Package ‘islasso’

November 16, 2021

Type Package

Title The Induced Smoothed Lasso

Version 1.4.1

Date 2021-11-03

Author Gianluca Sottile [aut, cre],
      Giovanna Cilluffo [aut, ctb],
      Vito MR Muggeo [aut, cre]

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

Depends glmnet (>= 4.0), Matrix (>= 1.0-6), R (>= 3.6.0)

Suggests knitr, lars, xfun, rmarkdown


License GPL (>= 2)

VignetteBuilder knitr

Encoding UTF-8

URL https://journals.sagepub.com/doi/abs/10.1177/0962280219842890

NeedsCompilation yes

LazyLoad yes

Repository CRAN

Date/Publication 2021-11-16 06:50:02 UTC
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: 2.0.0
Date: 2021-10-15
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.

islasso.path is used to fit Fit a generalized linear model via the Induced Smoothed Lasso. The
regularization path is computed for the lasso or elasticnet penalty at a grid of values for the regularization parameter lambda. Along with coefficients profile, the main advantage is to return also the standard errors profile. The resulting standard errors can be used to make inference in the lasso framework. The main function is `islasso.path` and the corresponding fitter function `islasso.path.fit`, and many auxiliary functions are implemented to summarize and visualize results: `summary.islasso.path`, `predict.islasso.path`, `coef.islasso.path`, `fitted.islasso.path`, `logLik.islasso.path`, `deviance.islasso.path`, `residuals.islasso.path`.

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 <- 30
p1 <- 10 # number of nonzero coefficients
coeff <- sort(round(c(seq(.5, 3, l=p1/2), seq(-1, -2, l=p1/2)), 2))
sigma <- 1
coeff <- c(coeff, rep(0, p-p1))

X <- matrix(rnorm(n*p), n, p)
mu <- drop(X%*%coeff)
y <- mu + rnorm(n, 0, sigma)
o <- islasso.path(y ~ ., data = data.frame(y = y, X),
  family = gaussian())
temp <- GoF.islasso.path(o)
lambda <- temp$lambda.min["BIC"]
o <- islasso(y ~ ., data = data.frame(y = y, X),
  family = gaussian(), lambda = lambda)
o
summary(o, pval = .05)
```
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", "AICc", "GCV", "GIC"),
            interval, g = 0, y, X,
            intercept = FALSE, family = gaussian(), alpha = 1, offset,
            weights, unpenalized, control = is.control(), trace = TRUE)
```

Arguments

- `object`: a fitted model object of class "islasso".
- `method`: the criterion to optimize, AIC, BIC, AICc, GCV, GIC.
- `interval`: the lower and upper limits of \( \lambda \) 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).
- `g`: a value belonging to the interval \([0, 1]\). Classical BIC is returned by letting \( g = 0 \) (default value), whereas extended BIC corresponds to the case \( g = 0.5 \).
- `y`: if `object` is missing, the response vector of length \( n \).
- `X`: if `object` is missing, the design matrix of dimension \( n \times p \).
- `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).
- `trace`: Should the iterative procedure be printed? `TRUE` is the default value.
Details

Minimization of the Akaike Information Criterion (AIC), or Bayesian Information Criterion (BIC) or several other criteria are 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 all methods are computed as trace of the hat matrix at convergence.

Value

the optimal lambda value is returned

Author(s)

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
p1 <- 20  # number of nonzero coefficients
coeff.veri <- sort(round(c(seq(.5, 3, 1=p1/2), seq(-1, -2, 1=p1/2)), 2))
sigma <- 1
coeff <- c(coeff.veri, rep(0, p-p1))
X <- matrix(rnorm(n*p), n, p)
mu <- drop(X%*%coeff)
y <- mu + rnorm(n, 0, sigma)
o <- islasso(y ~ ., data = data.frame(y = 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")
#overwrites the evaluation interval for lambda using eBIC criterion
lambda_ebic <- aic.islasso(o, interval = c(.1, 30), method = "BIC", g = .5)
## End(Not run)
```
Description

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

Usage

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

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.
- **ci**: optionally, a two columns matrix of estimated confidence intervals for the estimated coefficients.
- **...**: 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

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>

Examples

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

X <- matrix(rnorm(n*p), n, p)
eta <- drop(X %*% coef)
mu <- eta
y <- mu + rnorm(n, 0, sigma)
o <- islasso(y ~ . - 1, data = data.frame(y = y, 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 <- coef.true
anova(o, A, b)

## Not run:
# ci <- confint(o, type.ci = "score")
anova(o, A, b, ci = ci)
## 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 agressive and non-agressive 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**

