

Package ‘lassoscore’

February 20, 2015

Title High-Dimensional Inference with the Penalized Score Test

Description Use the lasso regression method to perform approximate inference in high dimensions, by penalizing the effects of nuisance parameters.

Version 0.6

Author Arie Voorman <arie.voorman@gmail.com>

Maintainer Arie Voorman <arie.voorman@gmail.com>

Depends R (>= 2.10), glasso, glmnet, Matrix

Suggests covTest, lars

License GPL (>= 2)

LazyData true

NeedsCompilation no

Repository CRAN

Date/Publication 2014-10-28 08:10:02

R topics documented:

diabetes	1
glassoscore	2
lassoscore	4
qqpval	6

Index	8
--------------	----------

diabetes	<i>Blood and other measurements in diabetics.</i>
----------	---

Description

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

Usage

```
data(diabetes)
```

Format

A data frame with 442 observations on the following 3 variables.

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 variance 1 in each column and zero mean. The matrix x2 consists of x plus certain interactions.

Source

Data can be found in the ‘lars’ package.

References

Voorman, A, Shojaie, A, and Witten D. Inference in high-dimensions with the penalized score test. In preparation.

See Also

[lassoscore](#)

Examples

```
data(diabetes)
mod <- with(diabetes,lassoscore(y,x,lambda=0.02))
```

glassoscore

Penalized score test, for the graphical lasso.

Description

Test whether an element of the precision matrix is zero, using the graphical lasso to approximate the parameters in remainder of the precision matrix.

Usage

```
glassoscore(x, lambda, subset=NULL, penalize.diagonal=FALSE, tol=1e-8)
mboscore(x, lambda, subset=NULL, tol=1e-8,...)
```

Arguments

<code>x</code>	data matrix. Unlike <code>glasso</code> , this function requires the original data, not just the covariance matrix.
<code>lambda</code>	a non-negative tuning parameter
<code>subset</code>	An <code>ncol(x)</code> by <code>ncol(x)</code> logical matrix, giving a subset of edges to test.
<code>penalize.diagonal</code>	logical. Whether or not to penalize the diagonal in the graphical lasso. Defaults to <code>FALSE</code> .
<code>tol</code>	convergence tolerance for <code>glasso</code> or <code>glmnet</code>
<code>...</code>	for <code>mboscore</code> , additional arguments to be passed to <code>lassoscore</code>

Details

This function tests for pairwise association between features, using the graphical lasso (`glassoscore`) or neighborhood selection (`mboscore`). Tests are based on the penalized score statistic $T[\lambda]$, described in Voorman *et al* (2014). Note that a feature is non-zero in the (graphical) lasso solution if and only if

$$|T[\lambda]| > \lambda\sqrt{n},$$

where $T[\lambda]$ is penalized the score statistic.

Calculating the variance of $T[\lambda]$ can be computationally expensive for `glassoscore`. If there are q non-zero parameters in the graphical lasso solution, it will (roughly) require construction, and inversion, of a $q \times q$ matrix for each of the q non-zero parameters. That is, complexity is roughly q^4 .

For `mboscore`, the results are typically not symmetric. For instance, `p.sand[-i,i]` contains the p-values produced by `lassoscore(x[,i], x[,-i], lambda)`, i.e. using `x[,i]` as the outcome variable, and thus `p.sand[i,-i]` contains p-values associated with feature `i` when used as the a predictor variable.

Value

for an object of class either ‘`glassoscore`’ or ‘`mboscore`’, containing

<code>scores</code>	the penalized score statistics
<code>scorevar.model</code>	the variance of the score statistics, estimated using a model-based variance estimate
<code>scorevar.sand</code>	the variance of the score statistics, using a conservative variance estimate
<code>p.model</code>	p-value, using the model-based variance
<code>p.sand</code>	p-value, using the sandwich variance
<code>beta</code>	for <code>mboscore</code> , the <code>beta[-i,i]</code> contains the coefficients from lasso regression of <code>x[,i]</code> on <code>x[,-i]</code> .

In addition, `glassoscore` contains the output from ‘`glasso`’ applied to `x`.

Author(s)

Arend Voorman

References

Jerome Friedman, Trevor Hastie and Robert Tibshirani (2007). Sparse inverse covariance estimation with the lasso. *Biostatistics* 2007. <http://www-stat.stanford.edu/~tibs/ftp/graph.pdf>

N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the lasso. *Annals of Statistics*, 34(3):1436-1462, 2006.

See Also

[lassoscore](#), [glasso](#)

Examples

```
set.seed(100)

x<-matrix(rnorm(50*20),ncol=20)

gl <- glassoscore(x,0.2)
mb <- mbscore(x,0.2)

par(mfrow=c(1,2))
plot(gl)
plot(mb)
```

lassoscore

Lasso penalized score test

Description

Test for the association between y and each column of X , adjusted for the other columns using a lasso regression, as described in Voorman et al (2014).

