Package: alqrfe (via r-universe)

September 28, 2024

Type Package

Title Adaptive Lasso Quantile Regression with Fixed Effects

Version 1.1

Date 2022-11-30

Description Quantile regression with fixed effects solves longitudinal data, considering the individual intercepts as fixed effects. The parametric set of this type of problem used to be huge. Thus penalized methods such as Lasso are currently applied. Adaptive Lasso presents oracle proprieties, which include Gaussianity and correct model selection. Bayesian information criteria (BIC) estimates the optimal tuning parameter lambda. Plot tools are also available.

License GPL (>= 2)

Imports Rcpp (>= 1.0.5), MASS (>= 7.3-49)

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.2.1

NeedsCompilation yes

Author Ian Meneghel Danilevicz [aut, cre] (<https://orcid.org/0000-0003-4541-0524>), Pascal Bondon [aut], Valderio A. Reisen [aut]

Maintainer Ian Meneghel Danilevicz <iandanilevicz@gmail.com>

Repository CRAN

Date/Publication 2022-11-30 21:00:02 UTC

Contents

`alqrfe-package` *Adaptive Lasso Quantile Regression with Fixed Effects*

Description

Quantile regression with fixed effects solves longitudinal data, considering the individual intercepts as fixed effects. The parametric set of this type of problem used to be huge. Thus penalized methods such as Lasso are currently applied. Adaptive Lasso presents oracle proprieties, which include Gaussianity and correct model selection. Bayesian information criteria (BIC) estimates the optimal tuning parameter lambda. Plot tools are also available.

Package Content

Index of help topics:

`alqrfe-package`	`Adaptive Lasso Quantile Regression with Fixed Effects`
`bic_hat`	`Bayesian Information Criteria`
`clean_data`	`Clean missings`
`df_hat`	`degrees of fredom`
`f_den`	`Kernel density`
`f_tab`	`Tabular function`
`loss_alqr`	`Loss adaptive lasso quantile regression with fixed effects`
`loss_lqr`	`Loss lasso quantile regression with fixed effects`

loss_qr	Loss quantile regression
loss_qrfe	Loss quantile regression with fixed effects
make_z	Incident matrix Z
mqr	multiple penalized quantile regression
mqr_alpha	multiple penalized quantile regression - alpha
optim_alqr	optim adaptive lasso quantile regression with fixed effects
optim_lqr	optim lasso quantile regression with fixed effects
optim_qr	optim quantile regression
optim_qrfe	optim quantile regression with fixed effects
plot_alpha	plot multiple penalized quantile regression - alpha
plot_taus	plot multiple penalized quantile regression
print.ALQRFE	Print an ALQRFE
q_cov	Covariance
qr	quantile regression
rho_koenker	Rho Koenker
sgf	Identify significance

Maintainer

NA

Author(s)

NA

`bic_hat` *Bayesian Information Criteria*

Description

Bayesian Information Criteria

Usage

```
bic_hat(res, theta, tau, N, p, inf)
```

Arguments

res	Numeric vector, residuals.
theta	Numeric vector, parameters.
tau	Numeric scalar, the percentile.
N	Numeric integer, sample size.
p	Numeric integer, parameter length.
inf	Numeric, internal small quantity.

Value

BIC value

`clean_data` *Clean missings*

Description

Clean missings

Usage

```
clean_data(y, x, id)
```

Arguments

`y`	Numeric vector, outcome.
`x`	Numeric matrix, covariates
`id`	Numeric vector, identifies the unit to which the observation belongs.

Value

list with the same objects y, x, id, but without missings.

