Package ‘elasticnet’

May 15, 2020

Version 1.3

Date 2020-05-15

Title Elastic-Net for Sparse Estimation and Sparse PCA

Author Hui Zou <zouxx019@umn.edu> and Trevor Hastie
<hastie@stanford.edu>

Maintainer Hui Zou <zouxx019@umn.edu>

Depends R (>= 2.10), lars

Description Provides functions for fitting the entire
solution path of the Elastic-Net and also provides functions
for doing sparse PCA.

URL http://users.stat.umn.edu/~zouxx019/

NeedsCompilation no

Repository CRAN

Date/Publication 2020-05-15 17:10:02 UTC

R topics documented:

`arrayspc` *Sparse PCs of Microarrays*

Description

Sparse PC by iterative SVD and soft-thresholding

Usage

```
arrayspc(x,K=1,para,use.corr=FALSE, max.iter=200,trace=FALSE,eps=1e-3)
```

Arguments

`x`	The microarray matrix.
`K`	Number of components. Default is 1.
`para`	The thresholding parameters. A vector of length K.
`use.corr`	Perform PCA on the correlation matrix? This option is only effective when the argument type is set "data".
`max.iter`	Maximum number of iterations.
`trace`	If TRUE, prints out its progress.
`eps`	Convergence criterion.

Details

The function is equivalent to a special case of spca() with the quadratic penalty=infinity. It is specifically designed for the case p»n, like microarrays.

Value

A "arrayspc" object is returned.

Author(s)

Hui Zou and Trevor Hastie

References

Zou, H., Hastie, T. and Tibshirani, R. (2006) "Sparse principal component analysis" *Journal of Computational and Graphical Statistics*, 15 (2), 265–286.

See Also

spca, princomp

cv.enet *Computes K-fold cross-validated error curve for elastic net*

Description

Computes the K-fold cross-validated mean squared prediction error for elastic net.

Usage

```
cv.enet(x, y, K = 10, lambda, s, mode,trace = FALSE, plot.it = TRUE, se = TRUE, ...)
```

Arguments

`x`	Input to lars
`y`	Input to lars
`K`	Number of folds
`lambda`	Quadratic penalty parameter
`s`	Abscissa values at which CV curve should be computed. A value, or vector of values, indexing the path. Its values depends on the mode= argument
`mode`	Mode="step" means the s= argument indexes the LARS-EN step number. If mode="fraction", then s should be a number between 0 and 1, and it refers to the ratio of the L1 norm of the coefficient vector, relative to the norm at the full LS solution. Mode="norm" means s refers to the L1 norm of the coefficient vector. Abbreviations allowed. If mode="norm", then s should be the L1 norm of the coefficient vector. If mode="penalty", then s should be the 1-norm penalty parameter.
`trace`	Show computations?
`plot.it`	Plot it?
`se`	Include standard error bands?
`...`	Additional arguments to `enet`

Value

Invisibly returns a list with components (which can be plotted using `plotCVLars`)

`fraction`	Values of s
`cv`	The CV curve at each value of fraction
`cv.error`	The standard error of the CV curve

Author(s)

Hui Zou and Trevor Hastie

References

Zou and Hastie (2005) "Regularization and Variable Selection via the Elastic Net" *Journal of the Royal Statistical Society, Series B,76,301-320.*

Examples

```
data(diabetes)
attach(diabetes)
## use the L1 fraction norm as the tuning parameter
cv.enet(x2,y,lambda=0.05,s=seq(0,1,length=100),mode="fraction",trace=TRUE,max.steps=80)
## use the number of steps as the tuning parameter
cv.enet(x2,y,lambda=0.05,s=1:50,mode="step")
detach(diabetes)
```

`diabetes` *Blood and other measurements in diabetics*

Description

The `diabetes` data frame has 442 rows and 3 columns. These are the data used in the Efron et al "Least Angle Regression" paper.

Format

This data frame contains the following columns:

x a matrix with 10 columns

y a numeric vector

x2 a matrix with 64 columns

Details

The x matrix has been standardized to have unit L2 norm in each column and zero mean. The matrix x2 consists of x plus certain interactions.

Source

http://www-stat.stanford.edu/~hastie/Papers/LARS/LeastAngle_2002.ps

References

Efron, Hastie, Johnstone and Tibshirani (2003) "Least Angle Regression" (with discussion) *Annals of Statistics*

enet *Fits Elastic Net regression models*

Description

Starting from zero, the LARS-EN algorithm provides the entire sequence of coefficients and fits.

