

Package ‘mpath’

September 23, 2014

Title Regularized Linear Models

Version 0.1-15

Author Zhu Wang, with contributions from Achim Zeileis, Simon Jackman, Brian Ripley, Trevor Hastie, Rob Tibshirani, Balasubramanian Narasimhan, Gil Chu and Patrick Breheny

Maintainer Zhu Wang <zwang@connecticutchildrens.org>

Description Algorithms for fitting model-based penalized coefficient paths. Currently the models include penalized Poisson, negative binomial, zero-inflated Poisson and zero-inflated negative binomial regression models. The penalties include least absolute shrinkage and selection operator (LASSO), smoothly clipped absolute deviation (SCAD) and minimax concave penalty (MCP), and each possibly combining with L₂ penalty.

Imports MASS,glmnet,pscl,numDeriv

Depends methods

License GPL-2

NeedsCompilation yes

Repository CRAN

Date/Publication 2014-09-23 17:37:35

R topics documented:

be.zeroinfl	2
breadReg	3
conv2glmreg	4
conv2zipath	4
cv.glmreg	5
cv.glmregNB	7
cv.glmreg_fit	9
cv.zipath	10
estfunReg	12
glmreg	13

glmregNB	16
glmreg_fit	19
hessianReg	22
meatReg	23
methods	24
plot.glmreg	25
predict.glmreg	26
predict.zipath	27
pval.zipath	29
rzi	30
sandwichReg	31
se	32
summary.glmregNB	33
tuning.zipath	34
vuong.test	36
zipath	37

Index	42
--------------	-----------

be.zeroinfl	<i>conduct backward stepwise variable elimination for zero inflated count regression</i>
-------------	--

Description

conduct backward stepwise variable elimination for zero inflated count regression from zeroinfl function

Usage

```
be.zeroinfl(object, data, dist=c("poisson", "negbin", "geometric"), alpha=0.05,
trace=FALSE)
```

Arguments

object	an object from function zeroinfl
data	argument controlling formula processing via model.frame .
dist	one of the distributions in zeroinfl function
alpha	significance level of variable elimination
trace	logical value, if TRUE, print detailed calculation results

Details

conduct backward stepwise variable elimination for zero inflated count regression from zeroinfl function

Value

an object of `zeroinfl` with all variables having p-values less than the significance level `alpha`

Author(s)

Zhu Wang <zwang@connecticutchildrens.org>

References

Zhu Wang, Shuangge Ma, Ching-Yun Wang, Michael Zappitelli, Prasad Devarajan and Chirag R. Parikh (2014) *EM for Regularized Zero Inflated Regression Models with Applications to Postoperative Morbidity after Cardiac Surgery in Children*, *Statistics in Medicine*, in press

Zhu Wang, Shuangge Ma and Ching-Yun Wang (2014) *Variable selection for zero-inflated and overdispersed data with application to health care demand in Germany*, manuscript

breadReg

Bread for Sandwiches in Regularized Estimators

Description

Generic function for extracting an estimator for the bread of sandwiches.

Usage

```
breadReg(x, which, ...)
```

Arguments

<code>x</code>	a fitted model object.
<code>which</code>	which penalty parameter(s)?
<code>...</code>	arguments passed to methods.

Value

A matrix containing an estimator for the penalized second derivative of log-likelihood function. Typically, this should be an $k \times k$ matrix corresponding to k parameters. The rows and columns should be named as in `coef` or `terms`, respectively.

Author(s)

Zhu Wang <zwang@connecticutchildrens.org>

References

Zhu Wang, Shuangge Ma and Ching-Yun Wang (2014) *Variable selection for zero-inflated and overdispersed data with application to health care demand in Germany*, manuscript

See Also

[meatReg](#), [sandwichReg](#)

Examples

```
data("bioChemists", package = "pscl")
fm_zinb <- zipath(art ~ . | ., data = bioChemists, family = "negbin", nlambda=10)
breadReg(fm_zinb, which=which.min(fm_zinb$bic))
```

conv2glmreg	<i>convert glm object to class glmreg</i>
-------------	---

Description

convert glm object to class glmreg, which then can be used for other purposes

Usage

```
conv2glmreg(object, family=c("poisson", "negbin"))
```

Arguments

object	an object of class glm
family	one of families in glm class

Value

an object of class glmreg

Author(s)

Zhu Wang <zwang@connecticutchildrens.org>

conv2zipath	<i>convert zeroinfl object to class zipath</i>
-------------	--

Description

convert zeroinfl object to class zipath, which then can be used to predict newdata

Usage

```
conv2zipath(object, family=c("poisson", "negbin", "geometric"))
```

Arguments

object	an object of class zeroinfl
family	one of families in zeroinfl class

Value

an object of class zipath

Author(s)

Zhu Wang <zwang@connecticutchildrens.org>

 cv.glmreg

Cross-validation for glmreg

Description

Does k-fold cross-validation for glmreg, produces a plot, and returns cross-validated loglikelihood values for lambda

Usage

```
## S3 method for class 'formula'
cv.glmreg(formula, data, weights, offset=NULL, ...)
## S3 method for class 'matrix'
cv.glmreg(x, y, weights, offset=NULL, ...)
## Default S3 method:
cv.glmreg(x, ...)
## S3 method for class 'cv.glmreg'
plot(x,se=TRUE,ylab=NULL, main=NULL, width=0.02, col="darkgrey", ...)
## S3 method for class 'cv.glmreg'
coef(object,which=object$lambda.which, ...)
```

Arguments

formula	symbolic description of the model, see details.
data	argument controlling formula processing via model.frame .
x	x matrix as in glmreg. It could be object of cv.glmreg.
y	response y as in glmreg.
weights	Observation weights; defaults to 1 per observation
offset	Not implemented yet
object	object of cv.glmreg
which	Indices of the penalty parameter lambda at which estimates are extracted. By default, the one which generates the optimal cross-validation value.

se	logical value, if TRUE, standard error curve is also plotted
ylab	ylab on y-axis
main	title of plot
width	width of lines
col	color of standard error curve
...	Other arguments that can be passed to glmreg.

Details

The function runs `glmreg` `nfolds+1` times; the first to compute the lambda sequence, and then to compute the fit with each of the folds omitted. The error or the log-likelihood value is accumulated, and the average value and standard deviation over the folds is computed. Note that `cv.glmreg` can be used to search for values for alpha: it is required to call `cv.glmreg` with a fixed vector `foldid` for different values of alpha.

Value

an object of class "`cv.glmreg`" is returned, which is a list with the ingredients of the cross-validation fit.

lambda	the values of lambda used in the fits.
cvm	The mean cross-validated error - a vector of length <code>length(lambda)</code> .
cvsd	estimate of standard error of <code>cvm</code> .
cvup	upper curve = <code>cvm+cvsd</code> .
cvlo	lower curve = <code>cvm-cvsd</code> .
nzero	number of non-zero coefficients at each lambda.
glmreg.fit	a fitted <code>glmreg</code> object for the full data.
lambda.min	value of lambda that gives minimum <code>cvm</code> .

Author(s)

Zhu Wang <zwang@connecticutchildrens.org>

References

Zhu Wang, Shuangge Ma, Michael Zappitelli, Chirag Parikh, Ching-Yun Wang and Prasad Devarajan (2014) *Penalized Count Data Regression with Application to Hospital Stay after Pediatric Cardiac Surgery*, *Statistical Methods in Medical Research*, in press

See Also

[glmreg](#) and [plot](#), [predict](#), and [coef](#) methods for "`cv.glmreg`" object.

Examples

```
## Not run:
data("bioChemists", package = "pscl")
fm_pois <- cv.glmreg(art ~ ., data = bioChemists, family = "poisson")
plot(fm_pois)
title("Poisson Family",line=2.5)

## End(Not run)
```

cv.glmregNB

Cross-validation for glmregNB

Description

Does k-fold cross-validation for glmregNB, produces a plot, and returns cross-validated loglikelihood values for lambda

Usage

```
cv.glmregNB(formula, data, weights, lambda=NULL,
n folds=10, foldid, plot.it=TRUE, se=TRUE, trace=FALSE,...)
```

Arguments

formula	symbolic description of the model
data	arguments controlling formula processing via model.frame .
weights	Observation weights; defaults to 1 per observation
lambda	Optional user-supplied lambda sequence; default is NULL, and glmregNB chooses its own sequence
n folds	number of folds - default is 10. Although n folds can be as large as the sample size (leave-one-out CV), it is not recommended for large datasets. Smallest value allowable is n folds=3
foldid	an optional vector of values between 1 and n fold identifying what fold each observation is in. If supplied, n fold can be missing.
plot.it	a logical value, to plot the estimated loglikelihood values if TRUE.
se	a logical value, to plot with standard errors.
trace	if TRUE, shows cross-validation progress
...	Other arguments that can be passed to glmregNB.

