Package 'pcLasso'

May 9, 2026

Type Package

Title Principal Components Lasso

Version 1.2

Imports svd

Author Jerome Friedman, Kenneth Tay, Robert Tibshirani

Maintainer Rob Tibshirani <tibs@stanford.edu>

Description A method for fitting the entire regularization path
of the principal components lasso for linear and
logistic regression models. The algorithm uses cyclic coordinate descent
in a path-wise fashion. See URL below for more information on the algorithm.
See Tay, K., Friedman, J. ,Tibshirani, R., (2014) 'Principal component-guided sparse regression'
<doi:10.48550/arXiv.1810.04651>.

License GPL-3

URL https://arxiv.org/abs/1810.04651

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation yes

Repository CRAN

Date/Publication 2020-09-03 22:22:19 UTC

Contents

cv.pcLasso 2
pcLasso 4
plot.cv.pcLasso 7
predict.cv.pcLasso 8
predict.pcLasso 9

Index **11**

cv.pcLasso *Cross-validation for pcLasso*

Description

Does k-fold cross-validation for pcLasso.

Usage

```
cv.pcLasso(x, y, w = rep(1, length(y)), ratio = NULL, theta = NULL,
  groups = vector("list", 1), family = "gaussian", nfolds = 10,
  foldid = NULL, keep = FALSE, verbose = FALSE, ...)
```

Arguments

x	x matrix as in pcLasso.
y	y matrix as in pcLasso.
w	Observation weights. Default is 1 for each observation.
ratio	Ratio of shrinkage between the second and first principal components in the absence of the ℓ_1 penalty. More convenient way to specify the strength of the quadratic penalty. A value between 0 and 1 (only 1 included). ratio = 1 corresponds to the lasso, 0.5-0.9 are good values to use. Default is NULL. Exactly one of ratio or theta must be specified.
theta	Multiplier for the quadratic penalty: a non-negative real number. theta = 0 corresponds to the lasso, and larger theta gives strong shrinkage toward the top principal components. Default is NULL. Exactly one of ratio or theta must be specified.
groups	A list describing which features belong in each group. The length of the list should be equal to the number of groups, with groups[[k]] being a vector of feature/column numbers which belong to group k. Each feature must be assigned to at least one group. Features can belong to more than one group. By default, all the features belong to a single group.
family	Response type. Either "gaussian" (default) for linear regression or "binomial" for logistic regression.
nfolds	Number of folds for CV (default is 10). Although nfolds can be as large as the sample size (leave-one-out CV), it is not recommended for large datasets. Smallest value allowable is nfolds = 3.
foldid	An optional vector of values between 1 and nfold identifying what fold each observation is in. If supplied, nfold can be missing.
keep	If keep = TRUE, a prevalidated array is returned containing fitted values for each observation at each value of lambda. This means these fits are computed with this observation and the rest of its fold omitted. Default is FALSE.
verbose	Print out progess along the way? Default is FALSE.
...	Other arguments that can be passed to pcLasso.

Details

This function runs pcLasso nfolds+1 times: the first to get the lambda sequence, and the remaining nfolds times to compute the fit with each of the folds omitted. The error is accumulated, and the mean error and standard deviation over the folds is compued. Note that cv.pcLasso does NOT search for values of theta or ratio. A specific value of theta or ratio should be supplied.

Value

An object of class "cv.pcLasso", which is a list with the ingredients of the cross-validation fit.

glmfit	A fitted pcLasso object for the full data.
theta	Value of theta used in model fitting.
lambda	The values of lambda used in the fits.
nzero	If the groups overlap, the number of non-zero coefficients in the model glmfit for the expanded feature space, at each value of lambda. Otherwise, the number of non-zero coefficients in the model glmfit for the original feature space.
orignzero	If the groups are overlapping, this is the number of non-zero coefficients in the model glmfit for the original feature space, at each lambda. If groups are not overlapping, it is NULL.
fit.preval	If keep=TRUE, this is the array of prevalidated fits.
cvm	The mean cross-validated error: a vector of length length(lambda).
cvse	Estimate of standard error of cvm.
cvlo	Lower curve = cvm - cvsd.
cvup	Upper curve = cvm + cvsd.
lambda.min	The value of lambda that gives minimum cvm.
lambda.1se	The largest value of lambda such that the CV error is within one standard error of the minimum.
foldid	If keep=TRUE, the fold assignments used.
name	Name of error measurement used for CV.
call	The call that produced this object.