Examples

```
n = 10
m = 4
d = 3
N = n*m
L = N*d
x = matrix(rnorm(L), ncol=d, nrow=N)
subj = rep(1:n, each=m)
alpha = rnorm(n)
beta = rnorm(d)
eps = rnorm(N)
y = x %*% beta  + matrix(rep(alpha, each=m) + eps)
y = as.vector(y)
x[1,3] = NA
clean_data(y=y, x=x, id=subj)
```

df_hat *degrees of fredom*

Description

This function estimates the degrees of fredom

Usage

```
df_hat(theta, N, p, inf)
```

Arguments

theta	Numeric vector, parameters to be test
N	Numeric integer, sample size.
p	Numeric integer, length of theta.
inf	Numeric, internal small quantity.

Value

degrees of fredom

f_den *Kernel density*

Description

Kernel density

Usage

```
f_den(x, inf)
```

Arguments

x	Numeric vector.
inf	Numeric, internal small quantity.

Value

y vector, kernel density estimation.

Examples

```
x = rnorm(10)
f_den(x, 0.0001)
```

f_tab *Tabular function*

Description

Tabular function

Usage

```
f_tab(N, n, d, theta, sig2, kind, inf, digt)
```

Arguments

N	sample size.
n	length of alpha.
d	length of beta.
theta	Numeric vector.
sig2	Numeric vector.
kind	Numeric, 1 means alpha, 2 means beta
inf	Numeric scalar, internal value, small value.
digt	Numeric integer, round.

loss_alqr *Loss adaptive lasso quantile regression with fixed effects*

Description

Loss adaptive lasso quantile regression with fixed effects

Usage

```
loss_alqr(theta, x, y, z, tau, n, d, mm, lambda, w)
```

Arguments

theta	initial values
x	design matrix
y	vector output
z	incident matrix
tau	percentile
n	N sample size
d	columns of x

mm	n columns of z
lambda	constriction parameter
w	weights

---

loss_lqr *Loss lasso quantile regression with fixed effects*

Description

Loss lasso quantile regression with fixed effects

Usage

```
loss_lqr(theta, x, y, z, tau, n, d, mm, lambda)
```

Arguments

theta	initial values
x	design matrix
y	vector output
z	incident matrix
tau	percentile
n	N sample size
d	columns of x
mm	n columns of z
lambda	constriction parameter

---

loss_qr *Loss quantile regression*

Description

Loss quantile regression

Usage

```
loss_qr(beta, x, y, tau, N, d)
```

Arguments

beta	initial values
x	design matrix
y	vector output
tau	percentile
N	sample size
d	columns of x

loss_qrfe *Loss quantile regression with fixed effects*

Description

Loss quantile regression with fixed effects

Usage

```
loss_qrfe(theta, x, y, z, tau, n, d, mm)
```

Arguments

theta	initial values
x	design matrix
y	vector output
z	incident matrix
tau	percentile
n	N sample size
d	columns of x
mm	n columns of z

make_z *Incident matrix Z*

Description

Create an Incident matrix Z

Usage

```
make_z(n, N, id)
```

Arguments

n	Numeric integer, number of incidents (subjects, units or individuals).
N	Numeric integer, sample size.
id	Numeric vector of integer, incident identification.

Value

Z matrix.

mqr *multiple penalized quantile regression*

Description

Estimate QR for several taus

Usage

```
mqr(x, y, subj, tau = 1:9/10, method = "qr", ngrid = 20, inf = 1e-08, digt = 4)
```

Arguments

x	Numeric matrix, covariates
y	Numeric vector, outcome.
subj	Numeric vector, identifies the unit to which the observation belongs.
tau	Numeric vector, identifies the percentiles.
method	Factor, "qr" quantile regression, "qrfe" quantile regression with fixed effects, "lqrfe" Lasso quantile regression with fixed effects, "alqr" adaptive Lasso quantile regression with fixed effects.
ngrid	Numeric scalar greater than one, number of BIC to test.
inf	Numeric scalar, internal value, small value.
digt	Numeric scalar, internal value greater than one, define "zero" coefficient.

Value

Beta Numeric array, with three dimmensions: 1) tau, 2) coef., lower bound, upper bound, 3) exploratory variables.

