoldcv(X, Y, m, modelType, Opts)
```

Input

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

For model type 'env', 'envseq', 'henv', 'ienv', 'senv', 'spl', 'sxenv', 'xenv' and 'xenvpls', 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.

m: A positive integer that is used to indicate m-fold cross validation.

modelType: A string of characters indicating the model, choices can be 'env', 'envseq', 'henv', 'ienv', 'penv', 'senv', 'spl', 'sxenv', 'xenv' or 'xenvpls'.

Opts: A list containing the optional input parameters. If one or several (even all) fields are not defined, the default settings are used.

- Opts.verbose: Flag to print out dimension selection process, logical 0 or 1. Default value: 0.
- Opts.table: Flag to tabulate the results, which contains cross validation error for each u. Logical 0 or 1. Default value: 0.
- Opts.perm: A positive integer indicating number permutations of the observations, m-fold cross validation is run on each permutation. If not specified, the division is based on the sequential order of the observations.
- Opts.seed: A real number that set the seeds for permutations. Default value is 1.

Output

SelectOutput: A list containing the results of the selection.

- **SelectOutput.u:** The dimension of the envelope subspace selected by m-fold cross validation. An integer between 0 and r.
- **SelectOutput.PreErr:** A vector containing prediction errors for each u if Opts.perm is not specified, or a matrix with the element in the ith row and jth column containing the prediction error for u=j-1 and ith permutation of the observations.

Description

This function implements m-fold cross validation to select the dimension of the envelope space, based on prediction performance. For each u, the data is partitioned into m parts, each part is in turn used for testing for the prediction performance while the rest m-1 parts are used for training. The dimension is selected as the one that minimizes the average prediction errors. If Y is multivariate, the identity inner product is used for computing the prediction errors.

Example

```
load wheatprotein.txt
X = wheatprotein(:, 8);
Y = wheatprotein(:, 1 : 6);
m = 5;
modelType = 'envpls';
SelectOutput = mfoldcv(X, Y, m, modelType)
```

SelectOutput =

```
    u: 0
PreErr: [88.3274 90.2842 90.4142 90.7036 90.7090 90.7103 90.2842]
```

1.4 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 'env', 'henv', 'ienv', 'senv', 'sxenv' 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 of characters indicating the model, choices can be 'env', 'henv', 'ienv', 'penv', 'senv', 'sxenv' and 'xenv'.

Opts: A list containing the optional input parameters, 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 to print out dimension selection process. Logical 0 or 1. Default value: 0.
- Opts.table: Flag to tabulate the results, which contains AIC and log likelihood for each u. 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', 'sxenv' 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);
X.X1 = fiberpaper(:, 7);
X.X2 = fiberpaper(:, 5 : 6);
modelType = 'penv';
u = modelselectaic(X, Y, modelType)
```

u =

3

1.5 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 'env', 'henv', 'ienv', 'senv', 'sxenv' 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 of characters indicating the model, choices can be 'env', 'henv', 'ienv', 'penv', 'senv', 'sxenv' and 'xenv'.

Opts: A list containing the optional input parameters, 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 to print out dimension selection process. Logical 0 or 1. Default value: 0.
- Opts.table: Flag to tabulate the results, which contains BIC and log likelihood for each u. 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', 'xenv' 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);
X.X1 = fiberpaper(:, 7);
X.X2 = fiberpaper(:, 5 : 6);
modelType = 'penv';
u = modelselectbic(X, Y, modelType)
```

u =

1

1.6 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 '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 of characters indicating the model, choices can be 'env', 'henv', 'ienv', 'penv' and 'xenv'.

Opts: A list containing the optional input parameters, 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 to print out dimension selection process. Logical 0 or 1. Default value: 0.
- Opts.table: Flag to tabulate the results, which contains log likelihood, test statistic, degrees of freedom and p-value for each test. 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);
X.X1 = fiberpaper(:, 7);
X.X2 = fiberpaper(:, 5 : 6);
alpha = 0.01;
modelType = 'penv';
u = modelselectlrt(X, Y, alpha, modelType)
```

u =

1

1.7 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', 'sxenv' 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 indicating the inference type, the choices can be 'estimation' or 'prediction'.

modelType: A string of characters indicating the model, choices can be 'env', 'henv', 'ienv', 'penv', 'senv', 'sxenv' 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', 'sxenv' 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 = prediction(ModelOutput, Xnew, 'estimation', modelType)
[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);
X.X1 = fiberpaper(:, 7);
X.X2 = fiberpaper(:, 5 : 6);
modelType = 'penv';
u = modelselectbic(X, Y, modelType);
ModelOutput = penv(X, Y, u);
Xnew.X1 = X.X1(1, :);
Xnew.X2 = X.X2(1, :);
PredictOutput = prediction(ModelOutput, Xnew, 'estimation', modelType)
PredictOutput.SE

```

PredictOutput =

```

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

```

```
ans =
```

```
1.4680
```

```
0.4234
```

```
0.7145
```

```
0.3161
```

1.8 testcoefficient

This function tests the null hypothesis $L * \beta * R = A$ versus the alternative hypothesis $L * \beta * R \neq A$, where β 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 of characters indicating the model, choices can be 'env', 'henv', 'ienv', 'penv', 'seenv', 'sxenv' 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 β 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 β 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, the p-value, and the covariance matrix of $\text{vec}(L\hat{\beta}R)$. 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].
- TestOutput.covMatrix: The covariance matrix of $\text{vec}(L\hat{\beta}R)$. A $d1 * d2$ by $d1 * d2$ matrix.

Description

This function tests for hypothesis $H_0 : L\beta R = A$, versus $H_\alpha : L\beta R \neq A$. The β 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', 'sxenv' and 'xenv'), or if the group main effects are all zeros (for 'henv'). The test statistics used is $\text{vec}(L\hat{\beta}R - A) \hat{\Sigma}^{-1} \text{vec}(L\hat{\beta}R - A)^T$, 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);
X.X1 = fiberpaper(:, 7);
X.X2 = fiberpaper(:, 5 : 6);
alpha = 0.01;
u = modelselectlrt(X, Y, alpha, 'penv');
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

2.1 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 parameters, 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 to print out dimension selection process. Logical 0 or 1. Default value: 0.
- `Opts.table`: Flag to tabulate the results, which contains AIC and log likelihood for each u . 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

2.2 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 parameters, 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 to print out dimension selection process. Logical 0 or 1. Default value: 0.
- Opts.table: Flag to tabulate the results, which contains BIC and log likelihood for each u. 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

2.3 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 parameters, 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 to print out the number of bootstrap samples, logical 0 or 1. Default value: 0.

Output

bootse: The standard error for elements in β 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 = modelselectlrt(X, Y, alpha, 'env')
```

```
u =
```

```
1
```

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

```
bootse =
```

```
0.2893
```

```
0.4260
```

```
0.3523
```

```
0.5628
```

```
0.1675
```

```
0.6192
```

2.4 **dF4env**

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

Contents

- Syntax
- Input
- Output
- Description

Syntax

$$\text{df} = \text{dF4env}(\mathbf{R}, \text{DataParameter})$$

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 \mathbf{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.

2.5 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.

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

Opts: A list containing the optional input parameters, 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 to print out Grassmann manifold optimization process, logical 0 or 1. Default value: 0.
- `Opts.init`: The initial value for the envelope subspace. An r by u matrix. Default value is the one generated by function `get_Init`.

Output

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

- `ModelOutput.beta`: The envelope estimator of the regression coefficients β . 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 β with respect to `Gamma`. A u by p matrix.
- `ModelOutput.Omega`: The coordinates of `Sigma` with respect to `Gamma`. A u by u matrix.

- `ModelOutput.Omega0`: The coordinates of Sigma with respect to Gamma0. An $r \times u$ by $r \times u$ 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.asySE`: The asymptotic standard error for elements in β 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 β . An r by p matrix.
- `ModelOutput.paramNum`: 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 are 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` 2.4.3 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 = modelselectlrt(X, Y, alpha, 'env');

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]
    asySE: [6x1 double]
    ratio: [6x1 double]
    paramNum: 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 =
```

```
28.0389
```


18.3983
23.5916
16.2928
65.7999
6.4555

2.6 F4env

Objective function for computing the envelope subspace.

Contents

- Syntax
- Input
- Output
- Description

Syntax

$$f = \text{F4env}(R, \text{DataParameter})$$

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.

2.7 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 parameters, 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 to print out dimension selection process. Logical 0 or 1. Default value: 0.
- `Opts.table`: Flag to tabulate the results, which contains log likelihood, test statistic, degrees of freedom and p-value for each test. 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 α .

Example

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

u =

1

2.8 mfoldcv_env

Select the dimension of the envelope subspace using m-fold cross validation.

Contents

- Syntax
- Input
- Output
- Description
- Example

Syntax

```
SelectOutput = mfoldcv_env(X, Y, m)
SelectOutput = mfoldcv_env(X, Y, m, Opts)
```

Input

X: Predictors. An n by p matrix, p is the number of predictors and n is number of observations.

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

m: A positive integer that is used to indicate m-fold cross validation.

Opts: A list containing the optional input parameters. If one or several (even all) fields are not defined, the default settings are used.

- Opts.verbose: Flag to print out dimension selection process, logical 0 or 1. Default value: 0.
- Opts.table: Flag to tabulate the results, which contains cross validation error for each u. Logical 0 or 1. Default value: 0.
- Opts.perm: A positive integer indicating number permutations of the observations, m-fold cross validation is run on each permutation. If not specified, the division is based on the sequential order of the observations.
- Opts.seed: A real number that set the seeds for permutations. Default value is 1.

Output

SelectOutput: A list containing the results of the selection.

- SelectOutput.u: The dimension of the envelope subspace selected by m-fold cross validation. An integer between 0 and r.
- SelectOutput.PreErr: A vector containing prediction errors for each u if Opts.perm is not specified, or a matrix with the element in the ith row and jth column containing the prediction error for u=j-1 and ith permutation of the observations.

Description

This function implements m-fold cross validation to select the dimension of the envelope space, based on prediction performance. For each u , the data is partitioned into m parts, each part is in turn used for testing for the prediction performance while the rest $m-1$ parts are used for training. The dimension is selected as the one that minimizes the average prediction errors. As Y is multivariate, the identity inner product is used for computing the prediction errors.

Example

```
load wheatprotein.txt
X = wheatprotein(:, 8);
Y = wheatprotein(:, 1 : 6);
Opts.table = 1; % Print out the table of average prediction error for each u
SelectOutput = mfoldcv_env(X, Y, 5, Opts);
```

u	CV error

0	88.327
1	88.222
2	88.229
3	88.343
4	90.176
5	89.449
6	90.284

SelectOutput.u

ans =

1

```
Opts.perm = 10; % Run 5-fold CV on 10 permutations
Opts.seed = 3; % Set seed for the permutations
Opts.table = 1;
SelectOutput = mfoldcv_env(X, Y, 5, Opts);
```

u	CV error

0	83.858
1	83.754
2	83.837
3	84.847
4	84.943

```

5      84.451
6      85.733
-----

```

The rows of PreErr corresponds to permutations, and the columns of PreErr corresponds to u.

```
SelectOutput.PreErr
```

```
ans =
```

```

83.8414  83.7358  83.7703  84.4435  84.9735  84.3271  86.0893
85.1301  85.0230  85.0517  87.3159  86.8758  85.7392  87.3486
82.8522  82.7447  83.1461  85.1392  84.6682  84.2123  85.6514
84.0505  83.9447  83.9708  83.9715  83.9716  85.0202  85.0807
83.2022  83.0987  83.1268  83.1323  83.7480  84.1540  84.1109
83.4689  83.3656  83.4763  87.8617  87.8145  84.4186  90.1180
84.9664  84.8683  84.9066  84.9103  85.5000  84.9104  85.5247
85.2560  85.1521  85.2284  85.2295  86.0263  86.0263  86.7444
82.6187  82.5142  82.5512  83.2121  82.7440  82.5517  83.4039
83.1972  83.0933  83.1446  83.2518  83.1093  83.1452  83.2577

```

```
mean(SelectOutput.PreErr) % Compute the average of prediction errors for each u
```

```
ans =
```

```

83.8583  83.7541  83.8373  84.8468  84.9431  84.4505  85.7330

```

```
std(SelectOutput.PreErr) % Compute the standard deviations of the prediction errors for each u
```

```
ans =
```

```

0.9663  0.9670  0.9323  1.6501  1.6368  1.0620  2.0453

```

2.9 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 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 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
```

2.10 predict2_env

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

Contents

- Syntax
- Input
- Output
- Description
- Example

Syntax

```
PredictOutput = predict2_env(X, Y, Xnew, infType)
```

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.

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

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 X_{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 envelope model at new value X_{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. Compared to `predict_env`, this function performs prediction through partial envelope model, which can be more accurate if the partial envelope is of smaller dimension and contains less variant material information.

References

1. The codes are implemented based on the following reference: R.D. Cook (2013) "Lecture Notes on Envelope Models and Methods." School of Statistics, University of Minnesota, Minneapolis.

Example

```

load fiberpaper.dat
Y = fiberpaper(:, 1 : 4);
X = fiberpaper(:, [7 5 6]);
alpha = 0.01;
u = lrt_env(X, Y, alpha);
ModelOutput = env(X, Y, u);
Xnew = X(10, :);
p1 = predict_env(ModelOutput, Xnew, 'estimation')

```

p1 =

```

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

```

p1.value

ans =

```

20.1120
6.9174
4.7911
0.6162

```

p1.SE

ans =

```

1.7555
0.4533
0.8248
0.3846

```

```

p2 = predict2_env(X, Y, Xnew, 'estimation')

```

p2 =

```

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

```

p2.value

```
ans =
```

```
20.1595  
6.9105  
4.8021  
0.6182
```

```
p2.SE
```

```
ans =
```

```
0.2591  
0.1001  
0.1459  
0.0741
```

```
p1.SE./p2.SE
```

```
ans =
```

```
6.7766  
4.5264  
5.6532  
5.1914
```

2.11 testcoefficient_env

This function tests the null hypothesis $L * \beta * R = A$ versus the alternative hypothesis $L * \beta * R \neq A$, where β 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 β 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 β 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, the p-value, and the covariance matrix of $\text{vec}(L\hat{\beta}R)$. 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 $[0, 1]$.
- TestOutput.covMatrix: The covariance matrix of $\text{vec}(L\hat{\beta}R)$. A d1 * d2 by d1 * d2 matrix.

