Package 'ncvreg'

July 12, 2014

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

Version 3.2-0

Date 2014-07-11

Author Patrick Breheny

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

Description Efficient algorithms for fitting regularization paths for linear or logistic regression models penalized by MCP or SCAD, with optional additional L2 penalty (``Mnet'').

License GPL-2

URL http://myweb.uiowa.edu/pbreheny/publications/Breheny2011.pdf

Depends

NeedsCompilation yes

Repository CRAN

Date/Publication 2014-07-12 07:23:57

R topics documented:

`ncvreg-package` *Regularization paths for SCAD- and MCP-penalized regression models*

Description

Efficient algorithms for fitting regularization paths for linear or logistic regression models penalized by MCP or SCAD, with optional additional L2 penalty.

Details

Package:	ncvreg
Type:	Package
Version:	3.2-0
Date:	2014-07-11
License:	GPL-2

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.

Author(s)

Patrick Breheny <patrick-breheny@uiowa.edu>

References

Breheny, P. and Huang, J. (2011) Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. Ann. Appl. Statist., 5: 232-253.

Examples

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

fit <- ncvreg(X,y)
plot(fit)

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

`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

```
cv.ncvreg(X, y, ..., nfolds=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`.
`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.
`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"`.

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 for `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`.

Author(s)

Patrick Breheny <patrick-breheny@uiowa.edu>

References

Breheny, P. and Huang, J. (2011) Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. Ann. Appl. Statist., 5: 232-253.

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]
```

---

fir *False inclusion rates for ncvreg (independence approximation)*

Description

Estimates false inclusion rates (FIR) for penalized regression models based on an approximation of independence between the predictors.

Usage

```
fir(fit)
```

Arguments

`fit`	An `ncvreg` object.

Details

The function estimates the false inclusion rate (FIR) for a penalized regression model. The calculation is based on an approximation of independence between the predictors, and is reasonably accurate in near-independent settings. However, the estimate is conservative when predictors are correlated. For a more accurate estimate of the false inclusion rate in the presence of correlated predictors, see `perm.ncvreg`.

Value

An object with S3 class "fir" 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.
FIR	The estimated false inclusion rate (EF/S).

Author(s)

Patrick Breheny <patrick-breheny@uiowa.edu>

See Also

ncvreg, plot.fir, perm.ncvreg

Examples

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

f <- fir(fit)
cbind(EF=f$EF, S=f$S, FIR=f$FIR)[1:20,]

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

---

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; either "Present" or "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

Usage

```
data(heart)
```

Format

A data frame with 462 observations on 10 variables

Source

http://www-stat-class.stanford.edu/~tibs/ElemStatLearn/

References

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer.

Rousseauw, J., et al. (1983). Coronary risk factor screening in three rural communities. *South African Medical Journal*, **64**, 430-436.

---

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

```
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=.001,
max.iter=1000, 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 threshhold. The algorithm iterates until the relative change in any coefficient is less than eps. Default is .001.
max.iter	Maximum number of iterations. Default is 1000.x
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.
...	Not used.

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.

Author(s)

Patrick Breheny <patrick-breheny@uiow.edu>

References

Breheny, P. and Huang, J. (2011) Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. Ann. Appl. Statist., 5: 232-253.

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,penalty="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)
plot(fit,main=expression(paste(alpha,"=",0.5)))
fit <- ncvreg(X,y,alpha=0.1)
plot(fit,main=expression(paste(alpha,"=",0.1)))

par(mfrow=c(2,2))
fit <- ncvreg(X,y)
plot(fir(fit))            ## Independence approximation
plot(fir(fit), type="EF") ## Independence approximation
perm.fit <- perm.ncvreg(X,y)
plot(perm.fit)
plot(perm.fit, type="EF")

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

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

par(mfrow=c(2,2))
fit <- ncvreg(X,y,family="binomial")
plot(fit,main=expression(paste(alpha,"=",1)))
```

```
fit <- ncvreg(X,y,family="binomial",alpha=0.9)
plot(fit,main=expression(paste(alpha,"=",0.9)))
fit <- ncvreg(X,y,family="binomial",alpha=0.5)
plot(fit,main=expression(paste(alpha,"=",0.5)))
fit <- ncvreg(X,y,family="binomial",alpha=0.1)
plot(fit,main=expression(paste(alpha,"=",0.1)))
```

`ncvreg_fit` *Internal C function for fitting ncvreg models*

Description

This function is a wrapper for the calls to C carried out by `ncvreg`. This function performs no checking, processing or standardization, so use `ncvreg` instead, unless you really know what you're doing.

