Package ‘lasso2’

Version 1.2-20
Date 2018-11-27
Author Justin Lokhorst, Bill Venables and Berwin Turlach;
  port to R, tests etc: Martin Maechler <maechler@stat.math.ethz.ch>
Maintainer Berwin Turlach <Berwin.Turlach@gmail.com>
Title L1 Constrained Estimation aka ‘lasso’
Description Routines and documentation for solving regression problems
  while imposing an L1 constraint on the estimates, based on
  the algorithm of Osborne et al. (1998).
Depends R (>= 3.2.3)
License GPL (>= 2)
NeedsCompilation yes
Repository CRAN
Date/Publication 2018-11-27 09:30:03 UTC

R topics documented:

  aux ................................................................. 2
  aux.l1celist ....................................................... 3
  coef.l1ce .......................................................... 3
  coef.l1celist ....................................................... 4
  deviance.gl1ce ..................................................... 4
  deviance.l1ce ....................................................... 5
  Extract.l1celist .................................................... 5
  fitted.l1ce ........................................................ 6
  gcv ................................................................. 6
  gcv.l1ce ............................................................ 7
  gl1ce ............................................................... 8
  gl1ce.object ....................................................... 10
  Iowa ............................................................... 11
  is.formula .......................................................... 12
aux

Extract Auxiliary Information From an Object

Description

Generic function for extracting auxiliary information from fitted model objects.

Usage

aux(object, ...)

Arguments

object fitted model object (here typically of class \texttt{l1celist}, see \texttt{aux.l1celist}).

... potentiilly further arguments passed to methods.

Details

See documentation (technical reports).

Value

a matrix with the bound(s) (relative [if used] and absolute) and the Lagrangian(s) for the fitted model(s).
Use `aux()` on a `l1celist` object

Description

This is a method for the function `aux()` for objects inheriting from class `l1celist`. See `aux` for the general behavior of this function and for the interpretation of `object`.

Usage

```r
## S3 method for class 'l1celist'
aux(object, ...)
```

Arguments

- `object`: fitted model object (here typically of class `l1celist`).
- `...`: potentially further arguments passed to methods.

Coefficients of an `l1ce` Object

Description

This is a method for `coef()` for objects inheriting from class `l1ce`. See `coef` for the general behavior of this function and for the interpretation of `object`.

Usage

```r
## S3 method for class 'l1ce'
coef(object, all=TRUE, constrained=FALSE, ...)
```

Arguments

- `object`: an object of class `l1ce`, see help on `l1ce.object`.
- `all`: logical; if false, then only the non-zero coefficients are returned.
- `constrained`: logical; if true, then only the coefficients that were constrained are returned.
- `...`: possibly further arguments (none at the moment).
**coef.\texttt{\texttt{l1celist}}** *Coefficients of an \texttt{\texttt{l1celist}} Object*

**Description**

This is a method for \texttt{coef()} for objects inheriting from class \texttt{l1celist}. See \texttt{coef} for the general behavior of this function and for the interpretation of \texttt{object}.

**Usage**

\begin{verbatim}
## S3 method for class \texttt{\texttt{l1celist}}
coef(object, all=TRUE, constrained=FALSE, ...)
\end{verbatim}

**Arguments**

- \texttt{object}: an object of class \texttt{l1celist}, see help on \texttt{l1celist.object}.
- \texttt{all}: logical; if false, then the coefficients that are zero in all fitted models of the list are not returned.
- \texttt{constrained}: logical; if true, then only the coefficients that were constrained are returned.
- \texttt{...}: possibly further arguments (none at the moment).

**deviance.\texttt{\texttt{gl1ce}}**  *Deviance Method for \texttt{\texttt{gl1ce}} Objects*

**Description**

This is a method of the generic function \texttt{deviance()} for objects inheriting from class \texttt{gl1ce} (see help(\texttt{gl1ce.object})).

**Usage**

\begin{verbatim}
## S3 method for class \texttt{\texttt{gl1ce}}
deviance(object, ...)\end{verbatim}

**Arguments**

- \texttt{object}: an object inheriting from class \texttt{gl1ce}.
- \texttt{...}: possibly further arguments (none at the moment).

**See Also**

\texttt{deviance} for the general behavior of this function and for the interpretation of \texttt{object}.
Deviance Method for ‘l1ce’ and ‘l1celist’ Objects

Description

These are methods of the generic function `deviance()` for objects inheriting from class `l1ce` or `l1celist` (see `help(l1ce.object)` and `help(l1celist.object)`).

Usage

```r
## S3 method for class 'l1ce'
deviance(object, ...)  
## S3 method for class 'l1celist'
deviance(object, ...)
```

Arguments

- `object`: an object inheriting from class `l1ce` or `l1celist`, respectively.
- `...`: possibly further arguments (none at the moment).

See Also

`deviance` for the general behavior of this function and for the interpretation of `object`.

Extract Parts of a ‘l1celist’ Object

Description

Allows the user to extract values from a `l1celist` object by using subscripts.

Usage

```r
## S3 method for class 'l1celist'
x[...], drop = TRUE]  
## S3 method for class 'l1celist'
x[...], drop = TRUE]]
```

Arguments

- `x`: an object inheriting from class "l1celist".
- `...`: a specification of indices – see `Extract`.
- `drop`: logical defaulting to `TRUE`. If only one model is subscribed, then it is returned as an object of class "l1ce". If `drop=F`, then an object of class "l1celist" is always returned.
Value

an object of class "l1celist" or class "l1ce" extracted from the original list.

---

### fitted.l1ce

*Fitted Values for 'l1ce', 'l1celist' and 'gl1ce' Objects*

**Description**

These are methods of the generic function `fitted()` for objects inheriting from class l1ce or l1celist (see help(l1ce.object) and help(l1celist.object)).