Description

This function tests for hypothesis $H_0 : L\beta R = A$, versus $H_\alpha : L\beta R \neq A$. The β 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 $\text{vec}(L\hat{\beta}R - A) \hat{\Sigma}^{-1} \text{vec}(L\hat{\beta}R - A)^T$, 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_env(ModelOutput, TestInput);
```

Test Hypothesis	Chisq Statistic	DF	P-value
L * beta * R = A	111.628	2	0.0000

envmean

3.1 aic_envmean

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

Contents

- Syntax
- Input
- Output
- Description
- Example

Syntax

```
u = aic_envmean(Y)
u = aic_envmean(Y, Opts)
```

Input

Y: Data matrix. An n by p matrix, p is the dimension of the variable and n is number of observations.

Opts: A list containing the optional input parameters, 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 to print out dimension selection process. Logical 0 or 1. Default value: 0.
- `Opts.table`: Flag to tabulate the results, which contains AIC and log likelihood for each u . Logical 0 or 1. Default value: 0.

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.

Example

```
load Adopted
Y = Adopted(:, 1 : 6);
u = aic_envmean(Y)
```

u =

3

3.2 bic_envmean

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

Contents

- Syntax
- Input
- Output
- Description
- Example

Syntax

```
u = bic_envmean(Y)
u = bic_envmean(Y, Opts)
```

Input

Y: Data matrix. An n by p matrix, p is the dimension of the variable and n is number of observations.

Opts: A list containing the optional input parameters, 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 to print out dimension selection process. Logical 0 or 1. Default value: 0.
- `Opts.table`: Flag to tabulate the results, which contains BIC and log likelihood for each u . Logical 0 or 1. Default value: 0.

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.

Example

```
load Adopted
Y = Adopted(:, 1 : 6);
u = bic_envmean(Y)
```

u =

3

3.3 bstrp_envmean

Compute bootstrap standard error for the envelope estimator of the multivariate mean.

Contents

- Syntax
- Input
- Output
- Description
- Example

Syntax

```
bootse = bstrp_envmean(Y, u, B)
bootse = bstrp_envmean(Y, u, B, Opts)
```

Input

Y: Data matrix. An n by p matrix, p is the dimension of the variable and n is number of observations.

u: Dimension of the envelope subspace. A positive integer between 0 and p .

B: Number of bootstrap samples. A positive integer.

Opts: A list containing the optional input parameters, 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 to print out the number of bootstrap samples, logical 0 or 1. Default value: 0.

Output

bootse: The standard error for elements in μ computed by bootstrap. A p dimensional column vector.

Description

This function computes the bootstrap standard errors for the envelope estimator of the multivariate mean by bootstrapping the residuals.

Example

```
load Adopted
Y = Adopted(:, 1 : 6);
u = bic_envmean(Y)
```

```
u =
```

```
3
```

```
B = 100;  
bootse = bstrp_envmean(Y, u, B)
```

```
bootse =
```

```
0.5155  
13.4169  
13.2268  
17.0221  
21.4016  
22.0329
```

3.4 dF4envmean

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

Contents

- Syntax
- Input
- Output
- Description

Syntax

```
df = dF4envmean(R, DataParameter)
```

Input

R: A p by u semi orthogonal matrix, $0 < u \leq p$.

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

Output

df: A p by u matrix containing the value of the derivative function evaluated at R .

Description

The objective function is derived by maximum likelihood estimation. This function is the derivative of the objective function.

3.5 envmean

Provide envelope estimator for the multivariate mean.

Contents

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

Syntax

```
ModelOutput = envmean(Y, u)
ModelOutput = envmean(Y, u, Opts)
```

Input

Y: Data matrix. An n by p matrix, p is the dimension of the variable and n is number of observations.

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

Opts: A list containing the optional input parameters, 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 to print out Grassmann manifold optimization process, logical 0 or 1. Default value: 0.
- `Opts.init`: The initial value for the envelope subspace. A p by u matrix. Default value is the one generated by function `get_Init4envmean`.

Output

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

- `ModelOutput.mu`: The envelope estimator of the multivariate mean μ . A p dimensional column vector.
- `ModelOutput.Sigma`: The envelope estimator of the error covariance matrix. A p by p matrix.
- `ModelOutput.Gamma`: The orthogonal basis of the envelope subspace. A p by u semi-orthogonal matrix.
- `ModelOutput.Gamma0`: The orthogonal basis of the complement of the envelope subspace. A p by $p-u$ semi-orthogonal matrix.
- `ModelOutput.eta`: The coordinates of μ with respect to `Gamma`. A u dimensional column vector.
- `ModelOutput.Omega`: The coordinates of `Sigma` with respect to `Gamma`. A u by u matrix.
- `ModelOutput.Omega0`: The coordinates of `Sigma` with respect to `Gamma0`. A $p-u$ by $p-u$ matrix.

- **ModelOutput.l:** The maximized log likelihood function. A real number.
- **ModelOutput.covMatrix:** The asymptotic covariance of μ . A p by p matrix. The covariance matrix returned are asymptotic. For the actual standard errors, multiply by $1/n$.
- **ModelOutput.asySE:** The asymptotic standard error for elements in μ under the envelope model. A p dimensional column vector. 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 μ . A p dimensional column vector.
- **ModelOutput.paramNum:** 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 provides an envelope estimator for the multivariate mean, with a given dimension of the envelope subspace u . The estimator is obtained using the maximum likelihood estimation. When the dimension is p , then the envelope model degenerates to the standard sample mean. When the dimension is 0, it means that Y has mean 0.

References

The Grassmann manifold optimization step calls the package `sg_min` 2.4.3 by Ross Lippert (<http://web.mit.edu/~ripper/www.sgmin.html>).

Example

```
load Adopted
Y = Adopted(:, 1 : 6);
u = bic_envmean(Y)

u =

    3

ModelOutput = envmean(Y, u)

ModelOutput =

    mu: [6x1 double]
    Sigma: [6x6 double]
    Gamma: [6x3 double]
    Gamma0: [6x3 double]
    eta: [3x1 double]
    Omega: [3x3 double]
    Omega0: [3x3 double]
    l: -1.3492e+03
    covMatrix: [6x6 double]
```

```

asySE: [6x1 double]
ratio: [6x1 double]
paramNum: 24
n: 62

```

ModelOutput.mu

```
ans =
```

```

12.3258
85.9841
115.5767
112.1291
114.4862
106.4240

```

ModelOutput.Sigma

```
ans =
```

```
Columns 1 through 5
```

```

8.3278    2.9150   -4.0008    0.1057    0.3731
2.9150  235.6587    5.8146   42.1613   59.7492
-4.0008    5.8146  179.2066   91.7228   92.9563
0.1057   42.1613   91.7228  167.1073  114.4203
0.3731   59.7492   92.9563  114.4203  184.3248
-0.8157   71.9394   72.1815  110.2918  161.8752

```

```
Column 6
```

```

-0.8157
71.9394
72.1815
110.2918
161.8752
233.9185

```


3.6 F4envmean

Objective function for computing the envelope subspace.

Contents

- Syntax
- Input
- Output
- Description

Syntax

```
f = F4envmean(R, DataParameter)
```

Input

R: A p 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 by maximum likelihood estimation. The columns of the semi-orthogonal matrix that minimizes this function span the estimated envelope subspace.

3.7 lrt_envmean

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

Contents

- Syntax
- Input
- Output
- Description
- Example

Syntax

```
u = lrt_envmean(Y, alpha)
u = lrt_envmean(Y, alpha, Opts)
```

Input

Y: Data matrix. An n by p matrix, p is the dimension of the variable and n is number of observations.

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 parameters, 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 to print out dimension selection process. Logical 0 or 1. Default value: 0.
- `Opts.table`: Flag to tabulate the results, which contains log likelihood, test statistic, degrees of freedom and p-value for each test. Logical 0 or 1. Default value: 0.

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, with pre-specified significance level α .

Example

```
load Adopted
Y = Adopted(:, 1 : 6);
alpha = 0.01;
u = lrt_envmean(Y, alpha)
```

u =

2

3.8 mfoldcv_envmean

Select the dimension of the envelope subspace using m-fold cross validation.

Contents

- Syntax
- Input
- Output
- Description
- Example

Syntax

```
SelectOutput = mfoldcv_envmean(Y, m)
SelectOutput = mfoldcv_envmean(Y, m, Opts)
```

Input

Y: Data matrix. An n by p matrix, p is the dimension of the variable and n is number of observations.

m: A positive integer that is used to indicate m-fold cross validation.

Opts: A list containing the optional input parameters. If one or several (even all) fields are not defined, the default settings are used.

- Opts.verbose: Flag to print out dimension selection process, logical 0 or 1. Default value: 0.
- Opts.table: Flag to tabulate the results, which contains cross validation error for each u. Logical 0 or 1. Default value: 0.
- Opts.perm: A positive integer indicating number permutations of the observations, m-fold cross validation is run on each permutation. If not specified, the division is based on the sequential order of the observations.
- Opts.seed: A real number that set the seeds for permutations. Default value is 1.

Output

SelectOutput: A list containing the results of the selection.

- SelectOutput.u: The dimension of the envelope subspace selected by m-fold cross validation. An integer between 0 and r.
- SelectOutput.PreErr: A vector containing prediction errors for each u if Opts.perm is not specified, or a matrix with the element in the ith row and jth column containing the prediction error for u=j-1 and ith permutation of the observations.

Description

This function implements m-fold cross validation to select the dimension of the envelope space, based on prediction performance. For each u, the data is partitioned into m parts, each part is in turn used for testing for the prediction performance while the rest m-1 parts are used for training. The dimension is selected as the one that minimizes the average prediction errors. As Y is multivariate, the identity inner product is used for computing the prediction errors.

Example

```
load Adopted
Y = Adopted(:, 1 : 6);
Opts.table = 1; % Print out the table of average prediction error for each u
SelectOutput = mfoldcv_envmean(Y, 5, Opts);
```

u	CV error
0	68.402
1	3.321
2	31.070
3	30.267
4	3.324
5	3.324
6	3.323

SelectOutput.u

ans =

1

```
Opts.perm = 10; % Run 5-fold CV on 10 permutations
Opts.seed = 3; % Set seed for the permutations
Opts.table = 1;
SelectOutput = mfoldcv_envmean(Y, 5, Opts);
```

u	CV error
0	68.381
1	23.071
2	11.282
3	5.411
4	2.838
5	2.838
6	2.839

The rows of PreErr corresponds to permutations, and the columns of PreErr corresponds to u.

SelectOutput.PreErr

```
ans =
```

```

68.4179 30.0985 31.6848 3.2555 3.2601 3.2623 3.2629
68.3949 3.0342 3.0239 3.0259 3.0180 3.0145 3.0191
68.4065 2.6861 2.6749 2.6867 2.6826 2.6790 2.6823
68.3404 2.4518 2.4496 2.4490 2.4441 2.4512 2.4517
68.3574 31.4960 3.9833 3.9791 3.9775 3.9760 3.9768
68.3996 30.1065 2.5022 2.5058 2.4999 2.5039 2.5026
68.3919 41.4619 2.0079 27.7202 2.0073 2.0114 2.0116
68.3628 43.1511 29.2499 2.1869 2.1839 2.1772 2.1792
68.3371 3.5819 32.5092 3.5791 3.5785 3.5790 3.5792
68.4042 42.6414 2.7339 2.7257 2.7235 2.7233 2.7229

```

```
mean(SelectOutput.PreErr) % Compute the average of prediction errors for each u
```

```
ans =
```

```

68.3813 23.0710 11.2819 5.4114 2.8375 2.8378 2.8388

```

```
std(SelectOutput.PreErr) % Compute the standard deviations of the prediction errors for each u
```

```
ans =
```

```

0.0292 18.0009 13.7414 7.8576 0.6220 0.6214 0.6214

```

3.9 predict_envmean

Perform estimation of the multivariate mean or prediction for a new observation.

Contents

- Syntax
- Input
- Output
- Description
- Example

Syntax

```
PredictOutput = predict_envmean(ModelOutput, infType)
```

Input

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

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 estimated multivariate mean or the prediction value. A p dimensional column vector.
- PredictOutput.covMatrix: The covariance matrix of PredictOutput.value. A p by p matrix.
- PredictOutput.SE: The standard error of elements in PredictOutput.value. A p dimensional column vector.

Description

If the inference type is prediction, this function predicts a new observation and gives its covariance matrix and standard errors of its elements. If the inference type is estimation, this function gives the estimation of the multivariate mean, its covariance matrix and standard errors of its elements.

Example

```
load Adopted
Y = Adopted(:, 1 : 6);
u = bic_envmean(Y);
ModelOutput = envmean(Y, u);
PredictOutput = predict_envmean(ModelOutput, 'prediction')
```

```
PredictOutput =
```

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

```
PredictOutput.value
```

```
ans =
```

```
12.3258  
85.9841  
115.5767  
112.1291  
114.4862  
106.4240
```

```
PredictOutput.SE
```

```
ans =
```

```
2.9090  
15.4745  
13.4943  
13.0308  
13.6857  
15.4172
```


3.10 testcoefficient_envmean

This function tests the null hypothesis $L * \mu = A$ versus the alternative hypothesis $L * \mu \neq A$, where μ is the multivariate mean.

Contents

- Syntax
- Input
- Output
- Description
- Example

Syntax

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

Input

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

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

- TestInput.L: The matrix multiplied to μ on the left. It is a d1 by p matrix, while d1 is less than or equal to p. Default value: identity matrix I_p .
- TestInput.A: The vector on the right hand side of the equation. It is a d1 dimensional column vector. Default value: d1 by d2 zero matrix.

Output

TestOutput: A list containing test statistics, degrees of freedom for the reference chi-squared distribution, the p-value, and the covariance matrix of $L\mu$. 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 $[0, 1]$.
- TestOutput.covMatrix: The covariance matrix of $L\mu$. A d1 dimensional column vector.

Description

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

Example

```

load Adopted
Y = Adopted(:, 1 : 6);
u = bic_envmean(Y);
ModelOutput = envmean(Y, u);
TestOutout = testcoefficient_envmean(ModelOutput);

```

Test Hypothesis	Chisq Statistic	DF	P-value
L * mu = A	8716.691	6	0.0000

```

p = size(Y, 2);
TestInput.L = rand(2, p);
TestInput.A = zeros(2, 1);
TestOutout = testcoefficient_envmean(ModelOutput, TestInput);

```

Test Hypothesis	Chisq Statistic	DF	P-value
L * mu = A	6543.508	2	0.0000

envseq

4.1 bstrp_envseq

Compute bootstrap standard errors of the envelope model using a sequential algorithm.

Contents

- Syntax
- Input
- Output
- Description
- Example

Syntax

```
bootse = bstrp_envseq(X, Y, u, B)
bootse = bstrp_envseq(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 p .

B: Number of bootstrap samples. A positive integer.

Opts: A list containing the optional input parameters. If one or several (even all) fields are not defined, the default settings are used.

- Opts.verbose: Flag to print out dimension selection process, logical 0 or 1. Default value: 0.

Output

bootse: The standard error for elements in β 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. The envelope model is applied for the reduction on X , using a sequential algorithm.

Example

```
load Rohwer
X = Rohwer(:, 4 : 5);
Y = Rohwer(:, 1 : 3);
m = 5;
u = mfoldcv_envseq(X, Y, m)

u =

    1

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

bootse =

    0.8738    0.6855
    0.5191    0.4404
    0.0961    0.0654
```

4.2 envseq

Fit the envelope model using a sequential algorithm.

Contents

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

Syntax

$$\text{ModelOutput} = \text{envseq}(X, Y, u)$$

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.

u: Dimension of the envelope. An integer between 0 and $(m-1) * n / m-1$.

Output

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

- **ModelOutput.beta:** The envelope estimator of the regression coefficients β . 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 β with respect to Gamma. A u by p matrix.
- **ModelOutput.Omega:** The coordinates of Sigma with respect to Gamma. A 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.paramNum:** 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

The codes are implemented based on the sequential algorithm in the lecture notes of Cook (2012).

Example

```
load Rohwer
X = Rohwer(:, 4 : 5);
Y = Rohwer(:, 1 : 3);
m = 5;
u = mfoldcv_envseq(X, Y, m)

u =

    1

ModelOutput = envseq(X, Y, u)

ModelOutput =

    beta: [3x2 double]
    Sigma: [3x3 double]
    Gamma: [3x1 double]
    Gamma0: [3x2 double]
    eta: [2.5708 1.1966]
    Omega: 752.8146
    Omega0: [2x2 double]
    alpha: [3x1 double]
    paramNum: 11
    n: 69

ModelOutput.Sigma

ans =

    587.2575    229.9647    37.2716
    229.9647    421.1372    42.9256
    37.2716    42.9256    12.2696
```

4.3 mfoldcv_envseq

Select the dimension of the envelope subspace using m-fold cross validation for envelope model using a sequential algorithm.

Contents

- Syntax
- Input
- Output
- Description
- Example

Syntax

```
SelectOutput = mfoldcv_envseq(X, Y, m)
SelectOutput = mfoldcv_envseq(X, Y, m, Opts)
```

Input

X: Predictors. An n by p matrix, p is the number of predictors and n is number of observations.

