Package 'glmpath'

October 13, 2022

Version 0.98

Date 2018-01-27

Title L1 Regularization Path for Generalized Linear Models and Cox
Proportional Hazards Model

Author Mee Young Park, Trevor Hastie

Maintainer Mee Young Park <meeyoung@google.com>

Depends survival, R (>= 2.0)

Description A path-following algorithm for L1 regularized generalized linear models and Cox propor-
tional hazards model.

License GPL (>= 2)

Repository CRAN

Date/Publication 2018-01-28 16:09:29 UTC

NeedsCompilation yes

R topics documented:

---

bootstrap.path   *Generates a set of bootstrap coefficients for glmpath or coxpath*

Description

This function generates a set of bootstrap coefficients for glmpath or coxpath. For each bootstrap run, the regularization parameter may be determined based on either aic or bic.

Usage

```
bootstrap.path(x, y, data, B, index = NULL,
               path = c("glmpath", "coxpath"),
               method = c("aic", "bic"), trace = FALSE, ...)
```

Arguments

x	matrix of features
y	response
data	a list of data components. If path=glmpath, data consists of x: a matrix of features and y: response. data is not needed if x and y are input separately. If path=coxpath, data must be provided, including x: a matrix of features, time: the survival time, and status: censor status with 1 if died and 0 if censored.
B	number of bootstrap runs
index	matrix (B rows, ncol(x) columns) of bootstrap sample indices. Each row is a vector of indices for a bootstrap run. If index=NULL, the indices are randomly chosen.
path	Bootstrap coefficients for either glmpath or coxpath are computed. Default is path=glmpath.
method	For each bootstrap run, the regularization parameter is determined based on either aic or bic. Default is aic.
trace	If TRUE, the number of bootstrap runs is printed out.
...	other options for glmpath or coxpath

Details

Fitting glmpath or coxpath gives a series of solution sets with a varying size of the active set. Once we select an appropriate value of the regularization parameter, and thus a set of coefficients, we may then validate the chosen coefficients through bootstrap analysis. plot.bootstrap summarizes the bootstrap results by generating the histograms or the pair scatter plots of the bootstrap coefficients.

Value

bootstrap.path returns a bootpath object, which is a matrix (B by ncol(x)) of bootstrap coefficients. Coefficients computed from the whole data are stored as an attribute coefficients.

Author(s)

Mee Young Park and Trevor Hastie

References

Bradley Efron and Robert Tibshirani (1993) *An Introduction to the Bootstrap* CHAPMAN & HALL/CRC, Boca Raton.

Mee Young Park and Trevor Hastie (2007) L1 regularization path algorithm for generalized linear models. *J. R. Statist. Soc.* B, 69, 659-677.

See Also

coxpath, glmpath, plot.bootstrap

Examples

```
data(heart.data)
attach(heart.data)
bootstrap.a <- bootstrap.path(x, y, B=5, method="bic")
detach(heart.data)
data(lung.data)
attach(lung.data)
bootstrap.b <- bootstrap.path(data=lung.data, B=5, path="coxpath")
detach(lung.data)
```

---

coxpath	*Fits the entire L1 regularization path for Cox proportional hazards model*

Description

This algorithm uses predictor-corrector method to compute the entire regularization path for Cox proportional hazards model with L1 penalty.

Usage

```
coxpath(data, nopenalty.subset = NULL, method = c("breslow", "efron"),
        lambda2 = 1e-5, max.steps = 10 * min(n, m), max.norm = 100 * m,
        min.lambda = (if (m >= n) 1e-3 else 0), max.vars = Inf,
        max.arclength = Inf, frac.arclength = 1, add.newvars = 1,
        bshoot.threshold = 0.1, relax.lambda = 1e-7,
        approx.Gram = FALSE, standardize = TRUE,
        eps = .Machine$double.eps, trace = FALSE)
```