Examples

```
n = 10
m = 5
d = 4
N = n*m
L = N*d
x = matrix(rnorm(L), ncol=d, nrow=N)
subj = rep(1:n, each=m)
alpha = rnorm(n)
beta = rnorm(d)
eps = rnorm(N)
y = x %*% beta  + matrix(rep(alpha, each=m) + eps)
y = as.vector(y)

Beta = mqr(x,y,subj,tau=1:9/10, method="qr", ngrid = 10)
Beta
```

mqr_alpha *multiple penalized quantile regression - alpha*

Description

Estimate QR intercepts for several taus

Usage

```
mqr_alpha(
  x,
  y,
  subj,
  tau = 1:9/10,
  method = "qr",
  ngrid = 20,
  inf = 1e-08,
  digt = 4
)
```

Arguments

x	Numeric matrix, covariates
y	Numeric vector, outcome.
subj	Numeric vector, identifies the unit to which the observation belongs.
tau	Numeric vector, identifies the percentiles.
method	Factor, "qr" quantile regression, "qrfe" quantile regression with fixed effects, "lqrfe" Lasso quantile regression with fixed effects, "alqr" adaptive Lasso quantile regression with fixed effects.
ngrid	Numeric scalar greater than one, number of BIC to test.
inf	Numeric scalar, internal value, small value.
digt	Numeric scalar, internal value greater than one, define "zero" coefficient.

Value

Alpha Numeric array, with three dimmensions: 1) tau, 2) coef., lower bound, upper bound, 3) exploratory variables.

Examples

```
n = 10
m = 5
d = 4
N = n*m
L = N*d
x = matrix(rnorm(L), ncol=d, nrow=N)
```

```
subj = rep(1:n, each=m)
alpha = rnorm(n)
beta = rnorm(d)
eps = rnorm(N)
y = x %*% beta  + matrix(rep(alpha, each=m) + eps)
y = as.vector(y)

Alpha = mqr(x,y,subj,tau=1:9/10, method="qr", ngrid = 10)
Alpha
```

---

optim_alqr *optim adaptive lasso quantile regression with fixed effects*

Description

This function solves an adaptive lasso quantile regression with fixed effects

Usage

```
optim_alqr(beta, alpha, wbeta, walpha, x, y, z, tau, N, d, n, ngrid, inf)
```

Arguments

beta	Numeric vector, initials values
alpha	Numeric vector, initials values
wbeta	Numeric vector, beta weigths
walpha	Numeric vector, alpha weigths
x	Numeric matrix, covariates.
y	Numeric vector, output.
z	Numeric matrix, incidents.
tau	Numeric scalar, the percentile.
N	Numeric integer, sample size.
d	Numeric integer, X number of columns.
n	Numeric integer, length of alpha.
ngrid	Numeric integer, number of iteractions of BIC.
inf	Numeric, internal small quantity.

Value

parametric vector and residuals

optim_lqr *optim lasso quantile regression with fixed effects*

Description

This function solves a lasso quantile regression with fixed effects

Usage

```
optim_lqr(beta, alpha, x, y, z, tau, N, d, n, ngrid, inf)
```

Arguments

beta	Numeric vector, initials values
alpha	Numeric vector, initials values
x	Numeric matrix, covariates.
y	Numeric vector, output.
z	Numeric matrix, incidents.
tau	Numeric scalar, the percentile.
N	Numeric integer, sample size.
d	Numeric integer, X number of columns.
n	Numeric integer, length of alpha.
ngrid	Numeric integer, number of iteractions of BIC.
inf	Numeric, internal small quantity.

Value

parametric vector and residuals

optim_qr *optim quantile regression*

Description

This function solves a quantile regression

Usage

```
optim_qr(beta, x, y, tau, N, d)
```

Arguments

beta	Numeric vector, initials values.
x	Numeric matrix, covariates.
y	Numeric vector, output.
tau	Numeric scalar, the percentile.
N	Numeric integer, sample size.
d	Numeric integer, X number of columns.

