

Package ‘islasso’

November 14, 2025

Type Package

Title The Induced Smoothed Lasso

Version 1.6.1

Date 2025-11-13

Depends glmnet (≥ 4.0), R ($\geq 4.1.0$)

Imports stats, utils, graphics, cli, gridExtra, ggplot2

Suggests knitr, lars, xfun, rmarkdown, roxygen2

Description

An implementation of the induced smoothing (IS) idea to lasso regularization models to allow estimation and inference on the model coefficients (currently hypothesis testing only). Linear, logistic, Poisson and gamma regressions with several link functions are implemented. The algorithm is described in the original paper; see [doi:10.1177/0962280219842890](https://doi.org/10.1177/0962280219842890) and discussed in a tutorial [doi:10.13140/RG.2.2.16360.11521](https://doi.org/10.13140/RG.2.2.16360.11521).

License GPL (≥ 2)

Encoding UTF-8

URL <https://gianluca-sottile.github.io/islasso/>

BugReports <https://github.com/gianluca-sottile/islasso/issues>

RoxygenNote 7.3.3

NeedsCompilation yes

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

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

LazyLoad yes

Repository CRAN

VignetteBuilder knitr

Date/Publication 2025-11-14 13:50:02 UTC

Contents

aic.islasso	2
anova.islasso	4
breast	5
confint.islasso	6
diabetes	7
GoF.islasso.path	8
is.control	10
islasso	11
islasso.path	14
plot.islasso	18
plot.islasso.path	19
predict.islasso	20
predict.islasso.path	22
Prostate	23
relax.islasso	24
simulXy	25
summary.islasso	27
summary.islasso.path	28
Index	30

aic.islasso	<i>Optimization for Lambda Selection</i>
-------------	--

Description

Minimizes information criteria to select the optimal tuning parameter lambda for [islasso](#) models. Supports AIC, BIC, AICc, GCV, and GIC.

Usage

```
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	Fitted model of class "islasso".
method	Criterion to minimize. Options are "AIC", "BIC", "AICc", "GCV", "GIC".
interval	Numeric vector (length 2) giving lower and upper bounds for lambda optimization. Optional if object includes prior cross-validation.
g	Numeric in $[0, 1]$. Governs BIC generalization: $g = 0$ is classic BIC, $g = 0.5$ is extended BIC.
y	Response vector. Required only if object is missing.
X	Design matrix. Required only if object is missing.
intercept	Logical. Whether to include intercept in X. Used if object is missing.
family	Error distribution. Accepted: gaussian, binomial, poisson. Uses canonical link.
alpha	Elastic-net mixing parameter, $0 \leq \alpha \leq 1$. Lasso: $\alpha = 1$; Ridge: $\alpha = 0$.
offset	Optional numeric vector. Adds known linear predictor component.
weights	Optional weights for observations. Defaults to 1.
unpenalized	Logical vector indicating variables to exclude from penalization.
control	List of control parameters. See is.control .
trace	Logical. If TRUE, prints progress of optimization. Default is TRUE.

Details

Instead of using cross-validation, this function selects the best lambda by minimizing criteria like AIC or BIC. Degrees of freedom are computed as the trace of the hat matrix (not necessarily an integer).

Value

Optimal lambda value as numeric.

Author(s)

Gianluca Sottile <gianluca.sottile@unipa.it>

See Also

[islasso](#), [islasso.path](#), [is.control](#), [simulxy](#)

Examples

```
set.seed(1)
n <- 100; p <- 100
beta <- c(rep(2, 20), rep(0, p - 20))
sim1 <- simulxy(n = n, p = p, beta = beta, seed = 1, family = gaussian())
o <- islasso(y ~ ., data = sim1$data, family = gaussian())
```

```
## 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(0.1, 30), method = "BIC")

# Overwrites the evaluation interval for lambda using eBIC criterion
lambda_ebic <- aic.islasso(o, interval = c(0.1, 30), method = "BIC", g = 0.5)

## End(Not run)
```

anova.islasso

General Linear Hypotheses for islasso Models

Description

Tests general linear hypotheses and computes confidence intervals for linear combinations of coefficients from a fitted [islasso](#) model.

Usage

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

Arguments

object	A fitted model object of class "islasso".
A	Hypothesis specification. Either: • A numeric matrix or vector with each row specifying a linear combination of coefficients, • Or a character vector with symbolic expressions (e.g. "X1 + X2 = 3").
b	Right-hand side vector for the null hypotheses $A\beta = b$. If omitted, defaults to zeros.
ci	Optional 2-column matrix of confidence intervals for coefficients.
...	Currently unused.

Details

The method tests the null hypothesis $H_0 : A\beta = b$, where A and b define a linear constraint on model coefficients.

Symbolic expressions support natural syntax: coefficients may be added/subtracted, constants may be multiplied (e.g. "2 * X1 + 3 * X2 = 7"). Equations with omitted = assume zero on the right-hand side. See examples for syntax flexibility.

Value

An object of class "anova.islasso" containing:

Estimate	Linear combination estimates
SE	Standard errors
Wald	Wald statistics
p-value	Associated p-values

Author(s)

Gianluca Sottile <gianluca.sottile@unipa.it>

See Also

[islasso](#), [summary.islasso](#), [confint.islasso](#), [predict.islasso](#), [logLik.islasso](#), [residuals.islasso](#)

Examples

```
n <- 100; p <- 100
beta <- c(runif(10, -2, 2), rep(0, p - 10))
sim <- simulXy(n = n, p = p, beta = beta, seed = 1, family = gaussian())
fit <- islasso(y ~ . -1, data = sim$data, family = gaussian())