Usage

```
enet(x, y, lambda, max.steps, normalize=TRUE, intercept=TRUE,
     trace = FALSE, eps = .Machine$double.eps)
```

Arguments

`x`	matrix of predictors
`y`	response
`lambda`	Quadratic penalty parameter. lambda=0 performs the Lasso fit.
`max.steps`	Limit the number of steps taken; the default is `50 * min(m,n-1)`, with m the number of variables, and n the number of samples. One can use this option to perform early stopping.
`trace`	If TRUE, prints out its progress
`normalize`	Standardize the predictors?
`intercept`	Center the predictors?
`eps`	An effective zero

Details

The Elastic Net methodology is described in detail in Zou and Hastie (2004). The LARS-EN algorithm computes the complete elastic net solution simultaneously for ALL values of the shrinkage parameter in the same computational cost as a least squares fit. The structure of enet() is based on lars() coded by Efron and Hastie. Some internel functions from the lars package are called. The user should install lars before using elasticnet functions.

Value

An "enet" object is returned, for which print, plot and predict methods exist.

Author(s)

Hui Zou and Trevor Hastie

References

Zou and Hastie (2005) "Regularization and Variable Selection via the Elastic Net" *Journal of the Royal Statistical Society, Series B, 67, 301-320.*

See Also

print, plot, and predict methods for enet

Examples

```
data(diabetes)
attach(diabetes)
##fit the lasso model (treated as a special case of the elastic net)
object1 <- enet(x,y,lambda=0)
plot(object1)
##fit the elastic net model with lambda=1.
object2 <- enet(x,y,lambda=1)
plot(object2)
##early stopping after 50 LARS-EN steps
object4 <- enet(x2,y,lambda=0.5,max.steps=50)
plot(object4)
detach(diabetes)
```

`pitprops` *Pitprops correlation data*

Description

The `pitprops` data is a correlation matrix that was calculated from 180 observations. There are 13 explanatory variables.

Usage

```
data(pitprops)
```

Details

Jeffers (1967) tried to interpret the first six PCs. This is a classical example showing the difficulty of interpreting principal components.

References

Jeffers, J. (1967) "Two case studies in the application of principal component", *Applied Statistics*, 16, 225-236.

plot.enet *Plot method for enet objects*

Description

Produce a plot of an enet fit. The default is a complete coefficient path.

Usage

```
## S3 method for class 'enet'
plot(x, xvar = c("fraction", "penalty", "L1norm", "step"),
use.color = FALSE, ...)
```

Arguments

`x`	enet object
`xvar`	The type of x variable against which to plot. `xvar=fraction` plots agains the fraction of the L1 norm of the coefficient vector (default). `xvar=penalty` plots against the 1-norm penalty parameter. `xvar=L1norm` plots against the L1 norm of the coefficient vector. `xvar=step` plots against the LARS-EN step number.
`use.color`	a colorful plot?
`...`	Additonal arguments for generic plot.

Value

NULL

Author(s)

Hui Zou and Trevor Hastie

References

Zou and Hastie (2005) "Regularization and Variable Selection via the Elastic Net" *Journal of the Royal Statistical Society, Series B,67,301-320.*

Examples

```
data(diabetes)
attach(diabetes)
object <- enet(x,y,lambda=1)
par(mfrow=c(2,2))
plot(object)
plot(object,xvar="step")
detach(diabetes)
```

predict.enet *Make predictions or extract coefficients from a fitted elastic net model*

Description

While enet() produces the entire path of solutions, predict.enet allows one to extract a prediction at a particular point along the path.

Usage

```
## S3 method for class 'enet'
predict(object, newx, s, type = c("fit", "coefficients"), mode =
c("step","fraction", "norm", "penalty"),naive=FALSE, ...)
```

Arguments

`object`	A fitted enet object
`newx`	If type="fit", then newx should be the x values at which the fit is required. If type="coefficients", then newx can be omitted.
`s`	a value, or vector of values, indexing the path. Its values depends on the mode= argument. By default (mode="step").
`type`	If type="fit", predict returns the fitted values. If type="coefficients", predict returns the coefficients. Abbreviations allowed.
`mode`	Mode="step" means the s= argument indexes the LARS-EN step number, and the coefficients will be returned corresponding to the values corresponding to step s. If mode="fraction", then s should be a number between 0 and 1, and it refers to the ratio of the L1 norm of the coefficient vector, relative to the norm at the full LS solution. Mode="norm" means s refers to the L1 norm of the coefficient vector. Abbreviations allowed. If mode="norm", then s should be the L1 norm of the coefficient vector. If mode="penalty", then s should be the 1-norm penalty parameter.
`naive`	IF naive is True, then the naive elastic net fit is returned.
`...`	Additonal arguments for generic print.

Details

Starting from zero, the LARS-EN algorithm provides the entire sequence of coefficients and fits.