See Also

pcLasso and plot.cv.pcLasso.

Examples

```
set.seed(1)
x <- matrix(rnorm(100 * 20), 100, 20)
y <- rnorm(100)
groups <- vector("list", 4)
for (k in 1:4) {
    groups[[k]] <- 5 * (k-1) + 1:5
}
cvfit1 <- cv.pcLasso(x, y, groups = groups, ratio = 0.8)
```

```
# change no. of CV folds
cvfit2 <- cv.pcLasso(x, y, groups = groups, ratio = 0.8, nfolds = 5)
# specify which observations are in each fold
foldid <- sample(rep(seq(5), length = length(y)))
cvfit3 <- cv.pcLasso(x, y, groups = groups, ratio = 0.8, foldid = foldid)

# keep=TRUE to have pre-validated fits and foldid returned
cvfit4 <- cv.pcLasso(x, y, groups = groups, ratio = 0.8, keep = TRUE)
```

---

pcLasso *Fit a model with principal components lasso*

Description

Fit a model using the principal components lasso for an entire regularization path indexed by the parameter lambda. Fits linear and logistic regression models.

Usage

```
pcLasso(x, y, w = rep(1, length(y)), family = c("gaussian",
  "binomial"), ratio = NULL, theta = NULL, groups = vector("list",
  1), lambda.min.ratio = ifelse(nrow(x) < ncol(x), 0.01, 1e-04),
  nlam = 100, lambda = NULL, standardize = F, SVD_info = NULL,
  nv = NULL, propack = T, thr = 1e-04, maxit = 1e+05,
  verbose = FALSE)
```

Arguments

x	Input matrix, of dimension nobs x nvars; each row is an observation vector.
y	Response variable. Quantitative for family = "gaussian". For family="binomial", should be a numeric vector consisting of 0s and 1s.
w	Observation weights. Default is 1 for each observation.
family	Response type. Either "gaussian" (default) for linear regression or "binomial" for logistic regression.
ratio	Ratio of shrinkage between the second and first principal components in the absence of the ℓ_1 penalty. More convenient way to specify the strength of the quadratic penalty. A value between 0 and 1 (only 1 included). ratio = 1 corresponds to the lasso, 0.5-0.9 are good values to use. Default is NULL. Exactly one of ratio or theta must be specified.
theta	Multiplier for the quadratic penalty: a non-negative real number. theta = 0 corresponds to the lasso, and larger theta gives strong shrinkage toward the top principal components. Default is NULL. Exactly one of ratio or theta must be specified.

`groups` A list describing which features belong in each group. The length of the list should be equal to the number of groups, with `groups[[k]]` being a vector of feature/column numbers which belong to group k. Each feature must be assigned to at least one group. Features can belong to more than one group. By default, all the features belong to a single group.

`lambda.min.ratio`
Smallest value for `lambda`, as a fraction of the largest `lambda` value. The default depends on the sample size `nobs` relative to the number of variables `nvars`. If `nobs >= nvars`, the default is `0.0001`, close to zero. If `nobs < nvars`, the default is `0.01`. This is only used when the user does not specify a `lambda` sequence.

`nlam` Number of `lambda` values; default is 100.

`lambda` A user supplied `lambda` sequence. Typical usage is to have the program compute its own `lambda` sequence based on `nlam` and `lambda.min.ratio`; supplying a value of lambda overrides this.

`standardize` If `TRUE`, the columns of the feature matrices are standardized before the algorithm is run. Default is `FALSE`.

`SVD_info` A list containing SVD information. Usually this should not be specified by the user: the function will compute it on its own by default. `SVD_info` is a list with three elements:

1. `aa`: A row-wise concatenation of the k quadratic penalty matrices.
2. `d`: A list where the kth element is the vector of squared singular values of the input matrix corresponding to group k.
3. `dd`: A list where the kth element is the vector of $d_{k1}^2 - d_{kj}^2$ for the input matrix corresponding to group k.

Since the initial SVD of `x` can be the largest part of the overall computation, we allow the user to compute it once and then re-use. See example below.

`nv` Number of singular vectors to use in the singular value decompositions. If not specified, the full SVD is used.

`propack` If `TRUE` (default), uses `propack.svd` from the `svd` package to perform the singular value decompositions. If not, uses `svd` from base R.

`thr` Convergence threhold for the coordinate descent algorithm. Default is `1e-4`.