```r
data(breast)
```

**Source**

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


---

**confint.islasso**  
*confint method for islasso objects*

**Description**

confint method for islasso objects

**Usage**

```r
## S3 method for class 'islasso'
confint(object, parm, level = 0.95, trace = FALSE, ...)
```

**Arguments**

- `object`: a fitted model object of class "islasso".
- `parm`: a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered.
- `level`: the confidence level required.
- `trace`: if TRUE (default) a bar shows the iterations status.
- `...`: additional argument(s) for methods.

**Author(s)**

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
p1 <- 10  # number of nonzero coefficients
coeff.veri <- sort(round(c(seq(.5, 3, l=p1/2), seq(-1, -2, l=p1/2)), 2))
```
sigma <- 1

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

X <- matrix(rnorm(n*p), n, p)
et <- drop(X %*% c)

mu <- et
y <- mu + rnorm(n, 0, sigma)

o <- islasso(y ~ ., data = data.frame(y = y, X),
            family = gaussian())

## Not run:
ci <- confint(o, parm = 1:10)
ci
plot(ci)
## End(Not run)

---

**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*
GoF.islasso.path  

Optimization for the selection of the tuning parameter

Description

This function extracts the value of the tuning parameter which minimizes the AIC/BIC/AICc/eBIC/GCV/GIC criterion in “islasso.path”.

Usage

GoF.islasso.path(object, plot = TRUE, ...)

Arguments

object  
a fitted model object of class "islasso.path".

plot  
a logical flag indicating if each criterion have to be plotted

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

Details

Minimization of the Akaike Information Criterion (AIC), or Bayesian Information Criterion (BIC) or several other criteria are 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 all methods are computed as trace of the hat matrix at convergence.

Value

A list of

gof  
the goodness of fit measures

minimum  
the position of the optimal lambda values

lambda.min  
the optimal lambda values

Author(s)

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

See Also

islasso.path, islasso.path.fit, coef.islasso.path, residuals.islasso.path, summary.islasso.path, logLik.islasso.path, fitted.islasso.path, predict.islasso.path and deviance.islasso.path methods.
**Examples**

```r
set.seed(1)
n <- 100
p <- 30
p1 <- 10  # number of nonzero coefficients
coeff.veri <- sort(round(c(seq(.5, 3, l=p1/2), seq(-1, -2, l=p1/2)), 2))
sigma <- 1

coeff <- c(coeff.veri, rep(0, p-p1))
X <- matrix(rnorm(n*p), n, p)
mu <- drop(X%*%coeff)
y <- mu + rnorm(n, 0, sigma)
o <- islasso.path(y ~ ., data = data.frame(y = y, X))
GoF.islasso.path(o)
```

**is.control**

*Auxiliary for controlling islasso model fitting*

**Description**

Auxiliary function for controlling the islasso model fitting.