Value

Either a vector/matrix of fitted values, or a vector/matrix of coefficients.

Author(s)

Hui Zou and Trevor Hastie

References

Zou and Hastie (2005) "Regularization and Variable Selection via the Elastic Net" *Journal of the Royal Statistical Society, Series B,67,301-320.*

See Also

print, plot, enet

Examples

```
data(diabetes)
attach(diabetes)
object <- enet(x,y,lambda=0.1)
### make predictions at the values in x, at each of the
### steps produced in object
fits <- predict.enet(object, x, type="fit")
### extract the coefficient vector with L1 norm=2000
coef2000 <- predict(object, s=2000, type="coef", mode="norm")
### extract the coefficient vector with L1 norm fraction=0.45
coef.45 <- predict(object, s=0.45, type="coef", mode="fraction")
detach(diabetes)
```

`print.arrayspc` *Print method for arrayspc objects*

Description

Print out an arrayspc fit.

Usage

```
## S3 method for class 'arrayspc'
print(x, ...)
```

Arguments

`x`	arrayspc object
`...`	Additonal arguments for generic print.

Value

NULL

Author(s)

Hui Zou and Trevor Hastie

References

Zou, H., Hastie, T. and Tibshirani, R. (2006) "Sparse principal component analysis" *Journal of Computational and Graphical Statistics*, 15 (2), 265–286.

`print.enet` *Print method for enet objects*

Description

Print out an enet fit.

Usage

```
## S3 method for class 'enet'
print(x, ...)
```

Arguments

`x`	enet object
`...`	Additonal arguments for generic print.

Value

NULL

Author(s)

Hui Zou and Trevor Hastie

References

Zou and Hastie (2005) "Regularization and Variable Selection via the Elastic Net" *Journal of the Royal Statistical Society, Series B,67,301-320*.

print.spca *Print method for spca objects*

Description

Print out a spca fit.

Usage

```
## S3 method for class 'spca'
print(x, ...)
```

Arguments

x	spca object
...	Additonal arguments for generic print.

Value

NULL

Author(s)

Hui Zou and Trevor Hastie

References

Zou, H., Hastie, T. and Tibshirani, R. (2006) "Sparse principal component analysis" *Journal of Computational and Graphical Statistics*, 15 (2), 265–286.

spca *Sparse Principal Components Analysis*

Description

Using an alternating minimization algorithm to minimize the SPCA criterion.

Usage

```
spca(x, K, para, type=c("predictor","Gram"),
     sparse=c("penalty","varnum"), use.corr=FALSE, lambda=1e-6,
     max.iter=200, trace=FALSE, eps.conv=1e-3)
```

Arguments

`x`	A matrix. It can be the predictor matrix or the sample covariance/correlation matrix.
`K`	Number of components
`para`	A vector of length K. All elements should be positive. If sparse="varnum", the elements integers.
`type`	If type="predictor", x is the predictor matrix. If type="Gram", the function asks the user to provide the sample covariance or correlation matrix.
`sparse`	If sparse="penalty", para is a vector of 1-norm penalty parameters. If sparse="varnum", para defines the number of sparse loadings to be obtained. This option is not discussed in the paper given below, but it is convenient in practice.
`lambda`	Quadratic penalty parameter. Default value is 1e-6.
`use.corr`	Perform PCA on the correlation matrix? This option is only effective when the argument type is set "data".
`max.iter`	Maximum number of iterations.
`trace`	If TRUE, prints out its progress.
`eps.conv`	Convergence criterion.

Details

PCA is shown to be equivalent to a regression-type optimization problem, then sparse loadings are obtained by imposing the 1-norm constraint on the regression coefficients. If x is a microarray matrix, use arrayspc().

Value

A "spca" object is returned. The below are some quantities which the user may be interested in:

`loadings`	The loadings of the sparse PCs
`pev`	Percentage of explained variance
`var.all`	Total variance of the predictors

Author(s)

Hui Zou and Trevor Hastie

References

Zou, H., Hastie, T. and Tibshirani, R. (2006) "Sparse principal component analysis" *Journal of Computational and Graphical Statistics*, 15 (2), 265–286.

See Also

princomp, arrayspc

Examples

```
data(pitprops)
out1<-spca(pitprops,K=6,type="Gram",sparse="penalty",trace=TRUE,para=c(0.06,0.16,0.1,0.5,0.5,0.5))
## print the object out1
out1
out2<-spca(pitprops,K=6,type="Gram",sparse="varnum",trace=TRUE,para=c(7,4,4,1,1,1))
out2
## to see the contents of out2
names(out2)
## to get the loadings
out2$loadings
```

Index