# Test if first 5 variables sum to -7.5
anova(fit, A = c("X1 + X2 + X3 + X4 + X5 = -7.5"))

# Test multiple hypotheses
anova(fit, A = c("X1 + X2 + X3 + X4 + X5", "X6 + X7 + X8 + X9 + X10"), b = c(-7.5, 8.75))

# Full diagonal comparison to true coefficients
anova(fit, A = diag(p), b = beta)
```

breast

Breast Cancer microarray experiment

Description

This data set details a microarray experiment for 52 breast cancer patients. The binary variable `status` indicates whether or not the patient died of breast cancer (`status = 0`: did not die, `status = 1`: died). The other variables represent amplification or deletion of specific genes.

Format

A data frame with 52 rows and multiple variables, including a binary status and gene-level measurements.

Details

Unlike gene expression studies, this experiment focuses on measuring gene amplification or deletion- the number of DNA copies for a given genomic sequence. The goal is to identify key genomic markers distinguishing aggressive from non-aggressive breast cancer.

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

Source

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

References

Augugliaro L., Mineo A.M. and Wit E.C. (2013). *dgLARS: a differential geometric approach to sparse generalized linear models*, Journal of the Royal Statistical Society. Series B, Vol 75(3), 471-498. Wit E.C. and McClure J. (2004). *Statistics for Microarrays: Design, Analysis and Inference*, Chichester: Wiley.

Examples

```
data(breast)
str(breast)
table(breast$status)

## Not run:
fit <- islasso.path(status ~ ., data = breast, family = binomial(),
  alpha = 0, control = is.control(trace = 2L))
temp <- GoF.islasso.path(fit)
lambda.aic <- temp$lambda.min["AIC"]
fit.aic <- islasso(status ~ ., data = breast, family = binomial(),
  alpha = 0, lambda = lambda.aic)
summary(fit.aic, pval = 0.05)

## End(Not run)
```

confint.islasso	<i>confint method for islasso objects</i>
-----------------	---

Description

Computes confidence intervals for islasso objects using a Wald-type approach.

Usage

```
## S3 method for class 'islasso'
confint(object, parm, level = 0.95, type.ci = "wald", trace = TRUE, ...)
```

Arguments

<code>object</code>	A fitted model object of class "islasso".
<code>parm</code>	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.
<code>level</code>	The confidence level required.
<code>type.ci</code>	Character. Only Wald-type confidence intervals are implemented yet! Set <code>type.ci = "wald"</code> to use estimates and standard errors to build the confidence interval.
<code>trace</code>	Logical. If TRUE (default), a bar shows the iterations status.
<code>...</code>	Additional arguments for methods.

Details

`confint` method for `islasso` objects

Author(s)

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

See Also

[islasso.fit](#), [summary.islasso](#), [residuals.islasso](#), [logLik.islasso](#), [predict.islasso](#), [deviance.islasso](#)

Examples

```
n <- 100; p <- 100; p1 <- 10
beta.veri <- sort(round(c(seq(0.5, 3, length.out = p1 / 2),
                        seq(-1, -2, length.out = p1 / 2)), 2))
beta <- c(beta.veri, rep(0, p - p1))
sim <- simuXy(n = n, p = p, beta = beta, seed = 1, family = gaussian())
o <- islasso(y ~ ., data = sim$data, family = gaussian())

ci <- confint(o, type.ci = "wald", parm = 1:11)
ci
plot(ci)
```

diabetes

Blood and other measurements in diabetics

Description

The diabetes data frame contains 442 observations used in the Efron et al. "Least Angle Regression" paper.

Format

A data frame with 442 rows and 3 columns:

x Matrix with 10 numeric columns (standardized)

y Numeric response vector

x2 Matrix with 64 columns including interactions

Details

The x matrix has been standardized to have unit L2 norm and zero mean in each column. The x2 matrix extends x by adding selected interaction terms.

Source

https://web.stanford.edu/~hastie/Papers/LARS/LeastAngle_2002.ps

References

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

Examples

```
data(diabetes)
str(diabetes)
summary(diabetes$y)

## Not run:
fit <- islasso(y ~ ., data = data.frame(y = diabetes$y, diabetes$x2),
              family = gaussian())
summary(fit, pval = 0.05)
lambda.aic <- aic.islasso(fit, interval = c(1, 100))
fit.aic <- update(fit, lambda = lambda.aic)
summary(fit.aic, pval = 0.05)

## End(Not run)
```

GoF.islasso.path

Select Optimal Lambda via Goodness-of-Fit Criteria

Description

Extracts the tuning parameter lambda minimizing multiple information criteria from a fitted `islasso.path` object. Supported criteria include AIC, BIC, AICc, eBIC, GCV, and GIC.

Usage

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


Arguments

<code>object</code>	A fitted model of class "islasso.path".
<code>plot</code>	Logical. If TRUE (default), displays plots for each criterion over the lambda path.
<code>...</code>	Additional arguments passed to lower-level plotting or diagnostic methods.

Details

This function identifies the optimal regularization parameter λ by minimizing various information-based selection criteria. Degrees of freedom are computed as the trace of the hat matrix, which may be fractional under induced smoothing. This provides a robust alternative to cross-validation, especially in high-dimensional settings.

Value

A list with components:

<code>gof</code>	Matrix of goodness-of-fit values across lambda values.
<code>minimum</code>	Index positions of the minimum for each criterion.
<code>lambda.min</code>	Optimal lambda values that minimize each criterion.

Author(s)

Gianluca Sottile <gianluca.sottile@unipa.it>

See Also

[islasso.path](#), [summary.islasso.path](#), [predict.islasso.path](#), [coef.islasso.path](#), [deviance.islasso.path](#), [logLik.islasso.path](#), [residuals.islasso.path](#), [fitted.islasso.path](#)

Examples