**Usage**

```r
is.control(sigma2 = -1, tol = 1E-05, itmax = 1E+3, stand = FALSE,
trace = 0, nfolds = 5, seed = NULL, adaptive = FALSE, g = .5,
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` tolerance value to declare convergence, default to 1e-5.
- `itmax` maximum number of iterations, default to 1000.
- `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, 3 = compact printing including Fisher scoring information (only used in glm family).
- `nfolds` if lambda is unspecified in islasso, the number of folds to be used to perform cross validation. 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.

adaptive experimental, if TRUE the adaptive LASSO is implemented.

g a value belonging to the interval \([0, 1]\). Classical BIC is returned by letting \(g = 0\) (default value), whereas extended BIC corresponds to the case \(g = 0.5\).

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, the default is \(c = -1\). \(c = -1\) means to compute it at each step of the iterative algorithm.

Author(s)

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

**islasso**

*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**

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

cfamily 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

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

alpha=1 is the lasso penalty, and alpha=0 the ridge penalty.
### Data
- **data**: an optional data frame, list or environment (or object coercible by `as.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
- **weights**: observation weights. Default is 1 for each observation.

### Subset
- **subset**: an optional vector specifying a subset of observations to be used in the fitting process.

### Offset
- **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
- **unpenalized**: optional. A vector of integers or characters indicating any covariate (in the formula) with coefficients not to be penalized. The intercept, if included in the model, is always unpenalized.

### Contrasts
- **contrasts**: an optional list. See the contrasts.arg of `model.matrix.default`.

### Control
- **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
- **residuals**: the working residuals
- **fitted.values**: the fitted values
- **rank**: the estimated degrees of freedom
- **family**: the family object used
- **linear.predictors**: the linear predictors
- **deviance**: the family deviance
- **aic**: the Akaike Information Criterion
- **null.deviance**: the family null deviance
- **iter**: the number of iterations of IWLS used
- **weights**: the working weights, that is the weights in the final iteration of the IWLS fit
- **df.residual**: the residual degrees of freedom
df.null  the degrees of freedom of a null model
converged logical. Was the IWLS algorithm judged to have converged?
model if requested (the default), the model frame used.
call the matched call
formula the formula supplied
terms the terms object used
data the data argument.
offset the offset vector used.
control the value of the control argument used
xlevels (where relevant) a record of the levels of the factors used in fitting.
lambda the lambda value used in the islasso algorithm
alpha the elasticnet mixing parameter
dispersion the estimated dispersion parameter
internal internal elements
contrasts (only where relevant) the contrasts used.

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
set.seed(1)
n <- 100
p <- 100
p1 <- 10  #number of nonzero coefficients
coeff.vari <- sort(round(c(seq(.5, 3, 1=p1/2), seq(-1, -2, 1=p1/2)), 2))
sigma <- 1
islasso

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

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

##### gaussian ######
mu <- eta
y <- mu + rnorm(n, 0, sigma)
o <- islasso(y ~ ., data = data.frame(y = y, X),
            family = gaussian())
o
summary(o)
coef(o)
fitted(o)
predict(o, type="response")
plot(o)
residuals(o)
deviance(o)
AIC(o)
logLik(o)

## Not run:
confint(o, parm = 1:10)

# for the interaction
o <- islasso(y ~ X1 * X2, data = data.frame(y = y, X),
            family = gaussian())

##### binomial ######
coef <- c(c(1,1,1), rep(0, p-3))
X <- matrix(rnorm(n*p), n, p)
eta <- drop(cbind(1, X)%*%c(-1, coef))
mu <- binomial()$linkinv(eta)
y <- rbinom(n, 100, mu)
y <- cbind(y, 100-y)
o <- islasso(cbind(y1, y2) ~ .,
            data = data.frame(y1 = y[,1], y2 = y[,2], X),
            family = binomial())

summary(o, pval = .05)

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

summary(o, pval = .05)
islasso.path

Description

islasso.path is used to fit a generalized linear model via induced smoothed lasso method. The regularization path is computed for the lasso or elasticnet penalty at a grid of values for the regularization parameter lambda. Fits linear, logistic, poisson and gamma regression models.

Usage

islasso.path(formula, family = gaussian(), lambda = NULL, nlambda = 100,
             lambda.min.ratio = ifelse(nobs < nvars, 1E-3, 1E-05), alpha = 1, data,
             weights, subset, offset, contrasts = NULL, unpenalized, control = is.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  A user supplied lambda sequence. Typical usage is to have the program compute its own lambda sequence based on nlambda and lambda.min.ratio. Supplying a value of lambda overrides this.

nlambda  The number of lambda values - default is 100.

lambda.min.ratio  Smallest value for lambda, as a fraction of lambda.max, the (data derived) entry value (i.e. the smallest value for which all coefficients are zero). The default depends on the sample size nobs relative to the number of variables nvars. If

```r
### Gamma ######
coef <- c(c(1,1,1), rep(0, p-3))
X <- matrix(rnorm(n*p), n, p)
eta <- 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 ~ ., data = data.frame(y = y, X),
            family = Gamma(link = "log"))
summary(o, pval = .05)
```

## End(Not run)
nobs > nvars, the default is 0.00001, close to zero. If nobs < nvars, the default is 0.001. A very small value of lambda.min.ratio will lead to a saturated fit in the nobs < nvars case.

alpha The elastic-net mixing parameter, with 0 ≤ α ≤ 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.

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.

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).

unpenalized optional. A vector of integers or characters indicating any covariate (in the formula) with coefficients not to be penalized. The intercept, if included in the model, is always unpenalized.

Details

The sequence of models implied by lambda is fit the islasso method. 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

call the matched call.

Info a named matrix containing information about lambda values, estimated degrees of freedom, estimated dispersion parameters, deviance, loglikelihood, number of iterations and convergence criteria.

GoF a named matrix containing information criteria, i.e., AIC, BIC, AICc, eBIC, GCV, GIC.

Coef a length(lambda) x nvars matrix of coefficients.
SE  
a length(lambda) x nvars matrix of standard errors.

Weights  
a length(lambda) x nvars matrix of the weight of the mixture in the induced smoothed lasso.

Linear.predictors  
a length(lambda) x nvars matrix of linear predictors

Fitted.values  
a length(lambda) x nvars matrix of fitted values

Residuals  
a length(lambda) x nvars matrix of working residuals

Input  
a named list containing several input arguments, i.e., the numbers of observations and predictors, if an intercept has to be estimated, the model matrix and the response vector, the observation weights, the offset, the family object used, The elasticnet mixing parameter and the vector used to specify the unpenalized estimators.

control  
the value of the control argument used.

formula  
the formula supplied.

model  
if requested (the default), the model frame used.

terms  
the terms object used.

data  
the data argument.

xlevels  
(where relevant) a record of the levels of the factors used in fitting.

contrasts  
(only where relevant) the contrasts used.

Author(s)

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

References


See Also

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

Examples

