Package ‘quantregGrowth’

February 20, 2015

Type Package
Title Growth charts via regression quantiles
Version 0.3-0
Date 2014-07-25
Maintainer Vito M. R. Muggeo <vito.muggeo@unipa.it>
Description Fits non-crossing regression quantiles as a function of linear covariates and a smooth terms via B-splines with quadratic penalties.
Depends quantreg, splines
License GPL
Author Vito M. R. Muggeo [aut, cre]
NeedsCompilation no
Repository CRAN
Date/Publication 2014-07-25 15:45:18

R topics documented:

quantregGrowth-package ........................................... 2
gcrq ................................................................. 3
growthData .......................................................... 6
ncross.rq.fitXB ...................................................... 7
plot.gcrq ............................................................. 8
predictQR ............................................................. 10
print.gcrq ........................................................... 11
ps ................................................................. 11
summary.gcrq ....................................................... 13
vcov.gcrq ........................................................... 14

Index 15
**quantregGrowth-package**

*Growth charts via regression quantiles*

**Description**

Fits noncrossing regression quantiles as a function of linear covariates and smooth terms via B-splines with quadratic penalties.

**Details**

Package: quantregGrowth  
Type: Package  
Version: 0.3-0  
Date: 2014-07-25  
License: GPL

Package quantregGrowth allows estimation of growth charts via quantile regression. Given a set of percentiles, gcrq estimates non-crossing quantile curves as a flexible function of a quantitative covariate (typically age), and possibly additional linear terms. To ensure flexibility, B-splines with a quadratic penalty are employed to estimate nonparametrically the curves; additionally monotonicity constraints may be also set. plot.gcrq displays the fitted lines.

**Author(s)**

Vito M.R. Muggeo  
Maintainer: Vito M.R. Muggeo <vito.muggeo@unipa.it>

**References**


Some references on growth charts (the first two papers employ the so-called LMS method)  


Some references on regression quantiles  

See Also

`gcrq, rq` in package quantreg

Examples

#see ?gcrq for some examples

---

`gcrq`  

**Growth charts regression quantiles**

Description

Estimation of nonparametric growth charts via quantile regression. Quantile curves are estimated via B-splines with a quadratic penalty on the spline coefficient differences, and non-crossing and monotonicity restrictions are set to obtain estimates more biologically plausible. Linear terms are allowed in the model specification.

Usage

```
gcrq(formula, tau = c(0.1, 0.25, 0.5, 0.75, 0.9), data, subset, weights, 
    na.action, transf=NULL, y = TRUE, interc=FALSE, 
    foldid = NULL, nfolds = 10, cv = FALSE, n.boot=0, eps=.0001, ...)
```

Arguments

- `formula`: a standard R formula to specify the response in the left hand side, and the co-variates in the right hand side. See Details.
- `tau`: a numeric vector to specify the percentiles of interest. Default to (.1, .25, .5, .75, .9).
- `data`: the dataframe where the variables required by the formula, subset and weights arguments are stored.
- `subset`: optional. A vector specifying a subset of observations to be used in the fitting process.
- `weights`: optional. A numeric vector specifying weights to be assigned to the observations in the fitting process. Currently unimplemented.
- `na.action`: a function which indicates how the possible 'NA's are handled.
- `transf`: an optional character string (with "y" as argument) meaning a function to apply to the response variable before fitting. E.g. "log(y+0.1)".
- `y`: logical. If TRUE (default) the returned object includes also the responses vector.
interc logical. If formula includes a "ps" term, interc=TRUE means that a model intercept is also estimated. If this is the case, a very small ridge penalty is exploited to allow estimation with a design matrix containing both a full B-spline basis and a column of ones. interc=TRUE overwrites the intercept specification in the formula (e.g., ~0+...), and it is ignored if the model does not include a "ps" term.

foldid optional. A numeric vector identifying the group labels to perform cross validation to select the smoothing parameter. Ignored if the lambda argument in ps() is not a vector.

