Package ‘ncvreg’

April 27, 2017

Title   Regularization Paths for SCAD and MCP Penalized Regression Models
Version 3.9-1
Date    2017-04-26
Author  Patrick Breheny [aut,cre]
Maintainer Patrick Breheny <patrick-breheny@uiowa.edu>
Suggests survival, parallel, knitr
VignetteBuilder knitr
Description Efficient algorithms for fitting regularization paths for linear or logistic regression models penalized by MCP or SCAD, with optional additional L2 penalty.
BugReports http://github.com/pbreheny/ncvreg/issues
License GPL-3
NeedsCompilation yes
Repository CRAN
Date/Publication 2017-04-27 17:20:47 UTC

R topics documented:

ncvreg-package ........................................ 2
AUC.cv.ncvsurv ........................................ 3
cv.ncvreg ........................................... 4
cv.ncvsurv ........................................... 6
fir ..................................................... 8
heart ................................................... 8
Lung ................................................... 9
mfdr .................................................... 10
ncvreg ............................................... 12
ncvsurv ............................................... 15
perm.ncvreg ......................................... 18

1
Regularization paths for SCAD- and MCP-penalized regression models

Description

Efficient algorithms for fitting regularization paths for a variety of regression models (linear, logistic, Poisson, survival) penalized by MCP or SCAD, with optional additional L2 penalty.

Details

Accepts a design matrix $X$ and vector of responses $y$, produces the regularization path over a grid of values for the tuning parameter $\lambda$. Also provides methods for plotting, cross-validation-based inference, and for determining locally convex regions of the coefficients paths.

See the "Quick start guide" for a brief overview of how the package works.

Author(s)

Patrick Breheny <patrick-breheny@uiowa.edu>

References


Examples

vignette("quick-start", package="ncvreg")
Calculates the cross-validated AUC (concordance) from a "cv.ncvsurv" object.

Usage

```r
## S3 method for class 'cv.ncvsurv'
AUC(obj, ...)
```

Arguments

- `obj`: A `cv.ncvsurv` object. You must run `cv.ncvsurv` with the option `returny=TRUE` in order for `AUC` to work.
- `...`: For S3 method compatibility.

Details

The area under the curve (AUC), or equivalently, the concordance statistic (C), is calculated according to the procedure outlined in the reference below. This calls the `survConcordance` function in the `survival` package, except the cross-validated linear predictors are used to guard against over-fitting. Thus, the values returned by `AUC.cv.ncvsurv` will be lower than those you would obtain with `survConcordance` if you fit the full (unpenalized) model.

Author(s)

Patrick Breheny, Brandon Butcher, and Lawrence Hunsicker

References


See Also

cv.ncvsurv, survConcordance

Examples

```r
data(Lung)
X <- Lung$X
y <- Lung$y
cvfit <- cv.ncvsurv(X, y, returny=TRUE)
head(AUC(cvfit))
ll <- log(cvfit$fit$lambda)
```
plot(ll, AUC(cvfit), xlim=rev(range(ll)), lwd=3, type='l',
    xlab=expression(log(lambda)), ylab='AUC')

---

### cv.ncvreg

**Cross-validation for ncvreg**

**Description**

Performs k-fold cross validation for MCP- or SCAD-penalized regression models over a grid of values for the regularization parameter lambda.

**Usage**

```r
cv.ncvreg(X, y, ..., cluster, nfolds=10, seed, cv.ind, returnY=FALSE, trace=FALSE)
```

**Arguments**

- `X` The design matrix, without an intercept, as in `ncvreg`.
- `y` The response vector, as in `ncvreg`.
- `...` Additional arguments to `ncvreg`.
- `cluster` `cv.ncvreg` can be run in parallel across a cluster using the `parallel` package. The cluster must be set up in advance using the `makeCluster` function from that package. The cluster must then be passed to `cv.ncvreg` (see example).
- `nfolds` The number of cross-validation folds. Default is 10.
- `cv.ind` Which fold each observation belongs to. By default the observations are randomly assigned by `cv.ncvreg`.
- `seed` You may set the seed of the random number generator in order to obtain reproducible results.
- `returnY` Should `cv.ncvreg` return the fitted values from the cross-validation folds? Default is FALSE; if TRUE, this will return a matrix in which the element for row i, column j is the fitted value for observation i from the fold in which observation i was excluded from the fit, at the jth value of lambda.
- `trace` If set to TRUE, `cv.ncvreg` will inform the user of its progress by announcing the beginning of each CV fold. Default is FALSE.

**Details**

The function calls `ncvreg` `nfolds` times, each time leaving out 1/nfolds of the data. The cross-validation error is based on the residual sum of squares when family="gaussian" and the binomial deviance when family="binomial" or family="poisson".

For family="binomial" models, the cross-validation fold assignments are balanced across the 0/1 outcomes, so that each fold has the same proportion of 0/1 outcomes (or as close to the same proportion as it is possible to achieve if cases do not divide evenly).
Value

An object with S3 class "cv.ncvreg" containing:

cve  The error for each value of lambda, averaged across the cross-validation folds.
cvse The estimated standard error associated with each value of cve.
lambda The sequence of regularization parameter values along which the cross-validation error was calculated.
fit  The fitted ncvreg object for the whole data.
min  The index of lambda corresponding to lambda.min.
lambda.min The value of lambda with the minimum cross-validation error.
null.dev The deviance for the intercept-only model.
pe   If family=binomial, the cross-validation prediction error for each value of lambda.
Y    If returnY=TRUE, the matrix of cross-validated fitted values (see above).

