Package ‘islasso’

February 25, 2020

Type Package
Title The Induced Smoothed Lasso
Version 1.1.1
Date 2020-02-20
Author Gianluca Sottile [aut, cre],
    Giovanna Cilluffo [aut, ctb],
    Vito MR Muggeo [aut, cre]
Maintainer Gianluca Sottile <gianluca.sottile@unipa.it>
Depends glmnet, Matrix, R (>= 2.10)
License GPL (>= 2)
URL https://journals.sagepub.com/doi/abs/10.1177/0962280219842890
LazyLoad yes
NeedsCompilation yes
Repository CRAN
RoxygenNote 6.1.1
Date/Publication 2020-02-25 13:40:02 UTC

R topics documented:

islasso-package .......................................................... 2
aic.islasso ................................................................. 3
anova.islasso ............................................................. 5


islasso-package

The Induced Smoothed lasso: A practical framework for hypothesis testing in high dimensional regression

Description

This package implements an induced smoothed approach for hypothesis testing in lasso regression.

Details

Package: islasso
Type: Package
Version: 1.1.1
Date: 2020-02-20
License: GPL-2

islasso is used to fit generalized linear models with a L1-penalty on (some) regression coefficients. Along with point estimates, the main advantage is to return the full covariance matrix of estimate. The resulting standard errors can be used to make inference in the lasso framework. The main function is islasso and the corresponding fitter function islasso.fit, and many auxiliary functions are implemented to summarize and visualize results: summary.islasso, predict.islasso, coef.islasso, fitted.islasso, logLik.islasso, AIC.islasso, deviance.islasso, residuals.islasso.

Author(s)

Gianluca Sottile based on some preliminary functions by Vito Muggeo.

Maintainer: Gianluca Sottile <gianluca.sottile@unipa.it>

References


Examples

```r
set.seed(1)
n <- 100
p <- 100
p1 <- 20  # number of nonzero coefficients
coef.veri <- sort(round(c(seq(.5, 3, l=p1/2), seq(-1, -2, l=p1/2)), 2))
sigma <- 1

coef <- c(coef.veri, rep(0, p-p1))

X <- matrix(rnorm(n*p), n, p)
mu <- drop(X%*%coef)
y <- mu + rnorm(n, 0, sigma)

lambda <- 2
o <- islasso(y~-1+X, family=gaussian, lambda=lambda)
o
summary(o)
```

---

**Description**

This function performs a minimization of the AIC/BIC criterion for selecting the tuning parameter in "islasso".

**Usage**

```r
aic.islasso(object, method = c("aic", "bic"), interval, y, X, intercept = FALSE, family = gaussian(), alpha = 1, offset, weights, unpenalized, control = is.control())
```

**Arguments**

- `object` a fitted model object of class "islasso".
- `method` the criterion to optimize, AIC or BIC.
- `interval` the lower and upper limits of λ wherein the AIC/BIC criterion should be optimized. Can be missing, if object has been obtained via cross-validation (and therefore includes the range of lambdas).
- `y` if object is missing, the response vector of length n.
- `X` if object is missing, the design matrix of dimension n * p.
intercept if object is missing, if TRUE the intercept is added to the model matrix.

family if object is missing, a description of the error distribution, family=gaussian, family=binomial and family=poisson are implemented with canonical link.

alpha The elasticnet mixing parameter, with $0 \leq \alpha \leq 1$. The penalty is defined as

$$(1 - \alpha)/2||\beta||_2^2 + \alpha||\beta||_1.$$  

alpha=1 is the lasso penalty, and alpha=0 the ridge penalty.

offset this can be used to specify an a priori known component to be included in the linear predictor during fitting. This should be NULL or a numeric vector of length equal to the number of cases.

weights observation weights. Default is 1 for each observation.

unpenalized a vector used to specify the unpenalized estimators; unpenalized has to be a vector of logicals.

control a list of parameters for controlling the fitting process (see islasso.control for more details).

