Package: plasso (via r-universe)

October 10, 2026

Type Package

Title Cross-Validated Post-Lasso

Version 0.1.3

Description Provides tools for cross-validated Lasso and Post-Lasso estimation. Built on top of the 'glmnet' package by Friedman, Hastie and Tibshirani (2010) <doi:10.18637/jss.v033.i01>, the main function plasso() extends the standard 'glmnet' output with coefficient paths for Post-Lasso models, while cv.plasso() performs cross-validation for both Lasso and Post-Lasso models and different ways to select the penalty parameter lambda as discussed in Knaus (2021) <doi:10.1111/rssa.12623>.

License GPL-3

VignetteBuilder knitr

Encoding UTF-8

URL https://github.com/MCKnaus/plasso

BugReports https://github.com/MCKnaus/plasso/issues

LazyData true

Imports glmnet, Matrix, methods, parallel, doParallel, foreach, iterators

RoxygenNote 7.3.3

Suggests testthat (>= 3.0.0), knitr, rmarkdown, xfun

Config/testthat/edition 3

Repository https://mcknaus.r-universe.dev

Date/Publication 2026-04-28 09:35:06 UTC

RemoteUrl https://github.com/mcknaus/plasso

RemoteRef HEAD

RemoteSha cf8208388919ca6c6be2d7625d7ea5f492b9654c

Contents

coef.cv.plasso . . . 2
coef.plasso . . . 3
cv.plasso . . . 4
plasso . . . 6
plot.cv.plasso . . . 7
plot.plasso . . . 8
predict.cv.plasso . . . 9
predict.plasso . . . 10
print.cv.plasso . . . 11
print.plasso . . . 12
print.summary.cv.plasso . . . 12
summary.cv.plasso . . . 13
summary.plasso . . . 14
toeplitz . . . 14

Index **16**

`coef.cv.plasso` *Extract coefficients from a* `cv.plasso` *object*

Description

Extract coefficients for both Lasso and Post-Lasso from a `cv.plasso` object.

Usage

```
## S3 method for class 'cv.plasso'
coef(object, ..., s = c("optimal", "all"), se_rule = 0)
```

Arguments

`object`	`cv.plasso` object
`...`	Pass generic `coef` options
`s`	Determines whether coefficients are extracted for all values of lambda ("all") or only for the optimal lambda ("optimal") according to the specified standard error-rule.
`se_rule`	If equal to 0, predictions from cross-validated MSE minimum (default). Negative values go in the direction of smaller models, positive values go in the direction of larger models (e.g. `se_rule=-1` creates the standard 1SE rule). This argument is not used for `s="all"`.

Value

List object containing coefficients for both the Lasso and Post-Lasso models respectively.

`lasso`	Sparse `dgCMatrix` with Lasso coefficients
`plasso`	Sparse `dgCMatrix` with Post-Lasso coefficients

Examples

```
# load toeplitz data
data(toeplitz)
# extract target and features from data
y = as.matrix(toeplitz[,1])
X = toeplitz[,-1]
# fit cv.plasso to the data
p.cv = plasso::cv.plasso(X,y)
# get estimated coefficients along whole lambda sequence
coefs = coef(p.cv, s="all")
head(coefs$plasso)
# get estimated coefficients for optimal lambda value according to 1-standard-error rule
coef(p.cv, s="optimal", se_rule=-1)
```

`coef.plasso` *Extract coefficients from a* `plasso` *object*

Description

Extract coefficients for both Lasso and Post-Lasso from a `plasso` object.

Usage

```
## S3 method for class 'plasso'
coef(object, ..., s = NULL)
```

Arguments

`object`	`plasso` object
`...`	Pass generic `coef` options
`s`	If Null, coefficients are returned for all lambda values. If a value is provided, the closest lambda value of the `plasso` object is used.

Value

List object containing coefficients that are associated with either all values along the lambda input sequence or for one specifically given lambda value for both the Lasso and Post-Lasso models respectively.

`lasso`	Sparse `dgCMatrix-class` object with Lasso coefficients
`plasso`	Sparse `dgCMatrix-class` object with Post-Lasso coefficients

Examples

```
# load toeplitz data
data(toeplitz)
# extract target and features from data
y = as.matrix(toeplitz[,1])
X = toeplitz[,-1]
# fit plasso to the data
p = plasso::plasso(X,y)
# get estimated coefficients along whole lambda sequence
coefs = coef(p)
head(coefs$plasso)
# get estimated coefficients for specific lambda approximation
coef(p, s=0.05)
```

`cv.plasso` *Cross-Validated Lasso and Post-Lasso*

Description

`cv.plasso` uses the `glmnet` package to estimate the coefficient paths and cross-validates least squares Lasso AND Post-Lasso.