Usage

```
ncvreg_fit(X, y, family=c("gaussian", "binomial", "poisson"),
penalty=c("MCP", "SCAD", "lasso"), gamma=3, alpha=1,
lambda.min=ifelse(n>p,0.001,0.05), nlambda=100, lambda, eps=.001,
max.iter=1000, dfmax=p+1, penalty.factor=rep(1, ncol(X)), warn=TRUE)
```

Arguments

`X`	The design matrix. Unlike in `ncvreg`, `ncvreg_fit` does not standardize the data. For Gaussian responses, no intercept is included. At least for now, an intercept is still included automatically for binomial and Poisson responses, as it is not clear (to me) that removing the intercept here would make sense.
`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, as in `ncvreg`.
`alpha`	Tuning parameter which controls the relative contributions from the MCP/SCAD penalty and the ridge, or L2 penalty, as in `ncvreg`.
`lambda.min`	Smallest value of lambda (as a fraction of lambda.max), as in `ncvreg`.
`nlambda`	Number of lambda values, as in `ncvreg`.
`lambda`	A user-specified sequence of lambda values, as in `ncvreg`.
`eps`	Convergence threshhold, as in `ncvreg`.
`max.iter`	Maximum number of iterations. Default is 1000.
`dfmax`	Upper bound for the number of nonzero coefficients, as in `ncvreg`.
`penalty.factor`	A multiplicative factor for the penalty applied to each coefficient, as in `ncvreg`.
`warn`	Return warning messages for failures to converge and model saturation? Default is TRUE.

Details

ncvreg_fit is supplied as a separate function in case developers wish to embed ncvreg's internal algorithms in a larger procedure. It should not be called directly unless you know exactly what you are doing. In particular, no standardization or processing of X and y are carried out, and the output will not work with any of other functions in the package such as plot.ncvreg or predict.ncvreg.

Value

A list with components:

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 the length of lambda.
loss	A vector containing either the residual sum of squares ("gaussian") or negative log-likelihood ("binomial" or "poisson") of the fitted model at each value of lambda.
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.

Author(s)

Patrick Breheny <patrick-breheny@uiowa.edu>

References

Breheny, P. and Huang, J. (2011) Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. Ann. Appl. Statist., 5: 232-253.

See Also

ncvreg

Examples

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

## These two results are NOT the same;
## No standardization is being done in the latter
ncvreg(X, y, lambda=c(0.5, 0.1, 0.05))$beta
ncvreg_fit(X, y, lambda=c(0.5, 0.1, 0.05))$beta
```

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.

Usage

```
perm.ncvreg(X, y, ..., 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.
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, 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 false inclusion rate (FIR).

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.
FIR	The estimated false inclusion 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>

See Also

ncvreg, plot.fir, fir

Examples

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

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

## 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$fit)
plot(pmfit$fit)
```

---

plot.cv.ncvreg *Plots the cross-validation curve from a "cv.ncvreg" object*

Description

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

Usage

```
## 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); snr plots an estimate of the signal-to-noise ratio; scale plots, for family="gaussian", an estimate of the scale parameter (standard deviation); pred plots, for 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-validaton 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.

Author(s)

Patrick Breheny <patrick-breheny@uiowa.edu>

References

Breheny, P. and Huang, J. (2011) Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. Ann. Appl. Statist., 5: 232-253.

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")
```

`plot.fir` *Plot false inclusion rate curves*

Description

Plot false inclusion rate curves from an `"fir"` or `"perm.ncvreg"` object.

Usage

```
## S3 method for class 'fir'
plot(x, type=c("FIR", "EF"), log.l=FALSE, legend=TRUE,
...)
```

Arguments

`x`	A `"perm.ncvreg"` or `"fir"` object.
`type`	What to plot on the vertical axis. `FIR` plots the false inclusion rate; `EF` plots the expected number of false inclusions along with the actual number of variables included in the model.
`log.l`	Should horizontal axis be on the log scale? Default is FALSE.
`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 inclusions? Default is `TRUE`.
`...`	Other graphical parameters to pass to `plot`

Author(s)

Patrick Breheny <patrick-breheny@uiowa.edu>

See Also

`fir`, `perm.ncvreg`

Examples

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

f <- fir(fit)
cbind(EF=f$EF, S=f$S, FIR=f$FIR)[1:20,]

## Comparison with perm.ncvreg
par(mfrow=c(2,2))
plot(f)
plot(f, type="EF")
pmfit <- perm.ncvreg(X, y)
plot(pmfit)
plot(pmfit, type="EF")
## Note that fir() 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

```
## S3 method for class 'ncvreg'
plot(x, alpha=1, log.l=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.l`	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

Breheny, P. and Huang, J. (2011) Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. Ann. Appl. Statist., 5: 232-253.

See Also

ncvreg

Examples

```
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)
```

---

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

```
## 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

Breheny, P. and Huang, J. (2011) Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. Ann. Appl. Statist., 5: 232-253.

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)
predict(fit, X, type="link", lambda=0.05)
predict(fit, X, type="response", lambda=0.05)
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))
```

---

prostate *Factors associated with prostate specific antigen*

Description

Data from a study by 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

http://www-stat-class.stanford.edu/~tibs/ElemStatLearn/

References

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer.

Stamey, T., et al. (1989). Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate. II. Radical prostatectomy treated patients. *Journal of Urology*, **16**: 1076-1083.

---

summary.cv.ncvreg *Summarizing inferences based on cross-validation*

Description

Summary method for `cv.ncvreg` objects

Usage

```
## 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

Breheny, P. and Huang, J. (2011) Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. Ann. Appl. Statist., 5: 232-253.

See Also

ncvreg, cv.ncvreg, plot.cv.ncvreg

Examples

```
## 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