Author(s)

Patrick Breheny <patrick-breheny@uiowa.edu>
Grant Brown helped with the parallelization support

References


See Also

ncvreg, plot.cv.ncvreg, summary.cv.ncvreg

Examples

data(prostate)
X <- as.matrix(prostate[,1:8])
y <- prostate$lpsa

cvfit <- cv.ncvreg(X, y)
plot(cvfit)
summary(cvfit)

fit <- cvfit$fit
plot(fit)
beta <- fit$beta[,cvfit$min]

## requires loading the parallel package
cv.ncvsurv

Cross-validation for ncvsurv

Description
Performs k-fold cross validation for MCP- or SCAD-penalized survival models over a grid of values for the regularization parameter lambda.

Usage
cv.ncvsurv(x, y, ..., cluster, nfolds=10, seed, returnY=FALSE, trace=FALSE)

Arguments
- `x`: The design matrix, as in ncvsurv.
- `y`: The response matrix, as in ncvsurv.
- `...`: Additional arguments to ncvsurv.
- `cluster`: cv.ncvsurv can be run in parallel across a cluster using the parallel package. The cluster must be set up in advance using the makeCluster function from that package. The cluster must then be passed to cv.ncvsurv (see example).
- `nfolds`: The number of cross-validation folds. Default is 10.
- `seed`: You may set the seed of the random number generator in order to obtain reproducible results.
- `returnY`: Should cv.ncvsurv return the linear predictors from the cross-validation folds? Default is FALSE; if TRUE, this will return a matrix in which the element for row i, column j is the fitted value for observation i from the fold in which observation i was excluded from the fit, at the jth value of lambda. NOTE: The rows of Y are ordered by time on study, and therefore do not correspond to the original order of observations passed to cv.ncvsurv.
- `trace`: If set to TRUE, cv.ncvsurv will inform the user of its progress by announcing the beginning of each CV fold. Default is FALSE.

Details
The function calls ncvsurv nfolds times, each time leaving out 1/nfolds of the data. Because of the semiparametric nature of Cox regression, cross-validation is not clearly defined. cv.ncvsurv uses the approach of calculating the full Cox partial likelihood using the cross-validated set of linear predictors. Unfortunately, using this approach there is no clear way (yet) of determining standard errors, so cv.ncvsurv, unlike cv.ncvreg, does not provide any.
Other approaches to cross-validation for the Cox regression model have been proposed; the strengths
and weaknesses of the various methods for penalized regression in the Cox model are not well un-
derstood. Because of this, the approach used by cv.ncvsurv may change in the future as additional
research is carried out.

Value

An object with S3 class "cv.ncvsurv" inheriting from "cv.ncvreg" and containing:

- cve: The error for each value of lambda, averaged across the cross-validation folds.
- cvse: NULL; see Details.
- lambda: The sequence of regularization parameter values along which the cross-validation
  error was calculated.
- fit: The fitted ncvsurv object for the whole data.
- min: The index of lambda corresponding to lambda.min.
- lambda.min: The value of lambda with the minimum cross-validation error.
- null.dev: The cross-validated deviance for the first model along the grid of lambda (i.e.,
  the cross-validated deviance for max(lambda), unless you have supplied your
  own lambda sequence, in which case this quantity is probably not meaningful).

Although the actual null deviance can be calculated, it cannot be compared with
the cross-validated deviance due to the manner in which deviance must be cal-
culated for Cox regression models (see details).

Author(s)

Patrick Breheny <patrick-breheny@uiowa.edu>

References

• Breheny P and Huang J. (2011) Coordinate descent algorithms for nonconvex penalized re-
gression, with applications to biological feature selection. Annals of Applied Statistics, 5:

  in Medicine, 12: 2305-2314.

See Also

ncvsurv, plot.cv.ncvreg, summary.cv.ncvreg

Examples

data(Lung)
X <- Lung$X
y <- Lung$y
cvfit <- cv.ncvsurv(X, y)
summary(cvfit)
plot(cvfit)
```
plot(cvfit, type="rsq")

## requires loading the parallel package
## Not run:
library(parallel)
cl <- makeCluster(4)
cvfit <- cv.ncvsurv(X, y, cluster=cl)
## End(Not run)
```

---

**fir**

*Marginal false discovery rates*

**Description**

Estimates the marginal false discovery rate (mFDR) of a penalized regression model.

**Usage**

```
fir(fit, ...)
```

**Arguments**

- `fit` An `ncvreg` or `ncvsurv` object.
- `...` Arguments to pass to `mfdr`.

**Details**

This function has been renamed and is currently deprecated. Use `mfdr` instead.

---

**heart**

*Risk factors associated with heart disease*

**Description**

Data from a subset of the Coronary Risk-Factor Study baseline survey, carried out in rural South Africa. The variables are as follows:

- `sbp`: Systolic blood pressure
- `tobacco`: Cumulative tobacco consumption, in kg
- `ldl`: Low-density lipoprotein cholesterol
- `adiposity`: Adipose tissue concentration
- `famhist`: Family history of heart disease (1=Present, 0=Absent)
- `typea`: Score on test designed to measure type-A behavior
- `obesity`: Obesity
- `alcohol`: Current consumption of alcohol
- `age`: Age of subject
- `chd`: Coronary heart disease at baseline; 1=Yes 0=No


**Lung**

**Usage**

data(heart)

**Format**

A data frame with 462 observations on 10 variables

**Source**


**References**


---

**Lung**

*VA lung cancer data set*

**Description**

Data from a randomised trial of two treatment regimens for lung cancer. This is a standard survival analysis data set from the classic textbook by Kalbfleisch and Prentice.