Arguments

data	a list consisting of x: a matrix of features, time: the survival time, and status: censor status with 1 if died and 0 if censored.
nopenalty.subset	
	a set of indices for the predictors that are not subject to the L1 penalty
method	approximation method for tied survival times. Approximations derived by Breslow (1974) and Efron (1977) are available. Default is breslow.
lambda2	regularization parameter for the L2 norm of the coefficients. Default is 1e-5.
max.steps	an optional bound for the number of steps to be taken. Default is 10 * min{nrow(x), ncol(x)}.
max.norm	an optional bound for the L1 norm of the coefficients. Default is 100 * ncol(x).
min.lambda	an optional (lower) bound for the size of λ. When ncol(x) is relatively large, the coefficient estimates are prone to numerical precision errors at extremely small λ. In such cases, early stopping is recommended. Default is 0 for ncol(x) < nrow(x) cases and 1e-3 otherwise.
max.vars	an optional bound for the number of active variables. Default is Inf.
max.arclength	an optional bound for arc length (L1 norm) of a step. If max.arclength is extremely small, an exact nonlinear path is produced. Default is Inf.
frac.arclength	Under the default setting, the next step size is computed so that the active set changes right at the next value of lambda. When frac.arclength is assigned some fraction between 0 and 1, the step size is decreased by the factor of frac.arclength in arc length. If frac.arclength=0.2, the step length is adjusted so that the active set would change after five smaller steps. Either max.arclength or frac.arclength can be used to force the path to be more accurate. Default is 1.
add.newvars	add.newvars candidate variables (that are currently not in the active set) are used in the corrector step as potential active variables. Default is 1.
bshoot.threshold	
	If the absolute value of a coefficient is larger than bshoot.threshold at the first corrector step it becomes nonzero (therefore when λ is considered to have been decreased too far), λ is increased again. i.e. A backward distance in λ that makes the coefficient zero is computed. Default is 0.1.
relax.lambda	A variable joins the active set if $|l'(\beta)| > \lambda*(1\text{-relax.lambda})$. Default is 1e-7. If no variable joins the active set even after many (>20) steps, the user should increase relax.lambda to 1e-6 or 1e-5, but not more than that. This adjustment is sometimes needed because of the numerical precision/error propagation problems. In general, the paths are less accurate with relaxed lambda.
approx.Gram	If TRUE, an approximated Gram matrix is used in predictor steps; each step takes less number of computations, but the total number of steps usually increases. This might be useful when the number of features is large.
standardize	If TRUE, predictors are standardized to have a unit variance.
eps	an effective zero
trace	If TRUE, the algorithm prints out its progress.

Details

This algorithm implements the predictor-corrector method to determine the entire path of the coefficient estimates as the amount of regularization varies; it computes a series of solution sets, each time estimating the coefficients with less regularization, based on the previous estimate. The coefficients are estimated with no error at the knots, and the values are connected, thereby making the paths piecewise linear.

Value

A `coxpath` object is returned.

`lambda`	vector of λ values for which the exact coefficients are computed
`lambda2`	λ_2 used
`step.length`	vector of step lengths in λ
`corr`	matrix of $l'(\beta)$ values (derivatives of the log-partial-likelihood)
`new.df`	vector of degrees of freedom (to be used in the plot function)
`df`	vector of degrees of freedom at each step
`loglik`	vector of log-partial-likelihood computed at each step
`aic`	vector of AIC values
`bic`	vector of BIC values
`b.predictor`	matrix of coefficient estimates from the predictor steps
`b.corrector`	matrix of coefficient estimates from the corrector steps
`new.A`	vector of boolean values indicating the steps at which the active set changed (to be used in the plot/predict functions)
`actions`	actions taken at each step
`meanx`	means of the columns of x
`sdx`	standard deviations of the columns of x
`xnames`	column names of x
`method`	method used
`nopenalty.subset`	
	nopenalty.subset used
`standardize`	TRUE if the predictors were standardized before fitting

Author(s)

Mee Young Park and Trevor Hastie

References

Mee Young Park and Trevor Hastie (2007) L1 regularization path algorithm for generalized linear models. *J. R. Statist. Soc.* B, 69, 659-677.

