

# Package ‘parcor’

September 4, 2014

**Type** Package

**Title** Regularized estimation of partial correlation matrices

**Version** 0.2-6

**Date** 2014-09-04

**Depends** MASS, glmnet, ppls, Epi, GeneNet

**Author** Nicole Kraemer, Juliane Schaefer

**Maintainer** Nicole Kraemer <kraemer\_r\_packages@yahoo.de>

**Description** The package estimates the matrix of partial correlations based on different regularized regression methods: lasso, adaptive lasso, PLS, and Ridge Regression. In addition, the package provides model selection for lasso, adaptive lasso and Ridge regression based on cross-validation.

**License** GPL (>= 2)

**LazyLoad** yes

**Repository** CRAN

**NeedsCompilation** no

**Date/Publication** 2014-09-04 13:43:28

## R topics documented:

parcor-package	2
adalasso	2
adalasso.net	4
Beta2parcor	5
lm.ridge.univariate	7
mylars	8
performance.pcor	9
pls.net	10
ridge.cv	12
ridge.net	13
sym2vec	14

**Index****15**


---

parcor-package	<i>Parcor: Estimation of partial correlations based on regularized regression.</i>
----------------	--

---

**Description**

This package contains different methods to estimate the matrix of partial correlations based on a  $(n \times p)$  matrix  $X$  of observation. For low-dimensional settings ( $p > n$ ), the matrix of partial correlations can be estimated based on  $p$  least-squares regression fits. However, in high-dimensional scenarios ( $p < n$ ), these least-squares problems are ill-posed and need to be regularized. This package contains four different regularized regression techniques for the estimation of the partial correlations: lasso, adaptive lasso, ridge regression, and Partial Least Squares. In addition, the package provides model selection for lasso, adaptive lasso and Ridge regression based on cross-validation.

**Details**

Package:	parcor
Type:	Package
Version:	0.2-6
Date:	2014-09-04
License:	GPL2 or newer
LazyLoad:	yes

**Author(s)**

Nicole Kraemer, Juliane Schaefer

Maintainer: Nicole Kraemer <kraemer\_r\_packages@yahoo.de>

**References**

N. Kraemer, J. Schaefer, A.-L. Boulesteix (2009) "Regularized Estimation of Large-Scale Gene Regulatory Networks with Gaussian Graphical Models", BMC Bioinformatics, 10:384

<http://www.biomedcentral.com/1471-2105/10/384/>

---

adalasso	<i>Adaptive Lasso</i>
----------	-----------------------

---

**Description**

This function computes the lasso and adaptive lasso solution based on  $k$ -fold cross-validation. The initial weights for adaptive lasso are computed from a lasso fit.

**Usage**

```
adalasso(X, y, k = 10, use.Gram = TRUE, both=TRUE, intercept=TRUE)
```

**Arguments**

X	matrix of input observations. The rows of X contain the samples, the columns of X contain the observed variables
y	vector of responses. The length of y must equal the number of rows of X
k	the number of splits in k-fold cross-validation. The same k is used for the estimation of the weights and the estimation of the penalty term for adaptive lasso. Default is k=10.
use.Gram	When the number of variables is very large, you may not want LARS to pre-compute the Gram matrix. Default is use.Gram=TRUE.
both	Logical. If both=FALSE, only the lasso coefficients are computed. Default is both=TRUE.
intercept	Should an intercept be included? Default is intercept=TRUE.

**Details**

In each of the k-fold cross-validation steps, the weights for adaptive lasso are computed in terms of a lasso fit. (The optimal value of the penalty term is selected via k-fold cross-validation). Note that this implies that a lasso solution is computed  $k \times k$  times!

**Value**

intercept.lasso	intercept for lasso. If intercept=FALSE was specified, the intercept is set to 0.
intercept.adalasso	intercept for adaptive lasso. If intercept=FALSE was specified, the intercept is set to 0.
coefficients.adalasso	regression coefficients for adaptive lasso.
coefficients.lasso	regression coefficients for lasso.
cv.lasso	cv error for the optimal lasso model.
cv.adalasso	cv error for the optimal adaptive lasso model.
lambda.lasso	optimal lambda value for lasso-
lambda.adalasso	optimal lambda value for adaptive lasso.

