Package ‘pqrBayes’

April 1, 2025

Type Package

Title Bayesian Penalized Quantile Regression

Version 1.1.2

Date 2025-03-31

Description Bayesian regularized quantile regression utilizing sparse priors to impose exact sparsity leads to efficient Bayesian shrinkage estimation, variable selection and statistical inference. In this package, we have implemented robust Bayesian variable selection with spike-and-slab priors under high-dimensional linear regression models (Fan et al. (2024) <doi:10.3390/e26090794> and Ren et al. (2023) <doi:10.1111/biom.13670>), and regularized quantile varying coefficient models (Zhou et al.(2023) <doi:10.1016/j.csda.2023.107808>). In particular, valid robust Bayesian inferences under both models in the presence of heavy-tailed errors can be validated on finite samples. Additional models including robust Bayesian group LASSO are also included. The Markov Chain Monte Carlo (MCMC) algorithms of the proposed and alternative models are implemented in C++.

Depends R (>= 3.5.0)

License GPL-2

Encoding UTF-8

URL `https://github.com/cenwu/pqrBayes`

BugReports `https://github.com/cenwu/pqrBayes/issues`

LazyData true

Imports Rcpp,glmnet,splines, stats

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.3.2

NeedsCompilation yes

Repository CRAN

Author Kun Fan [aut],
Cen Wu [aut, cre],
Jie Ren [aut],
Xiaoxi Li [aut],
Fei Zhou [aut]

Maintainer Cen Wu <`wucen@ksu.edu`>

Date/Publication 2025-04-01 03:00:02 UTC

Contents

`pqrBayes-package` *Bayesian penalized quantile regression for linear, group LASSO and varying coefficient models*

Description

In this package, we implement Bayesian penalized quantile regression for the linear regression model, group LASSO and quantile varying coefficient (VC) model. Point-mass spike-and-slab priors have been incorporated in the Bayesian hierarchical models to facilitate Bayesian shrinkage estimation with exact sparsity in these models. The three default methods are Bayesian regularized quantile regression with spike-and-slab priors under the linear model, group LASSO and VC model, correspondingly. In addition to default methods, users can also choose methods without robustness and/or spike–and–slab priors.

Details

The user friendly, integrated interface **pqrBayes()** allows users to flexibly choose fitting models by specifying the following parameters:

robust:	whether to fit a robust sparse quantile regression model or a quantile varying coefficient model or their non-robust counterparts.
sparse:	whether to use the spike-and-slab priors to impose exact sparsity.
model:	whether to fit a linear model, group LASSO or a varying coefficient model.

The function pqrBayes() returns a pqrBayes object that stores the posterior estimates of regression coefficients.

References

Fan, K., Subedi, S., Yang, G., Lu, X., Ren, J. and Wu, C. (2024). Is Seeing Believing? A Practitioner's Perspective on High-dimensional Statistical Inference in Cancer Genomics Studies. *Entropy*, 26(9).794 doi:10.3390/e26090794

Zhou, F., Ren, J., Ma, S. and Wu, C. (2023). The Bayesian regularized quantile varying coefficient model. *Computational Statistics & Data Analysis*, 107808 doi:10.1016/j.csda.2023.107808

Ren, J., Zhou, F., Li, X., Ma, S., Jiang, Y. and Wu, C. (2023). Robust Bayesian variable selection for gene-environment interactions. *Biometrics*, 79(2), 684-694 doi:10.1111/biom.13670

Wu, C., and Ma, S. (2015). A selective review of robust variable selection with applications in bioinformatics. *Briefings in Bioinformatics*, 16(5), 873–883 doi:10.1093/bib/bbu046

Zhou, F., Ren, J., Lu, X., Ma, S. and Wu, C. (2021). Gene–Environment Interaction: a Variable Selection Perspective. *Epistasis. Methods in Molecular Biology.* 2212:191–223 `https://link.springer.com/protocol/10.1007/978-1-0716-0947-7_13`

Ren, J., Zhou, F., Li, X., Chen, Q., Zhang, H., Ma, S., Jiang, Y. and Wu, C. (2020) Semi-parametric Bayesian variable selection for gene-environment interactions. *Statistics in Medicine*, 39: 617– 638 doi:10.1002/sim.8434

Ren, J., Zhou, F., Li, X., Wu, C. and Jiang, Y. (2019) spinBayes: Semi-Parametric Gene-Environment Interaction via Bayesian Variable Selection. R package version 0.1.0. `https://CRAN.R-project.org/package=spinBayes`