See Also

cv.coxpath, plot.coxpath, predict.coxpath, summary.coxpath

Examples

```
data(lung.data)
attach(lung.data)
fit.a <- coxpath(lung.data)
fit.b <- coxpath(lung.data, method="efron")
detach(lung.data)
```

---

cv.coxpath   *Computes cross-validated (minus) log-partial-likelihoods for coxpath*

Description

This function computes cross-validated (minus) log-partial-likelihoods for coxpath.

Usage

```
cv.coxpath(data, method = c("breslow", "efron"), nfold = 5,
           fraction = seq(0, 1, length = 100),
           mode = c("norm", "lambda"), plot.it = TRUE, se = TRUE, ...)
```

Arguments

data	a list consisting of x: a matrix of features, time: the survival time, and status: censor status with 1 if died and 0 if censored.
method	approximation method for tied survival times. Approximations derived by Breslow (1974) and Efron (1977) are available. Default is breslow.
nfold	number of folds to be used in cross-validation. Default is nfold=5.
fraction	the fraction of L1 norm or $\log(\lambda)$ with respect to their maximum values at which the CV errors are computed. Default is seq(0,1,length=100).
mode	If mode=norm, cross-validation is run at certain values of L1 norm. If mode=lambda, cross-validation is run at certain values of $\log(\lambda)$. Default is norm.
plot.it	If TRUE, CV curve is plotted.
se	If TRUE, standard errors are plotted.
...	other options for coxpath

Author(s)

Mee Young Park and Trevor Hastie

References

Mee Young Park and Trevor Hastie (2007) L1 regularization path algorithm for generalized linear models. *J. R. Statist. Soc.* B, 69, 659-677.

See Also

coxpath, plot.coxpath, predict.coxpath

Examples

```
data(lung.data)
attach(lung.data)
cv <- cv.coxpath(lung.data)
detach(lung.data)
```

---

cv.glmpath	*Computes cross-validated (minus) log-likelihoods or prediction errors for glmpath*

Description

This function computes cross-validated (minus) log-likelihoods or prediction errors for glmpath.

Usage

```
cv.glmpath(x, y, data, family = binomial, weight = rep(1, n),
           offset = rep(0, n), nfold = 10,
           fraction = seq(0, 1, length = 100),
           type = c("loglik", "response"), mode = c("norm", "lambda"),
           plot.it = TRUE, se = TRUE, ...)
```

Arguments

x	matrix of features
y	response
data	a list consisting of x: a matrix of features and y: response. data is not needed if above x and y are input separately.
family	name of a family function that represents the distribution of y to be used in the model. It must be binomial, gaussian, or poisson. For each one, the canonical link function is used; logit for binomial, identity for gaussian, and log for poisson distribution. Default is binomial.
weight	an optional vector of weights for observations
offset	an optional vector of offset. If a column of x is used as offset, the corresponding column must be excluded from x.
nfold	number of folds to be used in cross-validation. Default is nfold=10.
fraction	the fraction of L1 norm or $\log(\lambda)$ with respect to their maximum values at which the CV errors are computed. Default is seq(0,1,length=100).
type	If type=loglik, cross-validated minus log-likelihoods are computed. If type=response, cross-validated prediction errors are computed. Default is loglik.

mode	If mode=norm, cross-validation is run at certain values of L1 norm. If mode=lambda, cross-validation is run at certain values of $\log(\lambda)$. Default is norm.
plot.it	If TRUE, CV curve is plotted.
se	If TRUE, standard errors are plotted.
...	other options for glmpath

Author(s)

Mee Young Park and Trevor Hastie

References

Mee Young Park and Trevor Hastie (2007) L1 regularization path algorithm for generalized linear models. *J. R. Statist. Soc.* B, 69, 659-677.

See Also

glmpath, plot.glmpath, predict.glmpath

Examples

```
data(heart.data)
attach(heart.data)
cv.a <- cv.glmpath(x, y, family=binomial)
cv.b <- cv.glmpath(x, y, family=binomial, type="response")
detach(heart.data)
```

---

glmpath   *Fits the entire L1 regularization path for generalized linear models*

Description

This algorithm uses predictor-corrector method to compute the entire regularization path for generalized linear models with L1 penalty.