```
set.seed(1)
n <- 100; p <- 30
beta <- c(runif(10, -2, 2), rep(0, p - 10))
sim <- simulXy(n = n, p = p, beta = beta, seed = 1, family = gaussian())
fit <- islasso.path(y ~ ., data = sim$data, family = gaussian())
GoF.islasso.path(fit)
```

is.control

*Control Settings for islasso Model Fitting***Description**

Auxiliary function used to configure and customize the fitting process of [islasso](#) models.

Usage

```
is.control(
  sigma2 = -1,
  tol = 1e-05,
  itmax = 1000,
  stand = TRUE,
  trace = 0,
  nfolds = 5,
  seed = NULL,
  adaptive = FALSE,
  g = 0.5,
  b0 = NULL,
  V0 = NULL,
  c = 0.5
)
```

Arguments

sigma2	Numeric. Fixed value of the dispersion parameter. If -1 (default), it is estimated from data.
tol	Numeric. Tolerance level to declare convergence. Default is 1e-5.
itmax	Integer. Maximum number of iterations. Default is 1000.
stand	Logical. If TRUE (default), standardizes covariates before fitting. Returned coefficients remain on the original scale.
trace	Integer. Controls verbosity of the iterative procedure: • 0 - no printing, • 1 - compact printing, • 2 - detailed printing, • 3 - compact printing with Fisher scoring info (only for GLM).
nfolds	Integer. Number of folds for CV if lambda is missing in islasso. Defaults to 5.
seed	Optional. Integer seed for reproducibility in cross-validation.
adaptive	Logical. If TRUE, fits an adaptive LASSO. (Experimental)
g	Numeric in $[0, 1]$. Governs BIC selection: $g = 0$ is standard BIC; $g = 0.5$ is extended BIC.

b0	Optional. Starting values for regression coefficients. If NULL, uses glmnet estimates.
V0	Optional. Initial covariance matrix. Defaults to identity matrix if NULL.
c	Numeric. Controls the weight in the induced smoothed LASSO. Default is 0.5; use -1 to recompute at every iteration.

Value

A list of control parameters for use in [islasso](#).

Author(s)

Gianluca Sottile <gianluca.sottile@unipa.it>

See Also

[islasso](#), [islasso.path](#)

islasso	<i>The Induced Smoothed Lasso: A practical framework for hypothesis testing in high dimensional regression</i>
---------	--

Description

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

Fits regression models with a smoothed L1 penalty under the induced smoothing paradigm. Supports linear, logistic, Poisson, and Gamma responses. Enables reliable standard errors and Wald-based inference.

Usage

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

Arguments

<code>formula</code>	A symbolic formula describing the model.
<code>family</code>	Response distribution. Can be gaussian, binomial, poisson, or Gamma.
<code>lambda</code>	Regularization parameter. If missing, it is estimated via cv.glmnet .
<code>alpha</code>	Elastic-net mixing parameter ($0 \leq \alpha \leq 1$).
<code>data</code>	A data frame or environment containing the variables in the model.
<code>weights</code>	Observation weights. Defaults to 1.
<code>subset</code>	Optional vector specifying a subset of rows to include.
<code>offset</code>	Optional numeric vector of offsets in the linear predictor.
<code>unpenalized</code>	Vector indicating variables (by name or index) to exclude from penalization.
<code>contrasts</code>	Optional contrasts specification for factor variables.
<code>control</code>	A list of parameters to control model fitting. See is.control .

Details

Package: islasso
 Type: Package
 Version: 1.6.1
 Date: 2025-11-13
 License: GPL-2

[islasso](#) fits generalized linear models with an L1 penalty on selected coefficients. It returns both point estimates and full covariance matrices, enabling standard error-based inference. Related methods include: [summary.islasso](#), [predict.islasso](#), [logLik.islasso](#), [deviance.islasso](#), and [residuals.islasso](#).

[islasso.path](#) fits regularization paths using the Induced Smoothed Lasso. It computes coefficients and standard errors across a grid of lambda values. Companion methods include: [summary.islasso.path](#), [predict.islasso.path](#), [logLik.islasso.path](#), [residuals.islasso.path](#), [coef.islasso.path](#), and [fitted.islasso.path](#).

The non-smooth L1 penalty is replaced by a smooth approximation, enabling inference through standard errors and Wald tests. The approach controls type-I error and shows strong power in various simulation settings.

Value

A list with components such as:

<code>coefficients</code>	Estimated coefficients
<code>se</code>	Standard errors
<code>fitted.values</code>	Fitted values
<code>deviance, aic, null.deviance</code>	Model diagnostic metrics

