Package 'JPEN'

January 20, 2025

Type Package

Title Covariance and Inverse Covariance Matrix Estimation Using Joint Penalty

Version 1.0

Date 2015-08-20

Author Ashwini Maurya

Maintainer Ashwini Maurya <mauryaas@msu.edu>

Description A Joint PENalty Estimation of Covariance and Inverse Covariance Matrices.

Depends mvtnorm(>= 1.0-2), stats(>= 2.15.0),

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2015-09-16 10:05:02

Contents

JPEN-package *Covariance and Inverse Covariance Matrix Estimation Using Joint Penalty*

Description

A Joint PENalty Estimation of Covariance and Inverse Covariance Matrices.

Details

The DESCRIPTION file:

Package:	JPEN
Type:	Package
Title:	Covariance and Inverse Covariance Matrix Estimation Using Joint Penalty
Version:	1.0
Date:	2015-08-20
Author:	Ashwini Maurya
Maintainer:	Ashwini Maurya <mauryaas@msu.edu>
Description:	A Joint PENalty Estimation of Covariance and Inverse Covariance Matrices.
Depends:	mvtnorm(>= 1.0-2), stats(>= 2.15.0),
License:	GPL-2

Index of help topics:

Author(s)

Ashwini Maurya, Email: mauryaas@msu.edu. Ashwini Maurya Maintainer: Ashwini Maurya <mauryaas@msu.edu>

References

A Well Conditioned and Sparse Estimate of Covariance and Inverse Covariance Matrix Using Joint Penalty. Submitted. http://arxiv.org/pdf/1412.7907v2.pdf

See Also

jpen,jpen.inv

`f.K.fold` *Subset the data into K fold, training and test data.*

Description

K-fold subsetting.

Usage

```
f.K.fold(Nobs, K = 5)
```

Arguments

`Nobs`	n is number of observations
`K`	K is number of folds, typically 5 fold.

Details

K-fold subset of observations into training and test data.

Value

Returns the index for K-fold training and test data subsets.

Author(s)

Ashwini Maurya, Email: mauryaas@msu.edu

References

A Well Conditioned and Sparse Estimate of Covariance and Inverse Covariance Matrix Using Joint Penalty. Submitted. http://arxiv.org/pdf/1412.7907v2.pdf

Examples

```
n=100;K=5;cv=f.K.fold(n,K);
```

jpen *JPEN Estimate of covariance matrix*

Description

Estimate of covariance Matrix using Joint Penalty Method

Usage

```
jpen(S,  gam, lam=NULL)
```

Arguments

S	Sample covariance matrix.
gam	Tuning parameter gamma. gam is non-negative.
lam	Tuning parameter lambda. lam is non-negative.

Details

This function returns an estimate of covariance matrix using Joint Penalty method.

Value

Estimate of Covariance Matrix.

Author(s)

Ashwini Maurya, Email: mauryaas@msu.edu

References

A Well Conditioned and Sparse Estimate of Covariance and Inverse Covariance Matrix Using Joint Penalty. Submitted. http://arxiv.org/pdf/1412.7907v2.pdf

See Also

jpen.tune, jpen.inv

Examples

```
p=10;n=100;
Sig=diag(p);
y=rmvnorm(n,mean=rep(0,p),sigma=Sig);
gam=1.0;S=var(y);
lam=2/p;
Sighat=jpen(S,gam,lam);
```

jpen.inv *JPEN estimate of inverse cov matrix*

Description

A well conditioned and sparse estimate of inverse covariance matrix using Joint Penalty

Usage

```
jpen.inv(S, gam, lam=NULL)
```

Arguments

S	Sample cov matrix or a positive definite estimate based on covariance matrix.
gam	gam is tuning parameter for eigenvalues shrinkage.
lam	lam is tuning parameter for sparsity.

Details

Estimates a well conditioned and sparse inverse covariance matrix using Joint Penalty. If input matrix is singular or nearly singular, a JPEN estimate of covariance matrix is used in place of S.

Value

Returns a well conditioned and positive inverse covariance matrix.

Author(s)

Ashwini Maurya, Email: mauryaas@msu.edu.

References

A Well Conditioned and Sparse Estimate of Covariance and Inverse Covariance Matrix Using Joint Penalty. Submitted. http://arxiv.org/pdf/1412.7907v2.pdf

See Also

jpen,jpen.tune,jpen.inv.tune

Examples

```
p=10;n=100;
Sig=diag(p);
y=rmvnorm(n,mean=rep(0,p),sigma=Sig);
S=var(y);
gam=1.0;
lam=2*max(abs(S[col(S)!=row(S)]))/p;
Omghat=jpen.inv(var(y),gam,lam);
```

`jpen.inv.tune` *Tuning parameter Selection for inverse covariance matrix estimation based on minimization of Gaussian log-likelihood.*

Description

Returns optimal values of tuning parameters lambda and gamma

Usage

```
jpen.inv.tune(Ytr, gama, lambda=NULL)
```

Arguments

`Ytr`	Ytr is matrix of observations.
`gama`	A vector of gamma values.
`lambda`	Optional vector of values of lambda. If optional, the algorithm automatically calculates 10 values of lambda for each gamma and finds the optimal values of (lambda,gamma) that minimizes the negative of Gaussian likelihood function using K-fold cross validation.