Usage

```
glmpath(x, y, data, nopenalty.subset = NULL, family = binomial,
        weight = rep(1, n), offset = rep(0, n), lambda2 = 1e-5,
        max.steps = 10 * min(n, m), max.norm = 100 * m,
        min.lambda = (if (m >= n) 1e-6 else 0), max.vars = Inf,
        max.arclength = Inf, frac.arclength = 1, add.newvars = 1,
        bshoot.threshold = 0.1, relax.lambda = 1e-8,
        standardize = TRUE, eps = .Machine$double.eps,
        trace = FALSE)
```

Arguments

x	matrix of features
y	response
data	a list consisting of x: a matrix of features and y: response. data is not needed if x and y are input separately.
nopenalty.subset	
	a set of indices for the predictors that are not subject to the L1 penalty
family	name of a family function that represents the distribution of y to be used in the model. It must be binomial, gaussian, or poisson. For each one, the canonical link function is used; logit for binomial, identity for gaussian, and log for poisson distribution. Default is binomial.
weight	an optional vector of weights for observations
offset	an optional vector of offset. If a column of x is used as offset, the corresponding column must be removed from x.
lambda2	regularization parameter for the L2 norm of the coefficients. Default is 1e-5.
max.steps	an optional bound for the number of steps to be taken. Default is 10 * min{nrow(x), ncol(x)}.
max.norm	an optional bound for the L1 norm of the coefficients. Default is 100 * ncol(x).
min.lambda	an optional (lower) bound for the size of λ. Default is 0 for ncol(x) < nrow(x) cases and 1e-6 otherwise.
max.vars	an optional bound for the number of active variables. Default is Inf.
max.arclength	an optional bound for arc length (L1 norm) of a step. If max.arclength is extremely small, an exact nonlinear path is produced. Default is Inf.
frac.arclength	Under the default setting, the next step size is computed so that the active set changes right at the next value of lambda. When frac.arclength is assigned some fraction between 0 and 1, the step size is decreased by the factor of frac.arclength in arc length. If frac.arclength=0.2, the step length is adjusted so that the active set would change after five smaller steps. Either max.arclength or frac.arclength can be used to force the path to be more accurate. Default is 1.
add.newvars	add.newvars candidate variables (that are currently not in the active set) are used in the corrector step as potential active variables. Default is 1.
bshoot.threshold	
	If the absolute value of a coefficient is larger than bshoot.threshold at the first corrector step it becomes nonzero (therefore when λ is considered to have been decreased too far), λ is increased again. i.e. A backward distance in λ that makes the coefficient zero is computed. Default is 0.1.
relax.lambda	A variable joins the active set if $|l'(\beta)| > \lambda*$(1-relax.lambda). Default is 1e-8. If no variable joins the active set even after many (>20) steps, the user should increase relax.lambda to 1e-7 or 1e-6, but not more than that. This adjustment is sometimes needed because of the numerical precision/error propagation problems. In general, the paths are less accurate with relaxed lambda.
standardize	If TRUE, predictors are standardized to have a unit variance.
eps	an effective zero
trace	If TRUE, the algorithm prints out its progress.

Details

This algorithm implements the predictor-corrector method to determine the entire path of the coefficient estimates as the amount of regularization varies; it computes a series of solution sets, each time estimating the coefficients with less regularization, based on the previous estimate. The coefficients are estimated with no error at the knots, and the values are connected, thereby making the paths piecewise linear.

Value

A `glmpath` object is returned.

`lambda`	vector of λ values for which the exact coefficients are computed
`lambda2`	λ_2 used
`step.length`	vector of step lengths in λ
`corr`	matrix of $l'(\beta)$ values (derivatives of the log-likelihood)
`new.df`	vector of degrees of freedom (to be used in the plot function)
`df`	vector of degrees of freedom at each step
`deviance`	vector of deviance computed at each step
`aic`	vector of AIC values
`bic`	vector of BIC values
`b.predictor`	matrix of coefficient estimates from the predictor steps
`b.corrector`	matrix of coefficient estimates from the corrector steps
`new.A`	vector of boolean values indicating the steps at which the active set changed (to be used in the plot/predict functions)
`actions`	actions taken at each step
`meanx`	means of the columns of x
`sdx`	standard deviations of the columns of x
`xnames`	column names of x
`family`	family used
`weight`	weights used
`offset`	offset used
`nopenalty.subset`	
	nopenalty.subset used
`standardize`	TRUE if the predictors were standardized before fitting

Author(s)

Mee Young Park and Trevor Hastie

References

Mee Young Park and Trevor Hastie (2007) L1 regularization path algorithm for generalized linear models. *J. R. Statist. Soc.* B, 69, 659-677.