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

m: A positive integer that is used to indicate m-fold cross validation.

Opts: A list containing the optional input parameters. If one or several (even all) fields are not defined, the default settings are used.

- Opts.verbose: Flag to print out dimension selection process, logical 0 or 1. Default value: 0.
- Opts.table: Flag to tabulate the results, which contains cross validation error for each u. Logical 0 or 1. Default value: 0.
- Opts.perm: A positive integer indicating number permutations of the observations, m-fold cross validation is run on each permutation. If not specified, the division is based on the sequential order of the observations.
- Opts.seed: A real number that set the seeds for permutations. Default value is 1.

Output

SelectOutput: A list containing the results of the selection.

- SelectOutput.u: The dimension of the envelope subspace selected by m-fold cross validation. An integer between 0 and r.
- SelectOutput.PreErr: A vector containing prediction errors for each u if Opts.perm is not specified, or a matrix with the element in the ith row and jth column containing the prediction error for u=j-1 and ith permutation of the observations.

Description

This function implements m-fold cross validation to select the dimension of the envelope space, based on prediction performance. For each u , the data is partitioned into m parts, each part is in turn used for testing for the prediction performance while the rest $m-1$ parts are used for training. The dimension is selected as the one that minimizes the average prediction errors. As Y is multivariate, the identity inner product is used for computing the prediction errors.

Example

```
load Rohwer
X = Rohwer(:, 4 : 5);
Y = Rohwer(:, 1 : 3);
Opts.table = 1; % Print out the table of average prediction error for each u
SelectOutput = mfoldcv_envseq(X, Y, 5, Opts);
```

u	CV error
0	35.575
1	33.713
2	34.352
3	33.823

SelectOutput.u

ans =

1

```
Opts.perm = 10; % Run 5-fold CV on 10 permutations
Opts.seed = 3; % Set seed for the permutations
Opts.table = 1;
SelectOutput = mfoldcv_envseq(X, Y, 5, Opts);
```

u	CV error
0	34.721
1	33.190
2	33.764
3	33.314

The rows of PreErr corresponds to permutations, and the columns of PreErr corresponds to u .

SelectOutput.PreErr

ans =

34.3525	32.8549	33.3868	32.9159
34.8898	33.2683	34.4169	33.3090
34.5545	33.5600	34.1334	33.7670
34.8090	33.2963	33.9307	33.3307
34.6288	33.3523	33.6383	33.4140
34.8314	32.9893	33.5988	33.0860
34.8499	33.0738	33.5392	33.2285
34.9425	33.4004	33.7077	33.4434
34.6249	32.9303	33.4134	33.3155
34.7250	33.1717	33.8734	33.3336

mean>SelectOutput.PreErr) % Compute the average of prediction errors for each u

ans =

34.7208	33.1897	33.7638	33.3144
---------	---------	---------	---------

std>SelectOutput.PreErr) % Compute the standard deviations of the prediction errors for each u

ans =

0.1812	0.2258	0.3274	0.2237
--------	--------	--------	--------

henv

5.1 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.

Opts: A list containing the optional input parameters, 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 to print out dimension selection process. Logical 0 or 1. Default value: 0.
- `Opts.table`: Flag to tabulate the results, which contains AIC and log likelihood for each u . 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

5.2 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.

Opts: A list containing the optional input parameters, 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 to print out dimension selection process. Logical 0 or 1. Default value: 0.
- `Opts.table`: Flag to tabulate the results, which contains BIC and log likelihood for each u . 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

5.3 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.

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 parameters, 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 to print out the number of bootstrap samples, logical 0 or 1. Default value: 0.

Output

bootse: The standard error for elements in β 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
```

5.4 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 (2013) by using maximum likelihood estimation. This function is the derivative of the objective function.

5.5 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 (2013) 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.

5.6 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.

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

Opts: A list containing the optional input parameters, 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 to print out Grassmann manifold optimization process, logical 0 or 1. Default value: 0.
- `Opts.init`: The initial value for the heteroscedastic envelope subspace. An r by u matrix. Default value is the one generated by function `get_Init4henv`.

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 β with respect to `Gamma`. A 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`. A 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.paramNum`: 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.asySE`: The asymptotic standard errors for elements in β under the heteroscedastic envelope model. An r by p matrix. The standard errors returned are asymptotic, for actual standard errors, multiply by $1 / \text{sqrt}(n)$.
- `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 (2013). 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 are implemented based on the algorithm in Section 2.2 of Su and Cook (2013).
2. The Grassmann manifold optimization step calls the package `sg_min` 2.4.3 by Ross Lipert (<http://web.mit.edu/~ripper/www/sgmin.html>).

Example

The following codes produce the results of the water strider example in Su and Cook (2013).

```

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]
    paramNum: 98
    l: 1.0051e+03
    covMatrix: [32x32 double]
    asySE: [8x3 double]
    ratio: [8x3 double]
    ng: [3x1 double]

ans =

6.2553 10.8792 6.2856
4.4358 5.0351 4.6347
4.1925 4.7967 4.2595
4.5553 5.9242 5.0582
7.6349 12.1591 9.1945
9.0979 11.1701 10.9407
11.2834 15.0924 12.0360
6.6312 10.7068 9.7542

```

5.7 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.

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 parameters, 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 to print out dimension selection process. Logical 0 or 1. Default value: 0.
- `Opts.table`: Flag to tabulate the results, which contains log likelihood, test statistic, degrees of freedom and p-value for each test. 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 α .

Example

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

u =

6

5.8 mfoldcv_henv

Use m-fold cross validation to select the dimension of the envelope subspace for heteroscedastic envelope model.

Contents

- Syntax
- Input
- Output
- Description
- Example

Syntax

```
SelectOutput = mfoldcv_henv(X, Y, m)
SelectOutput = mfoldcv_henv(X, Y, m, 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.

m: A positive integer that is used to indicate m-fold cross validation.

Opts: A list containing the optional input parameters. If one or several (even all) fields are not defined, the default settings are used.

- Opts.verbose: Flag to print out dimension selection process, logical 0 or 1. Default value: 0.
- Opts.table: Flag to tabulate the results, which contains cross validation error for each u. Logical 0 or 1. Default value: 0.
- Opts.perm: A positive integer indicating number permutations of the observations, m-fold cross validation is run on each permutation. If not specified, the division is based on the sequential order of the observations.
- Opts.seed: A real number that set the seeds for permutations. Default value is 1.

Output

SelectOutput: A list containing the results of the selection.

- SelectOutput.u: The dimension of the envelope subspace selected by m-fold cross validation. An integer between 0 and r.
- SelectOutput.PreErr: A vector containing prediction errors for each u if Opts.perm is not specified, or a matrix with the element in the ith row and jth column containing the prediction error for u=j-1 and ith permutation of the observations.

Description

This function implements m-fold cross validation to select the dimension of the envelope space, based on prediction performance. For each u , the data is partitioned into m parts, each part is in turn used for testing for the prediction performance while the rest $m-1$ parts are used for training. The dimension is selected as the one that minimizes the average prediction errors. As Y is multivariate, the identity inner product is used for computing the prediction errors.

Example

```
load wheatprotein.txt
Opts.table = 1; % Print out the table of average prediction error for each u
SelectOutput = mfoldcv_henv(X, Y, 5, Opts);
```

u	CV error
0	1.177
1	1.173
2	1.171

SelectOutput.u

ans =

2

```
Opts.perm = 10; % Run 5-fold CV on 10 permutations
Opts.seed = 3; % Set seed for the permutations
Opts.table = 1;
SelectOutput = mfoldcv_henv(X, Y, 5, Opts);
```

u	CV error
0	1.088
1	1.083
2	1.081

The rows of PreErr corresponds to permutations, and the columns of PreErr corresponds to u .

SelectOutput.PreErr


```
ans =
```

1.0746	1.0704	1.0681
1.0921	1.0881	1.0852
1.0849	1.0803	1.0789
1.1077	1.1037	1.1015
1.0795	1.0756	1.0732
1.0908	1.0861	1.0843
1.0921	1.0880	1.0865
1.0845	1.0809	1.0776
1.0815	1.0769	1.0756
1.0878	1.0838	1.0815

```
mean(SelectOutput.PreErr) % Compute the average of prediction errors for each u
```

```
ans =
```

1.0876	1.0834	1.0812
--------	--------	--------

```
std(SelectOutput.PreErr) % Compute the standard deviations of the prediction errors for each u
```

```
ans =
```

0.0091	0.0091	0.0091
--------	--------	--------

5.9 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
```

5.10 testcoefficient_henv

This function tests the null hypothesis $L * \beta * R = A$ versus the alternative hypothesis $L * \beta * R \neq A$, where β 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 β 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 β 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 β 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, the p-value, and the covariance matrix of $\text{vec}(L\hat{\beta}R)$. 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].
- TestOutput.covMatrix: The covariance matrix of $\text{vec}(L\hat{\beta}R)$. A d1 * d2 by d1 * d2 matrix.

Description

This function tests for hypothesis $H_0 : L\beta R = A$, versus $H_a : L\beta R \neq A$. The β 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 $\text{vec}(L\hat{\beta}R - A)^T \hat{\Sigma}^{-1} \text{vec}(L\hat{\beta}R - A)$, 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

6.1 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 parameters, 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 to print out dimension selection process. Logical 0 or 1. Default value: 0.
- `Opts.table`: Flag to tabulate the results, which contains AIC and log likelihood for each u . 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

6.2 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 parameters, 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 to print out dimension selection process. Logical 0 or 1. Default value: 0.
- Opts.table: Flag to tabulate the results, which contains BIC and log likelihood for each u. 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

6.3 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 .

B: Number of bootstrap samples. A positive integer.

Opts: A list containing the optional input parameters, 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 to print out the number of bootstrap samples, logical 0 or 1. Default value: 0.

Output

bootse: The standard error for elements in β 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

6.4 **dfF4ienv**

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

Contents

- Syntax
- Input
- Output
- Description

Syntax

$$df = \text{dF4ienv}(R, \text{DataParameter})$$

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.

6.5 F4ienv

Objective function for computing the inner envelope subspace.

Contents

- Syntax
- Input
- Output
- Description

Syntax

$$f = \text{F4ienv}(R, \text{DataParameter})$$

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.

6.6 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 .

Opts: A list containing the optional input parameters, 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 to print out Grassmann manifold optimization process, logical 0 or 1. Default value: 0.
- `Opts.init`: The initial value for the inner envelope subspace. An r by u matrix. Default value is the one generated by function `get_Init`.

Output

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

- `ModelOutput.beta`: The envelope estimator of the regression coefficients β . 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 β with respect to Γ_1 . A p by u matrix.
- `ModelOutput.B`: An $(r - u)$ by $(p - u)$ semi-orthogonal matrix, so that $(\Gamma_1, \Gamma_1 * B)$ spans β .
- `ModelOutput.eta2`: The transpose of the coordinates of β with respect to Γ_0 . A p by $(p - u)$ matrix.
- `ModelOutput.Omega1`: The coordinates of Σ with respect to Γ_1 . A u by u matrix.
- `ModelOutput.Omega0`: The coordinates of Σ with respect to Γ_0 . 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.asySE`: Asymptotic standard error for elements in β 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 β . An r by p matrix.
- `ModelOutput.paramNum`: 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 are implemented based on the algorithm in Su and Cook (2012).
2. The Grassmann manifold optimization step calls the package `sg_min` 2.4.3 by Ross Lipert (<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 =`


```
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]
    paramNum: 16
        l: -1.4805e+03
    covMatrix: [8x8 double]
    asySE: [4x2 double]
    ratio: [4x2 double]
        n: 150
```

```
ans =
```

```
    0.0035    0.2111
    0.0166    0.0940
   -0.0062    0.1322
    0.0044    0.0178
```

6.7 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 parameters, 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 to print out dimension selection process. Logical 0 or 1. Default value: 0.
- `Opts.table`: Flag to tabulate the results, which contains log likelihood, test statistic, degrees of freedom and p-value for each test. Logical 0 or 1. Default value: 0.

Output

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

Description

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

Example

```
load irisf.mat

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

u =

1
```

6.8 mfoldcv_ienv

Select the dimension of the inner envelope subspace using m-fold cross validation.

Contents

- Syntax
- Input
- Output
- Description
- Example

Syntax

```
SelectOutput = mfoldcv_ienv(X, Y, m)
SelectOutput = mfoldcv_ienv(X, Y, m, Opts)
```

Input

X: Predictors. An n by p matrix, p is the number of predictors and n is number of observations.

Y: Responses. An n by r matrix, r is the number of responses. The number of the responses should be greater than the number of the predictors. And they must be continuous variables.

m: A positive integer that is used to indicate m-fold cross validation.

Opts: A list containing the optional input parameters. If one or several (even all) fields are not defined, the default settings are used.

- Opts.verbose: Flag to print out dimension selection process, logical 0 or 1. Default value: 0.
- Opts.table: Flag to tabulate the results, which contains cross validation error for each u . Logical 0 or 1. Default value: 0.
- Opts.perm: A positive integer indicating number permutations of the observations, m-fold cross validation is run on each permutation. If not specified, the division is based on the sequential order of the observations.
- Opts.seed: A real number that set the seeds for permutations. Default value is 1.

Output

SelectOutput: A list containing the results of the selection.

- SelectOutput.u: The dimension of the inner envelope subspace selected by m-fold cross validation. An integer between 0 and p .
- SelectOutput.PreErr: A vector containing prediction errors for each u if Opts.perm is not specified, or a matrix with the element in the i th row and j th column containing the prediction error for $u=j-1$ and i th permutation of the observations.

Description

This function implements m-fold cross validation to select the dimension of the inner envelope space, based on prediction performance. For each u , the data is partitioned into m parts, each part is in turn used for testing for the prediction performance while the rest $m-1$ parts are used for training. The dimension is selected as the one that minimizes the average prediction errors. As Y is multivariate, the identity inner product is used for computing the prediction errors.

Example

```
load irisf.mat
Opts.table = 1; % Print out the table of average prediction error for each u
SelectOutput = mfoldcv_ienv(X, Y, 5, Opts);
```

u	CV error
0	7.869
1	12.865
2	13.068

```
SelectOutput.u
```

```
ans =
```

```
0
```

```
Opts.perm = 10; % Run 5-fold CV on 10 permutations
Opts.seed = 3; % Set seed for the permutations
Opts.table = 1;
SelectOutput = mfoldcv_ienv(X, Y, 5, Opts);
```

u	CV error
0	7.871
1	13.987
2	8.660

The rows of PreErr corresponds to permutations, and the columns of PreErr corresponds to u .

```
SelectOutput.PreErr
```

```
ans =
```

```
7.8658    8.0292   10.2529
7.8439   15.3317    9.1913
7.8063   14.3470    7.9989
7.8566   16.2194    8.0552
7.8660   14.7267    8.0347
7.8460   17.5724    9.5129
7.8152   14.2335    8.0367
7.9323    8.2747    8.1148
7.8682   14.9348    8.0369
8.0062   16.1967    9.3622
```

```
mean(SelectOutput.PreErr) % Compute the average of prediction errors for each u
```

```
ans =
```

```
7.8707   13.9866    8.6596
```

```
std(SelectOutput.PreErr) % Compute the standard deviations of the prediction errors for each u
```

```
ans =
```

```
0.0587    3.2368    0.8372
```

6.9 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 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 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
```


6.10 testcoefficient_ienv

This function tests the null hypothesis $L * \beta * R = A$ versus the alternative hypothesis $L * \beta * R \neq A$, where β 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 β 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 β 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, the p-value, and the covariance matrix of $\text{vec}(L\hat{\beta}R)$. 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].
- TestOutput.covMatrix: The covariance matrix of $\text{vec}(L\hat{\beta}R)$. A d1 * d2 by d1 * d2 matrix.