Value

parametric vector and residuals.

---

optim_qrfe *optim quantile regression with fixed effects*

Description

This function solves a quantile regression with fixed effects

Usage

```
optim_qrfe(beta, alpha, x, y, z, tau, N, d, n)
```

Arguments

beta	Numeric vector, initials values
alpha	Numeric vector, initials values
x	Numeric matrix, covariates.
y	Numeric vector, output.
z	Numeric matrix, incidents.
tau	Numeric scalar, the percentile.
N	Numeric integer, sample size.
d	Numeric integer, X number of columns.
n	Numeric integer, length of alpha.

Value

parametric vector and residuals

plot_alpha *plot multiple penalized quantile regression - alpha*

Description

plot QR intercepts for several taus

Usage

```
plot_alpha(
  Beta,
  tau = 1:9/10,
  D,
  ylab = expression(alpha[1]),
  col = 2,
  lwd = 1,
  lty = 2,
  pch = 1,
  cex.axis = 1,
  cex.lab = 1,
  main = ""
)
```

Arguments

Beta	Numeric array, with three dimmensions: 1) tau, 2) coef., lower bound, upper bound, 3) exploratory variables.
tau	Numeric vector, identifies the percentiles.
D	intercept's number.
ylab	y legend
col	color.
lwd	line width.
lty	line type.
pch	point character.
cex.axis	cex axis length.
cex.lab	cex axis length.
main	title.

Examples

```
n = 10
m = 5
d = 4
N = n*m
```

```
L = N*d
x = matrix(rnorm(L), ncol=d, nrow=N)
subj = rep(1:n, each=m)
alpha = rnorm(n)
beta = rnorm(d)
eps = rnorm(N)
y = x %*% beta  + matrix(rep(alpha, each=m) + eps)
y = as.vector(y)

Beta = mqr_alpha(x,y,subj,tau=1:9/10, method="qr", ngrid = 10)
plot_alpha(Beta,tau=1:9/10,D=1)
```

`plot_taus` *plot multiple penalized quantile regression*

Description

plot QR for several taus

Usage

```
plot_taus(
  Beta,
  tau = 1:9/10,
  D,
  col = 2,
  lwd = 1,
  lty = 2,
  pch = 1,
  cex.axis = 1,
  cex.lab = 1,
  main = ""
)
```

Arguments

`Beta`	Numeric array, with three dimmensions: 1) tau, 2) coef., lower bound, upper bound, 3) exploratory variables.
`tau`	Numeric vector, identifies the percentiles.
`D`	covariate's number.
`col`	color.
`lwd`	line width.
`lty`	line type.
`pch`	point character.
`cex.axis`	cex axis length.
`cex.lab`	cex axis length.
`main`	title.

Examples

```
n = 10
m = 5
d = 4
N = n*m
L = N*d
x = matrix(rnorm(L), ncol=d, nrow=N)
subj = rep(1:n, each=m)
alpha = rnorm(n)
beta = rnorm(d)
eps = rnorm(N)
y = x %*% beta  + matrix(rep(alpha, each=m) + eps)
y = as.vector(y)

Beta = mqr(x,y,subj,tau=1:9/10, method="qr", ngrid = 10)
plot_taus(Beta,tau=1:9/10,D=1)
```

---

print.ALQRFE *Print an ALQRFE*

Description

Define the visible part of the object class ALQRFE

Usage

```
## S3 method for class 'ALQRFE'
print(x, ...)
```

Arguments

x	An object of class "ALQRFE"
...	further arguments passed to or from other methods.