```

residuals, weights          IWLS residuals and weights
df.residual, df.null, rank  Degrees of freedom
converged                   Logical; convergence status
model, call, terms, formula, data, offset
                             Model objects
xlevels, contrasts           Factor handling details
lambda, alpha, dispersion   Model parameters
internal                    Other internal values

```

Author(s)

Gianluca Sottile, based on preliminary work by Vito Muggeo. Maintainer: <gianluca.sottile@unipa.it>
 Gianluca Sottile <gianluca.sottile@unipa.it>

References

Cilluffo, G., Sottile, G., La Grutta, S., Muggeo, VMR (2019). *The Induced Smoothed lasso: A practical framework for hypothesis testing in high dimensional regression*, Statistical Methods in Medical Research. DOI: [doi:10.1177/0962280219842890](https://doi.org/10.1177/0962280219842890)

Sottile, G., Cilluffo, G., Muggeo, VMR (2019). *The R package islasso: estimation and hypothesis testing in lasso regression*. Technical Report on ResearchGate. DOI: [doi:10.13140/RG.2.2.16360.11521](https://doi.org/10.13140/RG.2.2.16360.11521)

Cilluffo G., Sottile G., La Grutta S., Muggeo V.M.R. (2019) *The Induced Smoothed Lasso: A practical framework for hypothesis testing in high dimensional regression*. Statistical Methods in Medical Research. DOI: [10.1177/0962280219842890](https://doi.org/10.1177/0962280219842890)

Sottile G., Cilluffo G., Muggeo V.M.R. (2019) *The R package islasso: estimation and hypothesis testing in lasso regression*. Technical Report. DOI: [10.13140/RG.2.2.16360.11521](https://doi.org/10.13140/RG.2.2.16360.11521)

See Also

Useful links:

- <https://gianluca-sottile.github.io/islasso/>
- Report bugs at <https://github.com/gianluca-sottile/islasso/issues>

[confint.islasso](#), [plot.islasso](#), [predict.islasso](#), [summary.islasso](#), [is.control](#), [aic.islasso](#), [anova.islasso](#), [relax.islasso](#), [islasso.path](#), [simulXy](#)

Examples

```

n <- 100; p <- 100

beta <- c(rep(1, 5), rep(0, p - 5))
sim1 <- simulXy(n = n, p = p, beta = beta, seed = 1, family = gaussian())
o <- islasso(y ~ ., data = sim1$data, family = gaussian())

summary(o, pval = 0.05)
coef(o)
fitted(o)
predict(o, type="response")
plot(o)
residuals(o)
deviance(o)
AIC(o)
logLik(o)
summary(relax.islasso(o), pval = 0.05)

## Not run:
# for the interaction
o <- islasso(y ~ X1 * X2, data = sim1$data, family = gaussian())

##### binomial #####
beta <- c(c(1,1,1), rep(0, p-3))
sim2 <- simulXy(n = n, p = p, beta = beta, inter = 1, seed = 1,
               size = 100, family = binomial())
o2 <- islasso(cbind(y.success, y.failure) ~ .,
              data = sim2$data, family = binomial())
summary(o2, pval = 0.05)

##### poisson #####
beta <- c(c(1,1,1), rep(0, p-3))
sim3 <- simulXy(n = n, p = p, beta = beta, inter = 1, seed = 1,
               family = poisson())
o3 <- islasso(y ~ ., data = sim3$data, family = poisson())
summary(o3, pval = 0.05)

##### Gamma #####
beta <- c(c(1,1,1), rep(0, p-3))
sim4 <- simulXy(n = n, p = p, beta = beta, inter = -1, seed = 1,
               dispersion = 0.1, family = Gamma(link = "log"))
o4 <- islasso(y ~ ., data = sim4$data, family = Gamma(link = "log"))
summary(o4, pval = 0.05)

## End(Not run)

```

Description

Fits a sequence of penalized regression models using the Induced Smoothing Lasso approach over a grid of lambda values. Supports elastic-net penalties and generalized linear models: Gaussian, Binomial, Poisson, and Gamma.

Usage

```
islasso.path(
  formula,
  family = gaussian(),
  lambda = NULL,
  nlambdas = 100,
  lambda.min.ratio = ifelse(nobs < nvars, 0.01, 1e-04),
  alpha = 1,
  data,
  weights,
  subset,
  offset,
  contrasts = NULL,
  unpenalized,
  control = is.control()
)
```

Arguments

formula	Model formula of type response ~ predictors.
family	Response distribution. Supported families: gaussian(), binomial(), poisson(), Gamma().
lambda	Optional numeric vector of lambda values. If not provided, a sequence is automatically generated.
nlambdas	Integer. Number of lambda values to generate if lambda is missing. Default is 100.
lambda.min.ratio	Smallest lambda as a fraction of lambda.max. Default: 1e-2 if nobs < nvars, else 1e-3.
alpha	Elastic-net mixing parameter: alpha = 1 is lasso, alpha = 0 is ridge.
data	Data frame containing model variables.
weights	Optional observation weights.
subset	Optional logical or numeric vector to subset observations.
offset	Optional vector of prior known components for the linear predictor.
contrasts	Optional contrast settings for factor variables.
unpenalized	Optional vector of variable names or indices excluded from penalization.
control	A list of control parameters via is.control .

Details

This function fits a regularization path of models using the induced smoothing paradigm, replacing the non-smooth L1 penalty with a differentiable surrogate. Standard errors are returned for all lambda points, allowing for Wald-based hypothesis testing. The regularization path spans a range of lambda values, either user-defined or automatically computed.

Value

A list with components:

call	Matched function call.
Info	Matrix with diagnostics: lambda, deviance, degrees of freedom, dispersion, iterations, convergence status.
GoF	Model goodness-of-fit metrics: AIC, BIC, AICc, GCV, GIC, eBIC.
Coef	Matrix of coefficients across lambda values.
SE	Matrix of standard errors.
Weights	Matrix of mixing weights for the smoothed penalty.
Gradient	Matrix of gradients for the smoothed penalty.
Linear.predictors, Fitted.values, Residuals	Matrices of fitted quantities across the path.
Input	List of input arguments and design matrix.
control, formula, model, terms, data, xlevels, contrasts	Standard model components.

Author(s)

Gianluca Sottile <gianluca.sottile@unipa.it>

References

Cilluffo G., Sottile G., La Grutta S., Muggeo V.M.R. (2019). *The Induced Smoothed Lasso: A practical framework for hypothesis testing in high dimensional regression*. Statistical Methods in Medical Research. DOI: 10.1177/0962280219842890

Sottile G., Cilluffo G., Muggeo V.M.R. (2019). *The R package islasso: estimation and hypothesis testing in lasso regression*. Technical Report. DOI: 10.13140/RG.2.2.16360.11521

See Also

[plot.islasso.path](#), [predict.islasso.path](#), [summary.islasso.path](#), [is.control](#), [GoF.islasso.path](#), [islasso](#), [simulXy](#)

Examples