Details

The function runs `glmregNB` `nfolds+1` times; the first to get the `lambda` sequence, and then the remainder to compute the fit with each of the folds omitted. The error is accumulated, and the average error and standard deviation over the folds is computed. Note that `cv.glmregNB` does NOT search for values for `alpha`. A specific value should be supplied, else `alpha=1` is assumed by default. If users would like to cross-validate `alpha` as well, they should call `cv.glmregNB` with a pre-computed vector `foldid`, and then use this same fold vector in separate calls to `cv.glmregNB` with different values of `alpha`.

Value

an object of class "`cv.glmregNB`" is returned, which is a list with the ingredients of the cross-validation fit.

<code>lambda</code>	the values of <code>lambda</code> used in the fits.
<code>cvm</code>	The mean cross-validated error - a vector of length <code>length(lambda)</code> .
<code>cvsd</code>	estimate of standard error of <code>cvm</code> .
<code>cvup</code>	upper curve = <code>cvm+cvsd</code> .
<code>cvlo</code>	lower curve = <code>cvm-cvsd</code> .
<code>nzero</code>	number of non-zero coefficients at each <code>lambda</code> .
<code>fit</code>	a fitted <code>glmregNB</code> object for the full data.
<code>lambda.min</code>	value of <code>lambda</code> that gives minimum <code>cvm</code> .

Author(s)

Zhu Wang <zwang@connecticutchildrens.org>

References

Zhu Wang, Shuangge Ma, Michael Zappitelli, Chirag Parikh, Ching-Yun Wang and Prasad Devarajan (2014) *Penalized Count Data Regression with Application to Hospital Stay after Pediatric Cardiac Surgery*, *Statistical Methods in Medical Research*, in press

See Also

[glmregNB](#) and [plot](#), [predict](#), and [coef](#) methods for "`cv.glmregNB`" object.

Examples

```
## Not run:
data("bioChemists", package = "pscl")
fm_nb <- cv.glmregNB(art ~ ., data = bioChemists)
plot(fm_nb)

## End(Not run)
```

cv.glmreg_fit

Internal function of cross-validation for glmreg

Description

Internal function to conduct k-fold cross-validation for glmreg, produces a plot, and returns cross-validated loglikelihood values for lambda

Usage

```
cv.glmreg_fit(x, y, weights, lambda=NULL, balance=TRUE,
family=c("gaussian", "binomial", "poisson", "negbin"),
n folds=10, foldid, plot.it=TRUE, se=TRUE, trace=FALSE, ...)
```

Arguments

x	x matrix as in glmreg.
y	response y as in glmreg.
weights	Observation weights; defaults to 1 per observation
lambda	Optional user-supplied lambda sequence; default is NULL, and glmreg chooses its own sequence
balance	for family="binomial" only
family	response variable distribution
n folds	number of folds ≥ 3 , default is 10
foldid	an optional vector of values between 1 and nfold identifying what fold each observation is in. If supplied, nfold can be missing and will be ignored.
plot.it	a logical value, to plot the estimated loglikelihood values if TRUE.
se	a logical value, to plot with standard errors.
trace	if TRUE, shows cross-validation progress
...	Other arguments that can be passed to glmreg.

Details

The function runs glmreg n folds+1 times; the first to compute the lambda sequence, and then to compute the fit with each of the folds omitted. The error or the log-likelihood value is accumulated, and the average value and standard deviation over the folds is computed. Note that cv.glmreg can be used to search for values for alpha: it is required to call cv.glmreg with a fixed vector foldid for different values of alpha.

Value

an object of class "cv.glmreg" is returned, which is a list with the ingredients of the cross-validation fit.

lambda	the values of lambda used in the fits.
cvm	The mean cross-validated error - a vector of length length(lambda).
cvsd	estimate of standard error of cvm.
cvup	upper curve = cvm+cvsd.
cvlo	lower curve = cvm-cvsd.
nzero	number of non-zero coefficients at each lambda.
glmreg.fit	a fitted glmreg object for the full data.
lambda.min	value of lambda that gives minimum cvm.

Author(s)

Zhu Wang <zwang@connecticutchildrens.org>

References

Zhu Wang, Shuangge Ma, Michael Zappitelli, Chirag Parikh, Ching-Yun Wang and Prasad Devarajan (2014) *Penalized Count Data Regression with Application to Hospital Stay after Pediatric Cardiac Surgery*, *Statistical Methods in Medical Research*, in press

See Also

[glmreg](#) and [plot](#), [predict](#), and [coef](#) methods for "cv.glmreg" object.

cv.zipath

Cross-validation for zipath

Description

Does k-fold cross-validation for zipath, produces a plot, and returns cross-validated loglikelihood values for lambda

Usage

```
cv.zipath(formula, data, weights, nlambda=100, lambda.count=NULL, lambda.zero=NULL,
nfolds=10, foldid, plot.it=TRUE, se=TRUE, trace=FALSE,...)
## S3 method for class 'cv.zipath'
coef(object, which=object$lambda.which, model = c("full", "count", "zero"), ...)
```

Arguments

formula	symbolic description of the model
data	arguments controlling formula processing via <code>model.frame</code> .
weights	Observation weights; defaults to 1 per observation
nlambda	number of lambda value, default value is 10.
lambda.count	Optional user-supplied lambda.count sequence; default is NULL
lambda.zero	Optional user-supplied lambda.zero sequence; default is NULL
nfolds	number of folds ≥ 3 , default is 10
foldid	an optional vector of values between 1 and nfolds identifying what fold each observation is in. If supplied, nfolds can be missing and will be ignored.
plot.it	a logical value, to plot the estimated loglikelihood values if TRUE.
se	a logical value, to plot with standard errors.
trace	if TRUE, shows cross-validation progress
...	Other arguments that can be passed to zipath.
object	object of class cv.zipath.
which	Indices of the pair of penalty parameters lambda.count and lambda.zero at which estimates are extracted. By default, the one which generates the optimal cross-validation value.
model	character specifying for which component of the model the estimated coefficients should be extracted.

Details

The function runs zipath nfolds+1 times; the first to compute the (lambda.count, lambda.zero) sequence, and then to compute the fit with each of the folds omitted. The log-likelihood value is accumulated, and the average value and standard deviation over the folds is computed. Note that cv.zipath can be used to search for values for count.alpha or zero.alpha: it is required to call cv.zipath with a fixed vector foldid for different values of count.alpha or zero.alpha.

The method for `coef` by default return a single vector of coefficients, i.e., all coefficients are concatenated. By setting the model argument, the estimates for the corresponding model components can be extracted.

Value

an object of class "cv.zipath" is returned, which is a list with the components of the cross-validation fit.

fit	a fitted zipath object for the full data.
residmat	matrix for cross-validated log-likelihood at each (count.lambda, zero.lambda) sequence
bic	The cross-validated BIC value - a vector of length length(count.lambda).
fraction	a sequence from 1:nlambda. nlambda is the same as the argument if any one of (count.lambda, zero.lambda) is missing; otherwise nlambda=length(count.lambda)

cv	The mean cross-validated log-likelihood - a vector of length <code>length(count.lambda)</code> .
cv.error	estimate of standard error of cv.
foldid	an optional vector of values between 1 and <code>nfold</code> identifying what fold each observation is in.
lambda.which	index of <code>(count.lambda, zero.lambda)</code> that gives minimum cv.
lambda.optim	value of <code>(count.lambda, zero.lambda)</code> that gives minimum cv.

Author(s)

Zhu Wang <zwang@connecticutchildrens.org>

References

Zhu Wang, Shuangge Ma, Michael Zappitelli, Chirag Parikh, Ching-Yun Wang and Prasad Devarajan (2014) *Penalized Count Data Regression with Application to Hospital Stay after Pediatric Cardiac Surgery*, *Statistical Methods in Medical Research*, in press

Zhu Wang, Shuangge Ma, Ching-Yun Wang, Michael Zappitelli, Prasad Devarajan and Chirag R. Parikh (2014) *EM for Regularized Zero Inflated Regression Models with Applications to Postoperative Morbidity after Cardiac Surgery in Children*, *Statistics in Medicine*, in press

Zhu Wang, Shuangge Ma and Ching-Yun Wang (2014) *Variable selection for zero-inflated and overdispersed data with application to health care demand in Germany*, manuscript

See Also

[zipath](#) and [plot](#), [predict](#), and [coef](#) methods for "cv.zipath" object.