Wu, C., Jiang, Y., Ren, J., Cui, Y. and Ma, S. (2018). Dissecting gene-environment interactions: A penalized robust approach accounting for hierarchical structures. *Statistics in Medicine*, 37:437–456 doi:10.1002/sim.7518

Wu, C., Shi, X., Cui, Y. and Ma, S. (2015). A penalized robust semiparametric approach for gene-environment interactions. *Statistics in Medicine*, 34 (30): 4016–4030 doi:10.1002/sim.6609

Wu, C., Cui, Y., and Ma, S. (2014). Integrative analysis of gene–environment interactions under a multi–response partially linear varying coefficient model. *Statistics in Medicine*, 33(28), 4988–4998 doi:10.1002/sim.6287

Wu, C., Zhong, P.S. and Cui, Y. (2018). Additive varying–coefficient model for nonlinear gene–environment interactions. *Statistical Applications in Genetics and Molecular Biology*, 17(2) doi:10.1515/sagmb-20170008

Wu, C., Zhong, P.S. and Cui, Y. (2013). High dimensional variable selection for gene-environment interactions. *Technical Report. Michigan State University.*

See Also

`pqrBayes`

`coverage` *95% empirical coverage probability for a pqrBayes object*

Description

Calculate 95% empirical coverage probabilities for regression coefficients under linear, group LASSO and VC models, respectively.

Usage

```
coverage(object,coefficient,u.grid=NULL,model="linear")
```

Arguments

`object`	the pqrBayes object.
`coefficient`	the vector of true regression coefficients under a linear model or the matrix of true varying coefficients evaluated on the grid points under a varying coefficient model.
`u.grid`	the vector of grid points under a varying coefficient model. When assessing empirical coverage probabilities under a linear model or group LASSO, u.grid = NULL.
`model`	the model to be fitted. Users can also choose "linear" for a linear model, "VC" for a varying coefficient model and "group" for group LASSO.

Value

c

See Also

`pqrBayes`

Examples

```
## The quantile regression model
data(data)
data = data$data_linear
g=data$g
y=data$y
e=data$e
coeff = data$coeff
fit1=pqrBayes(g,y,u=NULL,e,d=NULL,quant=0.5,spline=NULL,model="linear")
coverage=coverage(fit1,coeff,model="linear")
```

`data` *Simulated data under high-dimensional linear, group LASSO and quantile varying coefficient models*

Description

Simulated data under high-dimensional linear, group LASSO and quantile varying coefficient models

Format

The data_linear object consists of 4 components: g, y, e and coeff. coeff contains the true values of parameters used for generating the response variable y. The data_group object consists of 4 components: g, y, e and coeff. coeff contains the true values of parameters used for generating the response variable y. The data_varying object consists of five components: g, y, u, e and coeff. coeff contains the true values of parameters used for generating the response variable y.

Details

Generating Y using a sparse linear (quantile) regression model

The true data generating model under sparse linear regression:

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_3 X_{i3} + \epsilon_i,$$

where $\epsilon_i \sim N(0,1)$, $\beta_0 = 0$, $\beta_1 = 1$, $\beta_2 = 1.5$ and $\beta_3 = 2$.

Generating Y using a high-dimensional group LASSO model

The true data generating model under a group LASSO model:

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_3 X_{i3} + \beta_7 X_{i7} + \beta_8 X_{i8} + \beta_9 X_{i9} + \epsilon_i,$$

where $\epsilon_i \sim N(0,1)$, $\beta_0 = 0$, $\beta_1 = 0.6$, $\beta_2 = 0.7$,$\beta_3 = 0.8$,$\beta_7 = 0.65$, $\beta_8 = 0.75$ and $\beta_9 = 0.85$.

Generating Y using a (quantile) varying coefficient model

Data generation under sparse (quantile) VC model:

$$Y_i = \gamma_0(v_i) + \gamma_1(v_i)X_{i1} + \gamma_2(v_i)X_{i2} + \gamma_3(v_i)X_{i3} + \epsilon_i,$$

where $\epsilon_i \sim N(0,1)$, $\gamma_0(v_i) = 1.5\sin(0.2\pi * v_i)$, $\gamma_1(v_i) = 2\exp(0.2v_i - 1) - 1.5$, $\gamma_2(v_i) = 2 - 2v_i$ and $\gamma_3(v_i) = -4 + (v_i - 2)^3/6$.