Description

This function tests for hypothesis $H_0 : L\beta R = A$, versus $H_\alpha : L\beta R \neq A$. The β 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 $\text{vec}(L\hat{\beta}R - A)^T \hat{\Sigma}^{-1} \text{vec}(L\hat{\beta}R - A)$, 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

7.1 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 parameters, 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 to print out dimension selection process. Logical 0 or 1. Default value: 0.
- Opts.table: Flag to tabulate the results, which contains AIC and log likelihood for each u. 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);
X.X1 = fiberpaper(:, 7);
X.X2 = fiberpaper(:, 5 : 6);
u = aic_penv(X, Y)
```

u =

3

7.2 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 parameters, 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 to print out dimension selection process. Logical 0 or 1. Default value: 0.
- Opts.table: Flag to tabulate the results, which contains BIC and log likelihood for each u. 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);
X.X1 = fiberpaper(:, 7);
X.X2 = fiberpaper(:, 5 : 6);
u = bic_penv(X, Y)
```

u =

1

7.3 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 parameters, 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 to print out the number of bootstrap samples, logical 0 or 1. Default value: 0.

Output

bootse: The standard error for elements in β_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);
X.X1 = fiberpaper(:, 7);
X.X2 = fiberpaper(:, 5 : 6);
alpha = 0.01;
u = modelselectlrt(X, Y, alpha, 'penv')

u =

    1

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

bootse =

    0.0074
    0.0021
    0.0043
    0.0019
```


7.4 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 parameters, 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 to print out dimension selection process. Logical 0 or 1. Default value: 0.
- Opts.table: Flag to tabulate the results, which contains log likelihood, test statistic, degrees of freedom and p-value for each test. 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 α .

Example

```
load fiberpaper.dat
Y = fiberpaper(:, 1 : 4);
X.X1 = fiberpaper(:, 7);
X.X2 = fiberpaper(:, 5 : 6);
alpha = 0.01;
u = lrt_penv(X, Y, alpha)
```

u =

1

7.5 mfoldcv_penv

Select the dimension of the partial envelope subspace using m-fold cross validation.

Contents

- Syntax
- Input
- Output
- Description
- Example

Syntax

```
SelectOutput = mfoldcv_penv(X, Y, m)
SelectOutput = mfoldcv_penv(X, Y, m, 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.

m: A positive integer that is used to indicate m-fold cross validation.

Opts: A list containing the optional input parameters. If one or several (even all) fields are not defined, the default settings are used.

- Opts.verbose: Flag to print out dimension selection process, logical 0 or 1. Default value: 0.
- Opts.table: Flag to tabulate the results, which contains cross validation error for each u. Logical 0 or 1. Default value: 0.
- Opts.perm: A positive integer indicating number permutations of the observations, m-fold cross validation is run on each permutation. If not specified, the division is based on the sequential order of the observations.
- Opts.seed: A real number that set the seeds for permutations. Default value is 1.

Output

SelectOutput: A list containing the results of the selection.

- SelectOutput.u: The dimension of the partial envelope subspace selected by m-fold cross validation. An integer between 0 and r.
- SelectOutput.PreErr: A vector containing prediction errors for each u if Opts.perm is not specified, or a matrix with the element in the ith row and jth column containing the prediction error for u=j-1 and ith permutation of the observations.

Description

This function implements m-fold cross validation to select the dimension of the partial envelope space, based on prediction performance. For each u , the data is partitioned into m parts, each part is in turn used for testing for the prediction performance while the rest $m-1$ parts are used for training. The dimension is selected as the one that minimizes the average prediction errors. As Y is multivariate, the identity inner product is used for computing the prediction errors.

Example

```
load fiberpaper.dat
Y = fiberpaper(:, 1 : 4);
X.X1 = fiberpaper(:, 7);
X.X2 = fiberpaper(:, 5 : 6);
Opts.table = 1; % Print out the table of average prediction error for each u
SelectOutput = mfoldcv_penv(X, Y, 5, Opts);
```

u	CV error
0	3.384
1	4.423
2	4.485
3	3.471
4	4.486

SelectOutput.u

ans =

0

```
Opts.perm = 10; % Run 5-fold CV on 10 permutations
Opts.seed = 3; % Set seed for the permutations
Opts.table = 1;
SelectOutput = mfoldcv_penv(X, Y, 5, Opts);
```

u	CV error
0	2.416
1	2.460
2	2.502
3	2.493
4	2.513

The rows of PreErr corresponds to permutations, and the columns of PreErr corresponds to u.

```
SelectOutput.PreErr
```

```
ans =
```

```
2.3928  2.3917  2.4230  2.4304  2.4584
2.4007  2.4007  2.4873  2.4312  2.4871
2.3040  2.3031  2.3598  2.3176  2.3747
2.3118  2.3103  2.3880  2.3786  2.3875
2.4937  2.4927  2.5157  2.5055  2.5149
2.3756  2.3756  2.3892  2.3851  2.3883
2.4432  2.6921  2.6902  2.7041  2.7020
2.5305  2.5286  2.5467  2.5416  2.5595
2.4482  2.6476  2.6520  2.6662  2.6648
2.4580  2.4568  2.5684  2.5677  2.5933
```

```
mean>SelectOutput.PreErr) % Compute the average of prediction errors for each u
```

```
ans =
```

```
2.4159  2.4599  2.5020  2.4928  2.5130
```

```
std>SelectOutput.PreErr) % Compute the standard deviations of the prediction errors for each u
```

```
ans =
```

```
0.0734  0.1322  0.1140  0.1273  0.1162
```

7.6 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.

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

Opts: A list containing the optional input parameters, 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 to print out Grassmann manifold optimization process, logical 0 or 1. Default value: 0.
- Opts.init: The initial value for the partial envelope subspace. An r by u matrix. Default value is the one generated by function `get_Init`.

Output

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

- ModelOutput.beta1: The partial envelope estimator of β_1 , which is the regression coefficients for X1. An r by p1 matrix.
- ModelOutput.beta2: The partial envelope estimator of β_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 β_1 with respect to Gamma. A u by p_1 matrix.
- **ModelOutput.Omega**: The coordinates of Sigma with respect to Gamma. A 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 r_p by r_p matrix. The covariance matrix returned are asymptotic. For the actual standard errors, multiply by $1/n$.
- **ModelOutput.asySE**: Asymptotic standard error for elements in β_1 under the partial envelope model. An r by p_1 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 β_1 . An r by p_1 matrix.
- **ModelOutput.paramNum**: 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 X_1 and X_2 , 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 X_1 and X_2 as predictors. When the dimension is 0, X_1 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 are 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` 2.4.3 by Ross Lipert (<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);
X.X1 = fiberpaper(:, 7);
X.X2 = fiberpaper(:, 5 : 6);
alpha = 0.01;
u = modelselectlrt(X, Y, alpha, 'penv')
```

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
paramNum: 23
covMatrix: [12x12 double]
asySE: [4x1 double]
ratio: [4x1 double]
n: 62
```

ans =

0.0149

ans =

```
4.9819
0.0999
0.0050
```

ans =

```
65.9692
6.8217
10.4152
9.6228
```


7.7 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 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 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);
X.X1 = fiberpaper(:, 7);
X.X2 = fiberpaper(:, 5 : 6);
alpha = 0.01;
u = modelselectlrt(X, Y, alpha, 'penv');
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

7.8 testcoefficient_penv

This function tests the null hypothesis $L * \beta_1 * R = A$ versus the alternative hypothesis $L * \beta_1 * R \neq A$, where β_1 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 β_1 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 β_1 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, the p-value, and the covariance matrix of $\text{vec}(L\hat{\beta}_1 R)$. 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 $[0, 1]$.
- **TestOutput.covMatrix:** The covariance matrix of $\text{vec}(L\hat{\beta}_1 R)$. A $d_1 * d_2$ by $d_1 * d_2$ matrix.

Description

This function tests for hypothesis $H_0 : L\beta_1 R = A$, versus $H_a : L\beta_1 R \neq A$. The β_1 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 $\beta_1 = 0$. The test statistics used is $\text{vec}(L\hat{\beta}_1 R - A)^T \hat{\Sigma}^{-1} \text{vec}(L\hat{\beta}_1 R - A)$, and the reference distribution is chi-squared distribution with degrees of freedom $d_1 * d_2$.

Example

```

load fiberpaper.dat
Y = fiberpaper(:, 1 : 4);
X.X1 = fiberpaper(:, 7);
X.X2 = fiberpaper(:, 5 : 6);
alpha = 0.01;
u = modelselectlrt(X, Y, alpha, 'penv');
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

senv

8.1 aic_senv

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

Contents

- Syntax
- Input
- Output
- Description
- Example

Syntax

```
u = aic_senv(X, Y)
u = aic_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 parameters, 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 to print out dimension selection process. Logical 0 or 1. Default value: 0.
- `Opts.table`: Flag to tabulate the results, which contains AIC and log likelihood for each u . Logical 0 or 1. Default value: 0.
- `Opts.rep`: Number of replicates for scales. This option imposes special structure on scaling parameters. For example, if `Opts.rep = [3 4]`, this means that the first three responses have the same scale and the next four responses share a different scale. The elements of this vector should sum to r . If not specified, the default is `[]`, then all responses will be scaled differently. If all responses have the same scale, input `[r]`, then the regular envelope will be applied to the data. The input should be a row vector.

Output

u: Dimension of the scaled 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

8.2 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 parameters, 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 to print out dimension selection process. Logical 0 or 1. Default value: 0.
- `Opts.table`: Flag to tabulate the results, which contains BIC and log likelihood for each u . Logical 0 or 1. Default value: 0.
- `Opts.rep`: Number of replicates for scales. This option imposes special structure on scaling parameters. For example, if `Opts.rep = [3 4]`, this means that the first three responses have the same scale and the next four responses share a different scale. The elements of this vector should sum to r . If not specified, the default is `[]`, then all responses will be scaled differently. If all responses have the same scale, input `[r]`, then the regular envelope will be applied to the data. The input should be a row vector.

Output

u: Dimension of the scaled 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

8.3 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 parameters, 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 to print out the number of bootstrap samples, logical 0 or 1. Default value: 0.
- **Opts.rep:** Number of replicates for scales. This option imposes special structure on scaling parameters. For example, if `Opts.rep = [3 4]`, this means that the first three responses have the same scale and the next four responses share a different scale. The elements of this vector should sum to r . If not specified, the default is `[]`, then all responses will be scaled differently. If all responses have the same scale, input `[r]`, then the regular envelope will be applied to the data. The input should be a row vector.

Output

bootse: The standard error for elements in β 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
```

8.4 dF4senv

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

Contents

- Syntax
- Input
- Output
- Description

Syntax

$$df = \text{dF4senv}(R, \text{DataParameter})$$

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.

8.5 F4senv

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

Contents

- Syntax
- Input
- Output
- Description

Syntax

$f = \text{F4senv}(R, \text{DataParameter})$

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 (2013) using maximum likelihood estimation. The columns of the semi-orthogonal matrix that minimizes this function span the estimated envelope subspace.

8.6 mfoldcv_senv

Select the dimension of the scaled envelope subspace using m-fold cross validation.

Contents

- Syntax
- Input
- Output
- Description
- Example

Syntax

```
SelectOutput = mfoldcv_senv(X, Y, m)
SelectOutput = mfoldcv_senv(X, Y, m, Opts)
```

Input

X: Predictors. An n by p matrix, p is the number of predictors and n is number of observations.

Y: Responses. An n by r matrix, r is the number of responses. The number of the responses should be greater than the number of the predictors. And they must be continuous variables.

m: A positive integer that is used to indicate m-fold cross validation.

Opts: A list containing the optional input parameters. If one or several (even all) fields are not defined, the default settings are used.

- Opts.verbose: Flag to print out dimension selection process, logical 0 or 1. Default value: 0.
- Opts.table: Flag to tabulate the results, which contains cross validation error for each u. Logical 0 or 1. Default value: 0.
- Opts.rep: Number of replicates for scales. This option imposes special structure on scaling parameters. For example, if Opts.rep = [3 4], this means that the first three responses have the same scale and the next four responses share a different scale. The elements of this vector should sum to r. If not specified, the default is [], then all responses will be scaled differently. If all responses have the same scale, input [r], then the regular envelope will be applied to the data. The input should be a row vector.
- Opts.perm: A positive integer indicating number permutations of the observations, m-fold cross validation is run on each permutation. If not specified, the division is based on the sequential order of the observations.
- Opts.seed: A real number that set the seeds for permutations. Default value is 1.

Output

SelectOutput: A list containing the results of the selection.

- SelectOutput.u: The dimension of the scaled envelope subspace selected by m-fold cross validation. An integer between 0 and r.
- SelectOutput.PreErr: A vector containing prediction errors for each u if Opts.perm is not specified, or a matrix with the element in the ith row and jth column containing the prediction error for u=j-1 and ith permutation of the observations.

Description

This function implements m-fold cross validation to select the dimension of the scaled envelope space, based on prediction performance. For each u , the data is partitioned into m parts, each part is in turn used for testing for the prediction performance while the rest $m-1$ parts are used for training. The dimension is selected as the one that minimizes the average prediction errors. As Y is multivariate, the identity inner product is used for computing the prediction errors.

Example

```
load('sales.txt')
Y = sales(:, 4 : 7);
X = sales(:, 1 : 3);
Opts.table = 1; % Print out the table of average prediction error for each u
SelectOutput = mfoldcv_senv(X, Y, 5, Opts);
```

u	CV error
0	11.969
1	5.750
2	5.266
3	5.410
4	5.343

SelectOutput.u

ans =

2

```
Opts.perm = 10; % Run 5-fold CV on 10 permutations
Opts.seed = 3; % Set seed for the permutations
Opts.table = 1;
SelectOutput = mfoldcv_senv(X, Y, 5, Opts);
```

u	CV error
0	11.977
1	5.754
2	5.305
3	5.465
4	5.437

The rows of PreErr corresponds to permutations, and the columns of PreErr corresponds to u .

```
SelectOutput.PreErr
```

```
ans =
```

```
11.9619    5.7170    5.2153    5.4024    5.3363
12.1588    5.6580    5.1543    5.2932    5.3016
11.9315    5.8612    5.3559    5.5420    5.5057
11.9389    5.7846    5.3054    5.5447    5.4539
11.9335    5.8343    5.3847    5.9421    5.9300
11.8577    5.6179    5.2153    5.3673    5.3689
11.9126    5.7536    5.2937    5.2898    5.3091
12.2099    5.7924    5.3580    5.3046    5.2770
11.9684    5.8209    5.4442    5.4736    5.4700
11.9016    5.6978    5.3260    5.4891    5.4158
```

```
mean>SelectOutput.PreErr) % Compute the average of prediction errors for each u
```

```
ans =
```

```
11.9775    5.7538    5.3053    5.4649    5.4368
```

```
std>SelectOutput.PreErr) % Compute the standard deviations of the prediction errors for each u
```

```
ans =
```

```
0.1141    0.0796    0.0885    0.1944    0.1899
```


8.7 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 Cook and Su (2013) using maximum likelihood estimation.

8.8 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 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 X_{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_{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
```

8.9 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 parameters, 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 to print out number of iterations, logical 0 or 1. Default value: 0.
- `Opts.rep`: Number of replicates for scales. This option imposes special structure on scaling parameters. For example, if `Opts.rep = [3 4]`, this means that the first three responses have the same scale and the next four responses share a different scale. The elements of this vector should sum to r . If not specified, the default is `[]`, then all responses will be scaled differently. If all responses have the same scale, input `[r]`, then the regular envelope will be applied to the data. The input should be a row vector.