Usage

```
cv.plasso(x, y, w = NULL, kf = 10, parallel = FALSE, ...)
```

Arguments

`x`	Matrix of covariates (number of observations times number of covariates matrix)
`y`	Vector of outcomes
`w`	Vector of weights
`kf`	Number of folds in k-fold cross-validation
`parallel`	Set as TRUE for parallelized cross-validation. Default is FALSE.
`...`	Pass `glmnet` options

Value

cv.plasso object (using a list structure) including the base `glmnet` object and cross-validation results (incl. optimal Lambda values) for both Lasso and Post-Lasso model.

`call`	the call that produced this
`lasso_full`	base `glmnet` object
`kf`	number of folds in k-fold cross-validation
`cv_MSE_lasso`	cross-validated MSEs of Lasso model (for every iteration of k-fold cross-validation)

cv_MSE_plasso	cross-validated MSEs of Post-Lasso model (for every iteration of k-fold cross-validation)
mean_MSE_lasso	averaged cross-validated MSEs of Lasso model
mean_MSE_plasso	averaged cross-validated MSEs of Post-Lasso model
ind_min_l	index of MSE optimal lambda value for Lasso model
ind_min_pl	index of MSE optimal lambda value for Post-Lasso model
lambda_min_l	MSE optimal lambda value for Lasso model
lambda_min_pl	MSE optimal lambda value for Post-Lasso model
names_l	Names of active variables for MSE optimal Lasso model
names_pl	Names of active variables for MSE optimal Post-Lasso model
coef_min_l	Coefficients for MSE optimal Lasso model
coef_min_pl	Coefficients for MSE optimal Post-Lasso model
x	Input matrix of covariates
y	Matrix of outcomes
w	Matrix of weights

Examples

```
# load toeplitz data
data(toeplitz)
# extract target and features from data
y = as.matrix(toeplitz[,1])
X = toeplitz[,-1]
# fit cv.plasso to the data
p.cv = plasso::cv.plasso(X,y)
# get basic summary statistics
print(summary(p.cv, default=FALSE))
# plot cross-validated MSE curves and number of active coefficients
plot(p.cv, legend_pos="bottomleft")
# get coefficients at MSE optimal lambda value for both Lasso and Post-Lasso model
coef(p.cv)
# get coefficients at MSE optimal lambda value according to 1-standard-error rule
coef(p.cv, se_rule=-1)
# predict fitted values along whole lambda sequence
pred = predict(p.cv, s="all")
head(pred$plasso)
```

plasso *Lasso and Post-Lasso*

Description

plasso implicitly estimates a Lasso model using the glmnet package and additionally estimates coefficient paths for a subsequent Post-Lasso model.

Usage

```
plasso(x, y, w = NULL, ...)
```

Arguments

x	Matrix of covariates (number of observations times number of covariates matrix)
y	Vector of outcomes
w	Vector of weights
...	Pass glmnet options

Value

List including base glmnet (i.e. Lasso) object and Post-Lasso coefficients.

call	the call that produced this
lasso_full	base glmnet object
beta_plasso	matrix of coefficients for Post-Lasso model stored in sparse column format
x	Input matrix of covariates
y	Matrix of outcomes
w	Matrix of weights

Examples

```
# load toeplitz data
data(toeplitz)
# extract target and features from data
y = as.matrix(toeplitz[,1])
X = toeplitz[,-1]
# fit plasso to the data
p = plasso::plasso(X,y)
# plot coefficient paths for Post-Lasso model
plot(p, lasso=FALSE, xvar="lambda")
# plot coefficient paths for Lasso model
plot(p, lasso=TRUE, xvar="lambda")
# get coefficients for specific lambda approximation
coef(p, s=0.05)
# predict fitted values along whole lambda sequence
```

```
pred = predict(p)
head(pred$plasso)
```

`plot.cv.plasso` *Plot of cross-validation curves*

Description

Plot of cross-validation curves.

Usage

```
## S3 method for class 'cv.plasso'
plot(
  x,
  ...,
  legend_pos = c("bottomright", "bottom", "bottomleft", "left", "topleft", "top",
    "topright", "right", "center"),
  legend_size = 0.5,
  lasso = FALSE
)
```

Arguments

`x`	`cv.plasso` object
`...`	Pass generic `plot` options
`legend_pos`	Legend position. Only considered for joint plot (lass=FALSE).
`legend_size`	Font size of legend
`lasso`	If set as True, only the cross-validation curve for the Lasso model is plotted. Default is False.

Value

Plots the cross-validation curves for both Lasso and Post-Lasso models (incl. upper and lower standard deviation curves) for a fitted `cv.plasso` object.