Examples

```
## Not run:
data("bioChemists", package = "pscl")
fm_zip <- cv.zipath(art ~ . | ., data = bioChemists, family = "poisson", nlambda=10)
coef(fm_zip)
fm_znb <- cv.zipath(art ~ . | ., data = bioChemists, family = "negbin", nlambda=10)
coef(fm_znb)

## End(Not run)
```

estfunReg

Extract Empirical First Derivative of Log-likelihood Function

Description

Generic function for extracting the empirical first derivative of log-likelihood function of a fitted regularized model.

Usage

```
estfunReg(x, ...)
```

Arguments

x a fitted model object.
... arguments passed to methods.

Value

A matrix containing the empirical first derivative of log-likelihood functions. Typically, this should be an $n \times k$ matrix corresponding to n observations and k parameters. The columns should be named as in `coef` or `terms`, respectively.

Author(s)

Zhu Wang <zwang@connecticutchildrens.org>

References

Zhu Wang, Shuangge Ma and Ching-Yun Wang (2014) *Variable selection for zero-inflated and overdispersed data with application to health care demand in Germany*, manuscript

See Also

[zipath](#)

Examples

```
data("bioChemists", package = "pscl")  
fm_zinb <- zipath(art ~ . | ., data = bioChemists, family = "negbin", nlambda=10)  
res <- estfunReg(fm_zinb, which=which.min(fm_zinb$bic))
```

glmreg

fit a GLM with lasso (or elasticnet), snet or mnet regularization

Description

Fit a generalized linear model via penalized maximum likelihood. The regularization path is computed for the lasso (or elasticnet penalty), scad (or snet) and mcp (or mnet penalty), at a grid of values for the regularization parameter lambda. Fits linear, logistic, poisson and negative binomial (fixed scale parameter) regression models.

Usage

```
## S3 method for class 'formula'
glmreg(formula, data, weights, offset=NULL, contrasts=NULL,
x.keep=FALSE, y.keep=TRUE, ...)
## S3 method for class 'matrix'
glmreg(x, y, weights, offset=NULL, ...)
## Default S3 method:
glmreg(x, ...)
```

Arguments

formula	symbolic description of the model, see details.
data	argument controlling formula processing via <code>model.frame</code> .
weights	optional numeric vector of weights. If <code>standardize=TRUE</code> , weights are renormalized to <code>weights/sum(weights)</code> . If <code>standardize=FALSE</code> , weights are kept as original input
x	input matrix, of dimension <code>nobs x nvars</code> ; each row is an observation vector
y	response variable. Quantitative for <code>family="gaussian"</code> . Non-negative counts for <code>family="poisson"</code> or <code>family="negbin"</code> . For <code>family="binomial"</code> should be either a factor with two levels or a vector of proportions.
x.keep, y.keep	logical values: keep response variables or keep response variable?
offset	Not implemented yet
contrasts	the contrasts corresponding to levels from the respective models
...	Other arguments passing to <code>glmreg_fit</code>

Details

The sequence of models implied by `lambda` is fit by coordinate descent. For `family="gaussian"` this is the lasso, mcp or scad sequence if `alpha=1`, else it is the enet, mnet or snet sequence. For the other families, this is a lasso (mcp, scad) or elasticnet (mnet, snet) regularization path for fitting the generalized linear regression paths, by maximizing the appropriate penalized log-likelihood. Note that the objective function for "gaussian" is

$$1/2 * weights * RSS + \lambda * penalty,$$

if `standardize=FALSE` and

$$1/2 * \frac{weights}{\sum(weights)} * RSS + \lambda * penalty,$$

if `standardize=TRUE`. For the other models it is

$$- \sum(weights * loglik) + \lambda * penalty$$

if `standardize=FALSE` and

$$- \frac{weights}{\sum(weights)} * loglik + \lambda * penalty$$

if `standardize=TRUE`.

Value

An object with S3 class "glmreg" for the various types of models.

call	the call that produced this object
b0	Intercept sequence of length length(lambda)
beta	A $nvars \times length(lambda)$ matrix of coefficients.
lambda	The actual sequence of lambda values used
dev	The computed deviance (for "gaussian", this is the R-square). The deviance calculations incorporate weights if present in the model. The deviance is defined to be $2*(loglike_sat - loglike)$, where <code>loglike_sat</code> is the log-likelihood for the saturated model (a model with a free parameter per observation).
nulldev	Null deviance (per observation). This is defined to be $2*(loglike_sat - loglike(Null))$; The NULL model refers to the intercept model.
nobs	number of observations
p11	penalized log-likelihood values for standardized coefficients in the IRLS iterations.
p11res	penalized log-likelihood value for the estimated model on the original scale of coefficients

Author(s)

Zhu Wang <zwang@connecticutchildrens.org>

References

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

Zhu Wang, Shuangge Ma, Michael Zappitelli, Chirag Parikh, Ching-Yun Wang and Prasad Devarajan (2014) *Penalized Count Data Regression with Application to Hospital Stay after Pediatric Cardiac Surgery*, *Statistical Methods in Medical Research*, in press

See Also

[print](#), [predict](#), [coef](#) and [plot](#) methods, and the [cv.glmreg](#) function.

Examples

```
#binomial
x=matrix(rnorm(100*20),100,20)
g2=sample(0:1,100,replace=TRUE)
fit2=glmreg(x,g2,family="binomial")
#poisson and negative binomial
data("bioChemists", package = "pscl")
fm_pois <- glmreg(art ~ ., data = bioChemists, family = "poisson")
coef(fm_pois)
fm_nb1 <- glmreg(art ~ ., data = bioChemists, family = "negbin", theta=1)
coef(fm_nb1)
```

```
## Not run:
fm_nb2 <- glmregNB(art ~ ., data = bioChemists)
coef(fm_nb2)

## End(Not run)
```

glmregNB	<i>fit a negative binomial model with lasso (or elasticnet), snet and mnet regularization</i>
----------	---

Description

Fit a negative binomial linear model via penalized maximum likelihood. The regularization path is computed for the lasso (or elasticnet penalty), snet and mnet penalty, at a grid of values for the regularization parameter lambda.

Usage

```
glmregNB(formula, data, weights, nlambda = 100, lambda=NULL, lambda.min.ratio =
ifelse(nobs<nvars,0.05,0.001), alpha=1, gamma=3, rescale=TRUE, standardize = TRUE,
penalty.factor = rep(1, nvars), thresh = 0.001, maxit.theta = 25, maxit=1000,
eps=.Machine$double.eps, trace=FALSE, start = NULL, etastart = NULL, mustart = NULL,
theta.est=TRUE, theta0=NULL, init.theta=ifelse(theta.est, theta0[1],NULL),link=log,
penalty=c("enet","mnet","snet"), method="glmreg_fit", model=TRUE,
x.keep=FALSE, y.keep=TRUE, contrasts=NULL, convex=FALSE, ...)
```

Arguments

formula	formula used to describe a model.
data	argument controlling formula processing via <code>model.frame</code> .
weights	observation weights. Default is 1 for each observation
nlambda	The number of lambda values - default is 100.
lambda	A user supplied lambda sequence
lambda.min.ratio	Smallest value for lambda, as a fraction of lambda.max, the (data derived) entry value (i.e. the smallest value for which all coefficients are zero). The default depends on the sample size nobs relative to the number of variables nvars. If nobs > nvars, the default is 0.001, close to zero. If nobs < nvars, the default is 0.05.
alpha	The L2 penalty mixing parameter, with $0 \leq \alpha \leq 1$. alpha=1 is lasso (mcp, scad) penalty; and alpha=0 the ridge penalty.
gamma	The tuning parameter of the snet or mnet penalty.
rescale	logical value, if TRUE, adaptive rescaling of the penalty parameter for penalty="mnet" or penalty="snet" with family other than "gaussian". See reference

standardize	Logical flag for x variable standardization, prior to fitting the model sequence. The coefficients are always returned on the original scale. Default is standardize=TRUE. If variables are in the same units already, you might not wish to standardize.
penalty.factor	This is a number that multiplies lambda to allow differential shrinkage of coefficients. Can be 0 for some variables, which implies no shrinkage, and that variable is always included in the model. Default is same shrinkage for all variables.
thresh	Convergence threshold for coordinate descent. Defaults value is 1e-6.
maxit.theta	Maximum number of iterations for estimating theta scaling parameter
maxit	Maximum number of coordinate descent iterations for each lambda value; default is 1000.
eps	If a number is less than eps in magnitude, then this number is considered as 0
trace	If TRUE, fitting progress is reported
start, etastart, mustart, ...	arguments for the link{glmreg} function
init.theta	initial scaling parameter theta
theta.est	Estimate scale parameter theta? Default is TRUE
theta0	initial scale parameter vector theta, with length nlambda if theta.est=FALSE. Default is NULL
convex	Calculate index for which objective function ceases to be locally convex? Default is FALSE and only useful if penalty="mnet" or "snet".
link	link function, default is log
penalty	Type of regularization
method	estimation method
model, x.keep, y.keep	logicals. If TRUE the corresponding components of the fit (model frame, response, model matrix) are returned.
contrasts	the contrasts corresponding to levels from the respective models

Details

The sequence of models implied by lambda is fit by coordinate descent. This is a lasso (mcp, scad) or elasticnet (mnet, snet) regularization path for fitting the negative binomial linear regression paths, by maximizing the penalized log-likelihood. Note that the objective function is

$$-\sum(weights * loglik) + \lambda * penalty$$

if standardize=FALSE and

$$-\frac{weights}{\sum(weights)} * loglik + \lambda * penalty$$

if standardize=TRUE.