nfolds optional. If foldid is not provided, it is scalar specifying the number of ‘folds’ (groups) which should be used to perform cross validation to select the smoothing parameter. Default to 10, but it is ignored if the lambda argument in ps() is not a vector.

cv logical. If TRUE, the returned object includes also the matrix cv having number of rows equal to length of lambda and number of columns equal to nfolds. Ignored if the lambda argument in ps() is not a vector.

n.boot Number of nonparametric (cases resampling) bootstrap samples to be used. Notice that the smoothing parameter (if relevant) does change throughout the bootstrap replicates.

eps A small positive constant to ensure noncrossing curves. Use it at your risk! If eps is large, the resulting fitted quantile curves could appear unreasonable.

Details

The function fits regression quantiles at specified percentiles given in tau as a function of covariates specified in the formula argument. The formula may optionally include several ps terms to model nonlinear relationships with quantitative covariates, usually age in growth charts. When the lambda argument in ps() is scalar, it represents the actual smoothing parameter. When it is a vector, 'K-fold' cross validation is performed to select the ‘optimal’ lambda value and the model is fitted at such selected lambda value. To select the smoothing parameter via CV, foldid or nfolds may be supplied. If provided foldid overwrites nfolds, otherwise foldid is obtained via random extraction, namely sample(rep(seq(nfolds), length = n)). However selection of smoothing parameter is allowed with a unique ps() term in the formula.

Value

This function returns an object of class gcrq, that is a list with the following components

- coefficients The matrix of estimated regression parameters; the number of columns equals the number of the fitted quantile curves.
- B the design matrix of the final fit.
- df a vector reporting the df values for each quantile curve. See the section 'Warning' below.
- rho a vector including the values of the objective functions at the solution for each quantile curve.
**gcrq**

- **info.smooth**: some information on the smoothing term (if included in the formula via `ps`).
- **BB**: further information on the smoothing term (if present in the formula via `ps`), including stuff useful for plotting via `plot.gcrq()`.
- **Bderiv**: if the smooth term is included, the first derivative of the B spline basis.
- **boot.coef**: The array including the estimated coefficients at different bootstrap samples.
- **y**: the response vector (if `gcrq()` has been called with `y=TRUE`).
- **contrasts**: the contrasts used, when the model contains a factor.
- **xlevels**: the levels of the factors (when included) used in fitting.
- **taus**: a vector of values between 0 and 1 indicating the estimated quantile curves.
- **call**: the matched call.

**Warning**

The function (and underlying method) works pretty well in obtaining point estimates and displaying quantile curves accordingly. Typically this is the main (and unique) goal when dealing with growth charts. However from a statistical viewpoint there are some important limitations affecting the theory and the relevant package,

1. Computation of model degrees of freedom
2. Computation of standard errors

Currently the function does not return standard errors for the parameter estimates (unless `n.boot>0`) and degrees of freedom are roughly computed by summing the 'zero' residuals for model containing the smooth term, or simply by the number of parameters in linear models.

**Note**

This function is based upon the package quantreg by R. Koenker. Currently methods specific to the class "gcrq" are `plot.gcrq`, `print.gcrq` and `summary.gcrq`.