Details

Minimization of the Akaike Information Criterion (AIC) or Bayesian Information Criterion (BIC) is sometimes employed to select the tuning parameter as an alternative to the cross validation. The model degrees of freedom (not necessarily integers as in the plain lasso) used in AIC/BIC are computed as trace of the hat matrix at convergence.

Value

the optimal lambda value is returned

Author(s)

Gianluca Sottile
Maintainer: Gianluca Sottile <gianluca.sottile@unipa.it>

See Also

islasso.fit, coef.islasso, summary.islasso, residuals.islasso, AIC.islasso, logLik.islasso, fitted.islasso, predict.islasso and deviance.islasso methods.

Examples

```r
set.seed(1)
n <- 100
p <- 100
pl <- 20 #number of nonzero coefficients
coef.veri <- sort(round(c(seq(.5, 3, l=pl/2), seq(-1, -2, l=pl/2)), 2))
sigma <- 1
coeff <- c(coef.veri, rep(0, p-pl))
```
X <- matrix(rnorm(n*p), n, p)
mu <- drop(X%*%coef)
y <- mu + rnorm(n, 0, sigma)
o <- islasso(y ~ X)

## Not run:
#use the evaluation interval of the fit
lambda_aic <- aic.islasso(o, method="aic")

#overwrites the evaluation interval for lambda
lambda_bic <- aic.islasso(o, interval=c(.1, 30), method="bic")

## End(Not run)

---

**anova.islasso**  
*General Linear Hypotheses method for islasso objects*

**Description**

General linear hypotheses for linear combinations of the regression coefficients in islasso fits

**Usage**

```r
## S3 method for class 'islasso'
anova(object, A, b, ...)
```

**Arguments**

- **object**: a fitted model object of class "islasso".
- **A**: a specification of the linear hypotheses to be tested. Linear functions can be specified by either a single vector (of length p) or by a matrix (of dimension k x p) of one or more linear hypotheses.
- **b**: an optional numeric vector specifying the right hand side of the hypothesis. Can be a scalar.
- **...**: not used.

**Details**

For the islasso regression model with coefficients $\beta$, the null hypothesis is

$$H_0 : A\beta = b$$

where $A$ and $b$ are known matrix and vector. $A$ can be a vector and $b$ can be a scalar.
Examples

```r
## Not run:
set.seed(1)
N <- 100
p <- 100
p1 <- 10  # number of nonzero coefficients
coeff.true <- sort(round(c(seq(.5, 3, l=p1/2), seq(-1, -2, l=p1/2)), 2))
sigma <- 1

coeff <- c(coeff.true, rep(0, p-p1))
X <- matrix(rnorm(N*p), n, p)
eta <- drop(X %*% coeff)
mu <- eta
y <- mu + rnorm(N, 0, sigma)
o <- islasso(y ~ -1 + X, family = gaussian)
A <- rbind(rep(c(1,0), c(10, p-10)),
          rep(c(0,1), c(10, p-10)))
anova(o, A)
A <- cbind(diag(10), matrix(0, 10, p-10))
b <- coeff.true
anova(o, A, b)
## End(Not run)
```

breast

Breast Cancer microarray experiment

Description

This data set details microarray experiment for 52 breast cancer patients. The binary variable `status` is used to indicate whether or not the patient has died of breast cancer (`status = 0` = did not die of breast cancer, `status = 1` = died of breast cancer). The other variables contain the amplification or deletion of the considered genes.

Rather than measuring gene expression, this experiment aims to measure gene amplification or deletion, which refers to the number of copies of a particular DNA sequence within the genome. The aim of the experiment is to find out the key genomic factors involved in aggressive and non-aggressive forms of breast cancer.

The experiment was conducted by the Dr. John Bartlett and Dr. Caroline Witton in the Division of Cancer Sciences and Molecular Pathology of the University of Glasgow at the city’s Royal Infirmary.