- **X**: A design matrix with 137 observations (rows) and 9 predictor variables (columns). The columns are:
  - trt: Treatment indicator (1=control group, 2=treatment group)
  - karno: Karnofsky performance score (0=bad, 100=good)
  - diagtime: Time from diagnosis to randomization (months)
  - age: Age (years)
  - prior: Prior therapy (0=no, 1=yes)
  - squamous: Indicator for whether the cancer type is squamous cell carcinoma (0=no, 1=yes)
  - small: Indicator for whether the cancer type is small cell lung cancer (0=no, 1=yes)
  - adeno: Indicator for whether the cancer type is adenocarcinoma (0=no, 1=yes)
  - large: Indicator for whether the cancer type is large cell carcinoma (0=no, 1=yes)
- **y**: A two column matrix (*Surv object*) containing the follow-up time (in days) and an indicator variable for whether the patient died while on the study or not.

**Usage**

data(Lung)
**Format**

A list containing the design matrix \( X \) and response matrix \( y \)

**Source**

https://cran.r-project.org/package=survival

**References**


**See Also**

ncvsurv

---

**mfdr**  
*Marginal false discovery rates*

**Description**

Estimates the marginal false discovery rate (mFDR) of a penalized regression model.

**Usage**

\[
\text{mfdr}(\text{fit}, X)
\]

**Arguments**

- \( \text{fit} \)  
  An \texttt{ncvreg} or \texttt{ncvsurv} object.
- \( X \)  
  The model matrix corresponding to \( \text{fit} \). This is not necessary for linear regression, but in logistic and Cox regression, the mFDR depends on \( X \). It is not necessary to supply \( X \) if it is already contained in \( \text{fit} \); i.e., if \texttt{ncvreg/ncvsurv} was run with \texttt{returnX=TRUE}.

**Details**

The function estimates the marginal false discovery rate (mFDR) for a penalized regression model. The estimate tends to be accurate in most settings, but will be slightly conservative if predictors are highly correlated. For an alternative way of estimating the mFDR, typically more accurate in highly correlated cases, see \texttt{perm.ncvreg}. 
Value

An object with S3 class mfdr inheriting from data.frame and containing:

- **EF**
  - The number of variables selected at each value of lambda, averaged over the permutation fits.

- **S**
  - The actual number of selected variables for the non-permuted data.

- **mFDR**
  - The estimated marginal false discovery rate (EF/S).

Author(s)

Patrick Breheny <patrick-breheny@uiowa.edu>, with contributions from Ryan Miller <ryan-e-miller@uiowa.edu>

See Also

ncvreg, ncvsurv, plot.mfdr, perm.ncvreg

Examples

```r

### Linear regression

data(prostate)
X <- as.matrix(prostate[,1:8])
y <- prostate$lpsa
fit <- ncvreg(X, y)

obj <- mfdr(fit)

# Comparison with perm.ncvreg
par(mfrow=c(2,2))
plot(obj)
plot(obj, type="EF")

pmfit <- perm.ncvreg(X, y)
plot(pmfit)
plot(pmfit, type="EF")

# Note that mfdr() is more conservative

### Logistic regression

data(heart)
X <- as.matrix(heart[,1:9])
y <- heart$chd
fit <- ncvreg(X, y, family="binomial")
obj <- mfdr(fit, X)

head(obj)
plot(obj)
plot(obj, type="EF")

### Cox regression

data(Lung)
X <- Lung$X
y <- Lung$y
```

```
ncvreg

**Fit an MCP- or SCAD-penalized regression path**

**Description**

Fit coefficients paths for MCP- or SCAD-penalized regression models over a grid of values for the regularization parameter lambda. Fits linear and logistic regression models, with option for an additional L2 penalty.

**Usage**