**Usage**

```r
## S3 method for class 'l1ce'
fitted(object, ...)
## S3 method for class 'l1celist'
fitted(object, ...)
```

**Arguments**

- `object` an object inheriting from class l1ce or l1celist, e.g., an gl1ce one (see gl1ce.object).
- `...` further potential arguments passed to methods.

**See Also**

`fitted` for the general behavior of this function and for the interpretation of object.

---

### gcv

*Generalised Cross-Validation Score*

**Description**

Extracts the generalised cross-validation score(s) from fitted model objects.

**Usage**

```r
gcv(object, ...)
```

**Arguments**

- `object` fitted model object; see gcv methods for details.
- `...` arguments passed to methods.
Details

See documentation.

Value

A vector (or matrix) with the bound(s) (relative [if used] and absolute), the Lagrangian(s) and the generalised cross-validation score(s) for the fitted model(s).

Description

This is a method for the function `gcv()` for objects inheriting from class `l1ce` or `l1celist`.

Usage

```r
## S3 method for class 'l1ce'
gcv(object, type = c("OPT", "Tibshirani"),
     gen.inverse.diag = 0, ...)
## S3 method for class 'l1celist'
gcv(object, type = c("OPT", "Tibshirani"),
     gen.inverse.diag = 0, ...)
```

Arguments

- `object`: an object of class `l1ce` or `l1celist`.
- `type`: character (string) indicating whether to use the covariance formula of Osborne, Presnell and Turlach or the formula of Tibshirani.
- `gen.inverse.diag`: if Tibshirani’s formula for the covariance matrix is used, this value is used for the diagonal elements of the generalised inverse that appears in the formula that corresponds to parameters estimated to be zero. The default is 0, i.e. use the Moore-Penrose inverse. Tibshirani’s code uses `gen.inverse.diag = 1e11`.
- `...`: further potential arguments passed to methods.

Details

See documentation.

See Also

`gcv` for the general behavior of this function; `l1ce.object` and `l1celist.object` for description of the object argument.
**Description**

Fit a generalized regression problem while imposing an L1 constraint on the parameters. Returns an object of class `gl1ce`.

**Usage**

```r
gl1ce(formula, data = sys.parent(), weights, subset, na.action, 
  family = gaussian, control = glm.control(...), sweep.out = ~ 1, 
  x = FALSE, y = TRUE, contrasts = NULL, standardize = TRUE, 
  guess.constrained.coefficients = double(p), bound = 0.5, ...) 
```

## S3 method for class 'gl1ce'

`family(object, ...)`

**Arguments**

- `formula`: a formula, with the response on the left hand side of a `~` operator, and the terms, separated by a `+` operator, on the right hand side.
- `data`: a `data.frame` in which to interpret the variables named in the formula, the weights, the subset and the sweep.out argument. If this is missing, then the variables in the formula should be globally available.
- `weights`: vector of observation weights. The length of weights must be the same as the number of observations. The weights must be strictly positive, since zero weights are ambiguous, compared to use of the subset argument.
- `subset`: expression saying which subset of the rows of the data should be used in the fit. This can be a logical vector (which is replicated to have length equal to the number of observations), or a numeric vector indicating which observation numbers are to be included, or a character vector of the row names to be included. All observations are included by default.
- `na.action`: a function to be applied to the model.frame after any subset argument has been used. The default (with `na.fail`) is to create an error if any missing values are found. A possible alternative is `na.omit`, which deletes observations that contain one or more missing values.
- `family`: a `family` object - a list of functions and expressions for defining the link and variance functions, initialization and iterative weights. Families supported are gaussian, binomial, poisson, Gamma, inverse.gaussian and quasi. Functions like binomial produce a family object, but can be given without the parentheses. Family functions can take arguments, as in `binomial(link=probit)`.
- `control`: a list of iteration and algorithmic constants. See `glm.control` for their names and default values. These can also be set as arguments to `gl1ce` itself.
sweep.out a formula object, variables whose parameters are not put under the constraint are swept out first. The variables should appear on the right of a ~ operator and be separated by + operators. Default is ~1, i.e. the constant term is not under the constraint. If this parameter is NULL, then all parameters are put under the constraint.

x logical flag: if TRUE, the model matrix is returned in component x.

y logical flag: if TRUE, the response is returned in component y.

contrasts a list giving contrasts for some or all of the factors appearing in the model formula. The elements of the list should have the same name as the variable and should be either a contrast matrix (specifically, any full-rank matrix with as many rows as there are levels in the factor), or else a function to compute such a matrix given the number of levels.

standardize logical flag: if TRUE, then the columns of the model matrix that correspond to parameters that are constrained are standardized to have empirical variance one. The standardization is done after taking possible weights into account and after sweeping out variables whose parameters are not constrained.

guess.constrained.coefficients initial guess for the parameters that are constrained.

bound numeric, either a single number or a vector: the constraint(s) that is/are put onto the L1 norm of the parameters.

... potential arguments for glm.control, as default for the control argument above.

object an R object of class "gl1ce".

Value

an object of class gl1ce is returned by gl1ce(). See gl1ce.object for details.

References

See the references in l1ce.


See Also

glm for unconstrained generalized regression modeling.