```
set.seed(1)
n <- 100
p <- 30
p1 <- 10  # number of nonzero coefficients
coeff.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)
et <- drop(X%*%coef)

##### gaussian ######
mu <- et
y <- mu + rnorm(n, 0, sigma)
o <- islasso.path(y ~ ., data = data.frame(y = y, X),
    family = gaussian(), nlambda = 30L)
o
summary(o, lambda = 10)
coef(o, lambda = 10)
fitted(o, lambda = 10)
predict(o, type="response", lambda = 10)
plot(o, xvar = "coef")
residuals(o, lambda = 10)
deviance(o, lambda = 10)
logLik(o, lambda = 10)
GoF.islasso.path(o)

## Not run:

##### 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, coef))
mu <- binomial(linkinv(eta))
y <- rbinom(n, 100, mu)
y <- cbind(y, 100-y)
o <- islasso.path(cbind(y1, y2) ~ .,
    data = data.frame(y1 = y[,1], y2 = y[,2], X),
    family = binomial(), nlambda = 30L)
temp <- GoF.islasso.path(o)
summary(o, pval = .05, lambda = temp$lambda.min["BIC"])

##### poisson ######
coef <- c(c(1,1,1), rep(0, p-3))
X <- matrix(rnorm(n*p), n, p)
et <- drop(cbind(1, X)%*%c(1, coef))
mu <- poisson(linkinv(eta))
y <- rpois(n, mu)
o <- islasso.path(y ~ ., data = data.frame(y = y, X),
    family = poisson(), nlambda = 30L)
temp <- GoF.islasso.path(o)
summary(o, pval = .05, lambda = temp$lambda.min["BIC"])

##### Gamma ######
coef <- c(c(1,1,1), rep(0, p-3))
\begin{verbatim}
X <- matrix(rnorm(n*p), n, p)
eta <- drop(cbind(1, X)%*%c(-1, coef))
mu <- Gamma(link="log")$linkinv(eta)
shape <- 10
phi <- 1 / shape
y <- rgamma(n, scale = mu / shape, shape = shape)
o <- islasso.path(y ~ ., data = data.frame(y = y, X),
                 family = Gamma(link = "log"), nlambda = 30L)
temp <- GoF.islasso.path(o)
summary(o, pval = .05, lambda = temp$lambda.min["BIC"])
\end{verbatim}

---

**plot.islasso**

### Description

Diagnostics plots for Induced Smoothing Lasso Model

#### Usage

```
## 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

#### 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.

#### Author(s)

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
## Not run:
set.seed(1)
n <- 100
p <- 100
p1 <- 20 # number of nonzero coefficients
coeff.veri <- sort(round(c(seq(.5, 3, l=p1/2), seq(-1, -2, l=p1/2)), 2))
sigma <- 1
coeff <- c(coeff.veri, 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 ~ ., data = data.frame(y = y, X),
             family = gaussian(), lambda = lambda)
plot(o)
## End(Not run)
```

plot.islasso.path  
plot coefficient profile from a fitted "islasso.path" object.

Description

Produces a coefficient profile plot of the coefficient paths for a fitted "islasso.path" object.