```r
ncvreg(x, y, family=c("gaussian", "binomial", "poisson"),
penalty=c("MCP", "SCAD", "lasso"), gamma=switch(penalty, SCAD=3.7, 3),
alpha=1, lambda.min=ifelse(n>p,.001,.05), nlambda=100, lambda, eps=1e-4,
max.iter=10000, convex=TRUE, dfmax=p+1, penalty.factor=rep(1, ncol(X)),
warn=TRUE, returnX=FALSE, ...)
```

**Arguments**

- **x**
  - The design matrix, without an intercept. `ncvreg` standardizes the data and includes an intercept by default.
- **y**
  - The response vector.
- **family**
  - Either "gaussian", "binomial", or "poisson", depending on the response.
- **penalty**
  - The penalty to be applied to the model. Either "MCP" (the default), "SCAD", or "lasso".
- **gamma**
  - The tuning parameter of the MCP/SCAD penalty (see details). Default is 3 for MCP and 3.7 for SCAD.
- **alpha**
  - Tuning parameter for the Mnet estimator which controls the relative contributions from the MCP/SCAD penalty and the ridge, or L2 penalty. `alpha=1` is equivalent to MCP/SCAD penalty, while `alpha=0` would be equivalent to ridge regression. However, `alpha=0` is not supported; `alpha` may be arbitrarily small, but not exactly 0.
- **lambda.min**
  - The smallest value for lambda, as a fraction of lambda.max. Default is .001 if the number of observations is larger than the number of covariates and .05 otherwise.
- **nlambda**
  - The number of lambda values. Default is 100.
- **lambda**
  - A user-specified sequence of lambda values. By default, a sequence of values of length `nlambda` is computed, equally spaced on the log scale.
eps Convergence threshold. The algorithm iterates until the RMSD for the change in linear predictors for any coefficient is less than eps. Default is $1e^{-4}$.

max.iter Maximum number of iterations (total across entire path). Default is 1000.

convex Calculate index for which objective function ceases to be locally convex? Default is TRUE.

dfmax Upper bound for the number of nonzero coefficients. Default is no upper bound. However, for large data sets, computational burden may be heavy for models with a large number of nonzero coefficients.

penalty.factor A multiplicative factor for the penalty applied to each coefficient. If supplied, penalty.factor must be a numeric vector of length equal to the number of columns of $X$. The purpose of penalty.factor is to apply differential penalization if some coefficients are thought to be more likely than others to be in the model. In particular, penalty.factor can be 0, in which case the coefficient is always in the model without shrinkage.

warn Return warning messages for failures to converge and model saturation? Default is TRUE.

returnX Return the standardized design matrix? Default is FALSE.

Details

The sequence of models indexed by the regularization parameter lambda is fit using a coordinate descent algorithm. For logistic regression models, some care is taken to avoid model saturation; the algorithm may exit early in this setting. The objective function is defined to be

$$\frac{1}{2n} \text{RSS} + \text{penalty}$$

for "gaussian" and

$$-\frac{1}{n} \ell + \text{penalty}$$

for "binomial" or "poisson", where the likelihood is from a traditional generalized linear model assuming the canonical link (logit for "binomial"; log for "poisson").

This algorithm is stable, very efficient, and generally converges quite rapidly to the solution. For GLMs, adaptive rescaling (see reference) is used.

The convexity diagnostics rely on a fine covering of $(\lambda_{\min},\lambda_{\max})$; choosing a low value of nlambda may produce unreliable results.

Value

An object with S3 class "ncvreg" containing:

beta The fitted matrix of coefficients. The number of rows is equal to the number of coefficients, and the number of columns is equal to nlambda.

iter A vector of length nlambda containing the number of iterations until convergence at each value of lambda.
lambda The sequence of regularization parameter values in the path.
penalty Same as above.
family Same as above.
gamma Same as above.
alpha Same as above.
convex.min The last index for which the objective function is locally convex. The smallest value of lambda for which the objective function is convex is therefore lambda[convex.min], with corresponding coefficients beta[,convex.min].
loss A vector containing either the residual sum of squares ("gaussian") or negative log-likelihood ("binomial" and "poisson") of the fitted model at each value of lambda.
penalty.factor Same as above.

Author(s)
Patrick Breheny <patrick-breheny@uiow.edu>

References

See Also
plot.ncvreg, cv.ncvreg

Examples
## Linear regression
data(prostate)
X <- as.matrix(prostate[,1:8])
y <- prostate$lpsa

par(mfrow=c(2,2))
fit <- ncvreg(X,y)
plot(fit,main=expression(paste(gamma,"=",3)))
fit <- ncvreg(X,y,gamma=10)
plot(fit,main=expression(paste(gamma,"=",10)))
fit <- ncvreg(X,y,gamma=1.5)
plot(fit,main=expression(paste(gamma,"=",1.5)))
fit <- ncvreg(X,y,penal="SCAD")
plot(fit,main=expression(paste("SCAD,"",gamma,"=",3)))

par(mfrow=c(2,2))
fit <- ncvreg(X,y)
plot(fit,main=expression(paste(alpha,"=",1)))
fit <- ncvreg(X,y,alpha=0.9)
plot(fit,main=expression(paste(alpha,"=",0.9)))
fit <- ncvreg(X,y,alpha=0.5)
ncvsurv

Fit coefficients paths for MCP- or SCAD-penalized Cox regression models over a grid of values for the regularization parameter lambda, with option for an additional L2 penalty.

Usage

ncvsurv(X, y, penalty=c("MCP", "SCAD", "lasso"),
gamma=switch(penalty, SCAD=3.7, 3), alpha=1,
lambda.min=ifelse(n>p,.001,.05), nlambda=100, lambda, eps=1e-4,
max.iter=10000, convex=TRUE, dfmax=p, penalty.factor=rep(1, ncol(x)),
warn=TRUE, returnX=FALSE, ...)

Arguments

X          The design matrix of predictor values. nCVsurv standardizes the data prior to fitting.

y          The time-to-event outcome, as a two-column matrix or Surv object. The first column should be time on study (follow up time); the second column should be a binary variable with 1 indicating that the event has occurred and 0 indicating (right) censoring.

penalty    The penalty to be applied to the model. Either "MCP" (the default), "SCAD", or "lasso".

gamma      The tuning parameter of the MCP/SCAD penalty (see details). Default is 3 for MCP and 3.7 for SCAD.

alpha      Tuning parameter for the Mnet estimator which controls the relative contributions from the MCP/SCAD penalty and the ridge, or L2 penalty. alpha=1 is equivalent to MCP/SCAD penalty, while alpha=0 would be equivalent to ridge regression. However, alpah=0 is not supported; alpha may be arbitrarily small, but not exactly 0.

lambda.min The smallest value for lambda, as a fraction of lambda.max. Default is .001 if the number of observations is larger than the number of covariates and .05 otherwise.

dlambdas   The number of lambda values. Default is 100.

lambda     A user-specified sequence of lambda values. By default, a sequence of values of length nlambda is computed, equally spaced on the log scale.

eps        Convergence threshold. The algorithm iterates until the RMSD for the change in linear predictors for any coefficient is less than eps. Default is 1e-4.

max.iter   Maximum number of iterations (total across entire path). Default is 1000.

convex     Calculate index for which objective function ceases to be locally convex? Default is TRUE.

dfmax      Upper bound for the number of nonzero coefficients. Default is no upper bound. However, for large data sets, computational burden may be heavy for models with a large number of nonzero coefficients.

penalty.factor A multiplicative factor for the penalty applied to each coefficient. If supplied, penalty.factor must be a numeric vector of length equal to the number of columns of X. The purpose of penalty.factor is to apply differential penalization if some coefficients are thought to be more likely than others to be in the model. In particular, penalty.factor can be 0, in which case the coefficient is always in the model without any penalization/shrinkage.

warn       Return warning messages for failures to converge and model saturation? Default is TRUE.

returnX    Return the standardized design matrix? Default is FALSE.

...        Not used.