Examples

## example from base:
data(esoph)
summary(esoph)
## effects of alcohol, tobacco and interaction, age-adjusted
modEsO <- formula(cbind(ncases, ncontrols) ~ agegp + tobgp * alcgp)
glm.E <- glm(modEsO, data = esoph, family = binomial())
```r
# Initial code snippet

```r
glc.E <- gl1ce(modEso, data = esoph, family = binomial())
glc.E
plot(residuals(glc.E) ~ fitted(glc.E))

glc <- summary(glc.E)
glc

## Another comparison glm() / glc.E:
plot(predict(glm.E, type="link"), predict(glm.E, type="response"),
     xlim = c(-3,0))
points(predict(glm.E, type="link"), predict(glm.E, type="response"),
       col = 2, cex = 1.5)

labels(glc.E)#-- oops! empty!!
```

### gl1ce.object

**Generalized L1 Constrained Estimation Model Object**

---

**Description**

These are objects of class `gl1ce` They represent the fit of a generalized regression model under an L1 constraint on (some of) the parameters.

**Details**

The residuals, fitted values, coefficients, and effects should be extracted by the generic functions of the same name, rather than by the `$` operator.

**GENERATION**

This class of objects is returned from the `gl1ce` function to represent a fitted model.

**METHODS**

The `gl1ce` class of objects has methods for the following generic functions: deviance, predict, print, residuals, summary. Other generic functions are inherited from the class `l1ce`.

**STRUCTURE**

The following components must be included in a legitimate `gl1ce` object.

- `coefficients` the coefficients of the fit of the response to the columns of the model matrix. The names of the coefficients are the names of the columns of the model matrix.
- `residuals` the residuals from the fit. If weights were used, then the residuals are the raw residuals - the weights are not taken into account. If you need residuals that all have the same variance, then use the `residuals` function with `type="pearson"`.
- `fitted.values` the fitted values from the fit. If weights were used, the fitted values are not adjusted for the weights.
family the family of which the fitted regression model belongs, e.g., binomial(link=probit).
bound the (absolute) L1 constraint imposed on the parameters.
Lagrangian the value of the Lagrangian that enforces the constraint at the solution.
xtx the moment matrix of the variables that are under the constraint. (After taking weights, sweep-out variables and standardization into account).
xtr the product of the design matrix of the variables that are under the constraint (after taking weights, sweep-out variables and standardization into account) with the residual vector.
constrained.coefficients the coefficients on the scale on which they are constrained. Useful as initial value for further fits.
sweep.out information on the variables that are not under the constraint and on which the other variables and the response is projected first. Optional, not present if sweep.out = NULL.
assign the list of assignments of coefficients (and effects) to the terms in the model. The names of this list are the names of the terms. The i-th element of the list is the vector saying which coefficients correspond to the i-th term. It may be of length 0 if there were no estimable effects for the term. See also R.assign below.
terms an object of mode expression and class term summarizing the formula. Used by various methods, but typically not of direct relevance to users.
call an image of the call that produced the object, but with the arguments all named and with the actual formula included as the formula argument.
contrasts a list containing sufficient information to construct the contrasts used to fit any factors occurring in the model. The list contains entries that are either matrices or character vectors. When a factor is coded by contrasts, the corresponding contrast matrix is stored in this list. Factors that appear only as dummy variables and variables in the model that are matrices correspond to character vectors in the list. The character vector has the level names for a factor or the column labels for a matrix.
x optionally the model matrix, if x=T.
y optionally the response, if y=T.

See Also
glmce.coefficients.

---

**Iowa**

The Iowa Wheat Yield Data

**Description**

The data gives the pre-season and three growing months’ precipitation, the mean temperatures for the three growing months and harvest month, the year, and the yield of wheat for the USA state of Iowa, for the years 1930–1962.

**Usage**

data(Iowa)
Format

The data frame has the following components:

- Year: Year of measurement (surrogate for variety improvements)
- Rain0: Pre-season rainfall (in.)
- Temp1: Mean temperature for the first growing month (deg. F)
- Rain1: Rainfall for the first growing month (in.)
- Temp2: Mean temperature for the second growing month (deg. F)
- Rain2: Rainfall for the second growing month (in.)
- Temp3: Mean temperature for the third growing month (deg. F)
- Rain3: Rainfall for the third growing month (in.)
- Temp4: Mean temperature for the harvest month (deg. F)
- Yield: Yield of wheat in Iowa for the given year (bush./acre)

CATEGORY

Multiple regression; diagnostics.

Source


Examples

data(Iowa)
pairs(Iowa)

is.formula

Tests for Formula Objects

Description

is.formula returns TRUE if x is an object of class "formula", and FALSE otherwise.

Usage

is.formula(x)

Arguments

x an R object to be tested.
**Description**

Returns an object of class "l1ce" or "licelist" that represents fit(s) of linear models while imposing L1 constraint(s) on the parameters.

**Usage**