**Author(s)**

Nicole Kraemer, Juliane Schaefer

## References

H. Zou (2006) "The Adaptive Lasso and its Oracle Property", Journal of the American Statistical Association 101 (476): 1418-1429.

N. Kraemer, J. Schaefer, A.-L. Boulesteix (2009) "Regularized Estimation of Large-Scale Gene Regulatory Networks using Gaussian Graphical Models", BMC Bioinformatics, 10:384

<http://www.biomedcentral.com/1471-2105/10/384/>

## See Also

[Beta2parcor](#), [adalasso.net](#)

## Examples

```
n<-100 # number of observations
p<-60 # number of variables
X<-matrix(rnorm(n*p),ncol=p)
y<-rnorm(n)
ada.object<-adalasso(X,y,k=10)
```

---

adalasso.net

*Partial Correlations with (Adaptive) Lasso*

---

## Description

This function computes the matrix of partial correlations based on an estimation of the corresponding regression models via lasso and adaptive lasso respectively.

## Usage

```
adalasso.net(X, k = 10, use.Gram=FALSE, both=TRUE, verbose=FALSE, intercept=TRUE)
```

## Arguments

X	matrix of observations. The rows of X contain the samples, the columns of X contain the observed variables.
k	the number of splits in k-fold cross-validation. The same k is used for the estimation of the weights and the estimation of the penalty term for adaptive lasso. Default value is k=10.
use.Gram	When the number of variables is very large, you may not want LARS to pre-compute the Gram matrix. Default is use.Gram=FALSE.
both	Logical. If both=FALSE, only the lasso solution is computed. Default is both=TRUE.
verbose	Print information on conflicting signs etc. Default is verbose=FALSE
intercept	Should an intercept be included in the regression models? Default is intercept=TRUE.

**Details**

For each of the columns of  $X$ , a regression model based on (adaptive) lasso is computed. In each of the  $k$ -fold cross-validation steps, the weights for adaptive lasso are computed in terms of a lasso fit. (The optimal value of the penalty term is selected via  $k$ -fold cross-validation). Note that this implies that a lasso solution is computed  $k*k$  times! Finally, the results of the regression models are transformed via the function Beta2parcor.

**Value**

pcor.adalasso    estimated matrix of partial correlation coefficients for adaptive lasso.  
pcor.lasso        estimated matrix of partial correlation coefficients for lasso.  
...

**Author(s)**

Nicole Kraemer

**References**

H. Zou (2006) "The Adaptive Lasso and its Oracle Property", Journal of the American Statistical Association. 101 (476): 1418-1429.

N. Kraemer, J. Schaefer, A.-L. Boulesteix (2009) "Regularized Estimation of Large-Scale Gene Regulatory Networks using Gaussian Graphical Models", BMC Bioinformatics, 10:384

<http://www.biomedcentral.com/1471-2105/10/384/>

**See Also**

[Beta2parcor](#), [adalasso](#)

**Examples**

```
n<-20
p<-10
X<-matrix(rnorm(n*p),ncol=p)
pc<-adalasso.net(X,k=5)
```

---

Beta2parcor

*Computation of partial correlation coefficients*

---

**Description**

This function computes the matrix of partial correlation coefficients based on the results of the corresponding regression models.

**Usage**

```
Beta2parcor(Beta, verbose=FALSE)
```

**Arguments**

Beta	matrix of regression coefficients
verbose	print information on conflicting signs etc. Default is verbose=FALSE.

**Details**

A well-known result (Whittaker, 1990) shows that the matrix of partial correlation coefficients can be estimated by computing a least squares regression model for each variable. If there are more variables than observations, the least squares problem is ill-posed and needs regularization. The matrix Beta stores the regression coefficients of any user-defined regression method. The function Beta2parcor computes the corresponding matrix of partial correlations.

**Value**

matrix of partial correlation coefficients

**Note**

This is an internal function.