ncvsurv

Details

The sequence of models indexed by the regularization parameter \( \lambda \) is fit using a coordinate descent algorithm. In order to accomplish this, the second derivative (Hessian) of the Cox partial log-likelihood is diagonalized (see references for details). The objective function is defined to be

\[
-\frac{1}{n} L(\beta | X, y) + \text{penalty},
\]

where \( L \) is the partial log-likelihood from the Cox regression model.

Presently, ties are not handled by ncvsurv in a particularly sophisticated manner. This will be improved upon in a future release of ncvreg.

Adaptive rescaling (see references) is used for MCP and SCAD models. The convexity diagnostics rely on a fine covering of \((\lambda_{\text{min}}, \lambda_{\text{max}})\); choosing a low value of \( n_{\lambda} \) may produce unreliable results.

Value

An object with S3 class "ncvsurv" containing:

- \textbf{beta} The fitted matrix of coefficients. The number of rows is equal to the number of coefficients, and the number of columns is equal to \( n_{\lambda} \).
- \textbf{iter} A vector of length \( n_{\lambda} \) containing the number of iterations until convergence at each value of \( \lambda \).
- \textbf{lambda} The sequence of regularization parameter values in the path.
- \textbf{penalty} Same as above.
- \textbf{model} Same as above.
- \textbf{gamma} Same as above.
- \textbf{alpha} Same as above.
- \textbf{convex.min} The last index for which the objective function is locally convex. The smallest value of \( \lambda \) for which the objective function is convex is therefore \( \lambda[\text{convex.min}] \), with corresponding coefficients \( \beta[., \text{convex.min}] \).
- \textbf{loss} The negative partial log-likelihood of the fitted model at each value of \( \lambda \).
- \textbf{penalty.factor} Same as above.
- \textbf{n} The number of observations.

For Cox models, the following objects are also returned (and are necessary to estimate baseline survival conditional on the estimated regression coefficients), all of which are ordered by time on study. I.e., the \( i \)th row of \( \mathbf{w} \) does not correspond to the \( i \)th row of \( X \):

- \textbf{w} Matrix of \( \exp(\beta) \) values for each subject over all \( \lambda \) values.
- \textbf{time} Times on study.
- \textbf{fail} Failure event indicator.

Author(s)

Patrick Breheny <patrick-breheny@uiowa.edu>
perm.ncvreg

Permutation fitting for ncvreg

Description

Fits multiple penalized regression models in which the outcome is randomly permuted, thereby allowing estimation of the false inclusion rate.

References


See Also

plot.ncvreg, cv.ncvsurv

Examples

data(Lung)
X <- Lung$X
y <- Lung$y

par(mfrow=c(2,2))
fit <- ncvsurv(X, y)
plot(fit, main=expression(paste(gamma,"=",3)))
fit <- ncvsurv(X, y, gamma=10)
plot(fit, main=expression(paste(gamma,"=",10)))
fit <- ncvsurv(X, y, gamma=1.5)
plot(fit, main=expression(paste(gamma,"=",1.5)))
fit <- ncvsurv(X, y, penalty="SCAD")
plot(fit, main=expression(paste("SCAD","=",3)))

fit <- ncvsurv(X,y)
l1 <- log(fit$lambda)
par(mfrow=c(2,1))
plot(l1, BIC(fit), type="l", xlab=expression(range(l1)))
lam <- fit$lambda[which.min(BIC(fit))]
b <- coef(fit, lambda=lam)
b[b!=0]
plot(fit)
abline(v=lam)

S <- predict(fit, X, type='survival', lambda=lam)
par(mfrow=c(1,1))
plot(S, xlim=c(0,200))
Usage

perm.ncvreg(X, y, ..., permute=c("outcome", "residuals"), N=10, seed, trace=FALSE)

Arguments

x
  The design matrix, without an intercept, as in ncvreg.
y
  The response vector, as in ncvreg.
...
  Additional arguments to ncvreg.
permute
  What to permute. If 'outcome', the response vector, y, is permuted. If 'residuals',
  the residuals are permuted. This is only available for linear regression (i.e., for
  family='gaussian'). Note that permuting the residuals may take a long time,
  as the residuals differ for each value of lambda, so separate permutations are
  required at every value of lambda. See also permres.
N
  The number of permutation replications. Default is 10.
seed
  You may set the seed of the random number generator in order to obtain repro-
  ducible results.
trace
  If set to TRUE, perm.ncvreg will inform the user of its progress by announcing
  the beginning of each permutation fit. Default is FALSE.

Details

The function fits a penalized regression model to the actual data, then repeats the process N times
with a permuted version of the response vector. This allows estimation of the expected number of
variables included by chance for each value of lambda. The ratio of this expected quantity to the
number of selected variables using the actual (non-permuted) response is called the marginal false
discovery rate (mFDR).

Value

An object with S3 class "perm.ncvreg" containing:

EF
  The number of variables selected at each value of lambda, averaged over the
  permutation fits.
S
  The actual number of selected variables for the non-permuted data.
mFDR
  The estimated marginal false discovery rate (EF/S).
fit
  The fitted ncvreg object for the original (non-permuted) data.
loss
  The loss/deviance for each value of lambda, averaged over the permutation fits.
  This is an estimate of the explanatory power of the model under null conditions,
  and can be used to adjust the loss of the fitted model in a manner akin to the idea
  of an adjusted R-squared in classical regression.

Author(s)

Patrick Breheny <patrick-breheny@uiowa.edu>
permres

Permute residuals for a fitted ncvreg model

Description

Fits multiple penalized regression models in which the residuals are randomly permuted, thereby allowing estimation of the false inclusion rate.

Usage

permres(fit, ...)

## S3 method for class 'ncvreg'
permres(fit, lambda, N=10, seed, trace=FALSE, ...)

Examples

## Linear regression
data(prostate)
X <- as.matrix(prostate[,1:8])
y <- prostate$lpsa
pmfit <- perm.ncvreg(X, y)

par(mfcol=c(2,2))
plot(pmfit)
plot(pmfit, type="EF")
plot(pmfit$fit)

pmfit.r <- perm.ncvreg(X, y, permute='residuals')
plot(pmfit.r, col="red")
lines(pmfit$fit$lambda, pmfit$mFDR, col="gray60")

## Logistic regression
data(heart)
X <- as.matrix(heart[,1:9])
y <- heart$chd
pmfit <- perm.ncvreg(X, y, family="binomial")

par(mfrow=c(2,2))
plot(pmfit)
plot(pmfit, type="EF")
plot(pmfit$fit)

See Also

ncvreg, plot.mfdr, mfdr
Arguments

fit A fitted ncvreg model, as produced by \texttt{ncvreg()}. To use with \texttt{permres}, the model must be fit using the \texttt{returnX=TRUE} option.

lambda The regularization parameter to use for estimating residuals. Unlike \texttt{perm.ncvreg}, \texttt{permres} calculates EF and FIR for a specific \texttt{lambda} value, not an entire path. As a result, it runs much faster.

N The number of permutation replications. Default is 10.

seed You may set the seed of the random number generator in order to obtain reproducible results.

trace If set to TRUE, \texttt{perm.ncvreg} will inform the user of its progress by announcing the beginning of each permutation fit. Default is FALSE.

Details

The function fits a penalized regression model to the actual data, then repeats the process \texttt{N} times with a permuted version of the response vector. This allows estimation of the expected number of variables included by chance for each value of \texttt{lambda}. The ratio of this expected quantity to the number of selected variables using the actual (non-permuted) response is called the false inclusion rate (FIR).

Value

A list with the following components:

\textbf{EF} The number of variables selected at each value of \texttt{lambda}, averaged over the permutation fits.

\textbf{S} The actual number of selected variables for the non-permuted data.

\textbf{FIR} The estimated false inclusion rate (\texttt{EF}/\texttt{S}).

\textbf{loss} The loss/deviance, averaged over the permutation fits. This is an estimate of the explanatory power of the model under null conditions, and can be used to adjust the loss of the fitted model in a manner akin to the idea of an adjusted R-squared in classical regression.

Author(s)

Patrick Breheny <patrick-breheny@uiowa.edu>

See Also

\texttt{ncvreg}, \texttt{fir}, \texttt{perm.ncvreg}
Examples