See Also

`pqrBayes`

Examples

```
data(data)
data = data$data_linear
g=data$g
dim(g)
y=data$y
coeff=data$coeff
print(coeff)

data = data$data_group
g=data$g
dim(g)
y=data$y
coeff=data$coeff
print(coeff)
```

```
data = data$data_varying
g=data$g
dim(g)
coeff=data$coeff
print(coeff)
```

`estimation.pqrBayes` *Estimation and estimation accuracy for a pqrBayes object*

Description

Calculate estimated regression coefficients with estimation accuracy from linear and quantile VC models, respectively.

Usage

```
estimation.pqrBayes(object,coefficient,u.grid=NULL,model="linear")
```

Arguments

`object`	an object of class 'pqrBayes'.
`coefficient`	the vector of quantile regression coefficients under a linear model or the matrix of true varying coefficients evaluated on the grid points under a varying coefficient model.
`u.grid`	the vector of grid points under a varying coefficient model. When fitting a linear (quantile) regression model or group LASSO, u.grid = NULL.
`model`	the model to be fitted. Users can choose "linear" for a linear model, "VC" for a varying coefficient model or "group for group LASSO.

Value

an object of class 'pqrBayes.est' is returned, which is a list with components:

`error`	mean square error or integrated mean square errors and total integrated mean square error.
`coeff.est`	estimated values of the regression coefficients or the varying coefficients.

See Also

`pqrBayes`

Examples

```
## The quantile regression model
data(data)
data = data$data_linear
g=data$g
y=data$y
e=data$e
coeff = data$coeff
fit1=pqrBayes(g,y,u=NULL,e,d = NULL,quant=0.5,spline=NULL,model="linear")
estimation=estimation.pqrBayes(fit1,coeff,model="linear")
```

`pqrBayes` *fit Bayesian penalized quantile regression for linear, group LASSO, or varying coefficient models*

Description

fit Bayesian penalized quantile regression for linear, group LASSO, or varying coefficient models

Usage

```
pqrBayes(
  g,
  y,
  u = NULL,
  e,
  d = NULL,
  quant = 0.5,
  iterations = 10000,
  burn.in = NULL,
  spline = NULL,
  robust = TRUE,
  sparse = TRUE,
  model = "linear",
  hyper = NULL,
  debugging = FALSE
)
```

Arguments

`g`	the matrix of predictors (subject to selection). Users do not need to specify an intercept which will be automatically included.
`y`	the response variable. The current version only supports the continuous response.
`u`	a vector of effect modifying variable of the quantile varying coefficient model. When fitting a linear model or group LASSO, u = NULL.

`e`	a matrix of clinical covariates not subject to selection.
`d`	a positive integer denotes the group size. When fitting a linear model or varying coefficient model, d = NULL.
`quant`	the quantile level specified by users. The default value is 0.5.
`iterations`	the number of MCMC iterations. The default value is 10,000.
`burn.in`	the number of burn-in iterations. If NULL, the first half of MCMC iterations will be used as burn-ins.
`spline`	a list of the number of interior knots (kn) for B-spline and the degree of B-spline basis (degree). When fitting a linear model or group LASSO, spline = NULL.
`robust`	logical flag. If TRUE, robust methods will be used. Otherwise, non-robust methods will be used. The default value is TRUE.
`sparse`	logical flag. If TRUE, spike-and-slab priors will be adopted to impose exact sparsity on regression coefficients. Otherwise, Laplacian shrinkage will be adopted. The default value is TRUE.
`model`	the model to be fitted. Users can specify "linear" for a linear model, "group" for group LASSO and "VC" for a varying coefficient model.
`hyper`	a named list of hyper-parameters. The default value is NULL.
`debugging`	logical flag. If TRUE, progress will be output to the console and extra information will be returned. The default value is FALSE.

Details

The quantile regression model described in "`data`" is:

$$Y_i = \sum_{k=1}^{q} E_{ik}\beta_k + \sum_{j=0}^{p} X_{ij}\gamma_j + \epsilon_i,$$

where β_k's are the regression coefficients for clinical covariates and γ_j's are the regression coefficients of $\boldsymbol{X}$.

The group LASSO model described in "`data`" is:

$$Y_i = \sum_{k=1}^{q} E_{ik}\beta_k + X_{i0}\gamma_0 + \sum_{j=1}^{m} \boldsymbol{X_{ij}}^\top \boldsymbol{\gamma_j} + \epsilon_i,$$

where β_k's are the regression coefficients for clinical covariates and $\boldsymbol{\gamma_j} = (\gamma_{j1}, \ldots, \gamma_{jd})^\top$ is the vector of regression coefficients of the $n \times d$ matrix $\boldsymbol{X}_j$.