Usage

data(breast)
**Source**

Dr. John Bartlett and Dr. Caroline Witton, Division of Cancer Sciences and Molecular Pathology, University of Glasgow, Glasgow Royal Infirmary.

**References**


---

**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**


**References**

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

Auxiliary for controlling islasso model fitting

Description

Auxiliary function for controlling the islasso model fitting.

Usage

is.control(sigma2 = -1, tol = 1e-04, itmax = 500, stand = TRUE,
trace = 0, nfolds = 5, seed=NULL, debias = FALSE, adaptive = FALSE,
b0 = NULL, V0 = NULL, c = -1)

Arguments

sigma2 optional. The fixed value of dispersion parameter. If -1 (default) it is estimated from the data
tol tollerance value to declare convergence, default to 1e-4
itmax maximum number of iterations, default to 500
stand if TRUE, the covariates are standardized prior to fitting the model. However the coefficients are always returned on the original scale
trace Should the iterative procedure be printed? 0: no printing, 1 = compact printing, 2 = enlarged printing.
nfolds if lambda is unspecified in islasso, the number of folds to be used to perform cross valdation. Default to 5, and nfolds>2 is allowed. Although nfolds can be as large as the sample size (leave-one-out CV), it is not recommended for large datasets. nfolds is ignored if lambda is supplied.
seed optional, the seed to be used to split the dataframe and to perform cross validation. Useful to make reproducible the results.
debias experimental, if TRUE, unbiased estimates are also returned.
adaptive experimental, if TRUE the adaptive LASSO is implemented.
b0 optional, starting values for the regression coefficients. If NULL, the point estimates from glmnet are used.
V0 optional, starting value for the estimates covariance matrix, If NULL, the identity matrix is used.
c the weight of the mixture in the induced smoothed lasso. c=-1 means to compute it at each step of the iterative algorithm.
The Induced Smoothed lasso

Description

islasso is used to fit lasso regression models wherein the nonsmooth $L_1$ norm penalty is replaced by a smooth approximation justified under the induced smoothing paradigm. Simple lasso-type or elastic-net penalties are permitted and Linear, Logistic, Poisson and Gamma responses are allowed.

Usage

```r
islasso(formula, family = gaussian, lambda, alpha=1, data, weights, subset,
offset, unpenalized, contrasts = NULL, control = is.lasso.control())
```

Arguments

- **formula**: an object of class “formula” (or one that can be coerced to that class): the ‘usual’ symbolic description of the model to be fitted.
- **family**: the assumed response distribution. Gaussian, (quasi) Binomial, (quasi) Poisson, and Gamma are allowed. `family=gaussian` is implemented with identity link, `family=binomial` is implemented with logit or probit links, `family=poisson` is implemented with log link, and `family=Gamma` is implemented with inverse, log and identity links.
- **lambda**: Value of the tuning parameter in the objective. If missing, the optimal lambda is computed using `cv.glmnet`.
- **alpha**: The elastic-net mixing parameter, with $0 \leq \alpha \leq 1$. The penalty is defined as
  \[(1 - \alpha)/2||\beta||^2 + \alpha||\beta||_1.\]
  
  `alpha=1` is the lasso penalty, and `alpha=0` the ridge penalty.
- **data**: an optional data frame, list or environment (or object coercible by `as.data.frame` to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which `islasso` is called.
- **weights**: observation weights. Default is 1 for each observation.
- **subset**: an optional vector specifying a subset of observations to be used in the fitting process.
- **offset**: this can be used to specify an a priori known component to be included in the linear predictor during fitting. This should be `NULL` or a numeric vector of length equal to the number of cases.
- **unpenalized**: optional. A vector of integers or characters indicating the covariate coefficients not to penalize. The intercept, if included in the model, is always unpenalized.
- **contrasts**: an optional list. See the `contrasts.arg` of `model.matrix.default`.
- **control**: a list of parameters for controlling the fitting process (see `islasso.control` for more details).
Details

islasso estimates regression models by imposing a lasso-type penalty on some or all regression coefficients. However the nonsmooth $L_1$ norm penalty is replaced by a smooth approximation justified under the induced smoothing paradigm. The advantage is that reliable standard errors are returned as model output and hypothesis testing on linear combinations of the regression parameters can be carried out straightforwardly via the Wald statistic. Simulation studies provide evidence that the proposed approach controls type-I errors and exhibits good power in different scenarios.

