Package 'emBayes'

May 16, 2023

Type Package

Title Spike-and-Slab Quantile LASSO

Version 1.0

Date 2023-05-15

Author Yuwen Liu, Cen Wu

Maintainer Yuwen Liu <yuwen@ksu.edu>

Description Various variable selection methods have been developed from frequentist and Bayesian perspectives. However, current Bayesian Spike-and-slab LASSO method lack robustness. In this package, we provide the implementation of Spike-and-slab quantile LASSO (BQLSS). It is a variable selection and estimation method that is constructed by extending the spike-and slab LASSO method to Bayesian quantile regression. It combines the advantages of Bayesian spike-and-slab prior and frequentist LASSO penalty function while maintaining robustness. Alternative method Bayesian linear regression with spike-and-slab LASSO prior (BLSS) is also included. The core modules of the package have been developed in C++.

Depends R (>= 3.5.0)

License GPL-2

Encoding UTF-8

LazyData true

Imports Rcpp (>= 1.0.9), glmnet

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.2.1

NeedsCompilation yes

ExperimentalWindowsRuntime ucrt

Archs x64

R topics documented:

`cv.emBayes` *k-folds cross-validation for emBayes*

Description

This function performs cross-validation and returns the optimal values of the tuning parameters.

Usage

```
cv.emBayes(
  y,
  clin = NULL,
  X,
  quant,
  t0,
  t1,
  k,
  func,
  error = 0.01,
  maxiter = 200
)
```

Arguments

`y`	a vector of response variable.
`clin`	a matrix of clinical factors. It has default value NULL.
`X`	a matrix of genetic factors.
`quant`	value of quantile.
`t0`	a user-supplied sequence of the spike scale s_0.
`t1`	a user-supplied sequence of the slab scale s_1.
`k`	number of folds for cross-validation.
`func`	methods to perform variable selection. Two choices are available: "BLSS" and "BQLSS".
`error`	cutoff value for determining convergence. The algorithm reaches convergence if the difference in the expected log-likelihood of two iterations is less than the value of error. The default value is 0.01.
`maxiter`	the maximum number of iterations that is used in the estimation algorithm. The default value is 200.

Details

When performing cross-validation for emBayes, function cv.emBayes returns two sets of optimal tuning parameters and their corresponding cross-validation error matrices. The spike scale parameter $CL.s0$ and the slab scale parameter $CL.s1$ are obtained based on the quantile check loss. The spike scale parameter $SL.s0$ and the slab scale parameter $SL.s1$ are obtained based on the squared loss. The spike scale parameter $SIC.s0$ and the slab scale parameter $SIC.s1$ are obtained based on the Schwarz Information Criterion (SIC). Corresponding error matrices $CL.CV$, $SL.CV$ and $SIC.CV$ can also be obtained from the output.

Schwarz Information Criterion has the following form:

$$SIC = \log \sum_{i=1}^{n} L(y_i - \hat{y}_i) + \frac{\log n}{2n} edf$$

where $L(\cdot)$ is the check loss and edf is the number of close to zero residuals (≤ 0.001).

Value

a list with components:

`CL.s0`	the optimal spike scale under check loss.
`CL.s1`	the optimal slab scale under check loss.
`SL.s0`	the optimal slab scale under squared loss.
`SL.s1`	the optimal slab scale under squared loss.
`SIC.s0`	the optimal slab scale under SIC.
`SIC.s1`	the optimal slab scale under SIC.
`CL.CV`	cross-validation error matrix under check loss.
`SL.CV`	cross-validation error matrix under squared loss.
`SIC.CV`	cross-validation error matrix under SIC.

`emBayes` *fit a model with given tuning parameters*

Description

This function performs penalized variable selection based on spike-and-slab quantile LASSO (BQLSS) or Bayesian linear regression with spike-and-slab LASSO prior (BLSS). Typical usage is to first obtain the optimal spike scale and slab scale using cross-validation, then specify them in the emBayes function.