```r
l1ce(formula, data = sys.parent(), weights, subset, na.action,
      sweep.out = ~ 1, x = FALSE, y = FALSE,
      contrasts = NULL, standardize = TRUE,
      trace = FALSE, guess.constrained.coefficients = double(p),
      bound = 0.5, absolute.t = FALSE)
```

**Arguments**

- `formula`: a formula object, with the response on the left of a `~` operator, and the terms, separated by `+` operators, on the right.
- `data`: a `data.frame` in which to interpret the variables named in the formula, the weights, the subset and the sweep.out argument. If this is missing, then the variables in the formula should be globally available.
- `weights`: vector of observation weights. The length of `weights` must be the same as the number of observations. The weights must be nonnegative and it is strongly recommended that they be strictly positive, since zero weights are ambiguous, compared to use of the subset argument.
- `subset`: expression saying which subset of the rows of the data should be used in the fit. This can be a logical vector (which is replicated to have length equal to the number of observations), or a numeric vector indicating which observation numbers are to be included, or a character vector of the row names to be included. All observations are included by default.
- `na.action`: a function to filter missing data. This is applied to the `model.frame` after any subset argument has been used. The default (with `na.fail`) is to create an error if any missing values are found. A possible alternative is `na.omit`, which deletes observations that contain one or more missing values.
- `sweep.out`: a formula object, variables whose parameters are not put under the constraint are swept out first. The variables should appear on the right of a `~` operator and be separated by `+` operators. Default is `~ 1`, i.e. the constant term is not under the constraint. If this parameter is `NULL`, then all parameters are put under the constraint.
- `x`: logical indicating if the model matrix should be returned in component `x`.
- `y`: logical indicating if the response should be returned in component `y`. 
contrasts

a list giving contrasts for some or all of the factors appearing in the model formula. The elements of the list should have the same name as the variable and should be either a contrast matrix (specifically, any full-rank matrix with as many rows as there are levels in the factor), or else a function to compute such a matrix given the number of levels.

standardize

logical flag: if TRUE, then the columns of the model matrix that correspond to parameters that are constrained are standardized to have empirical variance one. The standardization is done after taking possible weights into account and after sweeping out variables whose parameters are not constrained; see vignette for details.

trace

logical flag: if TRUE, then the status during each iteration of the fitting is reported.

guess.constrained.coefficients

initial guess for the parameters that are constrained.

bound

numeric, either a single number or a vector: the constraint(s) that is/are put onto the L1 norm of the parameters.

absolute.t

logical flag: if TRUE, then bound is an absolute bound and all entries in bound can be any positive number. If FALSE, then bound is a relative bound and all entries must be between 0 and 1; see vignette for details.

Value

an object of class l1ce (if bound was a single value) or l1celist (if bound was a vector of values) is returned. See l1ce.object and l1celist.object for details.

References


Examples

data(Iowa)
l1c.I <- l1ce(Yield ~ ., Iowa, bound = 10, absolute.t=TRUE)
l1c.I

## The same, printing information in each step:
l1ce(Yield ~ ., Iowa, bound = 10, trace = TRUE, absolute.t=TRUE)

data(Prostate)
l1c.P <- l1ce(lpsa ~ ., Prostate, bound=(1:30)/30)
length(l1c.P)# 30 l1ce models
l1c.P # -- MM: too large; should do this in summary(.)!

plot(resid(l1c.I) ~ fitted(l1c.I))
Description

These are objects of class "l1ce". They represent the fit of a regression model under an L1 constraint on (some of) the parameters.

Details

The residuals, fitted values, coefficients, and effects should be extracted by the generic functions of the same name, rather than by the $ operator.

GENERATION

This class of objects is returned from the l1ce function to represent a fitted model.

METHODS

The "l1ce" class of objects has methods for the following generic functions:
coef, deviance, fitted, formula, gcv, labels, predict, print, residuals, summary, vcov.

STRUCTURE

The following components must be included in a legitimate l1ce object.

- **coefficients** the coefficients of the fit of the response to the columns of the model matrix. The names of the coefficients are the names of the columns of the model matrix.
- **residuals** the residuals from the fit. If weights were used, then the residuals are the raw residuals - the weights are not taken into account. If you need residuals that all have the same variance, then use the residuals function with type="pearson".
- **fitted.values** the fitted values from the fit. If weights were used, the fitted values are not adjusted for the weights.
- **bound** the (absolute) L1 constraint imposed on the parameters.
- **relative.bound** optional, the (relative) L1 constraint imposed on the parameters. Present if absolute = F.
- **Lagrangian** the value of the Lagrangian that enforces the constraint at the solution.
- **xtx** the moment matrix of the variables that are under the constraint. (After taking weights, sweep-out variables and standardization into account).
- **xtr** the product of the design matrix of the variables that are under the constraint (after taking weights, sweep-out variables and standardization into account) with the residual vector.
- **constrained.coefficients** the coefficients on the scale on which they are constrained. Useful as initial value for further fits.
sweep.out  Optional information on the variables that are not under the constraint and on which the other variables and the response is projected first. Optional, not present if sweep.out = NULL.

assign  the list of assignments of coefficients (and effects) to the terms in the model. The names of this list are the names of the terms. The i\textsuperscript{th} element of the list is the vector saying which coefficients correspond to the i\textsuperscript{th} term. It may be of length 0 if there were no estimable effects for the term. See also R.assign below.

terms  an object of mode expression and class term summarizing the formula. Used by various methods, but typically not of direct relevance to users.
call  an image of the call that produced the object, but with the arguments all named and with the actual formula included as the formula argument.

contrasts  a list containing sufficient information to construct the contrasts used to fit any factors occurring in the model. The list contains entries that are either matrices or character vectors. When a factor is coded by contrasts, the corresponding contrast matrix is stored in this list. Factors that appear only as dummy variables and variables in the model that are matrices correspond to character vectors in the list. The character vector has the level names for a factor or the column labels for a matrix.

x  optionally the model matrix, if x=TRUE.

y  optionally the response, if y=TRUE.

See Also

l1ce, coefficients.
STRUCTURE

An object of class \texttt{l1celist} is a list of lists. Each component of the list is a list with the information of an object of class \texttt{l1ce} that is unique for that information. All shared information is stored as attributes.

Each component of the list must include the following components for it to be a legitimate \texttt{l1celist} object.

- **coefficients** the coefficients of the fit of the response to the columns of the model matrix. The names of the coefficients are the names of the columns of the model matrix.
- **residuals** the residuals from the fit. If weights were used, then the residuals are the raw residuals - the weights are not taken into account. If you need residuals that all have the same variance, then use the \texttt{residuals} function with \texttt{type=\texttt{"pearson"}}.
- **fitted.values** the fitted values from the fit. If weights were used, the fitted values are not adjusted for the weights.
- **bound** the (absolute) L1 constraint imposed on the parameters.
- **relative.bound** optional, the (relative) L1 constraint imposed on the parameters. Present if \texttt{absolute.t=\texttt{F}}.
- **Lagrangian** the value of the Lagrangian that enforces the constraint at the solution.
- **xtr** the product of the design matrix of the variables that are under the constraint (after taking weights, sweep-out variables and standardization into account) with the residual vector.
- **constrained.coefficients** the coefficients on the scale on which they are constrained. Useful as initial value for further fits.

See Also

\texttt{l1ce, coefficients}.

### Description

This is a method for the \texttt{labels()} function for objects inheriting from class \texttt{l1ce} or \texttt{l1celist} (see \texttt{help(l1ce.object)} and \texttt{help(l1celist.object)}). See \texttt{labels} or \texttt{labels.default} for the general behavior of this function and for the interpretation of object.

### Usage

