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 Blood and other measurements in diabetics.

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. nference 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)
mbscore(x, lambda, subset=NULL, tol=1e-8,...)
Arguments

x data matrix. Unlike glasso, this function requires the original data, not just the covariance matrix.

lambda a non-negative tuning parameter

subset An ncol(x) by ncol(x) logical matrix, giving a subset of edges to test.

penalize.diagonal logical. Whether or not to penalize the diagonal in the graphical lasso. Defaults to FALSE.

tol convergence tolerance for glasso or glmnet

... for mbscore, additional arguments to be passed to lassoscore

Details

This function tests for pairwise association between features, using the graphical lasso (glassoscore) or neighborhood selection (mbscore). 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 mbscore, 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 ‘mbscore’, containing

scores the penalized score statistics

scorevar.model the variance of the score statistics, estimated using a model-based variance estimate

scorevar.sand the variance of the score statistics, using a conservative variance estimate

p.model p-value, using the model-based variance

p.sand p-value, using the sandwich variance

beta for mbscore, the beta[-i,i] contains the coefficients from lasso regression of x[,i] on x[,-i].

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

Author(s)

Arend Voorman
References

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
y  outcome variable
X  matrix of predictors
lambda  tuning parameter value (see details)
family  The family, for the likelihood.
tol,maxit  convergence tolerance and maximum number of iterations in glmnet
resvar  value for the residual variance, for "gaussian" family. If not specified, the residual variance from lasso regression on all features is used (see details).
verbose  whether or not to print progress bars (defaults to FALSE)
subset  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 $\text{RSS}/(2n)$ (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 $x^Tx/n = 1$, $y$ is centered, but not rescaled.

**Value**

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

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

**Author(s)**

Arend Voorman <voorma@uw.edu>

**References**

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 -log10 scale. It also shows pointwise-95% confidence bounds for the order statistics of a Uniform(0,1) distribution.
qqpval

Usage

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

Arguments

p         p-values
cone      Logical. Whether or not to print 95% confidence bounds for Uniform(0,1) order statistics
log       Logical. Whether or not to plot p-values on -log10 scale
add       logical. whether or not to add to an existing plot
...       other options to be passed to ‘plot’
col, pch   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