**Author(s)**

Vito M. R. Muggeo, <vito.muggeo@unipa.it>

**References**


**See Also**

`ps`, `plot.gcrq`
Examples

```r
## Not run:
data(growthData) # load data
tauass <- seq(.1, .9, by = .1) # fix the percentiles of interest

m1 <- gcrq(y ~ ps(x, mon = 0), tau = tauass, data = growthData) # unpenalized.. very wiggly curves
# strongly penalized models
m2 <- gcrq(y ~ ps(x, mon = 0, lambda = 1000, pdiff = 2), tau = tauass, data = growthData) # linear
m3 <- gcrq(y ~ ps(x, mon = 0, lambda = 1000, pdiff = 3), tau = tauass, data = growthData) # quadratic

# penalized model with monotonicity restrictions
m4 <- gcrq(y ~ ps(x, mon = 1, lambda = 10), tau = tauass, data = growthData)

# monotonicity constraints with varying penalty
m5 <- gcrq(y ~ ps(x, mon = 1, lambda = 10, var.pen = "(1:k)^3"), tau = tauass, data = growthData)

par(mfrow = c(2, 2))
plot(m1, pch = 20, y = TRUE)
plot(m2, pch = 20, y = TRUE)
plot(m3, add = TRUE, lwd = 2)
plot(m4, pch = 20, y = TRUE)
plot(m5, pch = 20, y = TRUE, legend = TRUE)

# select lambda via 'K-fold' CV
m6 <- gcrq(y ~ ps(x, lambda = seq(0, 100, l = 20)), tau = tauass, data = growthData)
par(mfrow = c(1, 2))
plot(m6, cv = TRUE) # display CV score versus lambda values
plot(m6, y = TRUE) # fitting at the best lambda value

## End(Not run)
```

---

growthData

*Simulated data to illustrate capabilities of the package*

Description

The `growthData` data frame has 200 rows and 3 columns.

Usage

```r
data(growthData)
```

Format

A data frame with 200 observations on the following 3 variables.

- `x` the supposed `age` variable.
- `y` the supposed growth variable (e.g. weight).
- `z` an additional variable to be considered in the model.
Details

Simulated data to illustrate capabilities of the package.

Examples

data(growthData)
with(growthData, plot(x,y))

ncross.rq.fitXB

Estimation of noncrossing regression quantiles with monotonicity restrictions.

Description

These are internal functions of package quantregGrowth and should be not called by the user.

Usage

ncross.rq.fitXB(y, x, B = NULL, X = NULL, taus, interc=FALSE, monotone = FALSE, adj.middle = FALSE, ndx = 10, lambda = 0, deg = 3, dif = 3, eps = 1e-04, plott = 0, var.pen = NULL, ...)

ncross.rq.fitX(y, X = NULL, taus, lambda = 0, adj.middle = FALSE, eps = 1e-04, ...)

gcrq.rq.cv(y, B, X, taus, interc=FALSE, monotone, ndx, lambda, deg, dif, var.pen = NULL, cv = TRUE, nfolds = 10, foldid = NULL, eps = 1e-04)

Arguments

<table>
<thead>
<tr>
<th>y</th>
<th>the responses vector. see gcrq</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>the covariate supposed to have a nonlinear relationship.</td>
</tr>
<tr>
<td>B</td>
<td>the B-spline basis.</td>
</tr>
<tr>
<td>X</td>
<td>the design matrix for the linear parameters.</td>
</tr>
<tr>
<td>taus</td>
<td>the percentiles of interest.</td>
</tr>
<tr>
<td>interc</td>
<td>should the model intercept be estimated? see the same argument in gcrq</td>
</tr>
<tr>
<td>monotone</td>
<td>numerical value (-1/0/+1) to define a non-increasing, unconstrained, and non-decreasing flexible fit, respectively.</td>
</tr>
<tr>
<td>adj.middle</td>
<td>ignore it!</td>
</tr>
<tr>
<td>ndx</td>
<td>number of internal intervals within the covariate range, see ps.</td>
</tr>
</tbody>
</table>
null