Value

An object with S3 class "glmreg", "glmregNB" for the various types of models.

call	the call that produced the model fit
b0	Intercept sequence of length length(lambda)
beta	A $nvars \times \text{length}(\text{lambda})$ matrix of coefficients.
lambda	The actual sequence of lambda values used
dev	The computed deviance. The deviance calculations incorporate weights if present in the model. The deviance is defined to be $2 * (\text{loglike_sat} - \text{loglike})$, where <code>loglike_sat</code> is the log-likelihood for the saturated model (a model with a free parameter per observation).
nulldev	Null deviance (per observation). This is defined to be $2 * (\text{loglike_sat} - \text{loglike}(\text{Null}))$; The NULL model refers to the intercept model.
nobs	number of observations

Author(s)

Zhu Wang <zwang@connecticutchildrens.org>

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.

Zhu Wang, Shuangge Ma, Michael Zappitelli, Chirag Parikh, Ching-Yun Wang and Prasad Devarajan (2014) *Penalized Count Data Regression with Application to Hospital Stay after Pediatric Cardiac Surgery*, *Statistical Methods in Medical Research*, in press

See Also

[print](#), [predict](#), [coef](#) and [plot](#) methods, and the [cv.glmregNB](#) function.

Examples

```
## Not run:
data("bioChemists", package = "pscl")
fm_nb <- glmregNB(art ~ ., data = bioChemists)
coef(fm_nb)

## End(Not run)
```

glmreg_fit	<i>Internal function to fit a GLM with lasso (or elasticnet), snet and mnet regularization</i>
------------	--

Description

Fit a generalized linear model via penalized maximum likelihood. The regularization path is computed for the lasso (or elasticnet penalty), snet and mnet penalty, at a grid of values for the regularization parameter lambda. Fits linear, logistic, poisson and negative binomial (fixed scale parameter) regression models.

Usage

```
glmreg_fit(x, y, weights, start=NULL, etastart=NULL, mustart=NULL,
nlambda=100, lambda=NULL, lambda.min.ratio=ifelse(nobs<nvars,.05,.001),alpha=1,
gamma=3, rescale=TRUE, standardize=TRUE, penalty.factor = rep(1, nvars), thresh=1e-6,
eps.bino=1e-5, maxit=1000, eps=.Machine$double.eps, theta,
family=c("gaussian", "binomial", "poisson", "negbin"), penalty=c("enet", "mnet", "snet"),
convex=FALSE, x.keep=FALSE, y.keep=TRUE, trace=FALSE)
```

Arguments

x	input matrix, of dimension nobs x nvars; each row is an observation vector.
y	response variable. Quantitative for family="gaussian". Non-negative counts for family="poisson" or family="negbin". For family="binomial" should be either a factor with two levels or a vector of proportions.
weights	observation weights. Can be total counts if responses are proportion matrices. Default is 1 for each observation
start	starting values for the parameters in the linear predictor.
etastart	starting values for the linear predictor.
mustart	starting values for the vector of means.
nlambda	The number of lambda values - default is 100. The sequence may be truncated before nlambda is reached if a close to saturated model is fitted. See also satu.
lambda	A user supplied lambda sequence
lambda.min.ratio	Smallest value for lambda, as a fraction of lambda.max, the (data derived) entry value (i.e. the smallest value for which all coefficients are zero except the intercept). Note, there is a closed formula for lambda.max for penalty="enet". If rescale=TRUE, lambda.max is the same for penatly="mnet" or "snet". Otherwise, some modifications are required. For instance, for small gamma value, half of the square root (if lambda.max is too small) of the computed lambda.max can be used when penalty="mnet" or "snet". The default of lambda.min.ratio depends on the sample size nobs relative to the number of variables nvars. If nobs > nvars, the default is 0.001, close to zero. If nobs < nvars, the default is 0.05.

alpha	The L2 penalty mixing parameter, with $0 \leq \alpha \leq 1$. alpha=1 is lasso (mcp, scad) penalty; and alpha=0 the ridge penalty.
gamma	The tuning parameter of the snet or mnet penalty.
rescale	logical value, if TRUE, adaptive rescaling of the penalty parameter for penalty="mnet" or penalty="snet" with family other than "gaussian". See reference
standardize	logical value for x variable standardization, prior to fitting the model sequence. The coefficients are always returned on the original scale. Default is standardize=TRUE.
penalty.factor	This is a number that multiplies lambda to allow differential shrinkage of coefficients. Can be 0 for some variables, which implies no shrinkage, and that variable is always included in the model. Default is same shrinkage for all variables.
thresh	Convergence threshold for coordinate descent. Defaults value is 1e-6.
eps.bino	a lower bound of probabilities to be claimed as zero, for computing weights and related values when family="binomial".
maxit	Maximum number of coordinate descent iterations for each lambda value; default is 1000.
eps	If a coefficient is less than eps in magnitude, then it is reported to be 0
convex	Calculate index for which objective function ceases to be locally convex? Default is FALSE and only useful if penalty="mnet" or "snet".
theta	an overdispersion scaling parameter for family="negbin"
family	Response type (see above)
penalty	Type of regularization
x.keep, y.keep	For glmreg: logical values indicating whether the response vector and model matrix used in the fitting process should be returned as components of the returned value. For glmreg_fit: x is a design matrix of dimension n * p, and x is a vector of observations of length n.
trace	If TRUE, fitting progress is reported

Details

The sequence of models implied by lambda is fit by coordinate descent. For family="gaussian" this is the lasso, mcp or scad sequence if alpha=1, else it is the enet, mnet or snet sequence. For the other families, this is a lasso (mcp, scad) or elasticnet (mnet, snet) regularization path for fitting the generalized linear regression paths, by maximizing the appropriate penalized log-likelihood. Note that the objective function for "gaussian" is

$$1/2 * weights * RSS + \lambda * penalty,$$

if standardize=FALSE and

$$1/2 * \frac{weights}{\sum(weights)} * RSS + \lambda * penalty,$$

if standardize=TRUE. For the other models it is

$$- \sum(weights * loglik) + \lambda * penalty$$

if standardize=FALSE and

$$-\frac{weights}{\sum(weights)} * loglik + \lambda * penalty$$

if standardize=TRUE.

Value

An object with S3 class "glmreg" for the various types of models.

call	the call that produced the model fit
b0	Intercept sequence of length length(lambda)
beta	A nvars x length(lambda) matrix of coefficients.
lambda	The actual sequence of lambda values used
satu	satu=1 if a saturated model (deviance/null deviance < 0.05) is fit. Otherwise satu=0. The number of nlambda sequence may be truncated before nlambda is reached if satu=1.
dev	The computed deviance (for "gaussian", this is the R-square). The deviance calculations incorporate weights if present in the model. The deviance is defined to be 2*(loglike_sat - loglike), where loglike_sat is the log-likelihood for the saturated model (a model with a free parameter per observation).
nulldev	Null deviance (per observation). This is defined to be 2*(loglike_sat - loglike(NULL)); The NULL model refers to the intercept model.
nobs	number of observations

Author(s)

Zhu Wang <zwang@connecticutchildrens.org>

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.

Zhu Wang, Shuangge Ma, Michael Zappitelli, Chirag Parikh, Ching-Yun Wang and Prasad Devarajan (2014) *Penalized Count Data Regression with Application to Hospital Stay after Pediatric Cardiac Surgery*, *Statistical Methods in Medical Research*, in press

See Also

[glmreg](#)

`hessianReg`*Hessian Matrix of Regularized Estimators*

Description

Constructing Hessian matrix for regularized regression parameters.

Usage

```
hessianReg(x, which, ...)
```

Arguments

<code>x</code>	a fitted model object.
<code>which</code>	which penalty parameter(s)?
<code>...</code>	arguments passed to the <code>meatReg</code> function.