```r
## S3 method for class 'l1ce'
labels(object, ...)

## S3 method for class 'l1celist'
labels(object, ...)
```

### Arguments

- **object** fitted model of class \texttt{"l1ce"} or \texttt{"l1celist"}, respectively.
- **...** potentially further arguments passed to method.
merge.formula  

Merge Formula With Right Hand Side of Second Formula

Description

This is method for formulas of the `merge` generic function. Here it is support for the function `l1ce` and not intended to be called directly by users.

Usage

```r
## S3 method for class 'formula'
merge(x, y, ...)  
```

Arguments

- `x, y` formulas.
- `...` potentially further arguments passed to methods.

Examples

```r
merge(y ~ x1, ~ x2) ## -> y ~ x1 + x2
f2 <- merge(y ~ x1*x2, z ~ (x2+x4)^3)
f. <- merge(y ~ x1*x2, ~ (x2+x4)^3) # no LHS for 2nd term
f2
stopifnot(f2 == f.)
```

plot.l1celist  

Plot Method for `l1celist` Objects

Description

Plots a `l1celist` object on the current graphics device.

Usage

```r
## S3 method for class 'l1celist'
plot(x, plot=TRUE, all=TRUE, constrained=FALSE,
     type = "b", xlab = "bounds", ylab = "coeff | bounds", ...)
```
Arguments

x fitted model object of class l1celist.
plot logical; if TRUE a matplot() of all the coefficients in the list against the relative bound (absolute bound if relative is not present) is plotted. Otherwise no plot is done.
all logical; if FALSE, then only the non-zero coefficients are returned.
constrained if TRUE then only the coefficients that were constrained are returned.
type, xlab, ylab, ...
  further arguments with useful defaults passed to matplot.

Value

A matrix with the bound(s) (relative [if used] and absolute), the Lagrangian(s) and coefficients of the fitted model(s).

Examples

data(Prostate)
l1c.P <- l1ce(lpsa ~ ., Prostate, bound=(1:20)/20)
length(l1c.P)# 20 l1ce models
plot(l1c.P)

predict.gl1ce

Prediction Method for a ‘gl1ce’ Object

Description

This is a method for the generic function predict for class "gl1ce", typically produced from gl1ce(). When newdata is missing, the fitted values are extracted, otherwise returns new predictions.

Usage

## S3 method for class 'gl1ce'
predict(object, newdata, type=c("link", "response"),
  se.fit = FALSE, ...)

Arguments

object a fitted gl1ce object.
newdata a data frame containing the values at which predictions are required. This argument can be missing, in which case predictions are made at the same values used to compute the object. Only those predictors referred to in the right side of the formula in object need be present by name in newdata.
predict.l1ce

Description

This is a method for the generic function `predict` for class "l1ce", typically produced from `l1ce()`. When newdata is missing, the fitted values are extracted, otherwise returns new predictions.

Usage