See Also

cv.glmpath, plot.glmpath, predict.glmpath, summary.glmpath

Examples

```
data(heart.data)
attach(heart.data)
fit.a <- glmpath(x, y, family=binomial)
fit.b <- glmpath(x, y, family=gaussian)
detach(heart.data)
```

---

heart.data   *Dataset for glmpath*

Description

South African Heart Disease dataset used to test glmpath algorithm

Usage

```
data(heart.data)
```

Format

A dataset with 462 observations on 9 variables and a binary response.

x x contains 9 columns of the following variables: sbp (systolic blood pressure); tobacco (cumulative tobacco); ldl (low density lipoprotein cholesterol); adiposity; famhist (family history of heart disease); typea (type-A behavior); obesity; alcohol (current alcohol consumption); age (age at onset)

y response, coronary heart disease

References

Hastie, T., Tibshirani, R., and Friedman, J. (2001) *Elements of Statistical Learning; Data Mining, Inference, and Prediction* Springer-Verlag, New York.

Examples

```
data(heart.data)
attach(heart.data)
fit <- glmpath(x, y, family=binomial)
detach(heart.data)
```

`lung.data`	*Dataset for coxpath*

Description

Lung cancer dataset used to test `coxpath` algorithm

Usage

```
data(lung.data)
```

Format

A dataset consisting of 137 observations with their survival time, censor status as well as 6 features.

x x contains 6 columns of the following variables: trt (1=standard treatment, and 2=test); celltype (1=squamous, 2=smallcell, 3=adeno, and 4=large); karno (Karnofsky performance score); diagtime (months from diagnosis to randomization); age (in years); prior (prior therapy 0=no, and 1=yes)

time survival time

status censor status

References

Kalbfleisch, J. and Prentice, R. (2002) *The Statistical Analysis of Failure Time Data* J. Wiley, Hoboken, N.J.

Examples

```
data(lung.data)
attach(lung.data)
fit <- coxpath(lung.data)
detach(lung.data)
```

`plot.bootpath`	*Generates the histograms or the pairwise scatter plots of the bootstrap coefficients computed from bootstrap.path*

Description

This function takes a `bootpath` object from `bootstrap.path` and generates the histograms or the pairwise scatter plots of the bootstrap coefficients.

Usage

```
   ## S3 method for class 'bootpath'
plot(x, type = c("histogram", "pairplot"),
     mfrow = NULL, mar = NULL, ...)
```

Arguments

x	a bootpath object from `bootstrap.path`.
type	If `type=histogram`, the histograms of bootstrap coefficients for individual features are generated. The red vertical bar indicates the coefficient computed using the whole data. The thick bar at zero indicates the frequency of the zero coefficients. If `type=pairplot`, the pairwise scatter plots of the bootstrap coefficients are generated. The red solid dot indicates the pair of coefficients computed using the whole data. Default is `histogram`.
mfrow	determines the numbers of rows and columns of the histograms on a page. 2 rows are generated as a default.
mar	margin relative to the current font size
...	other options for the plot

Details

Fitting `glmpath` or `coxpath` gives a series of solution sets with a varying size of the active set. Once we select an appropriate value of the regularization parameter, and, thus a set of coefficients, we may then validate the chosen coefficients through a bootstrap analysis. `plot.bootstrap` summarizes the bootstrap results by generating the histograms or the pairwise scatter plots of the bootstrap coefficients.

Author(s)

Mee Young Park and Trevor Hastie

References

Bradley Efron and Robert Tibshirani (1993) *An Introduction to the Bootstrap* CHAPMAN & HALL/CRC, Boca Raton.

Mee Young Park and Trevor Hastie (2007) L1 regularization path algorithm for generalized linear models. *J. R. Statist. Soc.* B, 69, 659-677.