Details

`hessianReg` is a function to compute the Hessian matrix estimate of non-zero regularized estimators. Implemented only for `zipath` object with `family="negbin"` in the current version.

Value

A matrix containing the Hessian matrix estimate for the non-zero parameters.

Author(s)

Zhu Wang <zwang@connecticutchildrens.org>

References

Zhu Wang, Shuangge Ma and Ching-Yun Wang (2014) *Variable selection for zero-inflated and overdispersed data with application to health care demand in Germany*, manuscript

See Also

[breadReg](#), [meatReg](#)

Examples

```
data("bioChemists", package = "pscl")
fm_zinb <- zipath(art ~ . | ., data = bioChemists, family = "negbin", nlambda=10)
hessianReg(fm_zinb, which=which.min(fm_zinb$bic))
```

meatReg	<i>Meat Matrix Estimator</i>
---------	------------------------------

Description

Estimating the variance of the first derivative of log-likelihood function

Usage

```
meatReg(x, which, ...)
```

Arguments

x	a fitted model object. Currently only implemented for zipath object with family="negbin"
which	which penalty parameter(s)?
...	arguments passed to the estfunReg function.

Details

See reference below

Value

A $k \times k$
covariance matrix of first derivative of log-likelihood function

Author(s)

Zhu Wang <zwang@connecticutchildrens.org>

References

Zhu Wang, Shuangge Ma and Ching-Yun Wang (2014) *Variable selection for zero-inflated and overdispersed data with application to health care demand in Germany*, manuscript

See Also

[sandwichReg](#), [breadReg](#), [estfunReg](#)

Examples

```
data("bioChemists", package = "pscl")
fm_zinb <- zipath(art ~ . | ., data = bioChemists, family = "negbin", nlambd=10)
meatReg(fm_zinb, which=which.min(fm_zinb$bic))
```

methods

Methods for mpath Objects

Description

Methods for models fitted by coordinate descent algorithms.

Usage

```
## S3 method for class 'glmreg'  
AIC(object, ..., k)  
## S3 method for class 'zipath'  
AIC(object, ..., k)  
## S3 method for class 'glmreg'  
BIC(object, ...)  
## S3 method for class 'zipath'  
BIC(object, ...)
```

Arguments

object	objects of class glmreg or zipath.
...	additional arguments passed to callies.
k	numeric, the <i>penalty</i> per parameter to be used; the default $k = 2$ is the classical AIC. k has been hard coded in the function and there is no impact to the value of AIC if k is changed

Author(s)

Zhu Wang <zwang@connecticutchildrens.org>

References

Zhu Wang, Shuangge Ma, Michael Zappitelli, Chirag Parikh, Ching-Yun Wang and Prasad Devarajan (2014) *Penalized Count Data Regression with Application to Hospital Stay after Pediatric Cardiac Surgery*, *Statistical Methods in Medical Research*, in press

Zhu Wang, Shuangge Ma, Ching-Yun Wang, Michael Zappitelli, Prasad Devarajan and Chirag R. Parikh (2014) *EM for Regularized Zero Inflated Regression Models with Applications to Postoperative Morbidity after Cardiac Surgery in Children*, *Statistics in Medicine*, in press

Zhu Wang, Shuangge Ma and Ching-Yun Wang (2014) *Variable selection for zero-inflated and overdispersed data with application to health care demand in Germany*, manuscript

plot.glmreg *plot coefficients from a "glmreg" object*

Description

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

Usage

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

Arguments

x	fitted "glmreg" model
xvar	What is on the X-axis. "norm" plots against the L1-norm of the coefficients, "lambda" against the log-lambda sequence, and "dev" against the percent deviance explained.
label	If TRUE, label the curves with variable sequence numbers.
shade	Should nonconvex region be shaded? Default is TRUE. Code developed for all weights=1 only
...	Other graphical parameters to plot

Details

A coefficient profile plot is produced.

Author(s)

Zhu Wang zwang@connecticutchildrens.org

See Also

glmreg, and print, predict and coef methods.

Examples

```
x=matrix(rnorm(100*20),100,20)  
y=rnorm(100)  
g2=sample(1:2,100,replace=TRUE)  
fit1=glmreg(x,y)  
plot(fit1)  
plot(fit1,xvar="lambda",label=TRUE)
```

predict.glmreg *Model predictions based on a fitted "glmreg" object.*

Description

This function returns predictions from a fitted "glmreg" object.

Usage

```
## S3 method for class 'glmreg'
predict(object,newx,which=1:length(object$lambda),
type=c("link","response","class","coefficients","nonzero"), na.action=na.pass, ...)
## S3 method for class 'glmreg'
coef(object,which=1:length(object$lambda),...)
```

Arguments

object	Fitted "glmreg" model object.
newx	Matrix of values at which predictions are to be made. Not used for type="coefficients"
which	Indices of the penalty parameter lambda at which predictions are required. By default, all indices are returned.
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.
na.action	action for missing data value
...	arguments for predict

Value

The returned object depends on type.

Author(s)

Zhu Wang <zwang@connecticutchildrens.org>

References

Zhu Wang, Shuangge Ma, Michael Zappitelli, Chirag Parikh, Ching-Yun Wang and Prasad Devarajan (2014) *Penalized Count Data Regression with Application to Hospital Stay after Pediatric Cardiac Surgery*, *Statistical Methods in Medical Research*, in press

See Also

[glmreg](#)

Examples

```
## Dobson (1990) Page 93: Randomized Controlled Trial :
counts <- c(18,17,15,20,10,20,25,13,12)
outcome <- gl(3,1,9)
treatment <- gl(3,3)
print(d.AD <- data.frame(treatment, outcome, counts))
fit <- glmreg(counts ~ outcome + treatment, data=d.AD, family="poisson")
summary(fit)
coef(fit)
```

predict.zipath *Methods for zipath Objects*

Description

Methods for extracting information from fitted penalized zero-inflated regression model objects of class "zipath".

Usage

```
## S3 method for class 'zipath'
predict(object, newdata, which = 1:object$nlambda,
        type = c("response", "prob", "count", "zero", "nonzero"), na.action = na.pass,
        at = NULL, ...)
## S3 method for class 'zipath'
residuals(object, type = c("pearson", "response"), ...)

## S3 method for class 'zipath'
coef(object, which=1:object$nlambda, model = c("full", "count", "zero"), ...)

## S3 method for class 'zipath'
terms(x, model = c("count", "zero"), ...)
## S3 method for class 'zipath'
model.matrix(object, model = c("count", "zero"), ...)
```

Arguments

object, x	an object of class "zipath" as returned by zipath .
newdata	optionally, a data frame in which to look for variables with which to predict. If omitted, the original observations are used.
which	Indices of the penalty parameters lambda at which predictions are required. By default, all indices are returned.
type	character specifying the type of predictions or residuals, respectively. For details see below.
na.action	function determining what should be done with missing values in newdata. The default is to predict NA.

at	optionally, if type = "prob", a numeric vector at which the probabilities are evaluated. By default $0:\max(y)$ is used where y is the original observed response.
model	character specifying for which component of the model the terms or model matrix should be extracted.
...	currently not used.

Details

Re-uses the design of function `zeroinfl` in package `pscl` (see reference). A set of standard extractor functions for fitted model objects is available for objects of class "zipath", including methods to the generic functions `print` and `summary` which print the estimated coefficients along with some further information. As usual, the `summary` method returns an object of class "summary.zipath" containing the relevant summary statistics which can subsequently be printed using the associated `print` method.

The methods for `coef` by default return a single vector of coefficients and their associated covariance matrix, respectively, i.e., all coefficients are concatenated. By setting the `model` argument, the estimates for the corresponding model components can be extracted.

Both the `fitted` and `predict` methods can compute fitted responses. The latter additionally provides the predicted density (i.e., probabilities for the observed counts), the predicted mean from the count component (without zero inflation) and the predicted probability for the zero component. The `residuals` method can compute raw residuals (observed - fitted) and Pearson residuals (raw residuals scaled by square root of variance function).