```

n <- 100; p <- 30; p1 <- 10 # number of nonzero coefficients

beta.veri <- sort(round(c(seq(.5, 3, length.out = p1/2),
                        seq(-1, -2, length.out = p1/2)), 2))
beta <- c(beta.veri, rep(0, p - p1))
sim1 <- simulXy(n = n, p = p, beta = beta, seed = 1, family = gaussian())
o <- islasso.path(y ~ ., data = sim1$data,
                 family = gaussian(), nlambdas = 30L)

o

summary(o, lambda = 10, pval = 0.05)
coef(o, lambda = 10)
fitted(o, lambda = 10)
predict(o, type = "response", lambda = 10)
plot(o, yvar = "coef")
residuals(o, lambda = 10)
deviance(o, lambda = 10)
logLik(o, lambda = 10)
GoF.islasso.path(o)

## Not run:
##### binomial #####
beta <- c(1, 1, 1, rep(0, p - 3))
sim2 <- simulXy(n = n, p = p, beta = beta, inter = 1, seed = 1,
               size = 100, family = binomial())
o2 <- islasso.path(cbind(y.success, y.failure) ~ ., data = sim2$data,
                  family = binomial(), lambda = seq(0.1, 100, l = 50L))
temp <- GoF.islasso.path(o2)
summary(o2, pval = 0.05, lambda = temp$lambda.min["BIC"])

##### poisson #####
beta <- c(1, 1, 1, rep(0, p - 3))
sim3 <- simulXy(n = n, p = p, beta = beta, inter = 1, seed = 1,
               family = poisson())
o3 <- islasso.path(y ~ ., data = sim3$data, family = poisson(), nlambdas = 30L)
temp <- GoF.islasso.path(o3)
summary(o3, pval = 0.05, lambda = temp$lambda.min["BIC"])

##### Gamma #####
beta <- c(1, 1, 1, rep(0, p - 3))
sim4 <- simulXy(n = n, p = p, beta = beta, inter = -1, seed = 1,
               family = Gamma(link = "log"))
o4 <- islasso.path(y ~ ., data = sim4$data, family = Gamma(link = "log"),
                  nlambdas = 30L)
temp <- GoF.islasso.path(o4)
summary(o4, pval = .05, lambda = temp$lambda.min["BIC"])

## End(Not run)

```

`plot.islasso`*Diagnostic Plots for islasso Models*

Description

Produces standard diagnostic plots for a fitted `islasso` model to assess residuals, model fit, and variance structure.

Usage

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

Arguments

`x` An object of class "islasso", typically created via `islasso`.

`...` Additional graphical parameters passed to the underlying `plot()` functions.

Details

Generates a 2x2 grid of diagnostic plots:

- Top-left: Deviance residuals vs fitted values.
- Top-right: Normal Q-Q plot of standardized deviance residuals (red line = reference).
- Bottom-left: Squared standardized Pearson residuals vs fitted values.
- Bottom-right: Working response vector vs linear predictor.

These plots help assess the assumptions of linearity, homoscedasticity, and residual normality in penalized regression.

Author(s)

Gianluca Sottile <gianluca.sottile@unipa.it>

See Also

`islasso`, `summary.islasso`, `residuals.islasso`, `logLik.islasso`, `predict.islasso`, `deviance.islasso`

Examples

```
## Not run:
set.seed(1)
n <- 100; p <- 100
beta <- c(runif(20, -3, 3), rep(0, p - 20))
sim <- simulXy(n = n, p = p, beta = beta, seed = 1, family = gaussian())
fit <- islasso(y ~ ., data = sim$data, family = gaussian(), lambda = 2)
plot(fit)
```

```
## End(Not run)
```

plot.islasso.path	<i>Coefficient Profile and Diagnostic Plots for islasso.path</i>
-------------------	--

Description

Generates plots of coefficient profiles, standard errors, gradients, weights, or goodness-of-fit criteria from a fitted `islasso.path` model.

Usage