Examples

```
# load toeplitz data
data(toeplitz)
# extract target and features from data
y = as.matrix(toeplitz[,1])
X = toeplitz[,-1]
# fit cv.plasso to the data
p.cv = plasso::cv.plasso(X,y)
# plot cross-validated MSE curves and number of active coefficients
plot(p.cv, legend_pos="bottomleft")
```

`plot.plasso` *Plot coefficient paths*

Description

Plot coefficient paths of (Post-) Lasso model.

Usage

```
## S3 method for class 'plasso'
plot(x, ..., lasso = FALSE, xvar = c("norm", "lambda", "dev"), label = FALSE)
```

Arguments

`x`	`plasso` object
`...`	Pass generic `plot` options
`lasso`	If set as True, coefficient paths for Lasso instead of Post-Lasso is plotted. Default is False.
`xvar`	X-axis variable: `norm` plots against the L1-norm of the coefficients, `lambda` against the log-lambda sequence, and `dev` against the percent deviance explained.
`label`	If TRUE, label the curves with variable sequence numbers

Value

Produces a coefficient profile plot of the coefficient paths for a fitted `plasso` object.

Examples

```
# load toeplitz data
data(toeplitz)
# extract target and features from data
y = as.matrix(toeplitz[,1])
X = toeplitz[,-1]
# fit plasso to the data
p = plasso::plasso(X,y)
# plot coefficient paths for Post-Lasso model
plot(p, lasso=FALSE, xvar="lambda")
# plot coefficient paths for Lasso model
plot(p, lasso=TRUE, xvar="lambda")
```

`predict.cv.plasso` *Predict after cross-validated (Post-) Lasso*

Description

Prediction for cross-validated (Post-) Lasso.

Usage

```
## S3 method for class 'cv.plasso'
predict(
  object,
  ...,
  newx = NULL,
  type = c("response", "coefficients"),
  s = c("optimal", "all"),
  se_rule = 0
)
```

Arguments

`object`	Fitted `cv.plasso` model object
`...`	Pass generic `predict` options
`newx`	Matrix of new values for x at which predictions are to be made. If no value is supplied, x from fitting procedure is used. This argument is not used for `type="coefficients"`.
`type`	Type of prediction required. `"response"` returns fitted values, `"coefficients"` returns beta estimates.
`s`	Determines whether prediction is done for all values of lambda (`"all"`) or only for the optimal lambda (`"optimal"`) according to the standard error-rule.
`se_rule`	If equal to 0, predictions from cross-validated MSE minimum (default). Negative values go in the direction of smaller models, positive values go in the direction of larger models (e.g. `se_rule=-1` creates the standard 1SE rule). This argument is not used for `s="all"`.

Value

List object containing either fitted values or coefficients for both the Lasso and Post-Lasso models respectively.

`lasso`	Matrix with Lasso predictions or coefficients
`plasso`	Matrix with Post-Lasso predictions or coefficients

Examples

```
# load toeplitz data
data(toeplitz)
# extract target and features from data
y = as.matrix(toeplitz[,1])
X = toeplitz[,-1]
# fit cv.plasso to the data
p.cv = plasso::cv.plasso(X,y)
# predict fitted values along whole lambda sequence
pred = predict(p.cv, s="all")
head(pred$plasso)
# predict fitted values for optimal lambda value (according to cross-validation)
pred_optimal = predict(p.cv, s="optimal")
head(pred_optimal$plasso)
# predict fitted values for new feature set X
X_new = head(X, 10)
pred_new = predict(p.cv, newx=X_new, s="optimal")
pred_new$plasso
# get estimated coefficients along whole lambda sequence
coefs = predict(p.cv, type="coefficients", s="all")
head(coefs$plasso)
# get estimated coefficients for optimal lambda value according to 1-standard-error rule
predict(p.cv, type="coefficients", s="optimal", se_rule=-1)
```

`predict.plasso` *Predict for (Post-) Lasso models*

Description

Prediction for (Post-) Lasso models.

Usage

```
## S3 method for class 'plasso'
predict(
  object,
  ...,
  newx = NULL,
  type = c("response", "coefficients"),
  s = NULL
)
```

Arguments

`object`	Fitted `plasso` model object
`...`	Pass generic `predict` options

newx	Matrix of new values for x at which predictions are to be made. If no value is supplied, x from fitting procedure is used. This argument is not used for type="coefficients".
type	Type of prediction required. "response" returns fitted values, "coefficients" returns beta estimates.
s	If Null, prediction is done for all lambda values. If a value is provided, the closest lambda value of the plasso object is used.

