Package ‘spinBayes’

June 6, 2019

Type Package

Title Semi-Parametric Gene-Environment Interaction via Bayesian Variable Selection

Version 0.1.0

Author Jie Ren, Fei Zhou, Xiaoxi Li, Cen Wu, Yu Jiang

Maintainer Jie Ren <jieren@ksu.edu>

Description Many complex diseases are known to be affected by the interactions between genetic variants and environmental exposures beyond the main genetic and environmental effects. Existing Bayesian methods for gene-environment (G×E) interaction studies are challenged by the high-dimensional nature of the study and the complexity of environmental influences. We have developed a novel and powerful semi-parametric Bayesian variable selection method that can accommodate linear and nonlinear G×E interactions simultaneously (Ren et al. (2019) <arXiv:1906.01057>). Furthermore, the proposed method can conduct structural identification by distinguishing nonlinear interactions from main effects only case within Bayesian framework. Spike-and-slab priors are incorporated on both individual and group level to shrink coefficients corresponding to irrelevant main and interaction effects to zero exactly. The Markov chain Monte Carlo algorithms of the proposed and alternative methods are efficiently implemented in C++.

Depends R (>= 3.5.0)

License GPL-2

Encoding UTF-8

LazyData true

LinkingTo Rcpp, RcppArmadillo

Imports Rcpp, splines, MASS, glmnet, utils

URL https://github.com/jrhub/spinBayes

BugReports https://github.com/jrhub/spinBayes/issues

RoxygenNote 6.1.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2019-06-06 14:40:03 UTC
Description

Within the Bayesian framework, we propose a partially linear varying coefficient model (PLVC) for G×E interactions. The varying coefficient functions capture the possible non-linear G×E interaction, and the linear component models the G×E interactions with linear assumptions. The changing of basis with B splines is adopted to separate the coefficient functions with varying, non-zero constant and zero forms, corresponding to cases of nonlinear interaction, main effect only (no interaction) and no genetic interaction at all.

Details

The user friendly, integrated interface BVCfit() allows users to flexibly choose the fitting methods they prefer. There are three arguments in BVCfit() that control the fitting method

- **sparse**: whether to use the spike-and-slab priors to achieve sparsity.
- **VC**: whether to separate the coefficient functions with varying effects and non-zero constant (main) effects.
- **structural**: whether to use varying coefficient functions for modeling non-linear GxE interactions.

BVCfit() returns a BVCfit object that contains the posterior estimates of each coefficients. S3 generic functions BVSelection(), predict() and print() are implemented for BVCfit objects. BVSelection() takes a BVCfit object and returns the variable selection results. predict() takes a BVCfit object and returns the predicted values for new observations.

References

01057


**See Also**

*BVCfit*

---

**BVCfit**

*fit a Semi-parametric Bayesian variable selection*

**Description**

*fit a Bayesian semi-parametric model for both linear and non-linear GxE interactions. Users can also specify all the interactions as linear and fit a Bayesian LASSO type of model.*

**Usage**

```r
BVCfit(X, Y, Z, E = NULL, clin = NULL, iterations = 10000,
      burn.in = NULL, sparse = TRUE, structural = TRUE, VC = TRUE,
      kn = 2, degree = 2, hyper = NULL, debugging = FALSE)
```

**Arguments**

- **X**
  
  the matrix of predictors (genetic factors) without intercept. Each row should be an observation vector. A column of 1 will be added to the X matrix as the intercept.
the response variable. The current version of BVCfit only supports continuous
response.

Z a vector of environmental factor for non-linear G×E interactions.

E a vector of environmental factor for linear G×E interactions.

clin a matrix of clinical variables. Clinical variables are not subject to penalty.

iterations the number of MCMC iterations.

burn.in the number of iterations for burn-in.

sparse logical flag. If TRUE, spike-and-slab priors will be used to shrink coefficients of
irrelevant covariates to zero exactly. 'sparse' has effect only when VC=TRUE.

structural logical flag. If TRUE, the coefficient functions with varying effects and con-
stant effects will be penalized separately. 'structural' has effect only when
VC=TRUE.