```
## S3 method for class 'islasso.path'
plot(
  x,
  yvar = c("coefficients", "se", "gradient", "weight", "gof"),
  gof = c("none", "AIC", "BIC", "AICc", "eBIC", "GCV", "GIC"),
  label = FALSE,
  legend = FALSE,
  ...
)
```

Arguments

<code>x</code>	An object of class "islasso.path", typically created via <code>islasso.path</code> .
<code>yvar</code>	Character. Specifies what to display on the y-axis. Choices are: "coefficients" - coefficient paths over $\log(\lambda)$, "se" - standard errors over $\log(\lambda)$, "gradient" - gradient values over $\log(\lambda)$, "weight" - mixture weights used in smoothing, "gof" - goodness-of-fit values.
<code>gof</code>	Character. Criterion used for highlighting active variables. Choices: "none", "AIC", "BIC", "AICc", "eBIC", "GCV", "GIC".
<code>label</code>	Logical. Whether to annotate curves with variable names.
<code>legend</code>	Logical. Whether to display a plot legend.
<code>...</code>	Additional graphical parameters, e.g. <code>main</code> , <code>xlab</code> , <code>ylab</code> , <code>xlim</code> , <code>ylim</code> , <code>lty</code> , <code>col</code> , <code>lwd</code> , <code>cex.axis</code> , <code>cex.lab</code> , <code>cex.main</code> , <code>gof_lty</code> , <code>gof_col</code> , <code>gof_lwd</code> .

Details

This function visualizes the behavior of the solution path across a sequence of λ values, helping diagnose coefficient shrinkage, influence of penalty, and variable selection stability.

Value

Produces plots. Does not return an object.

Author(s)

Gianluca Sottile <gianluca.sottile@unipa.it>

See Also

[islasso.path](#), [GoF.islasso.path](#), [summary.islasso.path](#), [coef.islasso.path](#), [fitted.islasso.path](#), [predict.islasso.path](#)

Examples

```
## Not run:
n <- 100; p <- 30
beta <- c(runif(10, -2, 2), rep(0, p - 10))
sim <- simulXy(n = n, p = p, beta = beta, seed = 1, family = gaussian())
fit <- islasso.path(y ~ ., data = sim$data, family = gaussian())

plot(fit, yvar = "coefficients", gof = "AICc", label = TRUE)
plot(fit, yvar = "se", gof = "AICc")
plot(fit, yvar = "gradient", gof = "AICc")
plot(fit, yvar = "gof", gof = "AICc")

## End(Not run)
```

predict.islasso

Prediction Method for islasso Objects

Description

Computes predictions from a fitted [islasso](#) model object. Multiple output types supported, including response scale, linear predictor, and coefficient values.

Usage

```
## S3 method for class 'islasso'
predict(
  object,
  newdata = NULL,
  type = c("link", "response", "coefficients", "class", "terms"),
  se.fit = FALSE,
  ci = NULL,
  type.ci = c("wald", "score"),
  level = 0.95,
  terms = NULL,
```

```

    na.action = na.pass,
    ...
)

```

Arguments

<code>object</code>	A fitted model of class "islasso".
<code>newdata</code>	Optional data frame containing predictors for prediction. If omitted, the fitted model matrix is used.
<code>type</code>	Character. Specifies the prediction scale: • "link" (default): linear predictor scale; • "response": original response scale; • "coefficients": estimated coefficients; • "class": predicted class (only for <code>binomial()</code> family); • "terms": contribution of each term to the linear predictor.
<code>se.fit</code>	Logical. Whether to compute standard errors/confidence intervals.
<code>ci</code>	Optional. Precomputed matrix of confidence intervals (2 columns).
<code>type.ci</code>	Type of interval. Only "wald" is implemented.
<code>level</code>	Confidence level for intervals. Default is 0.95.
<code>terms</code>	If <code>type = "terms"</code> , optionally specify which terms to extract.
<code>na.action</code>	Function to handle missing values in <code>newdata</code> . Default: <code>na.pass</code> .
<code>...</code>	Additional arguments passed to downstream methods.

Value

A numeric vector, matrix, or list depending on `type`.

Author(s)

Gianluca Sottile <gianluca.sottile@unipa.it>

See Also

[islasso](#), [summary.islasso](#), [logLik.islasso](#), [residuals.islasso](#), [deviance.islasso](#)

Examples

```

set.seed(1)
n <- 100; p <- 100
beta <- c(runif(20, -3, 3), rep(0, p - 20))
sim <- simulXy(n = n, p = p, beta = beta, seed = 1, family = gaussian())
fit <- islasso(y ~ ., data = sim$data, family = gaussian(), lambda = 2)
predict(fit, type = "response")

```

predict.islasso.path *Prediction Method for islasso.path Objects*

Description

Generates predictions from a fitted [islasso.path](#) model at one or more lambda values. Supports various output types including linear predictors, response scale, class labels, and coefficients.

Usage

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

Arguments

object	A fitted model object of class "islasso.path".
newdata	Optional data frame containing covariates for prediction. If omitted, returns fitted values from the original model.
type	Character. Type of prediction: • "link" (default) - linear predictor scale, • "response" - original response scale, • "coefficients" - estimated coefficients, • "class" - predicted class labels (only for binomial models).
lambda	Numeric value(s). Specific lambda value(s) at which predictions are required. If missing, predictions are computed for the full lambda sequence.
...	Additional arguments passed to lower-level methods.

Value

A vector, matrix, or list depending on the type requested.

Author(s)

Gianluca Sottile <gianluca.sottile@unipa.it>

See Also

[islasso.path](#), [summary.islasso.path](#), [coef.islasso.path](#), [GoF.islasso.path](#), [fitted.islasso.path](#), [logLik.islasso.path](#), [residuals.islasso.path](#), [deviance.islasso.path](#)

Examples

```
## Not run:
set.seed(1)
n <- 100; p <- 30
beta <- c(runif(10, -3, 3), rep(0, p - 10))
sim <- simulXy(n = n, p = p, beta = beta, seed = 1, family = gaussian())
fit <- islasso.path(y ~ ., data = sim$data, family = gaussian())
optimal <- GoF.islasso.path(fit)
pred <- predict(fit, type = "response", lambda = optimal$lambda.min)