Usage

```r
## S3 method for class 'islasso.path'
plot(x, yvar = c("coefficients", "se", "gradient", "weight", "AIC", "BIC", "AICc", "eBIC", "GCV", "GIC"),
     label = FALSE, cex.lab = 1, ...)
```

Arguments

- `x` an object of class islasso, usually, a result of a call to `islasso.path`.
- `yvar` What is on the Y-axis. "coef" plot the log-lambda sequence against the coefficients; "se" plot the log-lambda sequence against the standard deviations; "gradient" plot the log-lambda sequence against the gradient; "weight" plot the log-lambda sequence against the mixture weight of the islasso method; "AIC", "BIC", "AICc", "eBIC", "GCV" and "GIC" plot the log-lambda sequence against the chosen criterion.
- `label` a logical flag indicating if some labels have to be added on each plot.
- `cex.lab` a numerical scalar giving the amount by which plotting labels should be scaled.
- `...` other graphical parameters for the plot
Details

A coefficient profile plot is produced for Induced Smoothing Lasso Model path.

Examples

```r
## Not run:
set.seed(1)
n <- 100
p <- 30
p1 <- 10 #number of nonzero coefficients
coeff.veri <- sort(round(c(seq(.5, 3, 1=p1/2), seq(-1, -2, 1=p1/2)), 2))
sigma <- 1

c <- c(coeff.veri, rep(0, p-p1))
X <- matrix(rnorm(n*p), n, p)
mu <- drop(X%*%c)
y <- mu + rnorm(n, 0, sigma)
o <- islasso.path(y ~ ., data = data.frame(y = y, X),
family = gaussian())
par(mfrow = c(2, 2))
plot(o, yvar = "coefficients")
plot(o, yvar = "se")
plot(o, yvar = "gradient")
plot(o, yvar = "weight")

par(mfrow = c(2, 3))
plot(o, yvar = "AIC")
plot(o, yvar = "BIC")
plot(o, yvar = "AICc")
plot(o, yvar = "eBIC")
plot(o, yvar = "GCV")
plot(o, yvar = "GIC")

## 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, newdata, type = c("link", "response", 
"coefficients", "class"), se.fit = FALSE, level = .95, ...)
```
**predict.islasso**

Arguments

- **object**: a fitted object of class "islasso".
- **newdata**: optionally, a data frame in which to look for variables with which to predict. If omitted, the fitted linear predictors are used.
- **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.
- **se.fit**: logical switch indicating if confidence intervals are required.
- **level**: the confidence level required.
- **...**: further arguments passed to or from other methods.

Value

An object depending on the type argument

Author(s)

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
coeff.VERI <- sort(round(c(seq(.5, 3, l=pl/2), seq(-1, -2, l=pl/2)), 2))
sigma <- 1
coeff <- c(coeff.VERI, rep(0, p-pl))
X <- matrix(rnorm(n*p), n, p)
mu <- drop(X%*%coeff)
y <- mu + rnorm(n, 0, sigma)
lambda <- 2
o <- islasso(y ~ ., data = data.frame(y = y, X), lambda = lambda)
predict(o, type = "response")

## Not run:
predict(o, type = "response", se.fit = TRUE)

## End(Not run)
```
predict.islasso.path  Prediction method for islasso.path fitted objects

Description

Prediction method for islasso fitted objects

Usage

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

Arguments

object  a fitted object of class "islasso.path".
newdata optionally, a data frame in which to look for variables with which to predict. If omitted, the fitted linear predictors are used.
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.
lambda Value(s) of the penalty parameter lambda at which predictions are required. Default is the entire sequence used to create the model.
... further arguments passed to or from other methods.

Value

An object depending on the type argument

Author(s)

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

See Also

islasso.path, islasso.path.fit, coef.islasso.path, residuals.islasso.path, GoF.islasso.path, logLik.islasso.path, fitted.islasso.path, summary.islasso.path and deviance.islasso.path methods.
## Examples

```r
## Not run:
set.seed(1)
N <- 100
p <- 30
p1 <- 10  # number of nonzero coefficients
do coef.veri <- sort(round(c(seq(.5, 3, l=p1/2), seq(-1, -2, l=p1/2)), 2))
sigma <- 1

do 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)

do o <- islasso.path(y ~ ., data = data.frame(y = y, X),
                      family = gaussian())
temp <- GoF.islasso.path(o)
predict(o, type = "response", lambda = temp$lambda.min)

## End(Not run)
```

### 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

```r
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
Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate: II. radical

---

simulXy

Simulate model matrix and response

---

Description
Simulate model matrix and response from a specified distribution.