VC logical flag. If TRUE, varying coefficient functions will be used for modeling
the interactions between Z and X. If FALSE, interactions between Z and X will
be modeled as linear interactions.

kn the number of interior knots for B-spline.

degree the degree of B spline basis.

hyper a named list of hyperparameters.

debugging logical flag. If TRUE, progress will be output to the console and extra informa-
tion will be returned.

Details

By default, varying coefficient functions are used for modeling the nonlinear interactions between
Z and X. Assuming both E and clin are NULL, the model can be expressed as

\[ Y = \beta_0(Z) + \sum \beta_j(Z)X_j + \epsilon \]

The basis expansion and changing of basis with B splines will be done automatically:

\[ \beta_j(\cdot) \approx \gamma_{j1} + \sum_{k=2}^{q} B_{jk}(\cdot)\gamma_{jk} \]

where \( B_{jk}(\cdot) \) represents B spline basis. \( \gamma_{j1} \) and \( (\gamma_{j2}, \ldots, \gamma_{jq})^T \) correspond to the constant and
varying parts of the coefficient functional, respectively. \( q=kn+\text{degree}+1 \) is the number of basis
functions. By default, \( kn=\text{degree}=2 \). User can change the values of \( kn \) and \( \text{degree} \) to any other
positive integers. If E is provided, the linear interactions between E and X will be added modeled
as pairwise-products:

\[ Y = \beta_0(Z) + \sum \beta_j(Z)X_j + \zeta_0E + \sum \zeta_jEX_j + \epsilon \]

If clin is provided, clinical variables will be added to the model.

If VC=FALSE, all interactions are treated as linear and a Bayesian LASSO model will be used.

With non-null values of E and clin, the full linear model is:

\[ Y \sim Z + ZX + clin + E + EX \]
Please check the references for more details about the model.

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

- a.c, a.v, a.e: shape parameters of the Gamma priors on $\lambda_c$, $\lambda_v$, and $\lambda_e$, respectively.
- b.c, b.v, b.e: rate parameters of the Gamma priors on $\lambda_c$, $\lambda_v$, and $\lambda_e$, respectively.
- r.c, r.v, r.e: shape parameters of the Beta priors $(\pi_r^{-1}(1-\pi)^{w_r^{-1}})$ on $\pi_c$, $\pi_v$, and $\pi_e$, respectively.
- w.c, w.v, w.e: shape parameters of the Beta priors on $\pi_c$, $\pi_v$, and $\pi_e$, respectively.
- s: shape parameters of the Inverse-gamma prior on $\sigma^2$.
- h: scale parameters of the Inverse-gamma prior on $\sigma^2$.

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

Value

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

- posterior: posterior samples from the MCMC
- coefficients: a list of posterior estimates of coefficients
- burn.in: the number of iterations for burn-in
- iterations: the number of MCMC iterations.

References


Examples

data(gExp)

## default method
spbayes=bvcfit(X, Y, Z, E, clin)
spbayes

## non-structural
structural=FALSE
spbayes=bvcfit(X, Y, Z, E, clin, structural=structural)
spbayes

## non-sparse
sparse=FALSE
spbayes=bvcfit(X, Y, Z, E, clin, sparse=sparse)
spbayes
Variable selection for a BVCfit object

Description

Variable selection for a BVCfit object

Usage

BVSel ection(obj, ...)

## S3 method for class 'BVCSparse'
BVSel ection(obj, burn.in = obj$burn.in,
   prob = 0.95, ...)

## S3 method for class 'BVCSparse'
BVSel ection(obj, burn.in = obj$burn.in, ...)

Arguments

obj BVCfit object.