```r
data(prostate)
X <- as.matrix(prostate[,1:8])
y <- prostate$lpsa
fit <- ncvreg(X, y, returnX = TRUE, N=50)
permres(fit, lambda=0.15)
```

Description

Plots the cross-validation curve from a "cv.ncvreg" object, along with standard error bars.

Usage

```r
# S3 method for class 'cv.ncvreg'
plot(x, log.l=TRUE, type=c("cve", "rsq", "scale", "snr", "pred", "all"), selected=TRUE, vertical.line=TRUE, col="red", ...)
```

Arguments

- `x`: A "cv.ncvreg" object.
- `log.l`: Should horizontal axis be on the log scale? Default is TRUE.
- `type`: What to plot on the vertical axis. cve plots the cross-validation error (deviance); rsq plots an estimate of the fraction of the deviance explained by the model (R-squared); scale plots an estimate of the signal-to-noise ratio; snr plots an estimate of the scale parameter (standard deviation); pred plots, family="binomial", the estimated prediction error; all produces all of the above.
- `selected`: If TRUE (the default), places an axis on top of the plot denoting the number of variables in the model (i.e., that have a nonzero regression coefficient) at that value of lambda.
- `vertical.line`: If TRUE (the default), draws a vertical line at the value where cross-validation error is minimized.
- `col`: Controls the color of the dots (CV estimates).
- `...`: Other graphical parameters to plot

Details

Error bars representing approximate 68% confidence intervals are plotted along with the estimates at value of lambda. For rsq and snr, these confidence intervals are quite crude, especially near zero, and will hopefully be improved upon in later versions of ncvreg.
## Description

Plot marginal false discovery rate curves from an "mfdr" or "perm.ncvreg" object.

## Usage

```r
## S3 method for class 'mfdr'
plot(x, type=c("mFDR", "EF"), log.l=FALSE, selected=TRUE, legend=TRUE, ...)```

**Author(s)**

Patrick Breheny <patrick-breheny@uiowa.edu>

**References**


**See Also**

ncvreg, cv.ncvreg

**Examples**

```
## Linear regression
data(prostate)
X <- as.matrix(prostate[,1:8])
y <- prostate$lpsa
cvfit <- cv.ncvreg(X, y)
plot(cvfit)
par(mfrow=c(2,2))
plot(cvfit, type="all")

## Logistic regression
data(heart)
X <- as.matrix(heart[,1:9])
y <- heart$chd
cvfit <- cv.ncvreg(X, y, family="binomial")
plot(cvfit)
par(mfrow=c(2,2))
plot(cvfit, type="all")
```
Arguments

- **x**: A "perm.ncvreg" or "mfdr" object.
- **type**: What to plot on the vertical axis. mFDR plots the marginal false discovery rate; EF plots the expected number of false discoveries along with the actual number of variables included in the model.
- **log.l**: Should horizontal axis be on the log scale? Default is FALSE.
- **selected**: If TRUE (the default), places an axis on top of the plot denoting the number of variables in the model (i.e., that have a nonzero regression coefficient) at that value of lambda.
- **legend**: For type="EF" plots, draw a legend to indicate which line is for the actual selections and which line is for the expected number of false discoveries? Default is TRUE.
- **...**: Other graphical parameters to pass to plot

Author(s)

Patrick Breheny <patrick-breheny@uiowa.edu>

See Also

- `mfdr`, `perm.ncvreg`

Examples

```r
data(prostate)
X <- as.matrix(prostate[,1:8])
y <- prostate$lpsa
fit <- ncvreg(X, y)

obj <- mfdr(fit)
obj[1:10,]

# Some plotting options
plot(obj)
plot(obj, type="EF")
plot(obj, log=TRUE)

# Comparison with perm.ncvreg
par(mfrow=c(2,2))
plot(obj)
plot(obj, type="EF")
pmfit <- perm.ncvreg(X, y)
plot(pmfilt)
plot(pmfilt, type="EF")
## Note that mfdr() is more conservative
```
plot.ncvreg  

Plot coefficients from a "ncvreg" object

Description

Produces a plot of the coefficient paths for a fitted "ncvreg" object.