See Also

bootstrap.path, coxpath, glmpath

Examples

```
data(heart.data)
attach(heart.data)
bootstrap.a <- bootstrap.path(x, y, B=5)
plot(bootstrap.a)
```

```
plot(bootstrap.a, type="pairplot")
detach(heart.data)
```

---

plot.coxpath   *Plots the regularization path computed from coxpath*

Description

This function takes a coxpath object and visualizes the regularization path. The horizontal axis can be norm, lambda or step. The vertical axis can be coefficients, aic or bic.

Usage

```
  ## S3 method for class 'coxpath'
plot(x, xvar = c("norm", "lambda", "step"),
     type = c("coefficients", "aic", "bic"),
     plot.all.steps = FALSE, xlimit = NULL, predictor = FALSE,
     omit.zero = TRUE, breaks = TRUE, mar = NULL, main = NULL,
     eps = .Machine$double.eps, ...)
```

Arguments

x	a coxpath object
xvar	horizontal axis. xvar=norm plots against the L1 norm of the coefficients (to which L1 norm penalty was applied); xvar=lambda plots against λ; and xvar=step plots against the number of steps taken. Default is norm.
type	type of the plot, or the vertical axis. Default is coefficients.
plot.all.steps	If TRUE, all the steps taken along the path are marked on the plot. If FALSE, which is the default, only the steps at which the active set changed are shown on the plot.
xlimit	When the user wants to visualize a (beginning) sub-part of the plot, xlimit sets an upper limit to the L1 norm or the number of steps, or a lower limit to λ.
predictor	If TRUE and type=coefficients, the predictor step estimates are connected with dotted lines. If FALSE, only the corrector step estimates are connected with solid lines.
omit.zero	If TRUE, the predictors that were never in the active set are omitted.
breaks	If TRUE, vertical lines are drawn at the points where the active set changes and numbered with the degrees of freedom.
mar	margin relative to the current font size
main	title of the plot
eps	an effective zero
...	other options for the plot

Author(s)

Mee Young Park and Trevor Hastie

References

Mee Young Park and Trevor Hastie (2007) L1 regularization path algorithm for generalized linear models. *J. R. Statist. Soc.* B, 69, 659-677.

See Also

cv.coxpath, coxpath, predict.coxpath

Examples

```
data(lung.data)
attach(lung.data)
fit <- coxpath(lung.data)
par(mfrow=c(3, 2))
plot(fit)
plot(fit, xvar="lambda")
plot(fit, xvar="step")
plot(fit, xvar="step", xlimit=8)
plot(fit, type="aic")
plot(fit, type="bic")
detach(lung.data)
```

---

plot.glmpath *Plots the regularization path computed from glmpath*

Description

This function takes a glmpath object and visualizes the regularization path. The horizontal axis can be norm, lambda or step. The vertical axis can be coefficients, aic or bic.

Usage

```
   ## S3 method for class 'glmpath'
plot(x, xvar = c("norm", "lambda", "step"),
     type = c("coefficients", "aic", "bic"),
     plot.all.steps = FALSE, xlimit = NULL, predictor = FALSE,
     omit.zero = TRUE, breaks = TRUE, mar = NULL,
     eps = .Machine$double.eps, main = NULL, ...)
```

Arguments

x	a `glmpath` object
xvar	horizontal axis. `xvar=norm` plots against the L1 norm of the coefficients (to which L1 norm penalty was applied); `xvar=lambda` plots against λ; and `xvar=step` plots against the number of steps taken. Default is `norm`.
type	type of the plot, or the vertical axis. Default is `coefficients`.
plot.all.steps	If `TRUE`, all the steps taken along the path are marked on the plot. If `FALSE`, which is the default, only the steps at which the active set changed are shown on the plot.
xlimit	When the user wants to visualize a (beginning) sub-part of the plot, `xlimit` sets an upper limit to the L1 norm or the number of steps, or a lower limit to λ.
predictor	If `TRUE` and `type=coefficients`, the predictor step estimates are connected with dotted lines. If `FALSE`, only the corrector step estimates are connected with solid lines.
omit.zero	If `TRUE` and `type=coefficients`, the predictors that were never in the active set are omitted.
breaks	If `TRUE`, vertical lines are drawn at the points where the active set changes and numbered with the degrees of freedom.
mar	margin relative to the current font size
eps	an effective zero
main	title of the plot
...	other options for the plot

Author(s)

Mee Young Park and Trevor Hastie

References

Mee Young Park and Trevor Hastie (2007) L1 regularization path algorithm for generalized linear models. *J. R. Statist. Soc.* B, 69, 659-677.