Output

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

- `ModelOutput.beta`: The scaled envelope estimator of the regression coefficients β . 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 β with respect to `Gamma`. A u by p matrix.
- `ModelOutput.Omega`: The coordinates of Σ with respect to `Gamma`. A u by u matrix.
- `ModelOutput.Omega0`: The coordinates of Σ 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.asySE`: Asymptotic standard error for elements in β 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 β . An r by p matrix.
- `ModelOutput.paramNum`: 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 (2013). 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 are implemented based on the algorithm in Section 4.1 of Cook and Su (2013).
2. The Grassmann manifold optimization step calls the package `sg_min` 2.4.3 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 (2013).

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

`u =`

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

```
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]
    paramNum: 23
    l: -386.1900
    covMatrix: [12x12 double]
    asySE: [4x3 double]
    ratio: [4x3 double]
    n: 50
```

```
ModelOutput.Lambda
```

```
ans =
```

```

1.0000    0    0    0
    0  0.9729    0    0
    0    0  0.8067    0
    0    0    0  1.7016
```

```
1 - 1 ./ ModelOutput.ratio
```

```
ans =
```

```

0.6823  0.4901  0.6287
0.6119  0.3188  0.5594
0.4036  0.1267  0.3484
0.5329  0.4585  0.5180
```

This example demonstrates the use of Opts.rep. In this example, the first six responses are measured in mg/ml, and the next responses are measured in ug / ml.

```

load('Urine.txt')
Y = Urine(:, 3 : 11);
X = Urine(:, 12 : 14);
Opts.rep = [6 3];
u = bic_senv(X, Y, Opts)

```

```
u =
```

```
1
```

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

```
ModelOutput =
```

```

    beta: [9x3 double]
    Sigma: [9x9 double]
    Lambda: [9x9 double]
    Gamma: [9x1 double]
    Gamma0: [9x8 double]
    eta: [0.1064 0.1326 0.1465]
    Omega: 0.0043
    Omega0: [8x8 double]
    alpha: [9x1 double]
    paramNum: 58
    l: -500.3789
    covMatrix: [27x27 double]
    asySE: [9x3 double]
    ratio: [9x3 double]
    n: 45

```

```
ModelOutput.Lambda
```

```
ans =
```

```

1.0000    0    0    0    0    0    0    0    0
    0 1.0000    0    0    0    0    0    0    0
    0    0 1.0000    0    0    0    0    0    0
    0    0    0 1.0000    0    0    0    0    0
    0    0    0    0 1.0000    0    0    0    0
    0    0    0    0    0 1.0000    0    0    0
    0    0    0    0    0    0 0.4356    0    0
    0    0    0    0    0    0    0 0.4356    0
    0    0    0    0    0    0    0    0 0.4356

```

8.10 testcoefficient_senv

This function tests the null hypothesis $L * \beta * R = A$ versus the alternative hypothesis $L * \beta * R \neq A$, where β 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 β 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 β 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, the p-value, and the covariance matrix of $\text{vec}(L\hat{\beta}R)$. 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 $[0, 1]$.
- **TestOutput.covMatrix:** The covariance matrix of $\text{vec}(L\hat{\beta}R)$. A $d1 * d2$ by $d1 * d2$ matrix.

Description

This function tests for hypothesis $H_0 : L\beta R = A$, versus $H_\alpha : L\beta R \neq A$. The β 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 $\text{vec}(L\hat{\beta}R - A) \hat{\Sigma}^{-1} \text{vec}(L\hat{\beta}R - A)^T$, 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	4660.161	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

9.1 aic_sxenv

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

Contents

- Syntax
- Input
- Output
- Description
- Example

Syntax

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

Input

X: Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.

Y: Responses. An n by r matrix, r is the number of responses. The response can be univariate or multivariate and must be continuous variable.

Opts: A list containing the optional input parameters, 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 to print out dimension selection process. Logical 0 or 1. Default value: 0.
- `Opts.table`: Flag to tabulate the results, which contains BIC and log likelihood for each u . Logical 0 or 1. Default value: 0.
- `Opts.rep`: Number of replicates for scales. This option imposes special structure on scaling parameters. For example, if `Opts.rep = [3 4]`, this means that the first three responses have the same scale and the next four responses share a different scale. The elements of this vector should sum to r . If not specified, the default is `[]`, then all responses will be scaled differently. If all responses have the same scale, input `[r]`, then the regular envelope will be applied to the data. The input should be a row vector.

Output

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

Description

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

Example

```
load('chemo.mat')
X = X(:, [6 11 21 22]);
Opts.verbose = 1;
Opts.table = 1;
u = aic_sxenv(X, Y, Opts)
```

Current dimension 0
Current dimension 1
Current dimension 2
Current dimension 3

u	log likelihood	AIC
0	348.898	-615.795
1	503.949	-907.898
2	550.180	-988.361
3	591.501	-1059.002
4	596.093	-1062.186

u =

4

9.2 bic_sxenv

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

Contents

- Syntax
- Input
- Output
- Description
- Example

Syntax

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

Input

X: Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.

Y: Responses. An n by r matrix, r is the number of responses. The response can be univariate or multivariate and must be continuous variable.

Opts: A list containing the optional input parameters, 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 to print out dimension selection process. Logical 0 or 1. Default value: 0.
- `Opts.table`: Flag to tabulate the results, which contains BIC and log likelihood for each u . Logical 0 or 1. Default value: 0.
- `Opts.rep`: Number of replicates for scales. This option imposes special structure on scaling parameters. For example, if `Opts.rep = [3 4]`, this means that the first three responses have the same scale and the next four responses share a different scale. The elements of this vector should sum to r . If not specified, the default is `[]`, then all responses will be scaled differently. If all responses have the same scale, input `[r]`, then the regular envelope will be applied to the data. The input should be a row vector.

Output

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

Description

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

Example

```
load('chemo.mat')
X = X(:, [6 11 21 22]);
Opts.verbose = 1;
Opts.table = 1;
u = bic_sxenv(X, Y, Opts)
```

Current dimension 0
Current dimension 1
Current dimension 2
Current dimension 3

u	log likelihood	BIC
0	348.898	-532.756
1	503.949	-806.630
2	550.180	-874.941
3	591.501	-933.430
4	596.093	-930.539

u =

3

9.3 bstrp_spls

Compute bootstrap standard error for the scaled SIMPLS algorithm.

Contents

- Syntax
- Input
- Output
- Description
- Example

Syntax

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

Input

X: Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.

Y: Responses. An n by r matrix, r is the number of responses. The response can be univariate or multivariate and must be continuous variable.

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

B: Number of bootstrap samples. A positive integer.

Opts: A list containing the optional input parameters. If one or several (even all) fields are not defined, the default settings are used.

- Opts.verbose: Flag for print out the number of bootstrap samples, logical 0 or 1. Default value: 0.
- Opts.rep: Number of replicates for scales. This option imposes special structure on scaling parameters. For example, if $\text{Opts.rep} = [3\ 4]$, this means that the first three responses have the same scale and the next four responses share a different scale. The elements of this vector should sum to r . If not specified, the default is $[]$, then all responses will be scaled differently. If all responses have the same scale, input $[r]$, then the regular envelope will be applied to the data. The input should be a row vector.
- Opts.Gamma: The initial value for the envelope subspace. A p by u matrix. Default value is the one obtained from `xenv`, with $\Lambda^{-1}X$ being the predictor and Y being the response.
- Opts.Lambda: The initial value for the scales. A p by p diagonal matrix. Default value is the identity matrix.

Output

bootse: The standard error for elements in β computed by bootstrap. A p by r matrix.

Description

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

Example

```

load('chemo.mat')
X = X(:, [6 11 21 22]);
ModelOutput = spls(X, Y, 3);

Opts.Gamma = ModelOutput.Gamma;
Opts.verbose = 1;
B = 10;
bootse = bstrp_spls(X, Y, 3, B, Opts)

```

```

Current number of bootstrap sample 1
Current number of bootstrap sample 2
Current number of bootstrap sample 3
Current number of bootstrap sample 4
Current number of bootstrap sample 5
Current number of bootstrap sample 6
Current number of bootstrap sample 7
Current number of bootstrap sample 8
Current number of bootstrap sample 9
Current number of bootstrap sample 10

```

If convergence is not reached for a bootstrap sample,
it is still used in computing bootse.

bootse =

0.1841	0.4632	0.0080	0.8161	0.0003	0.0034
0.3393	0.8850	0.0041	1.2737	0.0005	0.0048
0.0050	0.0132	0.0002	0.0243	0.0000	0.0001
0.0895	0.2325	0.0077	0.2336	0.0001	0.0010

9.4 bstrp_sxenv

Compute bootstrap standard error for the scaled predictor envelope model.

Contents

- Syntax
- Input
- Output
- Description
- Example

Syntax

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

Input

X: Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.

Y: Responses. An n by r matrix, r is the number of responses. The response can be univariate or multivariate and must be continuous variable.

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

B: Number of bootstrap samples. A positive integer.

Opts: A list containing the optional input parameters, 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.
- **Opts.rep:** Number of replicates for scales. This option imposes special structure on scaling parameters. For example, if `Opts.rep = [3 4]`, this means that the first three responses have the same scale and the next four responses share a different scale. The elements of this vector should sum to r . If not specified, the default is `[]`, then all responses will be scaled differently. If all responses have the same scale, input `[r]`, then the regular envelope will be applied to the data. The input should be a row vector.
- **Opts.Gamma:** The initial value for the envelope subspace. A p by u matrix. Default value is the one obtained from `xenv`, with $\Lambda^{-1}X$ being the predictor and Y being the response.
- **Opts.Lambda:** The initial value for the scales. A p by p diagonal matrix. Default value is the identity matrix.

Output

bootse: The standard error for elements in β computed by bootstrap. A p by r matrix.

Description

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

Example

```

load('chemo.mat')
X = X(:, [6 11 21 22]);
ModelOutput = sxenv(X, Y, 3);
ModelOutput.asySE / sqrt(ModelOutput.n)

ans =

    0.1639    0.5200    0.0051    0.6208    0.0002    0.0024
    0.3105    0.8149    0.0109    1.1854    0.0005    0.0047
    0.0053    0.0140    0.0002    0.0203    0.0000    0.0001
    0.1408    0.3474    0.0051    0.5388    0.0002    0.0022

    Opts.Gamma = ModelOutput.Gamma;
    Opts.verbose = 1;
    B = 10;
    bootse = bstrp_sxenv(X, Y, 3, B, Opts)

Current number of bootstrap sample 1
Current number of bootstrap sample 2
Current number of bootstrap sample 3
Current number of bootstrap sample 4
Current number of bootstrap sample 5
Current number of bootstrap sample 6
Current number of bootstrap sample 7
Current number of bootstrap sample 8
Current number of bootstrap sample 9
Current number of bootstrap sample 10

If convergence is not reached for a bootstrap sample,
it is still used in computing bootse.

bootse =

    0.1641    0.5550    0.0060    0.4402    0.0002    0.0018
    0.3548    1.1077    0.0098    1.0674    0.0005    0.0038
    0.0041    0.0083    0.0001    0.0104    0.0000    0.0000
    0.1593    0.4480    0.0052    0.4429    0.0002    0.0019

```

9.5 dF4sxenv

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

Contents

- Syntax
- Input
- Output
- Description

Syntax

$$df = \text{dF4sxenv}(R, \text{DataParameter})$$

Input

R: A p by u semi orthogonal matrix, $0 < u \leq p$.

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

Output

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

Description

The objective function is derived in Section 2.2 of Cook and Su (2015) by using maximum likelihood estimation. This function is the derivative of the objective function.

9.6 F4sxenv

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

Contents

- Syntax
- Input
- Output
- Description

Syntax

$$f = \text{F4sxenv}(R, \text{DataParameter})$$

Input

R: A p 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 2.2 of Cook and Su (2015) using maximum likelihood estimation. The columns of the semi-orthogonal matrix that minimizes this function span the estimated envelope subspace.

9.7 mfoldcv_spls

Select the dimension in the scaled SIMPLS algorithm using m-fold cross validation.

Contents

- Syntax
- Input
- Output
- Description
- Example

Syntax

```
SelectOutput = mfoldcv_spls(X, Y, m)
SelectOutput = mfoldcv_spls(X, Y, m, Opts)
```

Input

X: Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.

Y: Responses. An n by r matrix, r is the number of responses. The response can be univariate or multivariate and must be continuous variable.

m: A positive integer that is used to indicate m-fold cross validation.

Opts: A list containing the optional input parameters. If one or several (even all) fields are not defined, the default settings are used.

- Opts.verbose: Flag to print out dimension selection process, logical 0 or 1. Default value: 0.
- Opts.table: Flag to tabulate the results, which contains cross validation error for each u . Logical 0 or 1. Default value: 0.
- Opts.rep: Number of replicates for scales. This option imposes special structure on scaling parameters. For example, if `Opts.rep = [3 4]`, this means that the first three responses have the same scale and the next four responses share a different scale. The elements of this vector should sum to r . If not specified, the default is `[]`, then all responses will be scaled differently. If all responses have the same scale, input `[r]`, then the regular envelope will be applied to the data. The input should be a row vector.
- Opts.perm: A positive integer indicating number permutations of the observations, m-fold cross validation is run on each permutation. If not specified, the division is based on the sequential order of the observations.
- Opts.seed: A real number that set the seeds for permutations. Default value is 1.

Output

SelectOutput: A list containing the results of the selection.

- SelectOutput.u: The dimension of the scaled predictor envelope subspace selected by m-fold cross validation. An integer between 0 and p .
- SelectOutput.PreErr: A vector containing prediction errors for each u if Opts.perm is not specified, or a matrix with the element in the i th row and j th column containing the prediction error for $u=j-1$ and i th permutation of the observations.

Description

This function implements m-fold cross validation to select the dimension in the scaled SIMPLS algorithm, based on prediction performance. For each u , the data is partitioned into m parts, each part is in turn used for testing for the prediction performance while the rest $m-1$ parts are used for training. The dimension is selected as the one that minimizes the average prediction errors. As Y is multivariate, the identity inner product is used for computing the prediction errors.

Example

```
load('chemo.mat')
X = X(:, [6 11 21 22]);
Opts.table = 1; % Print out the table of average prediction error for each u
SelectOutput = mfoldcv_spls(X, Y, 2, Opts);
SelectOutput.u
```

u	CV error
0	10.999
1	7.668
2	3.296
3	2.037
4	2.134

ans =

3

9.8 mfoldcv_sxenv

Select the dimension of the scaled predictor envelope subspace using m-fold cross validation.

Contents

- Syntax
- Input
- Output
- Description
- Example

Syntax

```
SelectOutput = mfoldcv_sxenv(X, Y, m)
SelectOutput = mfoldcv_sxenv(X, Y, m, Opts)
```

Input

X: Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.

Y: Responses. An n by r matrix, r is the number of responses. The response can be univariate or multivariate and must be continuous variable.

m: A positive integer that is used to indicate m-fold cross validation.

Opts: A list containing the optional input parameters. If one or several (even all) fields are not defined, the default settings are used.

- Opts.verbose: Flag to print out dimension selection process, logical 0 or 1. Default value: 0.
- Opts.table: Flag to tabulate the results, which contains cross validation error for each u . Logical 0 or 1. Default value: 0.
- Opts.rep: Number of replicates for scales. This option imposes special structure on scaling parameters. For example, if `Opts.rep = [3 4]`, this means that the first three responses have the same scale and the next four responses share a different scale. The elements of this vector should sum to r . If not specified, the default is `[]`, then all responses will be scaled differently. If all responses have the same scale, input `[r]`, then the regular envelope will be applied to the data. The input should be a row vector.
- Opts.perm: A positive integer indicating number permutations of the observations, m-fold cross validation is run on each permutation. If not specified, the division is based on the sequential order of the observations.
- Opts.seed: A real number that set the seeds for permutations. Default value is 1.

Output

SelectOutput: A list containing the results of the selection.

- SelectOutput.u: The dimension of the scaled predictor envelope subspace selected by m-fold cross validation. An integer between 0 and p .