Value

A list of

- `coefficients` a named vector of coefficients
- `se` a named vector of standard errors
- `res` the working residuals
- `fitted.values` the fitted values
- `linear.predictors` the linear predictors
- `rank` the estimated degrees of freedom
- `family` the family object used
- `deviance` the family deviance
- `null.deviance` the family null deviance
- `aic` the Akaike Information Criterion
- `df.null` the degrees of freedom of a null model
- `phi` the estimated dispersion parameter
- `beta.unbias` unbiased coefficients
- `se.unbias` unbiased standard errors
- `internal` internal elements
- `control` the value of the control argument used
- `model` if requested (the default), the model frame used.
- `terms` the terms object used.
- `contrasts` (only where relevant) the contrasts used.
- `call` the matched call
- `formula` the formula supplied

Author(s)

The main function of the same name was inspired by the R function previously implemented by Vito MR Muggeo. Maintainer: Gianluca Sottile <gianluca.sottile@unipa.it>
References


See Also

islasso.fit, coef.islasso, summary.islasso, residuals.islasso, AIC.islasso, logLik.islasso, fitted.islasso, predict.islasso and deviance.islasso methods.

Examples

```r
set.seed(1)
n <- 100
p <- 100
p1 <- 10  # number of nonzero coefficients
coef.veri <- sort(round(c(seq(.5, 3, l=p1/2), seq(-1, -2, l=p1/2)), 2))
sigma <- 1
coe <- c(coef.veri, rep(0, p-p1))

X <- matrix(rnorm(n*p), n, p)
eta <- drop(X%*%coefficient)

##### gaussian ######
mu <- eta
y <- mu + rnorm(n, 0, sigma)
o <- islasso(y~-1+X, family=gaussian)

summary(o)
coef(o)
fitted(o)
predict(o, type="response")
plot(o)
residuals(o)
deviance(o)
AIC(o)
logLik(o)

## Not run:
# for the interaction
o <- islasso(y~-1+X[,1]*X[,2], family=gaussian)

##### binomial ######
coef <- c(c(1,1,1), rep(0, p-3))
X <- matrix(rnorm(n*p), n, p)
et <- drop(cbind(1, X)%*%c(1, coefficient))
```
mu <- binomial()$linkinv(eta)
y <- rbinom(n, 100, mu)
y <- cbind(y, 100-y)
o <- islasso(y~X, family=binomial)

##### poisson #####
coef <- c(c(1,1,1), rep(0, p-3))
X <- matrix(rnorm(n*p), n, p)
etta <- drop(cbind(1, X) %*% c(1, coef))
mu <- poisson()$linkinv(eta)
y <- rpois(n, mu)
o <- islasso(y~X, family=poisson)

##### Gamma #####
coef <- c(c(1,1,1), rep(0, p-3))
X <- matrix(rnorm(n*p), n, p)
etta <- drop(cbind(1, X) %*% c(1, coef))
mu <- Gamma(link="log")$linkinv(eta)
shape <- 10
phi <- 1 / shape
y <- rgamma(n, scale = mu / shape, shape = shape)
o <- islasso(y~X, family=Gamma(link="log"))

## End(Not run)

---

**plot.islasso**  
*Diagnostics plots for Induced Smoothing Lasso Model*

**Description**  
Diagnostics plots for Induced Smoothing Lasso Model

**Usage**  
```r
## S3 method for class 'islasso'
plot(x, ...)
```

**Arguments**  
x an object of class islasso, usually, a result of a call to `islasso`.
...
other graphical parameters for the plot
predict.islasso

Details

The plot on the top left is a plot of the standard deviance residuals against the fitted values. The plot on the top right is a normal QQ plot of the standardized deviance residuals. The red line is the expected line if the standardized residuals are normally distributed, i.e. it is the line with intercept 0 and slope 1. The bottom two panels are plots of link and variance functions. On the left is squared standardized Pearson residuals against the fitted values. On the right working vector against the linear predictor.