**Author(s)**

Nicole Kraemer

**References**

J. Whittaker (1990) "Graphical models in applied multivariate statistics", Wiley, New York.  
N. Kraemer, J. Schaefer, A.-L. Boulesteix (2009) "Regularized Estimation of Large-Scale Gene Regulatory Networks with Gaussian Graphical Models", BMC Bioinformatics, 10:384  
<http://www.biomedcentral.com/1471-2105/10/384/>

**See Also**

[ridge.net](#), [adalasso.net](#), [pls.net](#)

**Examples**

```
# this is an internal function and should not be called by the user
```

---

lm.ridge.univariate *Ridge Regression for a single predictor variable*

---

### Description

This function computes the ridge regression coefficients for a model with a single predictor variable.

### Usage

```
lm.ridge.univariate(x,y,lambda=0,scale=TRUE)
```

### Arguments

x	vector of predictor observations.
y	vector of responses. The length of y must equal the length of x
lambda	vector of penalty terms. Default is lambda=0
scale	Scale x and y? Default is scale=TRUE.

### Details

This function replaces the R function `lm.ridge` if only one predictor variable is used, as the latter function produces an error message in this case.

### Value

A matrix with two columns and `length(lambda)` rows. In each row, the intercept (first column) and the regression coefficient (second column) is stored.

### Author(s)

Nicole Kraemer

### See Also

[ridge.net](#), [ridge.cv](#)

### Examples

```
n<-100 # number of observations
x<-rnorm(100)
y<-rnorm(n)
ridge.object<-lm.ridge.univariate(x,y,lambda=1:10)
```

---

mylars

*Cross-validation for Lasso*


---

## Description

This function computes the cross-validation-optimal regression coefficients for lasso.

## Usage

```
mylars(X, y, k = 10, use.Gram=TRUE, normalize=TRUE, intercept=TRUE)
```

## Arguments

X	matrix of observations. The rows of X contain the samples, the columns of X contain the observed variables
y	vector of responses. The length of y must equal the number of rows of X
k	the number of splits in k-fold cross-validation. Default is k=10.
use.Gram	When the number of variables is very large, you may not want LARS to pre-compute the Gram matrix. Default is use.Gram=TRUE.
normalize	Should the columns of X be scaled? Default is normalize=TRUE.
intercept	Should an intercept be included? Default is intercept=TRUE.

## Details

We use the `glmnet()` function from the `glmnet` package to compute the fit. Note that in Kraemer et. al. (2009), we used the `lars()` function from the `lars` package, which is much slower than `glmnet()`.

## Value

lambda	vector of parameter values from which the optimal parameter is selected
cv	cross-validated error for all lambda values
lambda.opt	cross-validation optimal parameter
cv.lasso	cv error for the optimal model.
intercept	cross-validation optimal intercept. If <code>intercept=FALSE</code> was specified, the intercept is set to 0.
coefficients	cross-validation optimal regression coefficients, without intercept

## Author(s)

Nicole Kraemer



## References

R. Tibshirani (1997) "Regression Shrinkage and Selection via the Lasso", Journal of the Royal Statistical Society B, 58 (1)

N. Kraemer, J. Schaefer, A.-L. Boulesteix (2009) "Regularized Estimation of Large-Scale Gene Regulatory Networks with Gaussian Graphical Models", BMC Bioinformatics, 10:384

<http://www.biomedcentral.com/1471-2105/10/384/>

## See Also

[Beta2parcor](#), [adalasso](#)

## Examples

```
n<-20
p<-50
X<-matrix(rnorm(n*p),ncol=p)
y<-rnorm(n)
dummy<-mylars(X,y)
```

---

performance.pcor

*Quality of estimated partial correlations*

---

## Description

This function computed various performance measures of the estimated matrix of partial correlations.

## Usage

```
performance.pcor(inferred.pcor, true.pcor=NULL,
fdr=TRUE, cutoff.ggm=0.8,verbose=FALSE,plot.it=FALSE)
```

## Arguments

inferred.pcor	matrix of estimated partial correlations
true.pcor	true matrix of partial correlations. Default is true.pcor=NULL
fdr	logical. If fdr=TRUE, the entries of inferred.pcor are tested for significance. Default is fdr=TRUE
cutoff.ggm	default cutoff for significant partial correlations. Default is cutoff.ggm=0.8
verbose	Print information on test results etc.. Default is verbose=FALSE
plot.it	Plot test results and ROC-curves. Default is plot.it=FALSE

**Details**

This function computes a range of performance measures: The function always returns the number of selected edges, the binary matrix that encodes the edges, the connectivity and the percentage of positive correlations. If `true.pcor` is provided, the function also returns the power (= true positive rate), the false positive rate and the positive predictive value. For non-sparse estimates that involve testing (i.e. `fdr=TRUE`) the function also returns the area under the curve, and a pair of vectors of false and true positive rates. The latter can e.g. be used to plot a ROC-curve.