`maxit` Maximum number of passes over the data for all lambda values; default is `1e5`.

`verbose` Print out progess along the way? Default is `FALSE`.

Details

The objective function for `"gaussian"` is

$$1/2RSS/nobs + \lambda * ||\beta||_1 + \theta/2 \sum quadraticpenalty for group k,$$

where the sum is over the feature groups $1, ..., K$. The objective function for `"binomial"` is

$$-loglik/nobs + \lambda * ||\beta||_1 + \theta/2 \sum quadraticpenalty for group k.$$

`pcLasso` can handle overlapping groups. In this case, the original `x` matrix is expanded to a `nobs` x `p_1+...+p_K` matrix (where `p_k` is the number of features in group k) such that columns

p_1+...+p_{k-1}+1 to p_1+...+p_k represent the feature matrix for group k. pcLasso returns the model coefficients for both the expanded feature space and the original feature space.

One needs to specify the strength of the quadratic penalty either by specifying ratio, which is the ratio of shrinkage between the second and first principal components in the absence of the ℓ_1 penalty, or by specifying the multiplier theta. ratio is unitless and is more convenient.

pcLasso always mean centers the columns of the x matrix. If standardize=TRUE, pcLasso will also scale the columns to have standard deviation 1. In all cases, the beta coefficients returned are for the original x values (i.e. uncentered and unscaled).

Value

An object of class "pcLasso".

beta	If the groups overlap, a p_1+..._p_K x length(lambda) matrix of coefficients in the expanded feature space. If not, a nvars x length(lambda) matrix of coefficients in the original feature space.
origbeta	If the groups overlap, a nvars x length(lambda) matrix of coefficients in the original feature space. NULL if the groups do not overlap.
a0	Intercept sequence of length length(lambda).
lambda	The actual sequence of lambda values used.
nzero	If the groups overlap, the number of non-zero coefficients in the expanded feature space for each value of lambda. Otherwise, the number of non-zero coefficients in the original feature space.
orignzero	If the groups are overlapping, this is the number of non-zero coefficients in the original feature space of the model for each lambda. If groups are not overlapping, it is NULL.
jerr	Error flag for warnings and errors (largely for internal debugging).
theta	Value of theta used in model fitting.
origgroups	If the groups parameter was passed to the function call, this is a copy of that parameter. Otherwise, it is a list of length 1, with the first element being a vector of integers from 1 to nvars.
groups	If the groups are not overlapping, this has the same value as groups. If the groups are overlapping, then groups[[k]] is the vector from p_1 + ... + p_{k-1} + 1 to p_1 + ... p_k, where p_k is the number of features in group k.
SVD_info	A list containing SVD information. See param SVD_info for more information.
mx	If groups overlap, column means of the expanded x matrix. Otherwise, column means of the original x matrix.
origmx	Column means of the original x matrix.
my	If family = "gaussian", mean of the responses y. Otherwise, it is NA.
overlap	A logical flag indicating if the feature groups were overlapping or not.
nlp	Actual number of passes over the data for all lambda values.
family	Response type.
call	The call that produced this object.

Examples

```
set.seed(1)
x <- matrix(rnorm(100 * 20), 100, 20)
y <- rnorm(100)

# all features in one group by default
fit1 <- pcLasso(x, y, ratio = 0.8)
# print(fit1)  # Not run
# features in groups
groups <- vector("list", 4)
for (k in 1:4) {
    groups[[k]] <- 5 * (k-1) + 1:5
}
fit2 <- pcLasso(x, y, groups = groups, ratio = 0.8)
# groups can be overlapping
groups[[1]] <- 1:8
fit3 <- pcLasso(x, y, groups = groups, ratio = 0.8)

# specify ratio or theta, but not both
fit4 <- pcLasso(x, y, groups = groups, theta = 10)

# family = "binomial"
y2 <- sample(0:1, 100, replace = TRUE)
fit5 <- pcLasso(x, y2, ratio = 0.8, family = "binomial")

# example where SVD is computed once, then re-used
fit1 <- pcLasso(x, y, ratio = 0.8)
fit2 <- pcLasso(x, y, ratio = 0.8, SVD_info = fit1$SVD_info)
```

---

plot.cv.pcLasso *Plot the cross-validation curve produced by "cv.pcLasso" object*

Description

Plots the cross-validation curve produced by a `cv.pcLasso` object, along with upper and lower standard deviation curves, as a function of the `lambda` values used.