Usage
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 model matrix be scaled? The default is TRUE
- seed: optional, the seed to generate the data.
- X: optional, the model matrix.
Examples

```r
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

```r
## 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. Possible unpenalized coefficients (including the intercept if in the model) are always printed, regardless of their p-value.
- `use.t`: if TRUE, the p-values are computed using the t-distribution with residual model degrees of freedom
- `...`: not used

### Author(s)

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`
- `deviance.islasso` methods.

### Examples

```r
## Not run:
# continues example from ?islasso
summary(o, pval = .1) # print just the "borderline" significant coefficients
summary(o, pval = .1, type.pval = "score") # print just the "borderline" significant coefficients
## End(Not run)
```
summary.islasso.path  summary method for islasso.path fitted objects

Description

summary method for islasso.path fitted objects

Usage

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

Arguments

object  
fitted "islasso.path" object  
pval  
a threshold p-value value indicating which coefficients should be printed. If pval = 0.10, say, only the variables/coefficients with \( p \text{-value} \leq 0.10 \) are printed. Possible unpenalized coefficients (including the intercept if in the model) are always printed, regardless of their p-value.
use.t  
if TRUE, the p-values are computed using the t-distribution with residual model degrees of freedom
lambda  
Value of the penalty parameter lambda at which summary are required.
...  
not used

Author(s)

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

See Also

islasso.path, islasso.path.fit, coef.islasso.path, residuals.islasso.path, GoF.islasso.path, logLik.islasso.path, fitted.islasso.path, predict.islasso.path and deviance.islasso.path methods.

Examples

## Not run:
#continues example from ?islasso.path
summary(o, pval = .1, lambda = 5) #print just the "borderline" significant coefficients

## End(Not run)
Index

* datasets
  breast, 7
  diabetes, 9
  Prostate, 25

* models
  aic.islasso, 4
  GoF.islasso.path, 10
  islasso, 12
  islasso.path, 16

* package
  islasso-package, 2

* regression
  aic.islasso, 4
  GoF.islasso.path, 10
  islasso, 12
  islasso.path, 16

AIC.islasso, 2, 5, 8, 14, 20, 23, 27
AIC.islasso(islasso), 12
aic.islasso, 4
anova.islasso, 6

breast, 7

coef.islasso, 2, 5, 8, 14, 20, 23, 27
coef.islasso(islasso), 12
coeff.islasso(path(islasso), 16
confint.islasso, 8
cv.glmmnet, 12

deviance.islasso, 2, 5, 8, 14, 20, 23, 27
deviance.islasso(islasso), 12
deviance.islasso.path, 3, 10, 18, 24, 28
deviance.islasso.path(islasso), 16
diabetes, 9

fitted.islasso, 2, 5, 8, 14, 20, 23, 27
fitted.islasso(islasso), 12
fitted.islasso.path, 3, 10, 18, 24, 28
fitted.islasso.path(islasso), 16

GoF.islasso.path, 10, 18, 24, 28
is.control, 11
islasso, 2, 12, 20
islasso-package, 2
islasso.fit, 2, 5, 8, 14, 20, 23, 27
islasso.path, 2, 3, 10, 16, 21, 24, 28
islasso.path.fit, 3, 10, 18, 24, 28
logLik.islasso, 2, 5, 8, 14, 20, 23, 27
logLik.islasso(islasso), 12
logLik.islasso.path, 3, 10, 18, 24, 28
logLik.islasso.path(islasso), 16

model.matrix.islasso(islasso), 12
model.matrix.islasso.path
  (islasso), 16
modelX(simulXy), 26

plot.confint.islasso(confint.islasso), 8
plot.islasso, 20
plot.islasso.path, 21
predict.islasso, 2, 5, 8, 14, 20, 22, 23, 27
predict.islasso.path, 3, 10, 18, 24, 28
print.anova.islasso(anova.islasso), 6
print.confint.islasso
  (confint.islasso), 8
print.islasso(islasso), 12
print.islasso.path(islasso), 16
print.logLik.islasso(islasso), 12
print.logLik.islasso.path
  (islasso), 16
print.summary.islasso
  (summary.islasso), 27
print.summary.islasso.path
  (summary.islasso), 28
Prostate, 25

residuals.islasso, 2, 5, 8, 14, 20, 23, 27
residuals.islasso(islasso), 12
residuals.islasso.path, 3, 10, 18, 24, 28
residuals.islasso.path(islasso.path), 16

simulXy, 26
summary.islasso, 2, 5, 8, 14, 20, 23, 27, 27
summary.islasso.path, 3, 10, 18, 24, 28

vcov.islasso(islasso), 12