**Value**

<code>num.selected</code>	number of selected edges
<code>adj</code>	binary matrix that encodes the existence of an edge between two nodes.
<code>connectivity</code>	vector of length <code>ncol(inferred.pcor)</code> . Its <i>ith</i> entry indicated the number of nodes that are connected to the <i>ith</i> node.
<code>positive.cor</code>	percentage of positive partial correlations out of all selected edges.
<code>power</code>	power (if <code>true.pcor</code> is provided)
<code>ppv</code>	positive predictive value (if <code>true.pcor</code> is provided)
<code>tpr</code>	true positive rate (=power) (if <code>true.pcor</code> is provided)
<code>fpr</code>	true positive rate (=power) (if <code>true.pcor</code> is provided)
<code>auc</code>	area under the curve (if <code>true.pcor</code> is provided and <code>fdr=TRUE</code> )
<code>TPR</code>	vector of true positive rates corresponding to varying cut-offs (if <code>true.pcor</code> is provided and <code>fdr=TRUE</code> )
<code>FPR</code>	vector of false positive rates corresponding to varying cut-offs (if <code>true.pcor</code> is provided and <code>fdr=TRUE</code> )

**Author(s)**

Juliane Schaefer, Nicole Kraemer

**References**

N. Kraemer, J. Schaefer, A.-L. Boulesteix (2009) "Regularized Estimation of Large-Scale Gene Regulatory Networks using Gaussian Graphical Models", BMC Bioinformatics, 10:384

<http://www.biomedcentral.com/1471-2105/10/384/>

**Description**

This function computes the matrix of partial correlations via an estimation of the corresponding regression models via Partial Least Squares.

**Usage**

```
pls.net(X, scale = TRUE, k = 10, ncomp = 15, verbose=FALSE)
```

**Arguments**

X	matrix of observations. The rows of X contain the samples, the columns of X contain the observed variables.
scale	Scale the columns of X? Default is scale=TRUE.
k	Number of splits in k-fold cross-validation. Default value is k=10.
ncomp	Maximal number of components. Default is 15.
verbose	Print information on conflicting signs etc. Default is verbose=FALSE

**Details**

For each of the columns of X, a regression model based on Partial Least Squares is computed. The optimal model is determined via cross-validation. The results of the regression models are transformed via the function `Beta2parcor`.

**Value**

pcor	estimated matrix of partial correlation coefficients.
m	optimal number of components for each of the <code>ncol(X)</code> regression models.

**Author(s)**

Nicole Kraemer

**References**

N. Kraemer, J. Schaefer, A.-L. Boulesteix (2009) "Regularized Estimation of Large-Scale Gene Regulatory Networks using Gaussian Graphical Models", *BMC Bioinformatics*, 10:384

<http://www.biomedcentral.com/1471-2105/10/384/>

**Examples**

```
n<-20
p<-40
X<-matrix(rnorm(n*p),ncol=p)
pc<-pls.net(X,ncomp=10,k=5)
```

---

`ridge.cv`*Ridge Regression.*

---

**Description**

This function computes the optimal ridge regression model based on cross-validation.

**Usage**

```
ridge.cv(X, y, lambda, scale = TRUE, k = 10, plot.it = FALSE)
```

**Arguments**

<code>X</code>	matrix of input observations. The rows of <code>X</code> contain the samples, the columns of <code>X</code> contain the observed variables
<code>y</code>	vector of responses. The length of <code>y</code> must equal the number of rows of <code>X</code>
<code>lambda</code>	Vector of penalty terms.
<code>scale</code>	Scale the columns of <code>X</code> ? Default is <code>scale=TRUE</code> .
<code>k</code>	Number of splits in <code>k</code> -fold cross-validation. Default value is <code>k=10</code> .
<code>plot.it</code>	Plot the cross-validation error as a function of <code>lambda</code> ? Default is <code>FALSE</code> .