The quantile varying coefficient model described in "`data`" is:

$$Y_i = \sum_{k=1}^{q} E_{ik}\beta_k + \sum_{j=0}^{p} \gamma_j(V_i)X_{ij} + \epsilon_i,$$

where β_k's are the regression coefficients for the clinical covariates, and γ_j's are the varying intercept and varying coefficients for predictors (e.g. genetic factors), respectively.

Users can modify the hyper-parameters by providing a named list of hyper-parameters via the argument ‘hyper’. The list can have the following named components

a0, b0: shape parameters of the Beta priors ($\pi^{a_0-1}(1-\pi)^{b_0-1}$) on π_0.

c1, c2: the shape parameter and the rate parameter of the Gamma prior on ν.

Please check the references for more details about the prior distributions.

Value

an object of class "pqrBayes" is returned, which is a list with components:

`obj`	a list of posterior samples from the MCMC and other parameters
`coefficients`	a list of posterior estimates of coefficients

References

Fan, K., Subedi, S., Yang, G., Lu, X., Ren, J. and Wu, C. (2024). Is Seeing Believing? A Practitioner's Perspective on High-dimensional Statistical Inference in Cancer Genomics Studies. *Entropy*, 26(9).794 doi:10.3390/e26090794

Zhou, F., Ren, J., Ma, S. and Wu, C. (2023). The Bayesian regularized quantile varying coefficient model. *Computational Statistics & Data Analysis*, 107808 doi:10.1016/j.csda.2023.107808

Ren, J., Zhou, F., Li, X., Ma, S., Jiang, Y. and Wu, C. (2023). Robust Bayesian variable selection for gene-environment interactions. *Biometrics*, 79(2), 684-694 doi:10.1111/biom.13670

Ren, J., Zhou, F., Li, X., Chen, Q., Zhang, H., Ma, S., Jiang, Y. and Wu, C. (2020) Semi-parametric Bayesian variable selection for gene-environment interactions. *Statistics in Medicine*, 39: 617– 638 doi:10.1002/sim.8434

Examples

```
## The quantile regression model
data(data)
data = data$data_linear
g=data$g
y=data$y
e=data$e

fit1=pqrBayes(g,y,u=NULL,e,d=NULL,quant=0.5,model="linear")


## non-sparse
sparse=FALSE
fit2=pqrBayes(g,y,u=NULL,e,d=NULL,quant=0.5,spline=NULL,sparse = sparse,model="linear")

## non-robust
robust = FALSE
fit3=pqrBayes(g,y,u=NULL,e,d=NULL,quant=0.5,spline=NULL,robust = robust,model="linear")



## The group LASSO model
data(data)
data = data$data_group
```

```
g=data$g
y=data$y
e=data$e

fit1=pqrBayes(g,y,u=NULL,e,d=3,quant=0.5,model="group")


## non-sparse
sparse=FALSE
fit2=pqrBayes(g,y,u=NULL,e,d=3,quant=0.5,spline=NULL,sparse = sparse,model="group")

## non-robust
robust = FALSE
fit3=pqrBayes(g,y,u=NULL,e,d=3,quant=0.5,spline=NULL,robust = robust,model="group")


## The quantile varying coefficient model
data(data)
data = data$data_varying
g=data$g
y=data$y
u=data$u
e=data$e

spline = list(kn=2,degree=2)
fit1=pqrBayes(g,y,u,e,quant=0.5,spline = spline,model="VC")



## non-sparse
sparse=FALSE
fit2=pqrBayes(g,y,u,e,quant=0.5,spline = spline,sparse = sparse,model="VC")

## non-robust
robust = FALSE
fit3=pqrBayes(g,y,u,e,quant=0.5,spline = spline,robust = robust,model="VC")
```

`pqrBayes.select` *Variable selection for a pqrBayes object*

Description

Variable selection for a pqrBayes object

Usage

```
pqrBayes.select(object,sparse=T,model="linear")
```

Arguments

`object`	a pqrBayes object.
`sparse`	logical flag. If TRUE, the sparse model is used for variable selection. The default value is TRUE.
`model`	the model to be fitted. Users can also choose "linear" for a linear model, "VC" for a varying coefficient model or "group for group LASSO.