---

qr *quantile regression*

Description

Estimate quantile regression with fixed effects for one tau

Usage

```
qr(x, y, subj, tau = 0.5, method = "qr", ngrid = 20, inf = 1e-08, digt = 4)
```

Arguments

x	Numeric matrix, covariates
y	Numeric vector, outcome.
subj	Numeric vector, identifies the unit to which the observation belongs.
tau	Numeric, identifies the percentile.
method	Factor, "qr" quantile regression, "qrfe" quantile regression with fixed effects, "lqrfe" Lasso quantile regression with fixed effects, "alqr" adaptive Lasso quantile regression with fixed effects.
ngrid	Numeric scalar greater than one, number of BIC to test.
inf	Numeric scalar, internal value, small value.
digt	Numeric scalar, internal value greater than one, define "zero" coefficient.

Value

alpha Numeric vector, intercepts' coefficients.

beta Numeric vector, exploratory variables' coefficients.

lambda Numeric, estimated lambda.

res Numeric vector, percentile residuals.

tau Numeric scalar, the percentile.

penalty Numeric scalar, indicate the chosen effect.

sig2_alpha Numeric vector, intercepts' standard errors.

sig2_beta Numeric vector, exploratory variables' standard errors.

Tab_alpha Data.frame, intercepts' summary.

Tab_beta Data.frame, exploratory variables' summary.

Mat_alpha Numeric matrix, intercepts' summary.

Mat_beta Numeric matrix, exploratory variables' summary.

method Factor, method applied.

References

Koenker, R. (2004) "Quantile regression for longitudinal data", J. Multivar. Anal., 91(1): 74-89, <doi:10.1016/j.jmva.2004.05.006>

Examples

```
# Example 1
n = 10
m = 5
d = 4
N = n*m
L = N*d
x = matrix(rnorm(L), ncol=d, nrow=N)
subj = rep(1:n, each=m)
```

```
alpha = rnorm(n)
beta = rnorm(d)
eps = rnorm(N)
y = x %*% beta  + matrix(rep(alpha, each=m) + eps)
y = as.vector(y)
m1 = qr(x,y,subj,tau=0.75, method="qrfe")
m1
m2 = qr(x,y,subj,tau=0.3, method="lqrfe", ngrid = 10)
m2

# Example 2, from MASS package
Rabbit = MASS::Rabbit
Rabbit$Treatment = ifelse(Rabbit$Treatment=="Control",0,1)
Rabbit$Animal = ifelse(Rabbit$Animal == "R1",1,ifelse(Rabbit$Animal == "R2",2,
ifelse(Rabbit$Animal == "R3",3,ifelse(Rabbit$Animal == "R4",4,5))))
X = matrix(cbind(Rabbit$Dose,Rabbit$Treatment), ncol=2)
m3 = qr(x=X, y=Rabbit$BPchange, subj=Rabbit$Animal,tau=0.5, method="alqrfe", ngrid = 10)
m3
```

---

q_cov *Covariance*

Description

Estimate Covariance matrix

Usage

```
q_cov(alpha, beta, d, inf, n, N, res, method, tau, X, Z)
```

Arguments

alpha	Numeric vector.
beta	Numeric vector.
d	length of beta.
inf	Numeric scalar, internal value, small value.
n	length of alpha.
N	sample size.
res	Numeric vector, residuals.
method	Factor, "qr" quantile regression, "qrfe" quantile regression with fixed effects, "lqrfe" Lasso quantile regression with fixed effects, "alqr" adaptive Lasso quantile regression with fixed effects.
tau	Numeric, identifies the percentile.
X	Numeric matrix, covariates.
Z	Numeric matrix, incident matrix.

rho_koenker *Rho Koenker*

Description

Rho Koenker

Usage

```
rho_koenker(x, tau)
```

Arguments

x	generic vector
tau	percentile

sgf *Identify significance*

Description

Identify significance

Usage

```
sgf(x)
```

Arguments

x	Numeric vector.

Value

y vector Factor, symbol flag of significant p-values.

Examples

```
n = 10
pvalue = rgamma(10,1,10)
sgf(pvalue)
```

Index