Usage

```
## S3 method for class 'cv.pcLasso'
plot(x, sign.lambda = 1, orignz = TRUE, ...)
```

Arguments

`x`	Fitted "`cv.pcLasso`" object.
`sign.lambda`	Either plot against `log(lambda)` (default) or `-log(lambda)` (if `sign.lambda = -1`).

orignz	If TRUE (default), prints the number of non-zero coefficients in the original feature space. If not, prints the number of non-zero coefficients in the expanded feature space. No effect if groups are not overlapping.
...	Other graphical paramters to plot.

Details

A plot is produced and nothing is returned.

See Also

pcLasso and cv.pcLasso.

Examples

```
set.seed(1)
x <- matrix(rnorm(100 * 20), 100, 20)
y <- rnorm(100)
groups <- vector("list", 4)
for (k in 1:4) {
    groups[[k]] <- 5 * (k-1) + 1:5
}
cvfit <- cv.pcLasso(x, y, ratio = 0.8, groups = groups)
plot(cvfit)
# plot flipped: x-axis tracks -log(lambda) instead
plot(cvfit, sign.lambda = -1)

# if groups overlap, orignz can be used to decide which space to count the
# number of non-zero coefficients at the top
groups[[1]] <- 1:8
cvfit <- cv.pcLasso(x, y, ratio = 0.8, groups = groups)
plot(cvfit)                  # no. of non-zero coefficients in original space
plot(cvfit, orignz = FALSE)  # no. of non-zero coefficients in expanded space
```

---

predict.cv.pcLasso *Make predictions from a "cv.pcLasso" object*

Description

This function returns the predictions for a new data matrix from a cross-validated pcLasso model by using the stored "glmfit" object and the optimal value chosen for lambda.

Usage

```
## S3 method for class 'cv.pcLasso'
predict(object, xnew, s = c("lambda.1se",
  "lambda.min"), ...)
```

Arguments

object	Fitted "cv.pcLasso" object.
xnew	Matrix of new values for x at which predictions are to be made.
s	Value of the penalty parameter lambda at which predictions are required. Default is the value s="lambda.1se" stored in the CV fit. Alternatively, s="lambda.min" can be used.
...	Potentially other arguments to be passed to and from methods; currently not in use.

Details

This function makes it easier to use the results of cross-validation to make a prediction. Note that xnew should have the same number of columns as the original feature space, regardless of whether the groups are overlapping or not.

Value

Predictions which the cross-validated model makes for xnew at the optimal value of lambda. Note that the default is the "lambda.1se" for lambda, to make this function consistent with cv.glmnet in the glmnet package. The output is predictions of $E(y|xnew)$: these are probabilities for the binomial family.

See Also

cv.pcLasso and predict.pcLasso.

Examples

```
set.seed(1)
x <- matrix(rnorm(100 * 20), 100, 20)
y <- rnorm(100)

cvfit <- cv.pcLasso(x, y, ratio = 0.8)
predict(cvfit, xnew = x[1:5, ])
predict(cvfit, xnew = x[1:5, ], s = "lambda.min")
```

---

predict.pcLasso *Make predictions from a "pcLasso" object*

Description

This function returns the predictions from a "pcLasso" object for a new data matrix.

Usage

```
## S3 method for class 'pcLasso'
predict(object, xnew, ...)
```

Arguments

object	Fitted "pcLasso" object.
xnew	Matrix of new values for x at which predictions are to be made.
...	Potentially other arguments to be passed to and from methods; currently not in use.

Details

Note that xnew should have the same number of columns as the original feature space, regardless of whether the groups are overlapping or not.

Value

Predictions of $E(y|xnew)$ which the model object makes at xnew. These are probabilities for the binomial family.

See Also

pcLasso.

Examples

```
set.seed(1)
x <- matrix(rnorm(100 * 20), 100, 20)

# family = "gaussian"
y <- rnorm(100)
fit1 <- pcLasso(x, y, ratio = 0.8)
predict(fit1, xnew = x[1:5, ])

# family = "binomial"
y2 <- sample(0:1, 100, replace = TRUE)
fit2 <- pcLasso(x, y2, ratio = 0.8, family = "binomial")
predict(fit2, xnew = x[1:5, ])
```

Index

cv.pcLasso, 2, *8, 9*

pcLasso, *3*, 4, *8*, *10*
plot.cv.pcLasso, *3*, 7
predict.cv.pcLasso, 8
predict.pcLasso, *9*, 9