lambda

deg

dif

eps

plot

var.pen

foldid

nfvolds

cv

...
Arguments

- **x**: a fitted "gcrq" object.
- **term**: the smooth variable name entering the model via ps. Relevant fitted quantile curves will be plotted. It may be missing if the model includes a single smooth term.
- **add**: logical. If TRUE the fitted quantile curves are added on the current plot.
- **y**: logical. If TRUE raw data are also displayed, provided that the object has been called with the argument y=TRUE.
- **legend**: logical. If TRUE a legend is drawn on on the right side of the fitted curves.
- **select.tau**: an optional numeric vector to draw only some of the fitted quantiles. Percentile values or integers 1 to length(tau) may be supplied.
- **deriv**: logical. If TRUE the first derivative of the curve is displayed.
- **cv**: logical. If TRUE and the "gcrq" object contains the matrix cv, then the cross-validation scores against the lambda values are plotted.
- **transf**: An optional character string (with "y" as argument) meaning a function to apply to the response variable before plotting. E.g. "(exp(y)-0.1)". See argument "transf" in gcrq().
- **lambda0**: logical. If cv=TRUE, should the CV plot include also the first CV value? Usually the first CV value is at lambda=0, and typically it is much bigger than the other values making the plot not easy to read. Default to FALSE to ignore the first CV value in the plot.
- **...**: Additional graphical parameters, such as 'xlab', 'ylab', and 'xlim'; 'lwd', 'col' and 'lty' for the fitted quantile lines; 'cex' for the legend (if legend=TRUE); 'cex.p', 'col.p', and 'pch.p' for the points (if y=TRUE).

Details

Takes a "gcrq" object and displays the fitted quantile curves. When the object contains the component cv, plot.gcrq can display cross-validation scores against the lambda values, see argument cv.

Value

The function simply generates a new plot or adds fitted curves to an existing one.

Author(s)

Vito M. R. Muggeo

See Also

gcrq

Examples

```r
## see ?gcrq
```
predictQR  

Prediction for "gcrq" objects

Description

Takes a "gcrq" objects and computes fitted values

Usage

predictQR(object, newdata, xreg)

Arguments

object a fitted "gcrq" object.
newdata a dataframe including all the covariates of the model. The smooth term is represented by a covariate and proper basis functions will be build accordingly. Ignored if xreg is provided.
xreg the design matrix for which predictions are requested. Note xreg has to include the basis functions of the B-spline.

Details

predictQR computes fitted quantiles as a function of observations included in newdata or xreg. Either newdata or xreg have to be supplied, but newdata is ignored when xreg is provided.

Value

A matrix of fitted values with number of rows equal to number of rows of input data and number of columns depending on the fitted quantile curves.

Note

This function is at a preliminary stage and it should be replaced by the method predict.gcrq. Please use it with care.

Author(s)

Vito M.R. Muggeo

See Also

gcrq

Examples

#see ?gcrq
**print.gcrq**

*Print method for the gcrq class*

---

**Description**

Printing the most important features of a gcrq model.

**Usage**

```r
## S3 method for class 'gcrq'
print(x, digits = max(3, getOption("digits") - 3), ...)
```

**Arguments**

- `x` object of class `gcrq`
- `digits` number of digits to be printed
- `...` arguments passed to other functions

**Author(s)**

Vito M.R. Muggeo

**See Also**

`summary.gcrq`

---

**ps**

*Specifying a smooth term in the gcrq formula.*

---

**Description**

Function used to define the smooth term (via P-splines) within the gcrq formula. The function actually does not evaluate a (spline) smooth, but simply it passes relevant information to proper fitter functions.

**Usage**

```r
ps(x, monotone = 0, lambda = 0, pdiff = 3, ndx = NULL, deg = 3,
   var.pen = NULL)
```
Arguments

- **x**: The quantitative covariate supposed to have a nonlinear relationship with the quantiles. In growth charts this variable is typically the age.
- **monotone**: Numeric value to set up monotonicity restrictions on the fitted smooth function. It can be set to:
  - 0: no constrain;
  - 1: non-decreasing smooth function;
  - -1: non-increasing smooth function.
- **lambda**: A supplied smoothing parameter for the smooth term. If it is a vector, cross-validation is performed to select the ‘best’ value.
- **pdiff**: The difference order of the penalty. Default to 3.
- **ndx**: The number of intervals of the covariate range used to build the B-spline basis. If NULL, default, the empirical rule of Ruppert is used, namely \( \min(n/4, 40) \).
- **deg**: The degree of the spline polynomial. Default to 3.
- **var.pen**: A character indicating the varying penalty. See Details.