Value

List object containing either fitted values or coefficients for both the Lasso and Post-Lasso models associated with all values along the lambda input sequence or for one specifically given lambda value.

lasso	Matrix with Lasso predictions or coefficients
plasso	Matrix with Post-Lasso predictions or coefficients

Examples

```
# load toeplitz data
data(toeplitz)
# extract target and features from data
y = as.matrix(toeplitz[,1])
X = toeplitz[,-1]
# fit plasso to the data
p = plasso::plasso(X,y)
# predict fitted values along whole lambda sequence
pred = predict(p)
head(pred$plasso)
# get estimated coefficients for specific lambda approximation
predict(p, type="coefficients", s=0.05)
```

---

print.cv.plasso *Print cross-validated (Post-) Lasso model*

Description

Printing main insights from cross-validated (Post-) Lasso model.

Usage

```
## S3 method for class 'cv.plasso'
print(x, ..., digits = max(3, getOption("digits") - 3))
```

Arguments

x	cv.plasso object
...	Pass generic print options
digits	Integer, used for number formatting

Value

Prints basic statistics for different lambda values of a fitted plasso object, i.e. cross-validated MSEs for both Lasso and Post-Lasso model as well as the number of active variables.

---

print.plasso *Print (Post-) Lasso model*

Description

Printing main insights from (Post-) Lasso model.

Usage

```
## S3 method for class 'plasso'
print(x, ..., digits = max(3, getOption("digits") - 3))
```

Arguments

x	plasso object
...	Pass generic print options
digits	Integer, used for number formatting

Value

Prints glmnet-like output.

---

print.summary.cv.plasso

Print summary of (Post-) Lasso model

Description

Prints summary information of cv.plasso object

Usage

```
## S3 method for class 'summary.cv.plasso'
print(x, ..., digits = max(3L, getOption("digits") - 3L))
```

Arguments

x	Summary of plasso object (either of class summary.cv.plasso or summary)
...	Pass generic R print options
digits	Integer, used for number formatting

Value

Prints information from summary.cv.plasso object into console.

---

summary.cv.plasso *Summary of cross-validated (Post-) Lasso model*

Description

Summary of cross-validated (Post-) Lasso model.

Usage

```
## S3 method for class 'cv.plasso'
summary(object, ..., default = FALSE)
```

Arguments

object	cv.plasso object
...	Pass generic summary summary options
default	TRUE for glmnet-like summary output, FALSE for more specific summary information

Value

For specific summary information: summary.cv.plasso object (using list structure) containing optimal lambda values and associated MSEs for both cross-validated Lasso and Post-Lasso model. For default: summaryDefault object.

Examples

```
# load toeplitz data
data(toeplitz)
# extract target and features from data
y = as.matrix(toeplitz[,1])
X = toeplitz[,-1]
# fit cv.plasso to the data
p.cv = plasso::cv.plasso(X,y)
# get informative summary statistics
print(summary(p.cv, default=FALSE))
# set default=TRUE for standard summary statistics
print(summary(p.cv, default=TRUE))
```

summary.plasso *Summary of (Post-) Lasso model*

Description

Summary of (Post-) Lasso model.

Usage

```
## S3 method for class 'plasso'
summary(object, ...)
```

Arguments

object	plasso object
...	Pass generic summary summary options

Value

Default summary object

toeplitz *Simulated 'Toeplitz' Data*

Description

Simulated data from a DGP with an underlying causal relationship between covariates X and the target y. The covariates matrix X consists of 10 variables whose effect size on target y is defined by the vector `c(1, -0.83, 0.67, -0.5, 0.33, -0.17, 0, ..., 0)` with the first six effect sizes decreasing in absolute terms continuously from 1 to 0 and alternating in their sign. The true causal effect of all other covariates is 0. The variables in X follow a normal distribution with mean zero while the covariance matrix follows a Toeplitz matrix. The target y is then a linear transformation of X plus a vector of standard normal random variables (i.e. error term). (See vignette for more details.)

Usage

```
data(toeplitz)
```

Format

An object of class `standardGeneric` of length 1.

Examples

```
# load toeplitz data
data(toeplitz)
# extract target and features from data
y = as.matrix(toeplitz[,1])
X = toeplitz[,-1]
# fit cv.plasso to the data
p.cv = plasso::cv.plasso(X,y)
```

Index

∗ **datasets**
toeplitz, 14

coef, *2*, *3*
coef.cv.plasso, 2
coef.plasso, 3
cv.plasso, *2*, 4, *4*, *7*, *9*, *12*, *13*

dgCMatrix, *2*

glmnet, *4*, *6*, *12*, *13*

plasso, *3*, 6, *6*, *8*, *10–12*, *14*
plot, *7*, *8*
plot.cv.plasso, 7
plot.plasso, 8
predict, *9*, *10*
predict.cv.plasso, 9
predict.plasso, 10
print, *12*, *13*
print.cv.plasso, 11
print.plasso, 12
print.summary.cv.plasso, 12

summary, *13*, *14*
summary.cv.plasso, 13, *13*
summary.plasso, 14
summaryDefault, *13*

toeplitz, 14