Author(s)

Zhu Wang <zwang@connecticutchildrens.org>

References

Zhu Wang, Shuangge Ma, Michael Zappitelli, Chirag Parikh, Ching-Yun Wang and Prasad Devarajan (2014) *Penalized Count Data Regression with Application to Hospital Stay after Pediatric Cardiac Surgery*, *Statistical Methods in Medical Research*, in press

Zhu Wang, Shuangge Ma, Ching-Yun Wang, Michael Zappitelli, Prasad Devarajan and Chirag R. Parikh (2014) *EM for Regularized Zero Inflated Regression Models with Applications to Postoperative Morbidity after Cardiac Surgery in Children*, *Statistics in Medicine*, in press

Zhu Wang, Shuangge Ma and Ching-Yun Wang (2014) *Variable selection for zero-inflated and overdispersed data with application to health care demand in Germany*, manuscript

See Also

[zipath](#)

Examples

```
## Not run:
data("bioChemists", package = "pscl")
fm_zip <- zipath(art ~ . | ., data = bioChemists, nlambda=10)
```

```

plot(residuals(fm_zip) ~ fitted(fm_zip))
coef(fm_zip, model = "count")
coef(fm_zip, model = "zero")
summary(fm_zip)
logLik(fm_zip)

## End(Not run)

```

pval.zipath	<i>compute p-values from penalized zero-inflated model with multi-split data</i>
-------------	--

Description

compute p-values from penalized zero-inflated Poisson, negative binomial and geometric model with multi-split data

Usage

```

pval.zipath(formula, data, weights, subset, na.action, offset, standardize=TRUE,
family = c("poisson", "negbin", "geometric"),penalty = c("enet", "mnet", "snet"),
gamma.count = 3, gamma.zero = 3, prop=0.5, trace=TRUE, B=10, ...)

```

Arguments

formula	symbolic description of the model, see details.
data	argument controlling formula processing via <code>model.frame</code> .
weights	optional numeric vector of weights. If <code>standardize=TRUE</code> , weights are renormalized to <code>weights/sum(weights)</code> . If <code>standardize=FALSE</code> , weights are kept as original input
subset	subset of data
na.action	how to deal with missing data
offset	Not implemented yet
standardize	logical value, should variables be standardized?
family	family to fit zipath
penalty	penalty considered as one of <code>enet</code> , <code>mnet</code> , <code>snet</code> .
gamma.count	The tuning parameter of the <code>snet</code> or <code>mnet</code> penalty for the count part of model.
gamma.zero	The tuning parameter of the <code>snet</code> or <code>mnet</code> penalty for the zero part of model.
prop	proportion of data split, default is 50/50 split
trace	logical value, if <code>TRUE</code> , print detailed calculation results
B	number of repeated multi-split replications
...	Other arguments passing to <code>glmreg_fit</code>

Details

compute p-values from penalized zero-inflated Poisson, negative binomial and geometric model with multi-split data

Value

count.pval	raw p-values in the count component
zero.pval	raw p-values in the zero component
count.pval.q	Q value for the count component
zero.pval.q	Q value for the zero component

Author(s)

Zhu Wang <zwang@connecticutchildrens.org>

References

Nicolai Meinshausen, Lukas Meier and Peter Buehlmann (2013) *p-Values for High-Dimensional Regression*, *Journal of the American Statistical Association*, **104(488)**, 1671–1681

Zhu Wang, Shuangge Ma, Ching-Yun Wang, Michael Zappitelli, Prasad Devarajan and Chirag R. Parikh (2014) *EM for Regularized Zero Inflated Regression Models with Applications to Postoperative Morbidity after Cardiac Surgery in Children*, *Statistics in Medicine*, in press

Zhu Wang, Shuangge Ma and Ching-Yun Wang (2014) *Variable selection for zero-inflated and overdispersed data with application to health care demand in Germany*, manuscript

rzi	<i>random number generation of zero-inflated count response</i>
-----	---

Description

random number generation of zero-inflated count response

Usage

```
rzi(n, x, z, a, b, theta=1, family=c("poisson", "negbin", "geometric"), infl=TRUE)
```

Arguments

n	sample size of random number generation
x	design matrix of count model
z	design matrix of zero model
a	coefficient vector for x, length must be the same as column size of x
b	coefficient vector for z, length must be the same as column size of z
theta	dispersion parameter for family="negbin"
family	distribution of count model
infl	logical value, if TRUE, zero-inflated count response

Details

random number generation of zero-inflated count response

Value

numeric vector of zero-inflated count response

Author(s)

Zhu Wang <zwang@connecticutchildrens.org>

References

Zhu Wang, Shuangge Ma, Ching-Yun Wang, Michael Zappitelli, Prasad Devarajan and Chirag R. Parikh (2014) *EM for Regularized Zero Inflated Regression Models with Applications to Postoperative Morbidity after Cardiac Surgery in Children*, *Statistics in Medicine*, in press

Zhu Wang, Shuangge Ma and Ching-Yun Wang (2014) *Variable selection for zero-inflated and overdispersed data with application to health care demand in Germany*, manuscript

sandwichReg

Making Sandwiches with Bread and Meat for Regularized Estimators

Description

Constructing sandwich covariance matrix estimators by multiplying bread and meat matrices for regularized regression parameters.

Usage

```
sandwichReg(x, breadreg.=breadReg, meatreg.=meatReg, which, log=FALSE, ...)
```

Arguments

x	a fitted model object.
breadreg.	either a breadReg matrix or a function for computing this via breadreg.(x).
meatreg.	either a breadReg matrix or a function for computing this via meatreg.(x, ...).
which	which penalty parameters(s) to compute?
log	if TRUE, the corresponding element is with respect to log(theta) in negative binomial regression. Otherwise, for theta
...	arguments passed to the meatReg function.

Details

sandwichReg is a function to compute an estimator for the covariance of the non-zero parameters. It takes a breadReg matrix (i.e., estimator of the expectation of the negative derivative of the penalized estimating functions) and a meatReg matrix (i.e., estimator of the variance of the log-likelihood function) and multiplies them to a sandwich with meat between two slices of bread. By default [breadReg](#) and [meatReg](#) are called. Implemented only for zipath object with family="negbin" in the current version.

Value

A matrix containing the sandwich covariance matrix estimate for the non-zero parameters.

Author(s)

Zhu Wang <zwang@connecticutchildrens.org>

References

Zhu Wang, Shuangge Ma and Ching-Yun Wang (2014) *Variable selection for zero-inflated and overdispersed data with application to health care demand in Germany*, manuscript

See Also

[breadReg](#), [meatReg](#)

Examples

```
data("bioChemists", package = "pscl")
fm_zinb <- zipath(art ~ . | ., data = bioChemists, family = "negbin", nlambda=10)
sandwichReg(fm_zinb, which=which.min(fm_zinb$bic))
```

 se

Standard Error of Regularized Estimators

Description

Generic function for computing standard errors of non-zero regularized estimators

Usage

```
se(x, which, log=TRUE, ...)
```

Arguments

x	a fitted model object.
which	which penalty parameter(s)?
log	if TRUE, the computed standard error is for log(theta) for negative binomial regression, otherwise, for theta.
...	arguments passed to methods.

Value

A vector containing standard errors of non-zero regularized estimators.

Author(s)

Zhu Wang <zwang@connecticutchildrens.org>

References

Zhu Wang, Shuangge Ma and Ching-Yun Wang (2014) *Variable selection for zero-inflated and overdispersed data with application to health care demand in Germany*, manuscript

See Also

[zipath](#)

Examples

```
data("bioChemists", package = "pscl")
fm_zinb <- zipath(art ~ . | ., data = bioChemists, family = "negbin", nlambd=10)
res <- se(fm_zinb, which=which.min(fm_zinb$bic))
```

summary.glmregNB

Summary Method Function for Objects of Class 'glmregNB'

Description

Summary results of fitted penalized negative binomial regression model

Usage

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

Arguments

object	fitted model object of class glmregNB.
...	arguments passed to or from other methods.

Details

This function is a method for the generic function `summary()` for class "glmregNB". It can be invoked by calling `summary(x)` for an object `x` of the appropriate class, or directly by calling `summary.glmregNB(x)` regardless of the class of the object.