Details

Returns the value of optimal tuning parameters. The function uses K-fold cross validation to select the best tuning parameter from among a set of of values of lambda and gamma.

Value

Returns the optimal values of lambda and gamma.

Author(s)

Ashwini Maurya, Email: mauryaas@msu.edu.

References

A Well Conditioned and Sparse Estimate of Covariance and Inverse Covariance Matrix Using Joint Penalty. Submitted. http://arxiv.org/pdf/1412.7907v2.pdf

See Also

jpen

Examples

```
p=10;n=100;
Sig=diag(p);
y=rmvnorm(n,mean=rep(0,p),sigma=Sig);
gama=c(0.5,1.0);
opt=jpen.inv.tune(var(y),gama);
```

`jpen.tune` *Tuning parameter selection based on minimization of 5 fold mean square error.*

Description

Returns optimal values of tuning parameters lambda and gamma which minimizes the K-fold cross-validation error on

Usage

```
jpen.tune(Ytr, gama, lambda=NULL)
```

Arguments

`Ytr`	Ytr is matrix of observations.
`gama`	gama is vector of gamma values. gamma is non-negative.
`lambda`	lambda is vector of lambda values. lambda is non-negative.

Details

Returns the value of optimal tuning parameters. The function uses K-fold cross validation to select the best tuning parameter from among a set of of values of lambda and gamma.

Value

Returns the optimal values of lambda and gamma.

Author(s)

Ashwini Maurya, Email: mauryaas@msu.edu.

References

A Well Conditioned and Sparse Estimate of Covariance and Inverse Covariance Matrix Using Joint Penalty. Submitted. http://arxiv.org/pdf/1412.7907v2.pdf

See Also

jpen

Examples

```
p=10;n=100;
Sig=diag(p);
y=rmvnorm(n,mean=rep(0,p),sigma=Sig);
gama=c(0.5,1.0);
opt=jpen.tune(Ytr=y,gama);
```

`lamvec` *returns a vector of values of lambda for given value of gamma*

Description

returns 10 values of lambda for each gamma.

Usage

```
lamvec(c, gam, p)
```

Arguments

`c`	c is absolute maximum of off-diagonal entries of sample covariance matrix.
`gam`	gamma is a non-negative constant.
`p`	p is number of rows/columns of matrix.

Details

The lamvec function retuns a 10 values of lambda for each value of gamma. A larger value of lambda yields sparse estimate but need not be positive definite, however at least one combination of (lambda, gamma) will yield a positive definite solution. If two different combination of (lambda, gamma) yeilds same cross validation error, a larger values of lambda will be selected which results in more sparse solution.

Value

A vector of values of lambda for each combination of gama. By choosing c as the maximum of off-diagonal elements of sample covariance matrix, the largest value of lambda yields an estimate which diagonal matrix with elements proportional to the diagonal elements of sample covariance matrix.

Author(s)

Ashwini Maurya, Email: mauryaas@msu.edu

References

A Well Conditioned and Sparse Estimate of Covariance and Inverse Covariance Matrix Using Joint Penalty. Submitted. http://arxiv.org/pdf/1412.7907v2.pdf

See Also

jpen, jpen.inv, jpen.tune, jpen.tune.inv

Examples

```
p=10;n=100;Sig=diag(p);
y=rmvnorm(n,mean=rep(0,p),sigma=Sig);
gam=c(0.5);
S=var(y);
c=max(abs(S[row(S)!=col(S)]));
lambda=lamvec(c,gam,p);
```

---

tr *Trace of matrix*

Description

Returns the trace of a matrix

Usage

```
tr(A)
```

Arguments

A	A is the input matrix.

Details

Returns the trace (sum of diagonal elements)of input matrix).

Value

Trace of input matrix.

Author(s)

Ashwini Maurya, Email: mauryaas@msu.edu

References

A Well Conditioned and Sparse Estimate of Covariance and Inverse Covariance Matrix Using Joint Penalty. Submitted. http://arxiv.org/pdf/1412.7907v2.pdf

Examples

```
p=10;n=100;Sig=diag(p);
y=rmvnorm(n,mean=rep(0,p),sigma=Sig);
S=var(y);
tr(S);
```

Index