- `SelectOutput.PreErr`: A vector containing prediction errors for each `u` if `Opts.perm` is not specified, or a matrix with the element in the `ith` row and `jth` column containing the prediction error for `u=j-1` and `ith` permutation of the observations.

Description

This function implements `m`-fold cross validation to select the dimension of the scaled predictor envelope space, based on prediction performance. For each `u`, the data is partitioned into `m` parts, each part is in turn used for testing for the prediction performance while the rest `m-1` parts are used for training. The dimension is selected as the one that minimizes the average prediction errors. As `Y` is multivariate, the identity inner product is used for computing the prediction errors.

Example

```
load('chemo.mat')
X = X(:, [6 11 21 22]);
Opts.table = 1; % Print out the table of average prediction error for each u
SelectOutput = mfoldcv_sxenv(X, Y, 2, Opts);
SelectOutput.u
```

```
Current dimension 1
Current dimension 2
Current dimension 3
Current dimension 4
Current dimension 5
```

u	CV error
0	10.999
1	5.189
2	2.554
3	2.236
4	2.134

```
ans =
```

```
4
```

9.9 objfun_spls

Objective function for computing a-star in the scaled SIMPLS algorithm.

Contents

- Syntax
- Input
- Output
- Description

Syntax

```
f = objfun_spls(a, Gamma1, Gamma2, DataParameter)
```

Input

a: A scalar between 0 and 1 containing the initial value of a.

Gamma1: A p by u matrix containing an orthogonal basis of the envelope subspace in the last iteration.

Gamma2: A p by u matrix containing an orthogonal basis of the envelope subspace in the current iteration.

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

Output

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

Description

The objective function is introduced in Section 3 of Cook and Su (2015) under the context of scaled SIMPLS algorithm (SPLS algorithm).

9.10 objfun_sxenv

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

Contents

- Syntax
- Input
- Output
- Description

Syntax

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

Input

d: A $(p - 1)$ by 1 vector containing the value of the scales (the second to the last diagonal element in Λ).

Gamma: A p by u matrix containing an orthogonal basis of the 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 2.2 of Cook and Su (2015) by using maximum likelihood estimation. This function is the derivative of the objective function.

9.11 predict_sxenv

Perform estimation or prediction under the scaled predictor envelope model.

Contents

- Syntax
- Input
- Output
- Description
- Example

Syntax

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

Input

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

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

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 X_{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 predictor envelope model at new value X_{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('chemo.mat')
X = X(:, [6 11 21 22]);
ModelOutput = sxenv(X, Y, 3);
Xnew = X(1, :)';
PredictOutput = predict_sxenv(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 =
```

```
    2.5127    2.7986
    3.6497    4.8150
    0.2168    0.2148
   29.6482   30.0626
    0.0165    0.0170
    0.1441    0.1473
```

```
ans =
```

```
    0.1073
    0.2652
    0.0039
    0.4103
    0.0002
    0.0016
```

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

```
PredictOutput =
```

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

```
ans =
```

```
    0.3405
    0.8402
    0.0122
    1.3028
    0.0005
    0.0052
```

9.12 spls

Implement the scaled SIMPLS algorithm.

Contents

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

Syntax

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

Input

X: Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.

Y: Responses. An n by r matrix, r is the number of responses. The response can be univariate or multivariate and must be continuous variable.

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

Opts: A list containing the optional input parameters. If one or several (even all) fields are not defined, the default settings are used.

- Opts.verbose: Flag for print out number of iterations, logical 0 or 1. Default value: 0.
- Opts.Lambda: The initial value for the scales. A p by p diagonal matrix. Default value is the identity matrix.

Output

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

- ModelOutput.beta: The scaled predictor envelope estimator of the regression coefficients β . An p by r matrix.
- ModelOutput.SigmaX: The scaled envelope estimator of the covariance matrix of X , Σ_X . A p by p matrix.
- ModelOutput.Lambda: The matrix of estimated scales. An p by p 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. A p by u semi-orthogonal matrix.
- ModelOutput.Gamma0: The orthogonal basis of the complement of the envelope subspace. A p by $p - u$ semi-orthogonal matrix.
- ModelOutput.eta: The coordinates of β with respect to Gamma. A u by r matrix.
- ModelOutput.Omega: The coordinates of Sigma with respect to Gamma. A u by u matrix.
- ModelOutput.Omega0: The coordinates of Sigma with respect to Gamma0. A $p - u$ by $p - u$ matrix.

- `ModelOutput.muX`: The estimated mean of the predictors in the scaled predictor envelope model. A p by 1 vector.
- `ModelOutput.muY`: The estimated mean of the responses in the scaled predictor envelope model. An r by 1 vector.
- `ModelOutput.sigYcX`: The estimated conditional covariance matrix of Y given X . An r by r matrix.
- `ModelOutput.paramNum`: The number of parameters in the scaled predictor envelope model. A positive integer.
- `ModelOutput.n`: The number of observations in the data. A positive integer.

Description

This function fits the scaled predictor 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 Cook and Su (2015). When the dimension is p , then the scaled predictor 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 are implemented based on the algorithm in Section 3 of Cook and Su (2015).

Example

The following codes uses a subset of the chemometrics example in Cook and Su (2015) to demonstrate the use of 'spls'.

```
load('chemo.mat')
X = X(:, [6 11 21 22]);
ModelOutput = spls(X, Y, 3);
ModelOutput.Lambda

ans =

    1.0000         0         0         0
         0    0.7100         0         0
         0         0    2.4526         0
         0         0         0    0.4401
```

This example demonstrates the use of `Opts.rep`. In this example, the first two predictors are temperatures measured at equal distances along the reactor, and the other two predictors are wall temperature of the reactor and the solvent feed rate.

```
load('chemo.mat')
X = X(:, [6 11 21 22]);
Opts.rep = [2 1 1];
ModelOutput = spls(X, Y, 3, Opts);
ModelOutput.Lambda
```

ans =

1.0000	0	0	0
0	1.0000	0	0
0	0	1.0539	0
0	0	0	0.7522

9.13 sxenv

Fit the scaled predictor envelope model.

Contents

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

Syntax

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

Input

X: Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.

Y: Responses. An n by r matrix, r is the number of responses. The response can be univariate or multivariate and must be continuous variable.

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

Opts: A list containing the optional input parameters, 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.
- **Opts.rep:** Number of replicates for scales. This option imposes special structure on scaling parameters. For example, if `Opts.rep = [3 4]`, this means that the first three responses have the same scale and the next four responses share a different scale. The elements of this vector should sum to r . If not specified, the default is `[]`, then all responses will be scaled differently. If all responses have the same scale, input `[r]`, then the regular envelope will be applied to the data. The input should be a row vector.
- **Opts.Gamma:** The initial value for the envelope subspace. A p by u matrix. Default value is the one obtained from `xenv`, with $\Lambda^{-1}X$ being the predictor and Y being the response.
- **Opts.Lambda:** The initial value for the scales. A p by p diagonal matrix. Default value is the identity matrix.

Output

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

- **ModelOutput.beta:** The scaled predictor envelope estimator of the regression coefficients β . An p by r matrix.

- `ModelOutput.SigmaX`: The scaled envelope estimator of the covariance matrix of X , Σ_X . A p by p matrix.
- `ModelOutput.Lambda`: The matrix of estimated scales. An p by p 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. A p by u semi-orthogonal matrix.
- `ModelOutput.Gamma0`: The orthogonal basis of the complement of the envelope subspace. A p by $p - u$ semi-orthogonal matrix.
- `ModelOutput.eta`: The coordinates of β with respect to `Gamma`. A u by r matrix.
- `ModelOutput.Omega`: The coordinates of `Sigma` with respect to `Gamma`. A u by u matrix.
- `ModelOutput.Omega0`: The coordinates of `Sigma` with respect to `Gamma0`. A $p - u$ by $p - u$ matrix.
- `ModelOutput.muX`: The estimated mean of the predictors in the scaled predictor envelope model. A p by 1 vector.
- `ModelOutput.muY`: The estimated mean of the responses in the scaled predictor envelope model. An r by 1 vector.
- `ModelOutput.sigYcX`: The estimated conditional covariance matrix of Y given X . An r by r matrix.
- `ModelOutput.covMatrix`: The asymptotic covariance of $\text{vec}(\beta)$. A pr by pr matrix. The covariance matrix returned are asymptotic. For the actual standard errors, multiply by $1 / n$.
- `ModelOutput.asySE`: Asymptotic standard error for elements in β under the scaled predictor envelope model. A p by r 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 predictor envelope estimator, for each element in β . A p by r matrix.
- `ModelOutput.paramNum`: The number of parameters in the scaled predictor envelope model. A positive integer.
- `ModelOutput.l`: The maximized log likelihood function. A real number.
- `ModelOutput.n`: The number of observations in the data. A positive integer.

Description

This function fits the scaled predictor 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 Cook and Su (2015). When the dimension is p , then the scaled predictor 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 are implemented based on the algorithm in Section 2.2 of Cook and Su (2015).
2. The Grassmann manifold optimization step calls the package `sg_min` 2.4.3 by Ross Lipert (<http://web.mit.edu/~ripper/www/sgmin.html>).

Example

The following codes uses a subset of the chemometrics example in Cook and Su (2015) to demonstrate the use of 'sxenv'.


```
load('chemo.mat')
X = X(:, [6 11 21 22]);
ModelOutput = sxenv(X, Y, 2);
ModelOutput.Lambda
ModelOutput.ratio
```

ans =

```
1.0000    0    0    0
    0  0.9410    0    0
    0    0  1.0351    0
    0    0    0  0.4929
```

ans =

```
4.1540  2.2048 581.7418 566.3622 375.6180 620.0116
3.3337  1.7682 245.5331 257.5329 207.5385 268.6982
1.4706  1.3487  1.5283  1.5283  1.5282  1.5283
1.0013  1.0010  1.0015  1.0015  1.0015  1.0015
```

This example demonstrates the use of Opts.rep. In this example, the first two predictors are temperatures measured at equal distances along the reactor, and the other two predictors are wall temperature of the reactor and the solvent feed rate.

```
load('chemo.mat')
X = X(:, [6 11 21 22]);
Opts.rep = [2 1 1];
ModelOutput = sxenv(X, Y, 2, Opts);
ModelOutput.Lambda
ModelOutput.ratio
```

ans =

```
1.0000    0    0    0
    0  1.0000    0    0
    0    0  1.0026    0
    0    0    0  0.5246
```

ans =

```
3.9752  2.1102 561.4176 545.4485 360.7323 597.6076
3.3356  1.7697 259.5508 268.1817 215.3125 279.6718
1.4661  1.3368  1.5281  1.5281  1.5281  1.5281
1.0013  1.0010  1.0015  1.0015  1.0015  1.0015
```

9.14 testcoefficient_sxenv

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

Contents

- Syntax
- Input
- Output
- Description
- Example

Syntax

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

Input

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

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 β on the left. It is a $d1$ by p matrix, while $d1$ is less than or equal to p . Default value: identity matrix I_p .
- **TestInput.R:** The matrix multiplied to β on the right. It is an r by $d2$ matrix, while $d2$ is less than or equal to r . Default value: identity matrix I_r .
- **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, the p-value, and the covariance matrix of $\text{vec}(L\hat{\beta}R)$. 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 $[0, 1]$.
- **TestOutput.covMatrix:** The covariance matrix of $\text{vec}(L\hat{\beta}R)$. A $d1 * d2$ by $d1 * d2$ matrix.

Description

This function tests for hypothesis $H_0 : L\beta R = A$, versus $H_a : L\beta R \neq A$. The β is estimated by the scaled predictor 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 $\text{vec}(L\hat{\beta}R - A)^T \hat{\Sigma}^{-1} \text{vec}(L\hat{\beta}R - A)$, and the reference distribution is chi-squared distribution with degrees of freedom $d1 * d2$.

Example

```

load('chemo.mat')
X = X(:, [6 11 21 22]);
ModelOutput = sxenv(X, Y, 3);
r = size(Y, 2);
p = size(X, 2);
TestInput.L = rand(2, p);
TestInput.R = rand(r, 1);
TestInput.A = zeros(2, 1);
TestOutout = testcoefficient_sxenv(ModelOutput, TestInput);

```

Test Hypothesis	Chisq Statistic	DF	P-value
L * beta * R = A	180.876	2	0.0000

10.1 aic_xenv

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

Contents

- Syntax
- Input
- Output
- Description
- Example

Syntax

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

Input

X: Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.

Y: Responses. An n by r matrix, r is the number of responses. The response can be univariate or multivariate and must be continuous variable.

Opts: A list containing the optional input parameters, 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 to print out dimension selection process. Logical 0 or 1. Default value: 0.
- `Opts.table`: Flag to tabulate the results, which contains AIC and log likelihood for each u . Logical 0 or 1. Default value: 0.

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)
```

u =

4

10.2 bic_xenv

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

Contents

- Syntax
- Input
- Output
- Description
- Example

Syntax

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

Input

X: Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.

Y: Responses. An n by r matrix, r is the number of responses. The response can be univariate or multivariate and must be continuous variable.

Opts: A list containing the optional input parameters, 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 to print out dimension selection process. Logical 0 or 1. Default value: 0.
- `Opts.table`: Flag to tabulate the results, which contains BIC and log likelihood for each u. Logical 0 or 1. Default value: 0.

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)
```

u =

4

10.3 bstrp_xenv

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

Contents

- Syntax
- Input
- Output
- Description
- Example

Syntax

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

Input

X: Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.

Y: Responses. An n by r matrix, r is the number of responses. The response can be univariate or multivariate and must be continuous variable.

u: Dimension of the envelope subspace. A positive integer between 0 and p .

B: Number of bootstrap samples. A positive integer.

Opts: A list containing the optional input parameters, 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 to print out the number of bootstrap samples, logical 0 or 1. Default value: 0.

Output

bootse: The standard error for elements in β computed by bootstrap. A p by r matrix.

Description

This function computes the bootstrap standard errors for the regression coefficients in the envelope model by bootstrapping the (X, Y) jointly. 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(X, Y, alpha)
```

u =

4

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

bootse =

```
0.0385
0.0915
0.1007
0.0268
0.0039
0.0203
```

10.4 dF4xenv

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

Contents

- Syntax
- Input
- Output
- Description

Syntax

$$df = \text{dF4xenv}(R, \text{DataParameter})$$

Input

R: A p by u semi orthogonal matrix, $0 < u \leq p$.

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

Output

df: A p by u matrix containing the value of the derivative function evaluated at R .

Description

The objective function is derived in Section 4.5.1 of Cook et al. (2013) by using maximum likelihood estimation. This function is the derivative of the objective function.

10.5 F4xenv

Objective function for computing the envelope subspace for the reduction on X.

Contents

- Syntax
- Input
- Output
- Description

Syntax

$f = \text{F4xenv}(R, \text{DataParameter})$

Input

R: A p 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 4.5.1 of Cook et al. (2013) using maximum likelihood estimation. The columns of the semi-orthogonal matrix that minimizes this function span the estimated envelope subspace.

10.6 lrt_xenv

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

Contents

- Syntax
- Input
- Output
- Description
- Example

Syntax

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

Input

X: Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.

Y: Responses. An n by r matrix, r is the number of responses. The response can be univariate or multivariate and must be continuous variable.

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

Opts: A list containing the optional input parameters, 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 to print out dimension selection process. Logical 0 or 1. Default value: 0.
- `Opts.table`: Flag to tabulate the results, which contains log likelihood, test statistic, degrees of freedom and p-value for each test. Logical 0 or 1. Default value: 0.

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 α .

Example

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

u =

4

10.7 mfoldcv_xenv

Use m-fold cross validation to select the dimension of the envelope subspace for the reduction on X.

Contents

- Syntax
- Input
- Output
- Description
- Example

Syntax

```
SelectOutput = mfoldcv_xenv(X, Y, m)
SelectOutput = mfoldcv_xenv(X, Y, m, Opts)
```

Input

X: Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.