Examples

```r
## Not run:
set.seed(1)
n <- 100
p <- 100
p1 <- 20  # number of nonzero coefficients
coefficients <- sort(round(c(seq(.5, 3, l=p1/2), seq(-1, -2, l=p1/2)), 2))
sigma <- 1
coeff <- c(coefficients, rep(0, p-p1))
X <- matrix(rnorm(n*p), n, p)
mu <- drop(X*coeff)
y <- mu + rnorm(n, 0, sigma)
lambda <- 2
o <- islasso(y~X, family=gaussian, lambda=lambda)
plot(o)
## End(Not run)
```

---

predict.islasso  
*Prediction method for islasso fitted objects*

Description

Prediction method for islasso fitted objects

Usage

```r
# S3 method for class 'islasso'
predict(object, type = c("link", "response", "coefficients", "class"), newdata, ...)
```

Arguments

- **object** a fitted object of class "islasso"
type

the type of prediction required. The default is on the scale of the linear predictors; the alternative "response" is on the scale of the response variable. Thus for a default binomial model the default predictions are of log-odds (probabilities on logit scale) and type = "response" gives the predicted probabilities. The coefficients option returns coefficients. Type "class" applies only to "binomial" models, and produces the class label.

newdata

optionally, a data frame in which to look for variables with which to predict. If omitted, the fitted linear predictors are used.

... further arguments passed to or from other methods.

Value

An object depending on the type argument

Examples

```r
## Not run:
set.seed(1)
n <- 100
p <- 100
p1 <- 20  # number of nonzero coefficients
coefficients <- sort(round(c(seq(.5, 3, l=p1/2), seq(-1, -2, l=p1/2)), 2))
sigma <- 1

coefficients <- coefficients

X <- matrix(rnorm(n*p), n, p)
mu <- drop(X%*%coefficients)
y <- mu + rnorm(n, 0, sigma)
lambda <- 2
o <- islasso(y~X, family=gaussian, lambda=lambda)
predict(o)

### prediction of new dataset
newdata <- matrix(rnorm(n*p), n, p)
predict(o, type="response", newdata=newdata)
## End(Not run)
```

Prostate Cancer Data

Description

These data come from a study that examined the correlation between the level of prostate specific antigen and a number of clinical measures in men who were about to receive a radical prostatectomy. It is data frame with 97 rows and 9 columns.
Usage

data(Prostate)

Format

The data frame has the following components:

- lcavol log(cancer volume)
- lweight log(prostate weight)
- age age
- lbph log(benign prostatic hyperplasia amount)
- svi seminal vesicle invasion
- lcp log(capsular penetration)
- gleason Gleason score
- pgg45 percentage Gleason scores 4 or 5
- lpsa log(prostate specific antigen)

Source


---

**simulXy**

*Simulate model matrix and response*

Description

Simulate model matrix and response from a specified distribution.