See Also

cv.glmpath, glmpath, predict.glmpath

Examples

```
data(heart.data)
attach(heart.data)
fit <- glmpath(x, y, family=binomial)
par(mfrow=c(3, 2))
plot(fit)
plot(fit, xvar="lambda")
plot(fit, xvar="step")
plot(fit, xvar="step", xlimit=8)
```

```
plot(fit, type="aic")
plot(fit, type="bic")
detach(heart.data)
```

---

predict.coxpath   *Makes predictions at particular points along the fitted coxpath*

Description

This function makes predictions at particular points along the fitted `coxpath`. The coefficients, log-partial-likelihood, linear predictor or the risk can be computed. A `coxph` object can be returned at one particular value of λ.

Usage

```
    ## S3 method for class 'coxpath'
predict(object, data, s, type = c("coefficients", "loglik",
        "lp", "risk", "coxph"), mode = c("step",
        "norm.fraction", "norm", "lambda.fraction", "lambda"),
        eps = .Machine$double.eps, ...)
```

Arguments

`object`	a `coxpath` object
`data`	a list containing `x`, `time`, and `status`, with which the predictions are made. If `type=lp` or `type=risk`, then `x` is required. If `type=loglik` or `type=coxph`, then `x`, `time`, and `status` are required.
`s`	the values of `mode` at which the predictions are made. If `type=coxph`, only the first element of `s` is used. If `s` is missing, then the steps at which the active set changed are used, and thus, `mode` is automatically switched to `step`.
`type`	If `type=coefficients`, the coefficients are returned; if `type=loglik`, log-partial-likelihoods are returned; if `type=lp`, linear predictors ($x'\beta$) are returned; if `type=risk`, risks ($e^{x'\beta}$) are returned; and if `type=coxph`, a `coxph` object (as in **survival** package) at the first element of `s` is returned. (i.e. the components of a `coxph` object such as coefficients, variance, and the test statistics are adjusted to the shrinkage corresponding to `s`. A `coxph` object can be further used as an argument to the functions in **survival** package.) Default is `coefficients`. The coefficients for the initial input variables are returned (rather than the standardized coefficients).
`mode`	what `mode=s` refers to. If `mode=step`, `s` is the number of steps taken; if `mode=norm.fraction`, `s` is the fraction of the L1 norm of the standardized coefficients (with respect to the largest norm); if `mode=norm`, `s` is the L1 norm of the standardized coefficients; if `mode=lambda.fraction`, `s` is the fraction of $\log(\lambda)$; and if `mode=lambda`, `s` is λ. Default is `step`.
`eps`	an effective zero
`...`	other options for the prediction

Author(s)

Mee Young Park and Trevor Hastie

References

Mee Young Park and Trevor Hastie (2007) L1 regularization path algorithm for generalized linear models. *J. R. Statist. Soc.* B, 69, 659-677.

See Also

cv.coxpath, coxpath, plot.coxpath

Examples

```
data(lung.data)
attach(lung.data)
fit <- coxpath(lung.data)
pred.a <- predict(fit, x, s = seq(0, 1, length=10),
                  mode = "norm.fraction")
library(survival)
pred.b <- predict(fit, lung.data, s = 0.5, type = "coxph",
                  mode = "lambda.fraction")
pred.s <- survfit(pred.b)
plot(pred.s)
detach(lung.data)
```

---

predict.glmpath   *Makes predictions at particular points along the fitted glmpath*

Description

This function makes predictions at particular points along the fitted glmpath. The linear predictor, estimated response, log-likelihood, or the coefficients can be computed.