Usage

```
lassoscore(y,X, lambda=0, family=c("gaussian","binomial","poisson"),
  tol = .Machine$double.eps, maxit=1000,
  resvar = NULL, verbose=FALSE, subset = NULL)
```

Arguments

<code>y</code>	outcome variable
<code>X</code>	matrix of predictors
<code>lambda</code>	tuning parameter value (see details)
<code>family</code>	The family, for the likelihood.
<code>tol,maxit</code>	convergence tolerance and maximum number of iterations in glmnet
<code>resvar</code>	value for the residual variance, for "gaussian" family. If not specified, the residual variance from lasso regression on all features is used (see details).
<code>verbose</code>	whether or not to print progress bars (defaults to FALSE)
<code>subset</code>	a subset of columns to test

Details

For each column of X , denoted by x^* , this function computes the score statistic

$$T[\lambda] = x^*{}^T (y - \hat{y})/\sqrt{n}$$

where \hat{y} are the fitted values from lasso regression of y on $X[-,x^*]$ (see Note 2).

The variance of the score statistic is estimated in 4 ways:

- (i) a model-based estimate
- (ii) a sandwich variance
- (iii/iv) conservative versions of (i) and (ii), which do not depend on the selected model

Note 1: in lasso regression of y on X , the coefficient of x^* is non-zero **if and only if**

$$|T[\lambda]| > \lambda\sqrt{n}$$

Note 2: For lasso regression of y on X , we minimize $-l(b) + \lambda\|b\|_1$ over vectors b , where $l(b)$ is either $RSS/(2n)$ (for the "gaussian" family), or the log-likelihood for a generalized linear model. See the details of [glmnet](#) for more information.

Note 3: Each feature x is rescaled to have mean zero and $x^T x/n = 1$, y is centered, but not rescaled.

Value

Object of class 'lassoscore', which is an R 'list', with elements:

<code>fit</code>	Elements of the fitted lasso regression of y on X (see glmnet for details.)
<code>scores</code>	the score statistics
<code>resvar</code>	the value used for the residual variance
<code>scorevar.model</code>	the variance of the score statistics, estimated using a model-based approximation
<code>scorevar.sand</code>	the variance of the score statistics, using an model-agnostic, or sandwich formula
<code>scorevar.model.cons</code> , <code>scorevar.sand.cons</code>	conservative versions of the variances
<code>p.model</code>	p-value, using a model-based variance
<code>p.sand</code>	p-value, using sandwich variance
<code>p.model.cons</code> , <code>p.sand.cons</code>	p-value, using conservative variance formulas

Author(s)

Arend Voorman <voorma@uw.edu>

References

Voorman, A, Shojaie, A, and Witten D. Inference in high dimensions with the penalized score test. <http://arxiv.org/abs/1401.2678>.

See Also

[glassoscore](#), [qqpval](#)

Examples

```
#Simulation from Voorman et al (2014)
set.seed(20)
n <- 300
p <- 100
q <- 10

set.seed(20)
beta <- numeric(p)
beta[sample(p,q)] <- 0.4

Sigma <- forceSymmetric(t(0.5^outer(1:p,1:p,"-")))
cSigma <- chol(Sigma)

x <- scale(replicate(p,rnorm(n))%*%cSigma)
y <- rnorm(n,x%*%beta,1)

mod <- lassoscore(y,x,0.02)
summary(mod)
plot(mod,type="all")

#test only features 10:20:
mod0 <- lassoscore(y,x,0.02, subset = 10:20)

##### Diabetes data set:
#Test features in the diabetes data set, using 2 different values of `lambda`,
#and compare results:
resvar <- with(lm(y~x,data=diabetes), sum(residuals^2)/df.residual)

mod2 <- with(diabetes,lassoscore(y,x,lambda=4,resvar=resvar))
mod3 <- with(diabetes,lassoscore(y,x,lambda=0.5,resvar=resvar))
data.frame(
  "variable"=colnames(diabetes$x),
  "lambda_4"=format(mod2$p.model,digits=2),
  "lambda_0.5"=format(mod3$p.model,digits=2))
```

qqpval

make a QQ plot of p-values.

Description

This function makes QQ-plots for p-values, by default on a $-\log_{10}$ scale. It also shows pointwise-95% confidence bounds for the order statistics of a Uniform(0,1) distribution.

Usage

```
qqpval(p, cone = TRUE, log = TRUE, add = FALSE, col=1, pch=1,...)
```

Arguments

<code>p</code>	p-values
<code>cone</code>	Logical. Whether or not to print 95% confidence bounds for Uniform(0,1) order statistics
<code>log</code>	Logical. Whether or not to plot p-values on -log10 scale
<code>add</code>	logical. whether or not to add to an existing plot
<code>...</code>	other options to be passed to 'plot'
<code>col,pch</code>	color and point type. See plot

Author(s)

Arend Voorman

Examples

```
p <- runif(1000)
qqpval(p)
```

Index

*Topic **datasets**

diabetes, [1](#)

diabetes, [1](#)

glasso, [3](#), [4](#)

glassoscore, [2](#), [6](#)

glmnet, [3-5](#)

lassoscore, [2-4](#), [4](#)

lassoscore-package (lassoscore), [4](#)

mbscore (glassoscore), [2](#)

plot, [7](#)

qqpval, [6](#), [6](#)