Y: Responses. An n by r matrix, r is the number of responses. The response can be univariate or multivariate and must be continuous variable.

m: A positive integer that is used to indicate m-fold cross validation.

Opts: A list containing the optional input parameters. If one or several (even all) fields are not defined, the default settings are used.

- Opts.verbose: Flag to print out dimension selection process, logical 0 or 1. Default value: 0.
- Opts.table: Flag to tabulate the results, which contains cross validation error for each u. Logical 0 or 1. Default value: 0.
- Opts.perm: A positive integer indicating number permutations of the observations, m-fold cross validation is run on each permutation. If not specified, the division is based on the sequential order of the observations.
- Opts.seed: A real number that set the seeds for permutations. Default value is 1.

Output

SelectOutput: A list containing the results of the selection.

- SelectOutput.u: The dimension of the envelope subspace selected by m-fold cross validation. An integer between 0 and p.
- SelectOutput.PreErr: A vector containing prediction errors for each u if Opts.perm is not specified, or a matrix with the element in the ith row and jth column containing the prediction error for u=j-1 and ith permutation of the observations.

Description

This function implements m-fold cross validation to select the dimension of the envelope space, based on prediction performance. For each u , the data is partitioned into m parts, each part is in turn used for testing for the prediction performance while the rest $m-1$ parts are used for training. The dimension is selected as the one that minimizes the average prediction errors. If Y is multivariate, the identity inner product is used for computing the prediction errors.

Example

```
load wheatprotein.txt
X = wheatprotein(:, 1 : 6);
Y = wheatprotein(:, 7);
Opts.table = 1; % Print out the table of average prediction error for each u
SelectOutput = mfoldcv_xenv(X, Y, 5, Opts);
```

u	CV error
0	1.474
1	0.281
2	0.271
3	0.288
4	0.219
5	0.221
6	0.221

SelectOutput.u

ans =

4

```
Opts.perm = 10; % Run 5-fold CV on 10 permutations
Opts.seed = 3; % Set seed for the permutations
Opts.table = 1;
SelectOutput = mfoldcv_xenv(X, Y, 5, Opts);
```

u	CV error
0	1.473
1	0.274
2	0.273
3	0.283
4	0.234
5	0.214

6 0.214

The rows of PreErr corresponds to permutations, and the columns of PreErr corresponds to u.

SelectOutput.PreErr

ans =

1.4825	0.2751	0.2748	0.2933	0.2273	0.2265	0.2265
1.4807	0.3163	0.3178	0.3178	0.2394	0.2398	0.2391
1.4686	0.2592	0.2585	0.2574	0.2207	0.2005	0.2006
1.4705	0.2530	0.2523	0.2699	0.1984	0.1972	0.1972
1.4658	0.3015	0.2941	0.3082	0.2232	0.2226	0.2221
1.4698	0.2525	0.2528	0.2511	0.2536	0.2237	0.2241
1.4656	0.2589	0.2589	0.2813	0.2648	0.2077	0.2077
1.4730	0.2833	0.2829	0.2988	0.2252	0.2174	0.2166
1.4896	0.2842	0.2845	0.3074	0.2825	0.1993	0.1993
1.4609	0.2548	0.2545	0.2474	0.2032	0.2028	0.2027

mean(SelectOutput.PreErr) % Compute the average of prediction errors for each u

ans =

1.4727	0.2739	0.2731	0.2833	0.2338	0.2138	0.2136
--------	--------	--------	--------	--------	--------	--------

std(SelectOutput.PreErr) % Compute the standard deviations of the prediction errors for each u

ans =

0.0089	0.0223	0.0218	0.0256	0.0265	0.0143	0.0142
--------	--------	--------	--------	--------	--------	--------

10.8 predict_xenv

Perform estimation or prediction under the envelope model for the reduction on X.

Contents

- Syntax
- Input
- Output
- Description
- Example

Syntax

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

Input

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

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

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 envelope model for the reduction on X 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(:, 1 : 6);
Y = wheatprotein(:, 7);
u = bic_xenv(X, Y);
ModelOutput = xenv(X, Y, u);
Xnew = X(1, :);
PredictOutput = predict_xenv(ModelOutput, Xnew, 'estimation')
```

```
[PredictOutput.value, Y(1, :)] % Compare the fitted value with the data
PredictOutput.SE

PredictOutput =

    value: 9.1751
 covMatrix: 16.8439
        SE: 4.1041

ans =

    9.1751    9.2300

ans =

    4.1041

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

PredictOutput =

    value: 9.1751
 covMatrix: 16.8760
        SE: 4.1080

ans =

    4.1080
```

10.9 testcoefficient_xenv

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

Contents

- Syntax
- Input
- Output
- Description
- Example

Syntax

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

Input

ModelOutput: A list containing the maximum likelihood estimators and other statistics inherited from 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 β on the left. It is a d1 by p matrix, while d1 is less than or equal to p. Default value: identity matrix I_p .
- TestInput.R: The matrix multiplied to β on the right. It is a r by d2 matrix, while d2 is less than or equal to r. Default value: identity matrix I_r .
- 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, the p-value, and the covariance matrix of $\text{vec}(L\hat{\beta}R)$. 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].
- TestOutput.covMatrix: The covariance matrix of $\text{vec}(L\hat{\beta}R)$. A d1 * d2 by d1 * d2 matrix.

Description

This function tests for hypothesis $H_0 : L\beta R = A$, versus $H_a : L\beta R \neq A$. The β is estimated by the envelope model for the reduction on X. 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 $\text{vec}(L\hat{\beta}R - A) \hat{\Sigma}^{-1} \text{vec}(L\hat{\beta}R - A)^T$, and the reference distribution is chi-squared distribution with degrees of freedom d1 * d2.

Example

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

Test Hypothesis	Chisq Statistic	DF	P-value
L * beta * R = A	3233.053	6	0.0000

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

Test Hypothesis	Chisq Statistic	DF	P-value
L * beta * R = A	33.578	2	0.0000

10.10 xenv

Fit the envelope model for the reduction on X.

Contents

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

Syntax

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

Input

X: Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.

Y: Responses. An n by r matrix, r is the number of responses. The response can be univariate or multivariate and must be continuous variable.

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

Opts: A list containing the optional input parameters, 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 to print out Grassmann manifold optimization process, logical 0 or 1. Default value: 0.
- `Opts.init`: The initial value for the envelope subspace. A p by u matrix. Default value is the one generated by function `get_Init`.

Output

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

- `ModelOutput.beta`: The envelope estimator of the regression coefficients β . A p by r matrix.
- `ModelOutput.SigX`: The envelope estimator of the covariance matrix of X, Σ_X . A p by p matrix.
- `ModelOutput.Gamma`: The orthogonal basis of the envelope subspace. A p by u semi-orthogonal matrix.
- `ModelOutput.Gamma0`: The orthogonal basis of the complement of the envelope subspace. A p by p-u semi-orthogonal matrix.
- `ModelOutput.eta`: The coordinates of β with respect to Gamma. A u by r matrix.
- `ModelOutput.Omega`: The coordinates of Σ_X with respect to Gamma. A u by u matrix.

- `ModelOutput.Omega0`: The coordinates of Σ_X with respect to `Gamma0`. An $p - u$ by $p - u$ matrix.
- `ModelOutput.mu`: The estimated intercept. An r by 1 vector.
- `ModelOutput.sigYcX`: The estimated conditional covariance matrix of Y given X . An r by r matrix.
- `ModelOutput.l`: The maximized log likelihood function. A real number.
- `ModelOutput.covMatrix`: The asymptotic covariance of $\text{vec}(\beta)$. An pr by pr matrix. The covariance matrix returned are asymptotic. For the actual standard errors, multiply by $1 / n$.
- `ModelOutput.asySE`: Asymptotic standard error for elements in β 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 β . A p by r matrix.
- `ModelOutput.paramNum`: 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 in the predictor's space, 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. (2013). 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 are implemented based on the algorithm in Section 4.5.1 of Cook et al (2013).
2. The Grassmann manifold optimization step calls the package `sg_min` 2.4.3 by Ross Lipert (<http://web.mit.edu/~ripper/www/sgmin.html>).

Example

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

p = size(X, 2);
ModelOutput = xenv(X, Y, p);

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

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

```
ans =
```

```
0.0321
```

```
ans =
```

```
0.0321
```

```
temp.betaOLS'
ModelOutput.beta
```

```
ans =
```

```
-0.0416
-0.0490
0.3368
-0.1981
0.0020
-0.0480
```

```
ans =
```

```
-0.0416
-0.0490
0.3368
-0.1981
0.0020
-0.0480
```

```
u = bic_xenv(X, Y);
ModelOutput = xenv(X, Y, u)
```

```
ModelOutput =
```

```
beta: [6x1 double]
SigX: [6x6 double]
Gamma: [6x4 double]
Gamma0: [6x2 double]
eta: [4x1 double]
Omega: [4x4 double]
Omega0: [2x2 double]
mu: 24.8863
sigYcX: 0.0321
l: -865.6407
```

```
covMatrix: [6x6 double]
  asySE: [6x1 double]
  ratio: [6x1 double]
paramNum: 27
  n: 50

% To compare with the results obtained by Partial Least Squares, use the
% plsregress command
[XL, YL, XS, YS, BETA, PCTVAR, MSE, stats] = plsregress(X, Y, u);
ModelOutput.beta
BETA(2 : end, :)

ans =

-0.0443
-0.0481
 0.3377
-0.1963
 0.0019
-0.0487

ans =

-0.0199
 0.1373
 0.1309
-0.1827
 0.0056
-0.0708
```


xenvpls

11.1 bstrp_xenvpls

Compute bootstrap standard error of the envelope model for the reduction on X using partial least squares algorithm.

Contents

- Syntax
- Input
- Output
- Description
- Example

Syntax

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

Input

X: Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.

Y: Responses. An n by r matrix, r is the number of responses. The response can be univariate or multivariate and must be continuous variable.

u: Dimension of the envelope subspace. A positive integer between 0 and p.

B: Number of bootstrap samples. A positive integer.

Opts: A list containing the optional input parameters. If one or several (even all) fields are not defined, the default settings are used.

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

Output

bootse: The standard error for elements in β computed by bootstrap. A 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 is applied for the reduction on X , using the partial least squares algorithm.

Example

```
load VocabGrowth
X = VocabGrowth(:, 1 : 3);
Y = VocabGrowth(:, 4);
m = 5;
u = mfoldcv_xenvpls(X, Y, m)

u =

    1

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

bootse =

    0.0230
    0.0235
    0.0273
```

11.2 mfoldcv_xenvpls

Select the dimension of the envelope subspace using m-fold cross validation for envelope model on the reduction on X using partial least squares algorithm.

Contents

- Syntax
- Input
- Output
- Description
- Example

Syntax

```
SelectOutput = mfoldcv_xenvpls(X, Y, m)
SelectOutput = mfoldcv_xenvpls(X, Y, m, Opts)
```

Input

X: Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.

Y: Responses. An n by r matrix, r is the number of responses. The response can be univariate or multivariate and must be continuous variable.

m: A positive integer that is used to indicate m-fold cross validation.

Opts: A list containing the optional input parameters. If one or several (even all) fields are not defined, the default settings are used.

- Opts.verbose: Flag to print out dimension selection process, logical 0 or 1. Default value: 0.
- Opts.table: Flag to tabulate the results, which contains cross validation error for each u. Logical 0 or 1. Default value: 0.
- Opts.perm: A positive integer indicating number permutations of the observations, m-fold cross validation is run on each permutation. If not specified, the division is based on the sequential order of the observations.
- Opts.seed: A real number that set the seeds for permutations. Default value is 1.

Output

SelectOutput: A list containing the results of the selection.

- SelectOutput.u: The dimension of the envelope subspace selected by m-fold cross validation. An integer between 0 and p.
- SelectOutput.PreErr: A vector containing prediction errors for each u if Opts.perm is not specified, or a matrix with the element in the ith row and jth column containing the prediction error for u=j-1 and ith permutation of the observations.

Description

This function implements m-fold cross validation to select the dimension of the envelope space, based on prediction performance. For each u , the data is partitioned into m parts, each part is in turn used for testing for the prediction performance while the rest $m-1$ parts are used for training. The dimension is selected as the one that minimizes the average prediction errors. If Y is multivariate, the identity inner product is used for computing the prediction errors.

Example

```
load VocabGrowth
X = VocabGrowth(:, 1 : 3);
Y = VocabGrowth(:, 4);
Opts.table = 1; % Print out the table of average prediction error for each u
SelectOutput = mfoldcv_xenvpls(X, Y, 5, Opts);
```

u	CV error
0	1.959
1	1.091
2	1.127
3	1.130

```
SelectOutput.u
```

```
ans =
```

```
1
```

```
Opts.perm = 10; % Run 5-fold CV on 10 permutations
Opts.seed = 3; % Set seed for the permutations
Opts.table = 1;
SelectOutput = mfoldcv_xenvpls(X, Y, 5, Opts);
```

u	CV error
0	1.942
1	1.087
2	1.108
3	1.110

The rows of PreErr corresponds to permutations, and the columns of PreErr corresponds to u .

SelectOutput.PreErr

ans =

1.9576	1.1076	1.1272	1.1265
1.9500	1.0664	1.0834	1.0856
1.9347	1.0878	1.0938	1.0947
1.9298	1.0747	1.1080	1.1122
1.9309	1.0956	1.1130	1.1187
1.9659	1.0995	1.1180	1.1163
1.9312	1.0920	1.1019	1.1023
1.9188	1.0582	1.0836	1.0850
1.9303	1.0798	1.1009	1.1007
1.9676	1.1114	1.1494	1.1558

mean>SelectOutput.PreErr) % Compute the average of prediction errors for each u

ans =

1.9417	1.0873	1.1079	1.1098
--------	--------	--------	--------

std>SelectOutput.PreErr) % Compute the standard deviations of the prediction errors for each u

ans =

0.0171	0.0174	0.0203	0.0213
--------	--------	--------	--------

11.3 xenvpls

Fit the envelope model for the reduction on X using partial least squares algorithm.

Contents

- Syntax
- Input
- Output
- Description
- Reference
- Example

Syntax

`ModelOutput = xenvpls(X, Y, u)`

Input

X: Predictors. An n by p matrix, p is the number of predictors and n is number of observations. The predictors must be continuous variables.

Y: Responses. An n by r matrix, r is the number of responses. The response can be univariate or multivariate and must be continuous variable.

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

Output

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

- `ModelOutput.beta`: The envelope estimator of the regression coefficients β . A p by r matrix.
- `ModelOutput.SigX`: The envelope estimator of the covariance matrix of X, Σ_X . A p by p matrix.
- `ModelOutput.Gamma`: The orthogonal basis of the envelope subspace. A p by u semi-orthogonal matrix.
- `ModelOutput.Gamma0`: The orthogonal basis of the complement of the envelope subspace. A p by $p-u$ semi-orthogonal matrix.
- `ModelOutput.eta`: The coordinates of β with respect to Gamma. A u by r matrix.
- `ModelOutput.Omega`: The coordinates of Σ_X with respect to Gamma. A u by u matrix.
- `ModelOutput.Omega0`: The coordinates of Σ_X with respect to Gamma0. An $p - u$ by $p - u$ matrix.
- `ModelOutput.mu`: The estimated intercept. An r by 1 vector.
- `ModelOutput.sigYcX`: The estimated conditional covariance matrix of Y given X. An r by r matrix.
- `ModelOutput.paramNum`: 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 in the predictor's space, by the partial least squares algorithm in Cook et al. (2013). In the population level, this algorithm is equivalent to that in `xenv.m`, which uses the maximum likelihood estimation. In the sample version, the two algorithms are different. And this algorithm is much faster, which provides a root n consistent starting value for the one in `xenv.m`.

Reference

The codes are implemented based on the algorithm in Section 4.3 of Cook et al (2013).

Example

```
load VocabGrowth
X = VocabGrowth(:, 1 : 3);
Y = VocabGrowth(:, 4);
m = 5;
u = mfoldcv_xenvpls(X, Y, m)

u =

1

ModelOutput = xenvpls(X, Y, u)

ModelOutput =

    beta: [3x1 double]
    SigX: [3x3 double]
    Gamma: [3x1 double]
    Gamma0: [3x2 double]
    eta: 5.2899
    Omega: 10.9286
    Omega0: [2x2 double]
    mu: 1.5683
    sigYcX: 1.0934
    paramNum: 9
    n: 64

ModelOutput.beta

ans =

0.2573
0.2741
0.3049
```

12.1 center

Subtract the mean of each column.

Contents

- Syntax
- Input
- Output
- Description

Syntax

$$XC = \text{center}(X)$$

Input

X: A matrix or a column vector.

Output

XC: A matrix or a column vector with the mean for each column equal to 0.

Description

This function centerizes a matrix or a vector, by subtracting each column by its column mean.

12.2 Contr

Compute the contraction matrix of dimension r .

Contents

- Syntax
- Input
- Output
- Description

Syntax

$$C = \text{Contr}(r)$$

Input

r: Dimension of the contraction matrix. A positive integer.

Output

C: Contraction matrix of dimension r . C is an $r(r + 1) / 2$ by r^2 matrix.

Description

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

12.3 Expan

Compute the expansion matrix of dimension r .

Contents

- Syntax
- Input
- Output
- Description

Syntax

$$E = \text{Expan}(r)$$

Input

r: Dimension of the expansion matrix. A positive integer.

Output

E: Expansion matrix of dimension r . E is an r^2 by $r(r+1)/2$ matrix.

Description

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

12.4 fit_OLS

Fit multivariate linear regression.

Contents

- Syntax
- Input
- Output
- Description
- Example

Syntax

```
ModelOutput = fit_OLS(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

ModelOutput: A list that contains the maximum likelihood estimators of regression coefficients and error covariance matrix.