Usage

```
  ## S3 method for class 'glmpath'
predict(object, newx, newy, s, type = c("link", "response",
        "loglik", "coefficients"), mode = c("step",
        "norm.fraction", "norm", "lambda.fraction", "lambda"),
        weight = NULL, offset = NULL,
        eps = .Machine$double.eps, ...)
```

Arguments

`object`	a glmpath object
`newx`	a matrix of features at which the predictions are made. If `type=link`, `type=response`, or `type=loglik`, `newx` is required.
`newy`	a vector of responses corresponding to `newx`. If `type=loglik`, `newy` is required.
`s`	the values of mode at which the predictions are made. If `s` is missing, then the steps at which the active set changed are used, and thus, mode is automatically switched to `step`.
`type`	If `type=link`, the linear predictors are returned; if `type=response`, the estimated responses are returned; if `type=loglik`, the log-likelihoods are returned, and if `type=coefficients`, the coefficients are returned. The coefficients for the initial input variables are returned (rather than the standardized coefficients). Default is `link`.
`mode`	what `mode=s` refers to. If `mode=step`, `s` is the number of steps taken; if `mode=norm.fraction`, `s` is the fraction of the L1 norm of the standardized coefficients (with respect to the largest norm); if `mode=norm`, `s` is the L1 norm of the standardized coefficients; if `mode=lambda.fraction`, `s` is the fraction of $\log(\lambda)$; and if `mode=lambda`, `s` is λ. Default is `step`.
`weight`	an optional vector of weights for observations. `weight` is effective only if `type=loglik`.
`offset`	If `offset` was used in `object`, `offset` must be provided for prediction, unless `type=coefficients`.
`eps`	an effective zero
`...`	other options for the prediction

Author(s)

Mee Young Park and Trevor Hastie

References

Mee Young Park and Trevor Hastie (2007) L1 regularization path algorithm for generalized linear models. *J. R. Statist. Soc.* B, 69, 659-677.

See Also

cv.glmpath, glmpath, plot.glmpath

Examples

```
data(heart.data)
attach(heart.data)
fit <- glmpath(x, y, family=binomial)
pred <- predict(fit, x, s = seq(0, 1, length=10), mode="norm.fraction")
detach(heart.data)
```

| summary.coxpath | *Produces an anova-type summary for a coxpath object* |

Description

This function produces an anova-type summary for a coxpath object.

Usage

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

Arguments

object   a coxpath object

...   additional arguments

Details

An anova type of summary is returned, including Df, Log-partial-likelihood, AIC, and BIC values
for the steps where the active set changed.

Value

A data.frame is returned, with the following components at transition points:

Df   degrees of freedom at each step

Log.p.lik   log-partial-likelihood at each step

AIC   AIC value at each step

BIC   BIC value at each step

Author(s)

Mee Young Park and Trevor Hastie

References

Mee Young Park and Trevor Hastie (2007) L1 regularization path algorithm for generalized linear
models. *J. R. Statist. Soc.* B, 69, 659-677.

See Also

coxpath, plot.coxpath, print.coxpath

Examples

```
data(lung.data)
attach(lung.data)
fit <- coxpath(lung.data)
summary(fit)
detach(lung.data)
```

---

summary.glmpath	*Produces an anova-type summary for a glmpath object*

Description

This function produces an anova-type summary for a glmpath object.

Usage

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

Arguments

object   a glmpath object

...   additional arguments

Details

An anova type of summary is returned, including Df, Deviance, AIC, and BIC values for the steps where the active set changed.

Value

A data.frame is returned, with the following components at transition points:

Df   degrees of freedom at each step

Deviance   deviance computed at each step

AIC   AIC value at each step

BIC   BIC value at each step

Author(s)

Mee Young Park and Trevor Hastie

References

Mee Young Park and Trevor Hastie (2007) L1 regularization path algorithm for generalized linear models. *J. R. Statist. Soc.* B, 69, 659-677.

See Also

glmpath, plot.glmpath, print.glmpath

Examples

```
data(heart.data)
attach(heart.data)
fit <- glmpath(x, y)
summary(fit)
detach(heart.data)
```

Index