```r
## S3 method for class 'l1ce'
predict(object, newdata,
        type = c("response"), se.fit = FALSE, ...)
```
Arguments

- **object**: a fitted l1ce object.
- **newdata**: a data frame containing the values at which predictions are required. This argument can be missing, in which case predictions are made at the same values used to compute the object. Only those predictors referred to in the right side of the formula in object need be present by name in newdata.
- **type**: currently only "response".
- **se.fit**: logical indicating if standard errors should be returned as well. **Not yet available**.
- **...**: further potential arguments passed to methods.

Value

a vector of predictions.

Warning

predict can produce incorrect predictions when the newdata argument is used if the formula in object involves transformations, such as poly(Age, 3) or sqrt(Age - min(Age)).

MM: Not sure this is true for R ....

Examples

data(Iowa)
l1c.I <- l1ce(Yield ~ ., Iowa, bound = 10, absolute.t=TRUE)
p10 <- predict(l1c.I, newdata = Iowa[10:19,])
stopifnot(all.equal(p10, fitted(l1c.I)[10:19]))
Arguments

x     fitted model of class "l1ce" or "l1celist", respectively.
...   potentially further arguments passed to method.

Prostate Cancer Data

Description

These data come from a study that examined the correlation between the level of prostate specific antigen and a number of clinical measures in men who were about to receive a radical prostatectomy. It is data frame with 97 rows and 9 columns.

Usage

data(Prostate)

Format

The data frame has the following components:

lcavol  log(cancer volume)
lweight log(prostate weight)
age    age
lbph   log(benign prostatic hyperplasia amount)
svi    seminal vesicle invasion
lcp    log(capsular penetration)
gleason Gleason score
pgg45  percentage Gleason scores 4 or 5
lpsa   log(prostate specific antigen)

Source

qr.rtr.inv

Examples

data(Prostate)
attach(Prostate)
pairs(Prostate, col = 1:svi, pch = gleason - 5,
       main = paste("Prostate data, n = ", nrow(Prostate)))
detach()

l1c.P <- l1ce(lcavol ~ ., data = Prostate)
coef(l1c.P)[coef(l1c.P) != 0] # only age, lcp, lpsa (+ intercept)
summary(l1c.P)

---

qr.rtr.inv

Reconstruct the Inverse of \( R' \mathbf{R} \) from a QR Object

Description

From a QR object, compute the inverse matrix which is implicitly (but not explicitly!) used to solve the underlying least squares problem.

Usage

\[
\text{qr.rtr.inv}(\mathbf{qr})
\]

Arguments

\( \mathbf{qr} \) \"\text{"qr\" object, typically resulting from \texttt{qr(\ldots).\texttt{\ldots}}.

Value

The \( p \times p \) matrix \((R'R)^{-1}\) or equivalently, the inverse of \( X'X \) (i.e. \texttt{t(X)} %*% \texttt{X} in \texttt{R}).

See Also

\texttt{qr, qr.R, backsolve}.

Examples

\[
\begin{align*}
(\mathbf{h3} & \leftarrow \text{1/outer(0:5, 1:3, "+")}) \\
\mathbf{rtr} & \leftarrow \text{qr.rtr.inv}(%(\mathbf{h3})) \\
\text{all.equal(c(rtr %*% 1:3), solve(crossprod(h3), 1:3))}
\end{align*}
\]
residuals.gl1ce  

Compute Residuals for 'gl1ce' Objects

Description

Computes one of the four types of residuals available for gl1ce objects. This is a method for the function residuals() for objects inheriting from class gl1ce. As several types of residuals are available for gl1ce objects, there is an additional optional argument type.

Usage

## S3 method for class 'gl1ce'
residuals(object,  
type = c("deviance", "pearson", "working", "response"), ...

Arguments

object an object inheriting from class gl1ce representing a fitted model.

type type of residuals, with choices "deviance", "pearson", "working" or "response"; the first is the default.

... possibly further arguments (none at the moment).

Value

A numeric vector of residuals. See Statistical Models in S for detailed definitions of each type of residual. The sum of squared deviance residuals add up to the deviance. The pearson residuals are standardized residuals on the scale of the response. The working residuals reside on the object, and are the residuals from the final fit. The response residuals are simply \( y - \text{fitted(object)} \). The summary() method for gl1ce objects produces deviance residuals. The residuals component of a gl1ce object contains the working residuals.

References


See Also

gl1ce for examples; fitted.gl1ce is used for fitted values.
residuals.l1ce

Residuals of ‘l1ce’ or ‘l1celist’ Objects

Description

This is a method for the function residuals for objects inheriting from class l1ce or l1celist (see help(l1ce.object) and help(l1celist.object)). See residuals for the general behavior of this function and for the interpretation of object and type.

Usage

## S3 method for class 'l1ce'
residuals(object, type, ...)

## S3 method for class 'l1celist'
residuals(object, type, ...)

Arguments

- **object**: an object inheriting from class l1ce representing a fitted model.
- **type**: type of residuals, with choices "deviance", "pearson", "working" or "response"; the first is the default.
- **...**: possibly further arguments (none at the moment).

summary.gl1ce

Summary Method for Generalized L1 Constrained Regression Models

Description

Returns (and prints) a summary list for fitted generalized L1 constrained regression models.

Usage

## S3 method for class 'gl1ce'
summary(object, dispersion = NULL, correlation = FALSE, ...)

## S3 method for class 'summary.gl1ce'
print(x, digits = max(3,getOption("digits") - 3), ...)

Arguments

- **object**: fitted gl1ce model object. This is assumed to be the result of some fit that produces an object inheriting from the class gl1ce, in the sense that the components returned by the gl1ce() function will be available.
- **dispersion**: prescribed dispersion, see summary.glm.
- **correlation**: logical indicating if the correlation matrix should be returned as well.
- **x**: object of class summary.gl1ce.
- **digits, ...**: further arguments for the print method.
Details

This function is a method for the generic function summary for class gl1ce. It can be invoked by calling summary for an object of the appropriate class, or directly by calling summary.gl1ce regardless of the class of the object.

Value

an object of class summary.gl1ce (for which there's a print method). It is basically a list with the following components:

- `call` as contained on object
- `residuals` the deviance residuals, as produced by residuals(object, type = "deviance").
- `coefficients` the coefficients of the model.
- `family` the family of models to which object belongs, along with the variance and link functions for that model.
- `bound` the bound used in fitting this model
- `Lagrangian` the Lagrangian of the model

summary.gl1ce  
Summary Method for “gl1ce” Objects (Regression with L1 Constraint)

Description

Returns a summary list for a regression model with an L1 constraint on the parameters. A null value will be returned if printing is invoked.

Usage