**Value**

<code>intercept</code>	cross-validation optimal intercept
<code>coefficients</code>	cross-validation optimal regression coefficients
<code>lambda.opt</code>	optimal value of <code>lambda</code> .

**Author(s)**

Nicole Kraemer

**See Also**

[ridge.net](#)

**Examples**

```
n<-100 # number of observations
p<-60 # number of variables
X<-matrix(rnorm(n*p),ncol=p)
y<-rnorm(n)
ridge.object<-ridge.cv(X,y)
```

---

`ridge.net`*Partial correlations with ridge regression.*

---

**Description**

This function computes the matrix of partial correlations via an estimation of the corresponding regression models via Ridge Regression.

**Usage**

```
ridge.net(X, lambda, plot.it = FALSE, scale = TRUE, k = 10, verbose=FALSE)
```

**Arguments**

<code>X</code>	matrix of observations. The rows of <code>X</code> contain the samples, the columns of <code>X</code> contain the observed variables.
<code>lambda</code>	Vector of penalty terms.
<code>scale</code>	Scale the columns of <code>X</code> ? Default is <code>scale=TRUE</code> .
<code>k</code>	Number of splits in <code>k</code> -fold cross-validation. Default value is <code>k=10</code> .
<code>plot.it</code>	Plot the cross-validation error as a function of <code>lambda</code> ? Default is <code>FALSE</code> .
<code>verbose</code>	Print information on conflicting signs etc. Default is <code>verbose=FALSE</code>

**Value**

<code>pcor</code>	estimated matrix of partial correlations.
<code>lambda.opt</code>	optimal value of <code>lambda</code> for each of the <code>ncol</code> regression models.

**Author(s)**

Nicole Kraemer

**References**

N. Kraemer, J. Schaefer, A.-L. Boulesteix (2009) "Regularized Estimation of Large-Scale Gene Regulatory Networks using Gaussian Graphical Models", BMC Bioinformatics, 10:384  
<http://www.biomedcentral.com/1471-2105/10/384/>

**See Also**

[ridge.cv](#)

**Examples**

```
n<-20
p<-40
X<-matrix(rnorm(n*p),ncol=p)
pc<-ridge.net(X,k=5)
```

---

`sym2vec`*Transform symmetric matrix to vector*

---

**Description**

This function transforms the upper triangle (without diagonal) of a matrix A into a vector.

**Usage**

```
sym2vec(A)
```

**Arguments**

A                   quadratic matrix of size  $p \times p$

**Details**

The entries of the matrix A are stacked line-by-line.

**Value**

vector of length  $p(p-1)/2$

**Note**

This is an internal function.

**Author(s)**

Nicole Kraemer

**Examples**

```
p<-7  
A<-matrix(rnorm(p*p),ncol=p)  
v<-sym2vec(A)
```

# Index

\*Topic **multivariate**

- [adalasso](#), [2](#)
- [adalasso.net](#), [4](#)
- [Beta2parcor](#), [5](#)
- [mylars](#), [8](#)
- [performance.pcor](#), [9](#)
- [pls.net](#), [10](#)
- [ridge.cv](#), [12](#)
- [ridge.net](#), [13](#)
- [sym2vec](#), [14](#)

\*Topic **package**

- [parcor-package](#), [2](#)

\*Topic **univariate**

- [lm.ridge.univariate](#), [7](#)

[adalasso](#), [2](#), [5](#), [9](#)

[adalasso.net](#), [4](#), [4](#), [6](#)

[Beta2parcor](#), [4](#), [5](#), [5](#), [9](#)

[lm.ridge.univariate](#), [7](#)

[mylars](#), [8](#)

[parcor \(parcor-package\)](#), [2](#)

[parcor-package](#), [2](#)

[performance.pcor](#), [9](#)

[pls.net](#), [6](#), [10](#)

[ridge.cv](#), [7](#), [12](#), [13](#)

[ridge.net](#), [6](#), [7](#), [12](#), [13](#)

[sym2vec](#), [14](#)