Usage

```r
## S3 method for class 'ncvreg'
plot(x, alpha=1, log.1=FALSE, shade=TRUE, ...)
```

Arguments

- `x`: Fitted "ncvreg" model.
- `alpha`: Controls alpha-blending, helpful when the number of covariates is large. Default is `alpha=1`.
- `log.1`: Should horizontal axis be on the log scale? Default is `FALSE`.
- `shade`: Should nonconvex region be shaded? Default is `TRUE`.
- `...`: Other graphical parameters to `plot`

Author(s)

Patrick Breheny <patrick-breheny@uiowa.edu>

References


See Also

- `ncvreg`

Examples

```r
data(prostate)
X <- as.matrix(prostate[,1:8])
y <- prostate$lpsa

fit <- ncvreg(X,y)
plot(fit)
plot(fit,col="black")
plot(fit,log=TRUE)
```
plot.ncvsurv.func  

Plot survival curve for ncvsurv model

Description

Plot survival curve for a model that has been fit using ncvsurv followed by a prediction of the survival function using predict.ncvsurv

Usage

## S3 method for class 'ncvsurv.func'
plot(x, alpha=1, ...)

Arguments

x  
A 'ncvsurv.func' object, which is returned by predict.ncvsurv if type='survival' is specified. See examples.

alpha  
Controls alpha-blending (i.e., transparency). Useful if many overlapping lines are present.

...  
Other graphical parameters to pass to plot

Author(s)

Patrick Breheny <patrick-brehe\@uiowa.edu>

See Also

ncvsurv, predict.ncvsurv

Examples

data(Lung)
X <- Lung$X
y <- Lung$y

fit <- ncvsurv(X, y)

# A single survival curve
S <- predict(fit, X[,1], type='survival', lambda=.15)
plot(S, xlim=c(0,200))

# Lots of survival curves
S <- predict(fit, X, type='survival', lambda=.15)
plot(S, xlim=c(0,200))
predict.ncvreg

Model predictions based on a fitted "ncvreg" object.

Description

Similar to other predict methods, this function returns predictions from a fitted "ncvreg" object.

Usage

```r
## S3 method for class 'ncvreg'
predict(object, X, type=c("link", "response", "class", "coefficients", "vars", "nvars"), lambda, which=1:length(object$lambda), ...)
## S3 method for class 'ncvreg'
coef(object, lambda, which=1:length(object$lambda), drop=TRUE, ...)
```

Arguments

- `object`: Fitted "ncvreg" model object.
- `X`: Matrix of values at which predictions are to be made. Not used for `type="coefficients"` or for some of the `type` settings in `predict`.
- `lambda`: Values of the regularization parameter lambda at which predictions are requested. For values of `lambda` not in the sequence of fitted models, linear interpolation is used.
- `which`: Indices of the penalty parameter lambda at which predictions are required. By default, all indices are returned. If `lambda` is specified, this will override `which`.
- `type`: Type of prediction: "link" returns the linear predictors; "response" gives the fitted values; "class" returns the binomial outcome with the highest probability; "coefficients" returns the coefficients; "vars" returns a list containing the indices and names of the nonzero variables at each value of `lambda`; "nvars" returns the number of nonzero coefficients at each value of `lambda`.
- `drop`: If coefficients for a single value of `lambda` are to be returned, reduce dimensions to a vector? Setting `drop=FALSE` returns a 1-column matrix.
- `...`: Not used.

Value

The object returned depends on `type`.

Author(s)

Patrick Breheny <patrick-breheny@uiowa.edu>
References


See Also

ncvreg

Examples

data(heart)
X <- as.matrix(heart[,1:9])
y <- heart$chd

fit <- ncvreg(X,y,family="binomial")
coef(fit, lambda=0.05)
head(predict(fit, X, type="link", lambda=0.05))
head(predict(fit, X, type="response", lambda=0.05))
head(predict(fit, X, type="class", lambda=0.05))
predict(fit, type="vars", lambda=c(0.05, 0.01))
predict(fit, type="nvars", lambda=c(0.05, 0.01))
Type of prediction: "link" returns the linear predictors; "response" gives the risk (i.e., exp(link)); "survival" returns the estimated survival function; "median" estimates median survival times. The other options are all identical to their ncvreg counterparts: "coefficients" returns the coefficients; "vars" returns a list containing the indices and names of the nonzero variables at each value of lambda; "nvars" returns the number of nonzero coefficients at each value of lambda.

Not used.

Details

Estimation of baseline survival function conditional on the estimated values of beta is carried out according to the method described in Chapter 4.3 of Kalbfleish and Prentice. In particular, it agrees exactly the results returned by survfit.coxph(..., type='kalbfleisch-prentice') in the survival package.

Value

The object returned depends on type.