Usage

```r
simulXy(n, p, interc = 0, beta, family = gaussian(), prop = 0.1, lim.b = c(-3, 3), sigma = 1, size = 1, rho = 0, scale = TRUE, seed, X)
```

Arguments

- `n` number of observations.
- `p` total number of covariates in the model matrix.
- `interc` the model intercept.
- `beta` the vector of p coefficients in the linear predictor.
family

a description of the error distribution and link function to be used in the model. This can be a character string naming a family function, a family function or the result of a call to a family function. Only gaussian, binomial or poisson are allowed.

prop

if beta is missing, prop represent the quote of non-null coefficients out of p. The default is 0.10 p.

lim.b

if beta is missing, the coefficients come from uniform variates in lim.b. The default is (-3,3).

sigma

if family is 'gaussian', the standard deviation of the response. The default is 1.

size

if family is 'binomial', the number of trials to build the response vector. The default is 1.

rho

correlation value to define the variance covariance matrix to build the model matrix, i.e., rho^|i-j| i,j = 1,...,p and i different from j. The default is 0.

scale

Should the columns of the mdoel matrix be scaled? The default is TRUE

seed

optional, the seed to generate the data.

X

optional, the model matrix.

Examples

n <- 100
p <- 100
beta <- c(runif(10, -3, 3), rep(0, p-10))
dat <- simulXy(n, p, beta = beta, seed=1234)

summary.islasso

summary method for islasso fitted objects

Description

summary method for islasso fitted objects

Usage

## S3 method for class 'islasso'
summary(object, pval = 1, use.t=FALSE, ...)

Arguments

object

fitted "islasso" object

pval

a threshold p-value value indicating which coefficients should be printed. If pval = 0.10, say, only the variables/coefficients with \( p - value \leq 0.10 \) are printed. If no variables have a p-value lower than pval the unpenalized variables are printed (e.g., the intercept). Moreover, if no variables are unpenalized the variable associated to the lowest p-value is reported.
summary.islasso

use.t

if TRUE, the p-values are computed using the t-distribution with residual model
degrees of freedom

... not used

Examples

## Not run:
#continues example from ?islasso
summary(o, pval=.1) #print just the "borderline" significant coefficients

## End(Not run)
Index

<table>
<thead>
<tr>
<th>Topic</th>
<th>datasets</th>
<th>breast, 6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>diabetes</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Prostate</td>
<td>14</td>
</tr>
<tr>
<td>Topic</td>
<td>models</td>
<td>aic.islasso, 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>islasso, 9</td>
</tr>
<tr>
<td>Topic</td>
<td>package</td>
<td>islasso-package, 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>regression</td>
</tr>
<tr>
<td></td>
<td>aic.islasso</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>islasso</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AIC.islasso, 2, 4, 11</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AIC.islasso(islasso), 9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>aic.islasso, 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>anova.islasso, 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>coef.islasso, 2, 4, 11</td>
</tr>
<tr>
<td></td>
<td></td>
<td>coef.islasso(islasso), 9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>cv.glmnet, 9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>deviance.islasso, 2, 4, 11</td>
</tr>
<tr>
<td></td>
<td></td>
<td>deviance.islasso(islasso), 9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>diabetes, 7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>fitted.islasso, 2, 4, 11</td>
</tr>
<tr>
<td></td>
<td></td>
<td>fitted.islasso(islasso), 9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>is.control, 8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>islasso, 2, 9, 12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>islasso-package, 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>islasso.fit, 2, 4, 11</td>
</tr>
<tr>
<td></td>
<td></td>
<td>logLik.islasso, 2, 4, 11</td>
</tr>
<tr>
<td></td>
<td></td>
<td>logLik.islasso(islasso), 9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>model.matrix.islasso(islasso), 9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>modelX(simulXy), 15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>plot.islasso, 12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>predict.islasso, 2, 4, 11, 13</td>
</tr>
<tr>
<td></td>
<td></td>
<td>print.anova.islasso(anova.islasso), 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>print.islasso(islasso), 9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>print.logLik.islasso(islasso), 9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>print.summary.islasso(summary.islasso), 16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>residuals.islasso, 2, 4, 11</td>
</tr>
<tr>
<td></td>
<td></td>
<td>residuals.islasso(islasso), 9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>simulXy, 15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>summary.islasso, 2, 4, 11, 16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>vcov.islasso(islasso), 9</td>
</tr>
</tbody>
</table>

18