```r
## S3 method for class 'l1ce'
summary(object, correlation = TRUE,
        type = c("OPT", "Tibshirani"),
        gen.inverse.diag = 0, sigma = NULL, ...)
## S3 method for class 'summary.l1ce'
print(x, digits = max(3,getOption("digits") - 3), ...)
```

Arguments

- `object` fitted model of class "l1ce".
- `correlation` logical indicating if the correlation matrix for the coefficients should be included in the summary.
- `type` character string specifying whether to use the covariance formula of Osborne, Presnell and Turlach or the formula of Tibshirani.
gen.inverse.diag
if Tibshirani's formula for the covariance matrix is used, this value is used for
the diagonal elements of the generalised inverse that appears in the formula that
corresponds to parameters estimated to be zero. The default is 0, i.e. use the
Moore-Penrose inverse. Tibshirani's code uses gen.inverse.diag=1e11.
sigma
the residual standard error estimate. If not provided, then it is estimated by the
deviance of the model divided by the error degrees of freedom.
x
an R object of class summary.11ce.
digits
number of significant digits to use.
...
further potential arguments passed to methods.

Details
This function is a method for the generic function summary() for class "11ce". It can be invoked by
calling summary(x) for an object x of the appropriate class, or directly by calling summary.11ce(x)
regardless of the class of the object.

Value
an object of class summary.11ce (for which there's a print method). It is basically a list with the
following components:
correlation the computed correlation coefficient matrix for the coefficients in the model.
cov.unscaled the unscaled covariance matrix; i.e, a matrix such that multiplying it by an es-
timate of the error variance produces an estimated covariance matrix for the
coefficients.
df the number of degrees of freedom for the model and for residuals.
coefficients a matrix with three columns, containing the coefficients, their standard errors
and the corresponding t statistic.
residuals the model residuals. These are the weighted residuals if weights were given in
the model.
sigma the residual standard error estimate.
terms the terms object used in fitting this model.
call the call object used in fitting this model.
bound the bound used in fitting this model.
relative.bound the relative bound used in fitting this model (may not be present).
Lagrangian the Lagrangian of the model.

See Also
11ce, 11ce.object, summary.
Examples

data(Prostate)
summary(l1ce(lpsa ~ ., Prostate))

# Produces the following output:
## Not run:
Call:
  l1ce(formula = lpsa ~ ., data = Prostate)
Residuals:
  Min  1Q Median  3Q Max
 -1.636 -0.4119  0.076  0.452  1.83

Coefficients:

|                | Value | Std. Error | Z value | Pr(>|z|) |
|----------------|-------|------------|---------|----------|
| (Intercept)    | 0.7285| 1.3898     | 0.5242  | 0.6002   |
| lcavol         | 0.4937| 0.0919     | 5.3711  | 0.0000   |
| lweight        | 0.2682| 0.1774     | 1.5115  | 0.1307   |
| age            | 0.0000| 0.0111     | 0.0000  | 1.0000   |
| lbph           | 0.0093| 0.0587     | 0.1581  | 0.8744   |
| svi            | 0.4551| 0.2525     | 1.8023  | 0.0715   |
| lcp            | 0.0000| 0.0947     | 0.0000  | 1.0000   |
| gleason        | 0.0000| 0.1685     | 0.0000  | 1.0000   |
| pgg45          | 0.0002| 0.0046     | 0.0391  | 0.9688   |

Residual standard error: 0.7595 on 88.36 degrees of freedom
The relative L1 bound was : 0.5
The absolute L1 bound was : 0.9219925
The Lagrangian for the bound is: 13.05806

Correlation of Coefficients:

<table>
<thead>
<tr>
<th></th>
<th>(Intercept)</th>
<th>lcavol</th>
<th>lweight</th>
<th>age</th>
<th>lbph</th>
<th>svi</th>
<th>lcp</th>
<th>gleason</th>
<th>pgg45</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>1.0000</td>
<td>-0.4815</td>
<td>-0.2071</td>
<td>-0.3938</td>
<td>-0.0603</td>
<td>-0.0974</td>
<td>-0.0624</td>
<td>-0.2273</td>
<td>-0.2273</td>
</tr>
<tr>
<td>lcavol</td>
<td>-0.4815</td>
<td>1.0000</td>
<td>0.01988</td>
<td>0.3629</td>
<td>0.0201</td>
<td>0.5165</td>
<td>0.1303</td>
<td>0.2273</td>
<td>0.2273</td>
</tr>
<tr>
<td>lweight</td>
<td>-0.2071</td>
<td>0.01988</td>
<td>1.0000</td>
<td>-0.3938</td>
<td>-0.0603</td>
<td>-0.0974</td>
<td>-0.0624</td>
<td>-0.2273</td>
<td>-0.2273</td>
</tr>
<tr>
<td>age</td>
<td>-0.3938</td>
<td>0.3629</td>
<td>-0.3938</td>
<td>1.0000</td>
<td>0.01988</td>
<td>0.3629</td>
<td>0.0201</td>
<td>0.5165</td>
<td>0.1303</td>
</tr>
<tr>
<td>lbph</td>
<td>-0.0603</td>
<td>-0.0201</td>
<td>-0.0603</td>
<td>0.01988</td>
<td>1.0000</td>
<td>0.3629</td>
<td>0.0201</td>
<td>0.5165</td>
<td>0.1303</td>
</tr>
<tr>
<td>svi</td>
<td>-0.0974</td>
<td>0.5165</td>