## End(Not run)
```

Prostate

Prostate Cancer Data

Description

This dataset originates from a study examining the correlation between prostate-specific antigen levels and various clinical measures in men scheduled for radical prostatectomy. It contains 97 rows and 9 variables.

Format

A data frame with 97 observations and 9 variables:

lcavol Log of cancer volume
lweight Log of prostate weight
age Age of the patient
lbph Log of benign prostatic hyperplasia amount
svi Seminal vesicle invasion (binary)
lcp Log of capsular penetration
gleason Gleason score
pgg45 Percentage of Gleason scores 4 or 5
lpsa Log of prostate-specific antigen

Source

Stamey, T.A., et al. (1989). Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate: II. radical prostatectomy treated patients. *Journal of Urology*, 141(5), 1076-1083.

References

Stamey, T.A., Kabalin, J.N., McNeal, J.E., Johnstone, I.M., Freiha, F., Redwine, E.A., and Yang, N. (1989). *Journal of Urology*, 141(5), 1076-1083.

Examples

```
data(Prostate)
summary(Prostate)
cor(Prostate$lpsa, Prostate$lcavol)
## Not run:
  fit <- islasso(lpsa ~ ., data = Prostate, family = gaussian())
  summary(fit, pval = 0.05)
  lambda.aic <- aic.islasso(fit, method = "AIC")
  fit.aic <- update(fit, lambda = lambda.aic)
  summary(fit.aic, pval = 0.05)

## End(Not run)
```

relax.islasso	<i>Relaxed islasso model fitting</i>
---------------	--------------------------------------

Description

Fits a relaxed islasso model by unpenalizing a subset of variables selected either by statistical significance or by explicit user specification.

Usage

```
relax.islasso(fit, id = NULL, pval = 0.05, ...)
```

Arguments

fit	An object of class <code>islasso</code> , representing a fitted model.
id	Optional. Either: • A numeric vector of indices referring to variables to be kept unpenalized, or • A character vector of variable names to be kept unpenalized. If not provided, selection is based on the significance level given in <code>alpha</code> .
pval	Significance threshold (default 0.05). Variables with p-values less than or equal to <code>pval</code> are selected if <code>id</code> is missing.
...	Further arguments passed to <code>update</code> .

Details

The function creates a new `islasso` model where the selected variables are excluded from penalization. Selection can be made either through explicit variable indices/names via `id`, or automatically by thresholding p-values using `alpha`.

Value

An object of class `islasso`, representing the updated relaxed model.

Examples

```
## Not run:
fit <- islasso(y ~ x1 + x2 + x3, data = dat)
# Relaxed model keeping only variables with p <= 0.01
fit_relaxed <- relax.islasso(fit, alpha = 0.01)

# Relaxed model keeping variable "x1" unpenalized
fit_relaxed2 <- relax.islasso(fit, id = "x1")

## End(Not run)
```

simulXy

*Simulate Model Matrix and Response Vector***Description**

Generates synthetic covariates and response vector from a specified distribution for simulation studies or method validation.

Usage

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

Arguments

n	Integer. Number of observations.
p	Integer. Total number of covariates in the model matrix.
interc	Numeric. Intercept to include in the linear predictor. Default is 0.
beta	Numeric vector of length p. Regression coefficients in the linear predictor.
family	Distribution and link function. Allowed: <code>gaussian()</code> , <code>binomial()</code> , <code>poisson()</code> and <code>Gamma()</code> . Can be a string, function, or family object.

prop	Numeric in $[0, 1]$. Used only if beta is missing; proportion of non-zero coefficients in p. Default is 0.1.
lim.b	Numeric vector of length 2. Range for coefficients if beta is missing. Default: <code>c(-3, 3)</code> .
sigma	Standard deviation of Gaussian response. Default is 1.
size	Integer. Number of trials for binomial response. Default is 1.
rho	Numeric. Correlation coefficient for generating covariates. Used to create AR(1)-type covariance: $\rho^{ i-j }$. Default is 0.
scale.data	Logical. Whether to scale columns of the model matrix. Default is TRUE.
seed	Optional. Integer seed for reproducibility.
X	Optional. Custom model matrix. If supplied, it overrides the internally generated X.
dispersion	Dispersion parameter of Gamma response. Default is 0.1.

Value

A list with components:

X	Model matrix of dimension $n \times p$
y	Simulated response vector
beta	True regression coefficients used
eta	Linear predictor

Author(s)

Gianluca Sottile <gianluca.sottile@unipa.it>

See Also

[islasso](#), [islasso.path](#)

Examples

```
n <- 100; p <- 100
beta <- c(runif(10, -3, 3), rep(0, p - 10))
sim <- simulXy(n = n, p = p, beta = beta, seed = 1234)
o <- islasso(y ~ ., data = sim$data, family = gaussian())
summary(o, pval = 0.05)
```

summary.islasso	<i>Summarize islasso Fitted Model</i>
-----------------	---------------------------------------

Description

Provides a concise summary of a fitted [islasso](#) model, including p-values and optional filtering.

Usage

```
## S3 method for class 'islasso'  
summary(object, pval = 1, which, use.t = FALSE, type.pval = "wald", ...)
```

Arguments

object	A fitted model of class "islasso".
pval	Numeric threshold for displaying coefficients. Only those with $p \leq \text{pval}$ are printed. Unpenalized coefficients (like intercepts) are always shown.
which	Optional. Specifies a subset of coefficients to test. If missing, all parameters are evaluated.
use.t	Logical. If TRUE, p-values are computed using the t-distribution and residual degrees of freedom.
type.pval	Character. Type of p-value approximation. Only "wald" (default) is implemented.
...	Additional arguments (not currently used).

Value

An object of class "summary.islasso" containing:

coefficients	Coefficient estimates and related statistics
pval	Threshold used to filter coefficients
call	Original model call

Author(s)

Gianluca Sottile <gianluca.sottile@unipa.it>

See Also

[islasso.fit](#), [residuals.islasso](#), [logLik.islasso](#), [predict.islasso](#), [deviance.islasso](#)

Examples