Usage

```
emBayes(y, clin = NULL, X, quant, s0, s1, func, error = 0.01, maxiter = 200)
```

Arguments

`y`	a vector of response variable.
`clin`	a matrix of clinical factors. It has default value NULL.
`X`	a matrix of genetic factors.
`quant`	value of quantile.
`s0`	value of the spike scale s_0.
`s1`	value of the slab scale s_1.
`func`	methods to perform variable selection. Two choices are available: "BLSS" and "BQLSS".
`error`	cutoff value for determining convergence. The algorithm reaches convergence if the difference in the expected log-likelihood of two iterations is less than the value of error. The default value is 0.01.
`maxiter`	the maximum number of iterations that is used in the estimation algorithm. The default value is 200.

Details

The current version of emBayes supports two types of methods: "BLSS" and "BQLSS".

- **BLSS:** Bayesian LASSO with spike-and-slab prior fits a Bayesian linear regression through the EM algorithm.
- **BQLSS:** spike-and-slab quantile LASSO fits a Bayesian quantile regression (based on asymmetric Laplace distribution) through the EM algorithm.

Users can choose the desired method by setting func="BLSS" or "BQLSS".

Value

a list with components:

`alpha`	a vector containing the estimated intercept and clinical coefficients.
`intercept`	value of the estimated intercept.
`clin.coe`	a vector of estimated clinical coefficients.
`beta`	a vector of estimated beta coefficients.
`sigma`	value of estimated asymmetric Laplace distribution scale parameter σ.
`theta`	value of estimated probability parameter θ.
`iter`	value of number of iterations.
`ll`	a vector of expectation of likelihood at each iteration.

Examples

```
data(genes)
##load the clinical factors, genetic factors, response and quantile data
clin=genes$clin
X=genes$X
y=genes$y
quant=genes$quant
##generate tuning vectors of desired range
t0 <- seq(0.01,0.015,length.out=2)
t1 <- seq(0.1,0.5,length.out=2)
##perform cross-validation and obtain tuning parameters based on check loss
CV <- cv.emBayes(y,clin,X,quant,t0,t1,k=5,func="BQLSS",error=0.01,maxiter=2000)
s0 <- CV$CL.s0
s1 <- CV$CL.s1
##perform BQLSS under optimal tuning and calculate value of TP and FP for selecting beta
EM <- emBayes(y,clin,X,quant,s0,s1,func="BQLSS",error=0.01,maxiter=2000)
fit <- EM$beta
coef <- genes$coef
tp <- sum(fit[coef!=0]!=0)
fp <- sum(fit[coef==0]!=0)
list(tp=tp,fp=fp)
```

genes *simulated gene expression example data*

Description

Simulated gene expression data for demonstrating the usage of emBayes.

Usage

```
data(genes)
```

Format

The genes file consists of five components: y, clin, X, quant, coef and clin.coe. The coefficients and clinical coefficients are the true values of parameters used for generating response y. They can be used for performance evaluation.

Details

The data model for generating response

Let y_i be the response of the i-th subject (1$\leq$ i$\leq$ n). We have $z_i = (1, z_{i1}, \ldots, z_{iq})^\top$ being a $(q+1)$-dimensional vector of which the last q components indicate clinical factors and $x_i = (x_{i1}, \ldots, x_{ip})^\top$ denoting a p-dimensional vector of genetic factors. The linear quantile regression model for the τ-th quantile $(0 < \tau < 1)$ is:

$$y_i = z_i^\top \alpha + x_i^\top \beta + \epsilon_i$$

where $\alpha = (\alpha_0, \cdots, \alpha_q)^\top$ contains the intercept and the regression coefficients for the clinical covariates. $\beta = (\beta_1, \cdots, \beta_p)^\top$ are the regression coefficients and random error $\epsilon_i = (\epsilon_1, ..., \epsilon_n)^\top$ is set to follow a T2 distribution and has value 0 at its τ-th quantile.

See Also

emBayes

Index