<td>0.3629</td>
<td>0.3629</td>
<td>0.01988</td>
<td>1.0000</td>
<td>0.01988</td>
<td>0.3629</td>
<td>0.0201</td>
</tr>
<tr>
<td>lcp</td>
<td>-0.0624</td>
<td>0.1303</td>
<td>0.2273</td>
<td>0.0201</td>
<td>0.3629</td>
<td>0.3629</td>
<td>1.0000</td>
<td>0.01988</td>
<td>0.3629</td>
</tr>
<tr>
<td>gleason</td>
<td>-0.2273</td>
<td>0.1303</td>
<td>0.2273</td>
<td>0.2273</td>
<td>0.3629</td>
<td>0.3629</td>
<td>0.01988</td>
<td>1.0000</td>
<td>0.3629</td>
</tr>
<tr>
<td>pgg45</td>
<td>-0.2273</td>
<td>0.2273</td>
<td>0.2273</td>
<td>0.2273</td>
<td>0.3629</td>
<td>0.3629</td>
<td>0.3629</td>
<td>0.01988</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

## End(Not run)
vcov.l1ce

Description

Calculates the trace of a matrix

Usage

\texttt{tr(mat)}

Arguments

\texttt{mat} \\
\hspace{1em} \text{a square matrix.}

Value

The trace of the matrix, i.e. the sum of its diagonal elements, is returned.

Examples

\texttt{tr(cbind(1:1:3,4:2)) \# 5}

---

vcov.l1ce \hspace{1em} \textit{Variance-Covariance Matrix of ‘l1ce’ or ‘l1celist’ Objects}

Description

This is a method for the function \texttt{vcov()} for objects inheriting from class \texttt{l1ce} or \texttt{l1celist} (see help(\texttt{l1ce.object}) and help(\texttt{l1celist.object})). See \texttt{vcov} for the general behavior of this function.

Usage

\begin{verbatim}
## S3 method for class ‘l1ce’
vcov(object, type = c("OPT", "Tibshirani"),
gen.inverse.diag = 0, ...)
## S3 method for class ‘l1celist’
vcov(object, type = c("OPT", "Tibshirani"),
gen.inverse.diag = 0, ...)
\end{verbatim}

Arguments

\begin{description}
\item[\texttt{object}] \hspace{1em} an object of class \texttt{l1ce} or \texttt{l1celist}.
\item[\texttt{type}] \hspace{1em} character indicating whether to use the covariance formula of Osborne, Presnell and Turlach or the formula of Tibshirani.
\item[\texttt{gen.inverse.diag}] \hspace{1em} if Tibshirani’s formula for the covariance matrix is used, this value is used for the diagonal elements of the generalised inverse that appears in the formula that corresponds to parameters estimated to be zero. The default is 0, i.e. use the Moore-Penrose inverse. Tibshirani’s code uses \texttt{gen.inverse.diag = 1e11}.
\end{description}

… \hspace{1em} further potential arguments passed to methods.
Index

*Topic algebra
  qr.rtr.inv, 23
*Topic array
  qr.rtr.inv, 23
*Topic classes
  gl1ce.object, 10
  l1ce.object, 15
  l1celist.object, 16
*Topic datasets
  Iowa, 11
  Prostate, 22
*Topic hplot
  plot.l1celist, 18
*Topic manip
  Extract.l1celist, 5
*Topic math
  tr, 28
*Topic methods
  gl1ce.object, 10
  l1ce.object, 15
  l1celist.object, 16
*Topic models
  coef.l1ce, 3
  coef.l1celist, 4
  deviance.gl1ce, 4
  deviance.l1ce, 5
  fitted.l1ce, 6
  gcv, 6
  gcv.l1ce, 7
  gl1ce, 8
  l1ce, 13
  predict.gl1ce, 19
  predict.l1ce, 20
  residuals.gl1ce, 24
  residuals.l1ce, 25
  summary.gl1ce, 25
  vcov.l1ce, 29
*Topic optimize
  gl1ce, 8

  l1ce, 13
*Topic print
  print.l1ce, 21
*Topic regression
  coef.l1ce, 3
  coef.l1celist, 4
  gl1ce, 8
  gl1ce.object, 10
  l1ce, 13
  l1ce.object, 15
  l1celist.object, 16
  summary.l1ce, 26
*Topic utilities
  aux, 2
  aux.l1celist, 3
  is.formula, 12
  labels.l1ce, 17
  merge.formula, 18
  [.l1celist (Extract.l1celist), 5
  [[.l1celist (Extract.l1celist), 5
    aux, 2, 3
    aux.l1celist, 2, 3
  backsolve, 23
  binomial, 11
  coef, 3, 4
  coef.l1ce, 3
  coef.l1celist, 4
  coefficients, 11, 16, 17
  data.frame, 8
  deviance, 4, 5
  deviance.gl1ce, 4
  deviance.l1ce, 5
  deviance.l1celist (deviance.l1ce), 5

  Extract, 5
  Extract.l1celist, 5
INDEX

family, 8
family.gl1ce (gl1ce), 8
fitted, 6
fitted.l1ce, 6, 24
fitted.l1celist (fitted.l1ce), 6
formula, 8
gcv, 6, 7
gcv.l1ce, 7
gcv.l1celist (gcv.l1ce), 7
gl1ce, 8, 11, 19, 24
gl1ce.object, 4, 6, 9, 10, 21
glm, 9
glm.control, 9

Iowa, 11
is.formula, 12

l1ce, 9, 13, 15–18, 20, 27
l1ce.object, 3, 5–7, 14, 15, 17, 21, 25, 27, 29
l1celist.object, 4–7, 14, 16, 17, 21, 25, 29
labels, 17
labels.default, 17
labels.l1ce, 17
labels.l1celist (labels.l1ce), 17

matplot, 19
merge, 18
merge.formula, 18
model.frame, 13

plot.l1celist, 18
predict, 19, 20
predict.gl1ce, 19
predict.l1ce, 20
print, 21
print.default, 21
print.gl1ce (print.l1ce), 21
print.l1ce, 21
print.l1celist (print.l1ce), 21
print.summary.gl1ce (summary.gl1ce), 25
print.summary.l1ce (summary.l1ce), 26
Prostate, 22

qr, 23
qr.R, 23
qr.rtr.inv, 23

residuals, 25
residuals.gl1ce, 24

summary, 27
summary.gl1ce, 25
summary.glm, 25
summary.l1ce, 26

tr, 28

vcov, 29
vcov.l1ce, 29
vcov.l1celist (vcov.l1ce), 29

vcov.l1ce, 25
residuals.l1ce, 25
residuals.l1celist (residuals.l1ce), 25

vcov, 29
vcov.l1ce, 29
vcov.l1celist (vcov.l1ce), 29

tr, 28

vcov, 29