```
## Not run:
# Assuming object `o` from an islasso fit
summary(o, pval = 0.1) # Show coefficients with p <= 0.1

## End(Not run)
```

summary.islasso.path *Summarize islasso.path Model at Specific Lambda*

Description

Extracts coefficient estimates, standard errors and p-values from an [islasso.path](#) fit at a given regularization level lambda.

Usage

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

Arguments

object	A fitted object of class "islasso.path".
pval	Numeric threshold for displaying coefficients. Only variables with p-value <= pval are printed. Unpenalized coefficients (like the intercept) are always shown.
use.t	Logical. If TRUE, p-values are computed using a t-distribution with residual degrees of freedom.
lambda	Numeric. Value of the regularization parameter at which the summary should be extracted.
...	Currently unused.

Value

An object of class "summary.islasso.path" containing filtered estimates and significance metrics.

Author(s)

Gianluca Sottile <gianluca.sottile@unipa.it>

See Also

[islasso.path](#), [GoF.islasso.path](#), [coef.islasso.path](#), [fitted.islasso.path](#), [predict.islasso.path](#), [residuals.islasso.path](#), [logLik.islasso.path](#), [deviance.islasso.path](#)

Examples

```
## Not run:  
# Assuming object `o` is from islasso.path  
summary(o, pval = 0.1, lambda = 5)  
  
## End(Not run)
```

Index

- * **datasets**
 - breast, [5](#)
 - diabetes, [7](#)
 - Prostate, [23](#)
- * **models**
 - islasso, [11](#)
- * **package**
 - islasso, [11](#)
- * **regression**
 - islasso, [11](#)

aic.islasso, [2](#), [13](#)
anova.islasso, [4](#), [13](#)

breast, [5](#)

coef.islasso.path, [9](#), [12](#), [20](#), [22](#), [28](#)
coef.islasso.path(islasso.path), [14](#)
confint.islasso, [5](#), [6](#), [13](#)
cooks.distance.islasso(islasso), [11](#)
cv.glmnet, [12](#)

deviance.islasso, [7](#), [12](#), [18](#), [21](#), [27](#)
deviance.islasso(islasso), [11](#)
deviance.islasso.path, [9](#), [22](#), [28](#)
deviance.islasso.path(islasso.path), [14](#)
diabetes, [7](#)

extractAIC.islasso(islasso), [11](#)

family.islasso(islasso), [11](#)
fitted.islasso.path, [9](#), [12](#), [20](#), [22](#), [28](#)
fitted.islasso.path(islasso.path), [14](#)
formula.islasso(islasso), [11](#)

GoF.islasso.path, [8](#), [16](#), [20](#), [22](#), [28](#)

influence.islasso(islasso), [11](#)
is.control, [3](#), [10](#), [12](#), [13](#), [15](#), [16](#)
islasso, [2–5](#), [10](#), [11](#), [11](#), [12](#), [16](#), [18](#), [20](#), [21](#),
[26](#), [27](#)

islasso-package(islasso), [11](#)
islasso.fit, [7](#), [27](#)
islasso.fit(islasso), [11](#)
islasso.path, [3](#), [8](#), [9](#), [11–13](#), [14](#), [19](#), [20](#), [22](#),
[26](#), [28](#)

logLik.islasso, [5](#), [7](#), [12](#), [18](#), [21](#), [27](#)
logLik.islasso(islasso), [11](#)
logLik.islasso.path, [9](#), [12](#), [22](#), [28](#)
logLik.islasso.path(islasso.path), [14](#)

model.frame.islasso(islasso), [11](#)
model.matrix.islasso(islasso), [11](#)
model.matrix.islasso.path
(islasso.path), [14](#)

nobs.islasso(islasso), [11](#)

plot.confint.islasso(confint.islasso),
[6](#)
plot.islasso, [13](#), [18](#)
plot.islasso.path, [16](#), [19](#)
predict.islasso, [5](#), [7](#), [12](#), [13](#), [18](#), [20](#), [27](#)
predict.islasso.path, [9](#), [12](#), [16](#), [20](#), [22](#), [28](#)
print.anova.islasso(anova.islasso), [4](#)
print.confint.islasso
(confint.islasso), [6](#)
print.islasso(islasso), [11](#)
print.islasso.path(islasso.path), [14](#)
print.logLik.islasso.path
(islasso.path), [14](#)
print.summary.islasso
(summary.islasso), [27](#)
print.summary.islasso.path
(summary.islasso.path), [28](#)
Prostate, [23](#)

relax.islasso, [13](#), [24](#)
residuals.islasso, [5](#), [7](#), [12](#), [18](#), [21](#), [27](#)
residuals.islasso(islasso), [11](#)
residuals.islasso.path, [9](#), [12](#), [22](#), [28](#)

`residuals.islasso.path(islasso.path),`
 14
`rstandard.islasso(islasso),` 11
`rstudent.islasso(islasso),` 11

`simulxy,` 3, 13, 16, 25
`summary.islasso,` 5, 7, 12, 13, 18, 21, 27
`summary.islasso.path,` 9, 12, 16, 20, 22, 28

`variable.names.islasso(islasso),` 11
`vcov.islasso(islasso),` 11

`weights.islasso(islasso),` 11