Details

When \( \lambda = 0 \) an unpenalized fit is obtained. The fit gets smoother as \( \lambda \) increases, and for a very large value of \( \lambda \) it approaches to a polynomial of degree \( \text{pdiff}-1 \). It is also possible to put a varying penalty to set a different amount of smoothing. For instance for a constant smoothing (\( \text{var.pen} = \text{NULL} \)) the penalty is \( \lambda \sum \Delta_k^2 \) where \( \Delta_k \) is the \( k \)-th difference (of order \( \text{pdiff} \)) of the spline coefficients. When a varying penalty is set, the penalty becomes \( \lambda \sum \Delta_k^2 w_k \). The weights \( w_k \) depend on \( \text{var.pen} \); for instance \( \text{var.pen}="((1:k)\^2)" \) results in \( w_k = k^2 \). See model m5 in examples of `gcrq`.

Value

The function simply returns the covariate with added attributes relevant to smooth term.

Author(s)

Vito M. R. Muggeo

References

For a general discussion on using B-spline and penalties in regression model see

See Also

gcrq

Examples

```r
## See ?gcrq
```
Description

summary and print methods for class gcrq

Usage

```r
## S3 method for class 'gcrq'
summary(object, digits = max(3,getOption("digits") - 3), ...)
```

```r
\method{print}{summary.gcrq}(x, 
```

Arguments

- `object`: An object of class "gcrq".
- `digits`: controls number of digits printed in output.
- `...`: further arguments.

Details

These methods are a very preliminary stage. Currently `print.gcrq` only warns that there exist no print method :-). `summary.gcrq` simply returns some information on the fitted object, such as the call, number of parameters and values of the objective functions at solution.

Author(s)

Vito M.R. Muggeo

See Also

gcrq

Examples

```r
## see ?gcrq
```
`vcov.gcrq`  
Variance-Covariance Matrix for a Fitted `cgrq` Model

Description

Returns the variance-covariance matrix of the parameter estimates of a fitted gcrq model object.

Usage

```r
## S3 method for class 'gcrq'
vcov(object, ...)  
```

Arguments

- `object` a fitted model object of class "gcrq" returned by `gcrq()`.
- `...` additional arguments.

Details

If the "gcrq" object includes results from bootstrap runs (namely the component `boot.coef` is not `NULL`), `vcov.gcrq()` computes the covariance matrix for the parameter estimates of each quantile curve.

Value

A list including the covariance matrices of the parameter estimates for each regression quantile curve.

Author(s)

Vito Muggeo

See Also

`summary.gcrq`
Index

*Topic **datasets**  
growthData, 6

*Topic **models**  
print.gcrq, 11  
quartregGrowth-package, 2

*Topic **model**  
gcrq, 3

*Topic **nonlinear**  
ncross.rq.fitXB, 7  
plot.gcrq, 8  
predictQR, 10  
summary.gcrq, 13

*Topic **package**  
quartregGrowth-package, 2

*Topic **regression**  
gcrq, 3  
ncross.rq.fitXB, 7  
plot.gcrq, 8  
predictQR, 10  
ps, 11  
quartregGrowth-package, 2  
summary.gcrq, 13  
vcov.gcrq, 14

*Topic **smooth**  
gcrq, 3  
ps, 11

gcrq, 3, 7–10, 12, 13  
gcrq.rq.cv (ncross.rq.fitXB), 7  
growthData, 6

ncross.rq.fitX (ncross.rq.fitXB), 7  
ncross.rq.fitXB, 7

plot.gcrq, 5, 8, 8  
predictQR, 10  
print.gcrq, 11  
ps, 4, 5, 7, 8, 11

quartregGrowth  
(quantregGrowth-package, 2  
rq, 3  
summary.gcrq, 11, 13, 14  
vcov.gcrq, 14