Details

For class 'Sparse', the median probability model (MPM) (Barbieri and Berger, 2004) is used to identify predictors that are significantly associated with the response variable. For class 'NonSparse', variable selection is based on 95% credible interval. Please check the references for more details about the variable selection.

Value

an object of class 'select' is returned, which includes the indices of the selected predictors (e.g. genetic factors).

References

Ren, J., Zhou, F., Li, X., Ma, S., Jiang, Y. and Wu, C. (2023). Robust Bayesian variable selection for gene-environment interactions. *Biometrics*, 79(2), 684-694 doi:10.1111/biom.13670

Barbieri, M.M. and Berger, J.O. (2004). Optimal predictive model selection. *Ann. Statist*, 32(3):870–897

See Also

`pqrBayes`

Examples

```
## The quantile regression model
data(data)
data = data$data_linear
g=data$g
y=data$y
e=data$e

fit1=pqrBayes(g,y,u=NULL,e,d = NULL,quant=0.5,spline=NULL,model="linear")
sparse=TRUE
select=pqrBayes.select(obj = fit1,sparse = sparse,model="linear")

## The quantile varying coefficient model
data(data)
data = data$data_varying
g=data$g
y=data$y
u=data$u
e=data$e
spline = list(kn=2,degree=2)
```

```
fit1=pqrBayes(g,y,u,e,d = NULL,quant=0.5,spline = spline,model="VC")
sparse=TRUE
select=pqrBayes.select(obj = fit1,sparse = sparse,model="VC")
select


## non-sparse
sparse=FALSE
spline = list(kn=2,degree=2)
fit2=pqrBayes(g,y,u,e,d = NULL,quant=0.5,spline = spline,sparse = sparse,model="VC")
select=pqrBayes.select(obj=fit2,sparse=FALSE,model="VC")
select
```

`predict_pqrBayes` *Make predictions from a pqrBayes object*

Description

Make predictions from a pqrBayes object

Usage

```
predict_pqrBayes(object, g.new, u.new, e.new, y.new, quant, model, ...)
```

Arguments

`object`	a pqrBayes object.
`g.new`	a matrix of new predictors (e.g. genetic factors) at which predictions are to be made. When being applied to the linear model or group LASSO, g.new = g.
`u.new`	a vector of new environmental factor at which predictions are to be made. When being applied to the linear model or group LASSO, u.new = NULL.
`e.new`	a vector or matrix of new clinical covariates at which predictions are to be made. When being applied to the linear model, e.new = e.
`y.new`	a vector of the response of new observations. When being applied to the linear model or group LASSO, y.new = y.
`quant`	the quantile level. The default is 0.5.
`model`	the model to be fitted. The default is "VC" for a quantile varying coefficient model. Users can also specify "linear" for a linear model and "group" for a group LASSO.
`...`	other predict arguments

Details

g.new (u.new) must have the same number of columns as g (u) used for fitting the model. By default, the clinical covariates are NULL unless provided. The predictions are made based on the posterior estimates of coefficients in the pqrBayes object.

Value

an object of class ‘pqrBayes.pred’ is returned, which is a list with components:

`error`	prediction error.
`y.pred`	predicted values of the new observations.

See Also

`pqrBayes`

Examples

```
## The quantile regression model
data(data)
data = data$data_linear
g=data$g
y=data$y
e=data$e
fit1=pqrBayes(g,y,u=NULL,e,d = NULL,quant=0.5,spline=NULL,model="linear")
prediction=predict_pqrBayes(fit1,g,u.new=NULL,e.new = e, y.new = y,model="linear")
```

`print.pqrBayes` *print a pqrBayes result*

Description

Print a pqrBayes result

Usage

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

Arguments

`x`	pqrBayes result.
`digits`	significant digits in printout.
`...`	other print arguments.

Value

No return value, called for side effects.

See Also

`pqrBayes`

`print.pqrBayes.pred` *print a pqrBayes.pred object*

Description

Print a summary of a pqrBayes.pred object

Usage

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

Arguments

`x`	pqrBayes.pred object.
`digits`	significant digits in printout.
`...`	other print arguments

Value

No return value, called for side effects.

See Also

`predict_pqrBayes`

`print.pqrBayes.select` *print a select.pqrBayes object*

Description

Print a summary of a select.pqrBayes object

Usage

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

Arguments

`x`	pqrBayes.select object.
`digits`	significant digits in printout.
`...`	other print arguments

Value

No return value, called for side effects.

See Also

`pqrBayes.select`

Index