Value

Summary of fitted penalized negative binomial model

Author(s)

Zhu Wang <zwang@connecticutchildrens.org>

References

Zhu Wang, Shuangge Ma, Michael Zappitelli, Chirag Parikh, Ching-Yun Wang and Prasad Devarajan (2014) *Penalized Count Data Regression with Application to Hospital Stay after Pediatric Cardiac Surgery*, *Statistical Methods in Medical Research*, in press

See Also

[summary](#), [glm.nb](#)

Examples

```
## Not run:
data(quine, package="MASS")
summary(glmregNB(Days ~ Eth*Age*Lrn*Sex, quine, link = log))

## End(Not run)
```

tuning.zipath

find optimal penalized zero-inflated model

Description

Fit penalized zero-inflated models, generate multiple paths with varying penalty parameters, therefore determine optimal penalty parameters

Usage

```
tuning.zipath(formula, data, weights, subset, na.action, offset, standardize=TRUE,
family = c("poisson", "negbin", "geometric"), penalty = c("enet", "mnet", "snet"),
lambdaCountRatio = .0001, lambdaZeroRatio = c(.1, .01, .001),
maxit.theta=1, gamma.count=3, gamma.zero=3, ...)
```

Arguments

formula	symbolic description of the model, see details.
data	argument controlling formula processing via model.frame .
weights	optional numeric vector of weights. If standardize=TRUE, weights are renormalized to weights/sum(weights). If standardize=FALSE, weights are kept as original input
subset	subset of data
na.action	how to deal with missing data

offset	Not implemented yet
standardize	logical value, should variables be standardized?
family	family to fit
penalty	penalty considered as one of enet, mnet, snet.
lambdaCountRatio, lambdaZeroRatio	Smallest value for lambda.count and lambda.zero, respectively, as a fraction of lambda.max, the (data derived) entry value (i.e. the smallest value for which all coefficients are zero except the intercepts). This lambda.max can be a surrogate value for penalty="mnet" or "snet"
maxit.theta	For family="negbin", the maximum iteration allowed for estimating scale parameter theta. Note, the default value 1 is for computing speed purposes, and is typically too small and less desirable in real data analysis
gamma.count	The tuning parameter of the snet or mnet penalty for the count part of model.
gamma.zero	The tuning parameter of the snet or mnet penalty for the zero part of model.
...	Other arguments passing to zipath

Details

find optimal lambdaZeroRatio for penalized zero-inflated Poisson, negative binomial and geometric model

Value

An object of class zipath with the optimal lambdaZeroRatio

Author(s)

Zhu Wang <zwang@connecticutchildrens.org>

References

Zhu Wang, Shuangge Ma, Michael Zappitelli, Chirag Parikh, Ching-Yun Wang and Prasad Devarajan (2014) *Penalized Count Data Regression with Application to Hospital Stay after Pediatric Cardiac Surgery*, *Statistical Methods in Medical Research*, in press

Zhu Wang, Shuangge Ma, Ching-Yun Wang, Michael Zappitelli, Prasad Devarajan and Chirag R. Parikh (2014) *EM for Regularized Zero Inflated Regression Models with Applications to Postoperative Morbidity after Cardiac Surgery in Children*, *Statistics in Medicine*, in press

Zhu Wang, Shuangge Ma and Ching-Yun Wang (2014) *Variable selection for zero-inflated and overdispersed data with application to health care demand in Germany*, manuscript

See Also

[zipath](#)

Examples

```
## Not run:
## data
data("bioChemists", package = "pscl")

## inflation with regressors
## ("art ~ . | ." is "art ~ fem + mar + kid5 + phd + ment | fem + mar + kid5 + phd + ment")
fm_zip2 <- tuning.zipath(art ~ . | ., data = bioChemists, nlambda=10)
summary(fm_zip2)
fm_zinb2 <- tuning.zipath(art ~ . | ., data = bioChemists, family = "negbin", nlambda=10)
summary(fm_zinb2)

## End(Not run)
```

 vuong.test

Vuong's non-nested hypothesis test

Description

Compares two models fit to the same data that do not nest via Vuong's non-nested test.

Usage

```
vuong.test(m1, s1, m2, s2, type=c("None", "AIC", "BIC"), digits = getOption("digits"))
```

Arguments

m1	model 1, an object inheriting from class glm, negbin, zeroinfl, glmreg, glmregNB or zipath
s1	an interger value between 1 and m1\$nlambda, only useful when model 1 is from glmreg, glmregNB or zipath
m2	model 2, as for model 1
s2	similar to s1
type	type of penalty on number of parameters when comparing models
digits	significant digits in printed result

Details

The Vuong non-nested test is based on a comparison of the predicted probabilities of two models that do not nest, subject to penalty on number of parameters. The test can be used for penalized or non-penalized Poisson (negative binomial) regression models, zero-inflated Poisson (negative binomial) models.

Value

The test-statistic and p value

Author(s)

Zhu Wang zwang@connecticutchildrens.org

References

Vuong, Q.H. 1989. Likelihood ratio tests for model selection and non-nested hypotheses. *Econometrica*. **57**, 307–333.

Achim Zeileis, Christian Kleiber and Simon Jackman (2008), *Regression Models for Count Data in R*, *Journal of Statistical Software*, **27(8)**, <http://www.jstatsoft.org/v27/i08/>

Examples

```
## Not run:
data("bioChemists", package="pscl")
## compare penalized Poisson GLM and ZIP
glm1 <- glmreg(art ~ ., data = bioChemists, x.keep=TRUE, family = "poisson")
zip <- zipath(art ~ . | ., data = bioChemists)
vuong.test(glm1, which.min(glm1$bic), zip, which.min(zip$bic))

## compare penallized negbin with zero-inflated negbin
nb1 <- glmregNB(art ~ ., data=bioChemists, x.keep=TRUE)
zinb <- zipath(art ~ . | ., data = bioChemists, family = "negbin")
vuong.test(nb1, which.min(nb1$bic), zinb, which.min(zinb$bic))

## End(Not run)
```

zipath

*Fit zero-inflated count data linear model with lasso (or elasticnet),
snet or mnet regularization*

Description

Fit zero-inflated regression models for count data via penalized maximum likelihood.

Usage

```
zipath(formula, data, weights, subset, na.action, offset,
standardize = TRUE, family = c("poisson", "negbin", "geometric"),
link = c("logit", "probit", "cloglog", "cauchit", "log"),
penalty = c("enet", "mnet", "snet"), start = NULL, model = TRUE,
y = TRUE, x = FALSE, nlambdas = 100, lambda.count = NULL, lambda.zero = NULL,
lambda.count.min.ratio = .0001, lambda.zero.min.ratio = .1,
alpha.count = 1, alpha.zero = alpha.count, gamma.count = 3,
gamma.zero = gamma.count, rescale=FALSE, init.theta, theta.fixed=FALSE,
EM = TRUE, maxit.em=200, convtype=c("count", "both"), maxit = 1000,
maxit.theta = 1, reltol = 1e-5, eps.bino=1e-5, shortlist=FALSE, trace = FALSE, ...)
```

Arguments

formula	symbolic description of the model, see details.
weights	optional numeric vector of weights.
data, subset, na.action	arguments controlling formula processing via <code>model.frame</code> .
offset	optional numeric vector with an a priori known component to be included in the linear predictor of the count model. See below for more information on offsets.
standardize	Logical flag for x variable standardization, prior to fitting the model sequence. The coefficients are always returned on the original scale. Default is <code>standardize=TRUE</code> .
family	character specification of count model family (a log link is always used).
link	character specification of link function in the binary zero-inflation model (a binomial family is always used).
model, y, x	logicals. If TRUE the corresponding components of the fit (model frame, response, model matrix) are returned.
penalty	penalty considered as one of <code>enet</code> , <code>mnet</code> , <code>snet</code> .
start	starting values for the parameters in the linear predictor.
nlambda	number of lambda value, default value is 100. The sequence may be truncated before nlambda is reached if a close to saturated model for the zero component is fitted.
lambda.count	A user supplied lambda.count sequence. Typical usage is to have the program compute its own lambda.count and lambda.zero sequence based on nlambda and lambda.min.ratio.
lambda.zero	A user supplied lambda.zero sequence.
lambda.count.min.ratio, lambda.zero.min.ratio	Smallest value for lambda.count and lambda.zero, respectively, as a fraction of lambda.max, the (data derived) entry value (i.e. the smallest value for which all coefficients are zero except the intercepts). Note, there is a closed formula for lambda.max for <code>penalty="enet"</code> . If <code>rescale=TRUE</code> , lambda.max is the same for <code>penalty="mnet"</code> or <code>"snet"</code> . Otherwise, some modifications are required. In the current implementation, for small gamma value, the square root of the computed lambda.zero[1] is used when <code>penalty="mnet"</code> or <code>"snet"</code> .
alpha.count	The elasticnet mixing parameter for the count part of model.
alpha.zero	The elasticnet mixing parameter for the zero part of model.
gamma.count	The tuning parameter of the snet or mnet penalty for the count part of model.
gamma.zero	The tuning parameter of the snet or mnet penalty for the zero part of model.
rescale	logical value, if TRUE, adaptive rescaling
init.theta	The initial value of theta for <code>family="negbin"</code> .
theta.fixed	Logical value only used for <code>family="negbin"</code> . If TRUE, theta is not updated.
EM	Using EM algorithm. Not implemented otherwise
convtype	convergence type, default is for count component only for speedy computation
maxit.em	Maximum number of EM algorithm

maxit	Maximum number of coordinate descent algorithm
maxit.theta	Maximum number of iterations for estimating theta scaling parameter if family="negbin". Default value maxit.theta may be increased, yet may slow the algorithm
eps.bino	a lower bound of probabilities to be claimed as zero, for computing weights and related values when family="binomial".
reltol	Convergence criteria, default value 1e-5 may be reduced to make more accurate yet slow
shortlist	logical value, if TRUE, limited results return
trace	If TRUE, progress of algorithm is reported
...	Other arguments which can be passed to from glmreg

Details

The algorithm fits penalized zero-inflated count data regression models using the coordinate descent algorithm within the EM algorithm. The returned fitted model object is of class "zipath" and is similar to fitted "glm" and "zeroinfl" objects. For elements such as "coefficients" a list is returned with elements for the zero and count component, respectively. For details see below.