- ModelOutput.betaOLS: An r by p matrix containing estimate of the regression coefficients β .
- ModelOutput.SigmaOLS: An r by r matrix containing estimate of the error covariance matrix.
- ModelOutput.alpha: An r by 1 vector containing estimate of the intercept.
- ModelOutput.asySE: The asymptotic standard error for elements in β . An r by p matrix. The standard errors returned are asymptotic, for actual standard errors, multiply by $1/\sqrt{n}$.
- ModelOutput.n: The number of observations in the data. A positive integer.

Description

In a multivariate linear model, Y and X follows the following relationship: $Y = \alpha + \beta X + \varepsilon$, where ε contains the errors. This function performs the ordinary least squares fit to the inputs, and returns the estimates of β and the covariance matrix of ε .

Example

```
load wheatprotein.txt
X = wheatprotein(:, 8);
Y = wheatprotein(:, 1 : 6);
ModelOutput = fit_OLS(X, Y)
ModelOutput.betaOLS
```

ModelOutput.SigmaOLS

ModelOutput =

```

    betaOLS: [6x1 double]
    SigmaOLS: [6x6 double]
    alpha: [6x1 double]
    n: 50

```

ans =

```

    3.2724
    8.0288
    7.5224
   -2.0609
    3.2244
    0.6538

```

ans =

```

1.0e+03 *
    1.1905    0.9759    1.0506    1.1524    1.5384    0.6335
    0.9759    0.8061    0.8657    0.9432    1.2636    0.5266
    1.0506    0.8657    0.9310    1.0164    1.3664    0.5640
    1.1524    0.9432    1.0164    1.1228    1.5234    0.6183
    1.5384    1.2636    1.3664    1.5234    2.3229    0.8360
    0.6335    0.5266    0.5640    0.6183    0.8360    0.3618

```

12.5 get_envelope

Construct the envelope subspace using a sequential algorithm.

Contents

- Syntax
- Input
- Output
- Description
- Reference

Syntax

$$W = \text{get_envelope}(S, M, u)$$

Input

S: An r by p matrix whose columns span the subspace, the rank of S cannot be greater than u .

M: An r by r positive semi-definite matrix.

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

Output

W: An r by u semi-orthogonal matrix that spans the M -envelope of $\text{span}(S)$.

Description

This function constructs the M -envelope of $\text{span}(S)$ using a sequential algorithm similar to partial least squares.

Reference

The codes are implemented based on the algorithm in the lecture notes of Cook (2012).

Example

```
S = [1 2 3]';
S0 = grams(nulbasis(S'));
M = S * S' + S0 * S0';
u = 1;
W = get_envelope(S, M, u)
```

W =

```
0.2673
0.5345
0.8018
```

12.6 get_Init

Generate starting value for the envelope subspace.

Contents

- Syntax
- Input
- Output
- Description
- Reference

Syntax

$$W_{\text{Init}} = \text{get_Init}(F, u, \text{DataParameter})$$

Input

F: Objective function of the envelope subspace.

u: Dimension of the envelope. An integer between 1 and $r - 1$.

DataParameter: A list containing commonly used statistics computed from the data.

Output

WInit: The initial estimate of the orthogonal basis of the envelope subspace. An r by u orthogonal matrix.

Description

We compute the eigenvectors for the covariance matrices of Y and the estimated errors, and get $2r$ vectors. Then we get all the combinations of u vectors out of the $2r$ vectors. If the number of $2r$ choose u is small (≤ 50), we search over all the combinations and find out the one that minimizes the objective function F . If that number is large, then we do it iteratively: we pick up any u eigenvectors, fix all of them except the first one. Then we search over all the vectors orthogonal to the fixed ones, and record the one that minimizes F . Next, we fix the first u eigenvectors again but this time search for the second one, then we record the vector. This goes on and on until the last one. We do it for 5 rounds and use the final set as our starting value.

Reference

The codes are implemented based on the algorithm in Section 3.5 of Su and Cook (2011).

12.7 get_Init4envmean

Generate starting value for the envelope subspace in estimating the multivariate mean.

Contents

- Syntax
- Input
- Output
- Description
- Reference

Syntax

```
WInit = get_Init4envmean(F, u, DataParameter)
```

Input

F: Objective function to get the envelope subspace.

u: Dimension of the envelope. An integer between 1 and $p - 1$.

DataParameter: A list containing commonly used statistics computed from the data.

Output

WInit: The initial estimate of the orthogonal basis of the envelope subspace. A p by u orthogonal matrix.

Description

We compute the eigenvectors for the estimated error covariance matrix, and get p vectors. Then we get all the combinations of u vectors out of the p vectors. If the number of p choose u is small (≤ 50), we search over all the combinations and find out the one that minimizes the objective function F . If that number is large, then we do it iteratively: we pick up any u eigenvectors, fix all of them except the first one. Then we search over all the vectors orthogonal to the fixed ones, and record the one that minimizes F . Next, we fix the first u eigenvectors again but this time search for the second one, then we record the vector. This goes on and on until the last one. We do it for 3 rounds and use the final set as our starting value.

Reference

The codes are implemented based on the algorithm in Section 3.5 of Su and Cook (2011).

12.8 get_Init4henv

Generate starting value for the heteroscedastic envelope subspace.

Contents

- Syntax
- Input
- Output
- Description
- Reference

Syntax

```
WInit = get_Init4henv(F, u, DataParameter)
```

Input

F: Objective function to get the heteroscedastic envelope subspace.

u: Dimension of the envelope. An integer between 1 and $r - 1$.

DataParameter: A list containing commonly used statistics computed from the data.

Output

WInit: The initial estimate of the orthogonal basis of the heteroscedastic envelope subspace. An r by u orthogonal matrix.

Description

We compute the eigenvectors for the estimated error covariance matrix, and get r vectors. Then we get all the combinations of u vectors out of the r vectors. If the number of r choose u is small (≤ 50), we search over all the combinations and find out the one that minimizes the objective function F . If that number is large, then we do it iteratively: we pick up any u eigenvectors, fix all of them except the first one. Then we search over all the vectors orthogonal to the fixed ones, and record the one that minimizes F . Next, we fix the first u eigenvectors again but this time search for the second one, then we record the vector. This goes on and on until the last one. We do it for 3 rounds and use the final set as our starting value.

Reference

The codes are implemented based on the algorithm in Section 3.5 of Su and Cook (2011).

12.9 Kpd

Compute the communication matrix Kpd.

Contents

- Syntax
- Input
- Output
- Description
- Reference

Syntax

$$k = \text{Kpd}(p, d)$$

Input

p, d: two positive integers represent the dimension parameters for the communication matrix.

Output

k: The communication matrix Kpd. An $p * d$ by $p * d$ matrix.

Description

For a p by d matrix A , $\text{vec}(A') = \text{Kpd} * \text{vec}(A)$, and Kpd is called a communication matrix.

Reference

The codes are implemented based on Definition 3.1 in Magnus and Neudecker (1979).

12.10 Lmatrix

Extract the 2nd to the last diagonal element of a matrix into a vector.

Contents

- Syntax
- Input
- Output
- Description

Syntax

$$L = \text{Lmatrix}(r)$$

Input

r: The dimension of the matrix being extracted. The matrix should be an r by r matrix.

Output

L: An $r - 1$ dimensional vector that contains all the diagonal elements but the first one of the matrix.

Description

Let A be an r by r matrix, and vec be the vector operator, then $\text{Lmatrix}(r) * \text{vec}(A)$ will give the 2nd to the r th diagonal elements of A , arranged in a column vector.

12.11 make_dF

Generic function to generate the derivative function of the objective function F

Contents

- Syntax
- Input
- Output
- Description

Syntax

```
dF = make_dF(dfun_method_handle, FParameters)
```

Input

- `dfun_method_handle`: A specific model derivative function of the objective function.
- `FParameters`: A structure that contains data parameters as input for the function `dfun_method_handle`.

Output

- `dF`: The generic derivative function of the objective function for computing the envelope subspace.

Description

Generic function to generate the derivative function of the objective function F. The function first sets a handle to the specific model function and fixes the data parameters from the sample needed for its computation. The handle fixed with those parameters is then evaluated at a given value for argument W. A generic derivative function dF is returned.

12.12 make_F

Generic function to generate the objective function F.

Contents

- Syntax
- Input
- Output
- Description

Syntax

`F = make_F(fun_method_handle, FParameters)`

Input

- `fun_method_handle`: A specific model objective function.
- `FParameters`: A structure that contains data parameters as input for the function `fun_method_handle`.

Output

- `F`: The generic objective function for computing the envelope subspace.

Description

Generic function to generate the objective function F. The function first sets a handle to the specific model function and fixes the data parameters from the sample needed for its computation. The handle fixed with those parameters is then evaluated at a given value for argument W. A generic objective function F is returned.

12.13 make_opts

Make optional input parameters for running the sg_min package.

Contents

- Syntax
- Input
- Output:
- Description

Syntax

```
Opts = make_opts(Opts)
```

Input

Opts: A list containing optional input parameters for sg_min.m specified by users. One or several (even all) fields could be empty.

- Opts.maxIter: Maximum number of iterations.
- Opts.ftol: Tolerance parameter for F
- Opts.gradtol: Tolerance parameter for dF
- Opts.verbose: Flag to print out output, logical 0 or 1.

Output:

Opts: A list containing optional input parameters for sg_min.m, specified by users or the default values 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 to print out output, logical 0 or 1. Default value: 0.

Description

The sg_min function has some optional input parameters that control the iteration process. These parameters include maximum number of iteration, tolerance parameters for convergence of the objective function F and the derivative of the objective function dF, and the print out of the iteration process. The user can set one or all of parameters, if not, default values will be used.

12.14 make_parameter

Compute summary statistics from the data.

Contents

- Syntax
- Input
- Output
- Description

Syntax

```
DataParameter = make_parameter(X, Y, method)
```

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 .

method: A string of characters indicating which member of the envelope family to be used, the choices can be 'env', 'ienv', 'henv', 'senv', 'sxenv', 'xenv' or 'xenvpls'.

Output

DataParameter: A list that contains summary statistics computed from the data. The output list can vary from method to method.

- **DataParameter.n:** The number of observations in the data. A positive integer.
- **DataParameter.ng:** A p by 1 vector containing the number of observations in each group. p is the number of groups. Only for 'henv'.
- **DataParameter.ncum:** A p by 1 vector containing the total number of observations till this group. Only for 'henv'.
- **DataParameter.ind:** An n by 1 vector indicating the sequence of the observations after sorted by groups.
- **DataParameter.p:** The number of predictors or number of groups for 'henv'. A positive integer.
- **DataParameter.r:** The number of responses. A positive integer.
- **DataParameter.XC:** Centered predictors. An n by p matrix with the i th row being the i th observation of X subtracted by the mean of X . Only for 'env' and 'ienv'.
- **DataParameter.YC:** Centered responses. An n by r matrix with the i th row being the i th observation of Y subtracted by the mean of Y . Only for 'env' and 'ienv'.
- **DataParameter.mX:** The mean of predictors. A p by 1 vector. For all method except 'henv'.
- **DataParameter.mY:** The mean of responses. An r by 1 vector.
- **DataParameter.mYg:** An r by p matrix with the i th column being the sample mean of the i th group.
- **DataParameter.sigX:** The sample covariance matrix of X . A p by p matrix.
- **DataParameter.sigY:** The sample covariance matrix of Y . An r by r matrix.

- `DataParameter.sigRes`: For 'env', 'senv', 'ienv': The sample covariance matrix of the residuals from the ordinary least squares regression of Y on X. An r by r matrix. For 'henv', an r by r by p three dimensional matrix with the ith depth is the ith sample covariance matrix for the ith group.
- `DataParameter.sigFit`: The sample covariance matrix of the fitted value from the ordinary least squares regression of Y on X. An r by r matrix. Only for method 'ienv'.
- `DataParameter.betaOLS`: The regression coefficients from the ordinary least squares regression of Y on X. An r by p matrix. For all methods except 'henv'.
- `DataParameter.invsigY`: The inverse of the sample covariance matrix of Y. An r by r matrix. For all methods except 'ienv'.
- `DataParameter.invsigRes`: The inverse of the sample covariance matrix of the residuals from the ordinary least squares regression of Y on X. An r by r matrix. Only for method 'ienv'.

Description

This function computes statistics that will be used frequently in the estimation for each method.

12.15 mtest

Perform Box's M test to check the homogeneity of the covariance matrices.

Contents

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

Syntax

TestOutput = mtest(X, Y, alpha)

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.

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

Output

TestOutput: A list containing the Box's M statistic, the approximation test statistic, degrees of freedom for the approximation statistic test, and the p-value. At the same time, a table is printed out.

- TestOutput.mStatistic: The Box's M statistic. A real number.
- TestOutput.approxStatistic: The approximation test statistic.
- TestOutput.df: The degrees of freedom of the approximation statistic test. A positive integer.
- TestOutput.pValue: p-value of the test. A real number in [0, 1].

Description

This function performs the Box's M test for homogeneity of the covariance matrices for different groups, indicated by X. If the groups sample-size is at least 20 (sufficiently large), Box's M test takes a Chi-square approximation; otherwise it takes an F approximation.

References

The codes are implemented based on Trujillo-Ortiz, A., R. Hernandez-Walls, K. Castro-Morales, A. Espinoza-Tenorio, A. Guia-Ramirez and R. Carmona-Pina. (2002). MBoxtest: Multivariate Statistical Testing for the Homogeneity of Covariance Matrices by the Box's M. A MATLAB file. [WWW document]. URL: <http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=2733&objectType=FILE>

Example

```
load waterstrider.mat
alpha = 0.01;
TestOutput = mtest(X, Y, alpha);
```

MBox	Chi-sqr.	df	P
157.5977	137.3361	72	0.0000

Covariance matrices are significantly different.