Author(s)

Patrick Breheny <patrick-breheny@uiowa.edu>

References


See Also

ncvsurv

Examples

data(Lung)
X <- Lung$X
y <- Lung$y

fit <- ncvsurv(X,y)
coef(fit, lambda=0.05)
head(predict(fit, X, type="link", lambda=0.05))
head(predict(fit, X, type="response", lambda=0.05))

# Survival function
S <- predict(fit, X[,1], type="survival", lambda=0.05)
S(100)
S <- predict(fit, X, type="survival", lambda=0.05)
plot(S, xlim=c(0,200))

# Medians
predict(fit, X[1,], type="median", lambda=0.05)
M <- predict(fit, X, type="median")
M[1:10, 1:10]

# Nonzero coefficients
predict(fit, type="vars", lambda=c(0.1, 0.01))
predict(fit, type="nvars", lambda=c(0.1, 0.01))

---

prostate  
Factors associated with prostate specific antigen

Description

Data from a study by Stamey et al. (1989) to examine the association between prostate specific antigen (PSA) and several clinical measures that are potentially associated with PSA in men who were about to receive a radical prostatectomy. The variables are as follows:

- lcavol: Log cancer volume
- lweight: Log prostate weight
- age: The man’s age
- lbph: Log of the amount of benign hyperplasia
- svi: Seminal vesicle invasion; 1=Yes, 0=No
- lcp: Log of capsular penetration
- gleason: Gleason score
- pgg45: Percent of Gleason scores 4 or 5
- lpsa: Log PSA

Usage

data(prostate)

Format

A data frame with 97 observations on 9 variables

Source


References

Description

The function `std` accepts a design matrix and returns a standardized version of that matrix (i.e., each column will have mean 0 and mean sum of squares equal to 1).

Usage

```
std(X)
```

Arguments

- `X` A matrix (or object that can be coerced to a matrix, such as a data frame).

Details

This function centers and scales each column of `X` so that

\[ \sum_{i=1}^{n} x_{ij} = 0 \]

and

\[ n^{-1} \sum_{i=1}^{n} x_{ij}^2 = 1 \]

for all `j`. This is usually not necessary to call directly, as `ncvreg` internally standardizes the design matrix, but inspection of the standardized design matrix can sometimes be useful. This differs from the base R function `scale` in two ways: (1) `scale` uses the sample standard deviation `sqrt(sum((x^2)/((n-1)))`, while `std` uses the root-mean-square, or population, standard deviation `sqrt(mean(sum(x^2)))`, and (2) `std` is faster.

Value

The standardized design matrix, with attributes "center" and "scale" corresponding to the mean and (population) standard deviation used to scale the columns.

Author(s)

Patrick Breheny

Examples

```
X <- matrix(rnorm(50), 10, 5)
S <- std(X)
apply(S, 2, sum)
apply(S, 2, function(x) mean(x^2))
```
Summary method for cv.ncvreg objects

Usage

```r
## S3 method for class 'cv.ncvreg'
summary(object, ...)
## S3 method for class 'summary.cv.ncvreg'
print(x, digits, ...)
```

Arguments

- `object`: A "cv.ncvreg" object.
- `x`: A "summary.cv.ncvreg" object.
- `digits`: Number of digits past the decimal point to print out. Can be a vector specifying different display digits for each of the five non-integer printed values.
- `...`: Further arguments passed to or from other methods.

Value

`summary.cv.ncvreg` produces an object with S3 class "summary.cv.ncvreg". The class has its own print method and contains the following list elements:

- `penalty`: The penalty used by `ncvreg`.
- `model`: Either "linear" or "logistic", depending on the `family` option in `ncvreg`.
- `n`: Number of observations.
- `p`: Number of regression coefficients (not including the intercept).
- `min`: The index of `lambda` with the smallest cross-validation error.
- `lambda`: The sequence of `lambda` values used by `cv.ncvreg`.
- `cve`: Cross-validation error (deviance).
- `r.squared`: Proportion of variance explained by the model, as estimated by cross-validation.
- `snr`: Signal to noise ratio, as estimated by cross-validation.
- `sigma`: For linear regression models, the scale parameter estimate.
- `pe`: For logistic regression models, the prediction error (misclassification error).

Author(s)

Patrick Breheny <patrick-breheny@uiowa.edu>
References


See Also

ncvreg, cv.ncvreg, plot.cv.ncvreg

Examples

```r
## Linear regression
data(prostate)
X <- as.matrix(prostate[,1:8])
y <- prostate$lpsa
cvfit <- cv.ncvreg(X, y)
summary(cvfit)

## Logistic regression
data(heart)
X <- as.matrix(heart[,1:9])
y <- heart$chd
cvfit <- cv.ncvreg(X, y, family="binomial")
summary(cvfit)
```
Index

*Topic **datasets**
  - heart, 8
  - Lung, 9
  - prostate, 30
*Topic **models**
  - plot.cv.ncvreg, 22
  - plot.ncvreg, 25
  - predict.ncvreg, 27
  - summary.cv.ncvreg, 32
*Topic **regression**
  - plot.cv.ncvreg, 22
  - plot.ncvreg, 25
  - predict.ncvreg, 27
  - summary.cv.ncvreg, 32

AUC (AUC.cv.ncvsurv), 3
AUC.cv.ncvsurv, 3

coeff.ncvreg(predict.ncvreg), 27
coeff.ncvsurv(predict.ncvreg), 28
cv.ncvreg, 4, 14, 23, 33
cv.ncvsurv, 3, 6, 18

fir, 8, 21

heart, 8
Lung, 9
mfdr, 8, 10, 20, 24

ncvreg, 5, 11, 12, 20, 21, 23, 25, 28, 33
ncvreg-package, 2
ncvsurv, 7, 10, 11, 15, 26, 29

perm.ncvreg, 10, 11, 18, 21, 24
permres, 19, 20
plot.cv.ncvreg, 5, 7, 22, 33
plot.mfdr, 11, 20, 23
plot.ncvreg, 14, 18, 25
plot.ncvsurv.func, 26

cv.n.cv.n.cvreg, 27
predict.ncvreg, 27
predict.ncvsurv, 26, 28
print.summary.cv.ncvreg
    (summary.cv.ncvreg), 32
prostate, 30
scale, 31
std, 31
summary.cv.n.cvreg, 5, 7, 32