A set of standard extractor functions for fitted model objects is available for objects of class "zipath", including methods to the generic functions [print](#), [coef](#), [logLik](#), [residuals](#), [predict](#). See [predict.zipath](#) for more details on all methods.

Value

An object of class "zipath", i.e., a list with components including

coefficients	a list with elements "count" and "zero" containing the coefficients from the respective models,
residuals	a vector of raw residuals (observed - fitted),
fitted.values	a vector of fitted means,
weights	the case weights used,
terms	a list with elements "count", "zero" and "full" containing the terms objects for the respective models,
theta	estimate of the additional θ parameter of the negative binomial model (if a negative binomial regression is used),
loglik	log-likelihood of the fitted model,
family	character string describing the count distribution used,
link	character string describing the link of the zero-inflation model,
linkinv	the inverse link function corresponding to link,
converged	logical value, TRUE indicating successful convergence of zipath, FALSE indicating otherwise
call	the original function call
formula	the original formula

levels	levels of the categorical regressors
contrasts	a list with elements "count" and "zero" containing the contrasts corresponding to levels from the respective models,
model	the full model frame (if model = TRUE),
y	the response count vector (if y = TRUE),
x	a list with elements "count" and "zero" containing the model matrices from the respective models (if x = TRUE),

Author(s)

Zhu Wang <zwang@connecticutchildrens.org>

References

Zhu Wang, Shuangge Ma, Michael Zappitelli, Chirag Parikh, Ching-Yun Wang and Prasad Devarajan (2014) *Penalized Count Data Regression with Application to Hospital Stay after Pediatric Cardiac Surgery*, *Statistical Methods in Medical Research*, in press

Zhu Wang, Shuangge Ma, Ching-Yun Wang, Michael Zappitelli, Prasad Devarajan and Chirag R. Parikh (2014) *EM for Regularized Zero Inflated Regression Models with Applications to Postoperative Morbidity after Cardiac Surgery in Children*, *Statistics in Medicine*, in press

Zhu Wang, Shuangge Ma and Ching-Yun Wang (2014) *Variable selection for zero-inflated and overdispersed data with application to health care demand in Germany*, manuscript

See Also

[glm](#), [glmreg](#), [glmregNB](#)

Examples

```
## Not run:
## data
data("bioChemists", package = "pscl")

## without inflation
## ("art ~ ." is "art ~ fem + mar + kid5 + phd + ment")
fm_pois <- glmreg(art ~ ., data = bioChemists, family = "poisson")
coef(fm_pois)
fm_nb <- glmregNB(art ~ ., data = bioChemists)
coef(fm_nb)
## with simple inflation (no regressors for zero component)
fm_zip <- zipath(art ~ . | 1, data = bioChemists, nlambda=10)
summary(fm_zip)
fm_zinb <- zipath(art ~ . | 1, data = bioChemists, family = "negbin", nlambda=10)
summary(fm_zinb)
## inflation with regressors
## ("art ~ . | ." is "art ~ fem + mar + kid5 + phd + ment | fem + mar + kid5 + phd + ment")
fm_zip2 <- zipath(art ~ . | ., data = bioChemists, nlambda=10)
summary(fm_zip2)
fm_zinb2 <- zipath(art ~ . | ., data = bioChemists, family = "negbin", nlambda=10)
```



```
summary(fm_zinb2)
### non-penalized regression, compare with zeroinfl
fm_zinb3 <- zipath(art ~ . | ., data = bioChemists, family = "negbin",
lambda.count=0, lambda.zero=0, reltol=1e-12)
summary(fm_zinb3)
fm_zinb4 <- zeroinfl(art ~ . | ., data = bioChemists, dist = "negbin")
summary(fm_zinb4)

## End(Not run)
```

Index

*Topic **methods**

methods, [24](#)

*Topic **models**

be.zeroinfl, [2](#)

cv.glmreg, [5](#)

cv.glmreg_fit, [9](#)

cv.glmregNB, [7](#)

cv.zipath, [10](#)

glmreg, [13](#)

glmreg_fit, [19](#)

glmregNB, [16](#)

plot.glmreg, [25](#)

predict.glmreg, [26](#)

pval.zipath, [29](#)

rzi, [30](#)

summary.glmregNB, [33](#)

tuning.zipath, [34](#)

vuong.test, [36](#)

*Topic **regression**

be.zeroinfl, [2](#)

breadReg, [3](#)

cv.glmreg, [5](#)

cv.glmreg_fit, [9](#)

cv.glmregNB, [7](#)

cv.zipath, [10](#)

estfunReg, [12](#)

glmreg, [13](#)

glmreg_fit, [19](#)

glmregNB, [16](#)

hessianReg, [22](#)

meatReg, [23](#)

plot.glmreg, [25](#)

predict.glmreg, [26](#)

predict.zipath, [27](#)

pval.zipath, [29](#)

rzi, [30](#)

sandwichReg, [31](#)

se, [32](#)

tuning.zipath, [34](#)

zipath, [37](#)

AIC.glmreg (methods), [24](#)

AIC.zipath (methods), [24](#)

be.zeroinfl, [2](#)

BIC.glmreg (methods), [24](#)

BIC.zipath (methods), [24](#)

breadReg, [3](#), [22](#), [23](#), [32](#)

coef, [3](#), [6](#), [8](#), [10–13](#), [15](#), [18](#), [28](#), [39](#)

coef.cv.glmreg (cv.glmreg), [5](#)

coef.cv.zipath (cv.zipath), [10](#)

coef.glmreg (predict.glmreg), [26](#)

coef.zipath (predict.zipath), [27](#)

conv2glmreg, [4](#)

conv2zipath, [4](#)

cv.glmreg, [5](#), [15](#)

cv.glmreg_fit, [9](#)

cv.glmregNB, [7](#), [18](#)

cv.zipath, [10](#)

deviance.glmreg (glmreg), [13](#)

estfunReg, [12](#), [23](#)

fitted, [28](#)

fitted.zipath (predict.zipath), [27](#)

glm, [40](#)

glm.nb, [34](#)

glmreg, [6](#), [10](#), [13](#), [21](#), [26](#), [40](#)

glmreg_fit, [19](#)

glmregNB, [8](#), [16](#), [40](#)

glmregNegbin (glmregNB), [16](#)

hessianReg, [22](#)

logLik, [39](#)

logLik.glmreg (glmreg), [13](#)

logLik.zipath (predict.zipath), [27](#)

meatReg, [4](#), [22](#), [23](#), [32](#)
methods, [24](#)
model.frame, [2](#), [5](#), [7](#), [11](#), [14](#), [16](#), [29](#), [34](#), [38](#)
model.matrix.zipath(predict.zipath), [27](#)

plot, [6](#), [8](#), [10](#), [12](#), [15](#), [18](#)
plot.cv.glmreg(cv.glmreg), [5](#)
plot.glmreg, [25](#)
predict, [6](#), [8](#), [10](#), [12](#), [15](#), [18](#), [28](#), [39](#)
predict.glmreg, [26](#)
predict.zipath, [27](#), [39](#)
predprob.zipath(predict.zipath), [27](#)
print, [15](#), [18](#), [28](#), [39](#)
print.summary.glmregNB
 (summary.glmregNB), [33](#)
print.summary.zipath(predict.zipath),
 [27](#)
print.zipath(zipath), [37](#)
pval.zipath, [29](#)

residuals, [28](#), [39](#)
residuals.zipath(predict.zipath), [27](#)
rzi, [30](#)

sandwichReg, [4](#), [23](#), [31](#)
se, [32](#)
summary, [28](#), [34](#)
summary.glmregNB, [33](#)
summary.zipath(predict.zipath), [27](#)

terms, [3](#), [13](#)
terms.zipath(predict.zipath), [27](#)
tuning.zipath, [34](#)

vuong.test, [36](#)

zipath, [12](#), [13](#), [27](#), [28](#), [33](#), [35](#), [37](#)