

Package ‘qfabs’

June 4, 2024

Type Package

Title qfabs

Version 0.0.1

Author Panpan Ren [aut,cre], Xu Liu [aut], Xiao Zhang [aut], Peng Zhan [aut], Tingting Qiu [aut]

Maintainer Panpan Ren <ren08067146@163.com>

Description High-dimensional quantile regression estimation with the regression coefficients difference shrinkage. The contrast penalty is to consider the similarity among the regression coefficients. The Lasso is used to yield a sparse estimator.

License GPL (>=2)

Imports mnormt

Repository github

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

R topics documented:

generator	1
mqfabs	2
qfabs	4
simulatedData	5
Index	6

generator	<i>Generate samples for multiple datasets</i>
-----------	---

Description

This function generates data including response and covariates for multiple datasets. Data can be heavy-tailed. Moreover, there may be a high-correlated pattern among covariates.

Usage

```
generator(n, p, beta, distr = "gaussian", rho = 0.5)
```

Arguments

n	The sample sizes in multiple datasets. An integer vector in R^M , where M is the number of datasets.
p	The number of covariates. The number of covariates in each dataset is the same.
beta	A $p \times M$ numeric matrix, which is the true coefficients for the M datasets.
distr	The error's distribution, including "gaussian", "t3", "cauchy" and "logistic".
rho	The strength of correlation among covariates.

Value

A list

- x - A $N \times p$ design matrix, where N is the total sample size in M datasets.
- y - A length N vector of the response for M datasets.

Examples

```
library("mnormt")
M = 2
n1 = n2 = 20
n = c(n1, n2)
p = 50
beta = matrix(0, p, M)
index1 = 1:10
index2 = 2:11
beta[index1, 1] = runif(10, 0.2, 1.0)
beta[index2, 2] = runif(10, 0.4, 1.4)
distr = "t3"
rho = 0.8
dat = generator(n, p, beta, distr, rho)
x = dat$x
y = dat$y
```

mqfabs

A minorization-maximization forward and backward stagewise algorithm for high-dimensional integrative quantile regression.

Usage

```
mqfabs(y, x, n, tau = 0.5, Lambda2 = NULL, nlambd2 = 50, Lambda2_thre = 1e-3, epsilon = 0.01,
       delta = 1e-8, xi = 1e-10, max.iter = 5000, gamma = 1)
```

Arguments

y	The response for M datasets, a numeric vector.
x	The design matrix for M datasets.
n	The sample sizes in multiple datasets. An integer vector in R^M , where M is the number of datasets.
tau	The quantile to be estimated, this is a number strictly between 0 and 1. Default value is 0.5.

Lambda2	The value of the contrast penalty parameter(s) that determine how much shrinkage in different regression coefficients is done. This should be either a scalar, or a vector. If not specified then a sequence will be automatically generated based on the data.
nlambda2	The number of lambda2s generated. This is ignored if lambda is set manually. Default is 50.
Lambda2_thre	Smallest value for lambda2, as a fraction of lambda2.max. Default is 1e-3.
epsilon	The step size for updating coefficients. Default is 0.01.
delta	A constant in pertubed loss function. Default is 1e-8.
xi	The threshold for mqfabs. Default is 1e-10.
max.iter	The maximum number of outer-loop iterations allowed. Default is 5000.
gamma	A tuning parameter in EBIC. Default is 1.

Value

A list.

- beta - The optimal solution that minimizes EBIC.
- lambda1 - lambda1 sequence generated by mqfabs.
- Lambda2 - Lambda2 sequence.
- lambda2 - lambda2 returned that minimizes the EBIC across the Lambda2 sequence.
- iter - Iterations.
- EBIC - The EBIC for each solution.
- opt - Position of the optimal lambda1 based on EBIC for lambda2.

Examples

```
data(simulatedData_gaussian)
y = data_gaussian$y
x = data_gaussian$x
n = data_gaussian$n
fit <- mqfabs(y, x, n)
```

```
data(simulatedData_t3)
y = data_t3$y
x = data_t3$x
n = data_t3$n
fit <- mqfabs(y, x, n)
```

```
data(simulatedData_cauchy)
y = data_cauchy$y
x = data_cauchy$x
n = data_cauchy$n
fit <- mqfabs(y, x, n)
```

```
data(simulatedData_logistic)
y = data_logistic$y
x = data_logistic$x
n = data_logistic$n
fit <- mqfabs(y, x, n)
```

qfabs	<i>A minorization-maximization forward and backward stagewise algorithm for high-dimensional quantile regression.</i>
-------	---

Usage

```
qfabs(y, x, tau = 0.5, epsilon = 0.01, delta = 1e-8, xi = 1e-10, max.iter = 5000, gamma = 1)
```

Arguments

y	The response, a numeric vector.
x	The design matrix.
tau	The quantile to be estimated, this is a number strictly between 0 and 1. Default value is 0.5.
epsilon	The step size for updating coefficients. Default is 0.01.
delta	A constant in perturbed loss function. Default is 1e-8.
xi	The threshold for qfabs. Default is 1e-10.
max.iter	The maximum number of outer-loop iterations allowed. Default is 5000.
gamma	A tuning parameter in EBIC. Default is 1.

Value

A list.

- beta - The optimal solution that minimizes EBIC.
- lambda - lambda sequence generated by qfabs.
- iter - Iterations.
- EBIC - The EBIC for each solution.
- opt - Position of the optimal lambda based on EBIC.

Examples

```
library("mnormt")
n = 200
p = 500
beta = matrix(rep(0, p), ncol = 1)
index = 1:5
beta[index, 1] = c(1.0, 1.1, 1.2, 1.3, 1.4)
distr = "t3"
rho = 0.8
dat = generator(n, p, beta, distr, rho)
x = dat$x
y = dat$y
tau = 0.2
fit <- qfabs(y, x, tau)
```

simulatedData	<i>Simulated datasets from linear models.</i>
---------------	---

Description

Simulated data from linear models, including family 'gaussian' (simulatedData_gaussian), 't3' (simulatedData_t3), 'cauchy' (simulatedData_cauchy), and 'logistic' (simulatedData_logistic).

Usage

```
data(simulatedData_gaussian)
```

Examples

```
data(simulatedData_gaussian)
x = data_gaussian$x
y = data_gaussian$y
n = data_gaussian$n
```

Index

generator, [1](#)

mqfabs, [2](#)

qfabs, [4](#)

simulatedData, [5](#)