... other BVSel ection arguments

burn.in MCMC burn-in.

prob probability for credible interval, between 0 and 1. e.g. prob=0.95 leads to 95% credible interval

Details

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

Value

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

method posterior samples from the MCMC

indices a list of indices and names of selected variables

summary a summary of selected variables

References


See Also
   BVCfit

Examples
   data(gexp)
   ## sparse
   spbayes=bvcfit(X, Y, Z, E, clin)
   spbayes
   selected = BVSelection(spbayes)
   selected$indices
   
   ## non-sparse
   spbayes=bvcfit(X, Y, Z, E, clin, sparse=FALSE)
   spbayes
   selected = BVSelection(spbayes)
   selected

---

data simulated data for demonstrating the features of BVCfit

Description

Simulated gene expression data for demonstrating the features of BVCfit.

Usage

   data("gExp")
   data("gExp.new")
   data("gExp.L")

Format

  gExp consists of five components: X, Y, Z, E and clin. gExp.new contains the data of new observations (X.new, Y.new, Z.new, E.new and clin.new) which can be used for evaluating the prediction performance.
  gExp.L contains larger datasets: X2, Y2, Z2, E2 and clin2

Details

  the same true model is used for generating Y, Y.new and Y2

  \[ Y = \beta_0(Z) + \beta_1(Z)X_1 + \beta_2(Z)X_2 + 1.5X_3 - X_5 + 1.3E - 1.2EX_2 + 1.3EX_4 - clin_1 + 1.5clin_2 + \epsilon \]

  where \( \epsilon \sim N(0, 1) \), \( \beta_0 = 2 \sin(2\pi \ast Z) \), \( \beta_1 = 2 \exp(2Z - 1) \) and \( \beta_2 = -6Z(1 - Z) \)
predict.BVCfit

See Also
BVCfit

Examples

data(gExp)
dim(X)

data(gExp.L)
dim(X)

predict.BVCfit make predictions from a BVCfit object

Description
make predictions from a BVCfit object

Usage

## S3 method for class 'BVCfit'
predict(object, X.new, Z.new, E.new = NULL,
        clin.new = NULL, Y.new = NULL, ...)

## S3 method for class 'VarLin'
predict(object, X.new, Z.new, E.new, clin.new = NULL,
        Y.new = NULL, ...)

## S3 method for class 'VarOnly'
predict(object, X.new, Z.new, clin.new = NULL,
        Y.new = NULL, ...)

## S3 method for class 'LinOnly'
predict(object, X.new, Z.new, E.new = NULL,
        clin.new = NULL, Y.new = NULL, ...)

Arguments

object BVCfit object.
X.new a matrix of new values for X at which predictions are to be made.
Z.new a vector of new values for Z at which predictions are to be made.
E.new a vector of new values for E at which predictions are to be made.
clin.new a vector or matrix of new values for clin at which predictions are to be made.
Y.new a vector of the response of new observations. If provided, the prediction mean
        squared error (PMSE) will be computed based on Y.new.
... other predict arguments
Details

\(X,\text{new} (\text{clin,}\text{new})\) must have the same number of columns as \(X (\text{clin})\) used for fitting the model. If \(E\) and \(\text{clin}\) are provided when fit the model, \(E,\text{new}\) and \(\text{clin,}\text{new}\) must not be \text{NULL}, and vice versa. The predictions are made based on the posterior estimates of coefficients in the \text{BVCfit}\ object. Note that the main effects of environmental exposures \(Z\) and \(E\) are not subject to selection.

Value

an object of class "\text{BVCfit,}\text{pred}" is returned, which is a list with components:

\begin{itemize}
\item \text{pmse} \quad \text{predictions mean squared error. pmse is NULL is } Y,\text{new}=\text{NULL}.
\item \text{y,}\text{pred} \quad \text{predicted values of the new observations.}
\end{itemize}

See Also

\text{BVCfit}

Examples

\begin{verbatim}
data(gExp)
spbayes=\text{BVCfit}(X, Y, Z, E, clin)
spbayes

data(gExp,new)
pred = \text{predict}(spbayes, X,\text{new}, Z,\text{new}, E,\text{new}, \text{clin,}\text{new}, Y,\text{new})
pred$pmse
# pred$y,\text{pred}
\end{verbatim}

print.BVCfit

\textit{print a BVCfit object}

Description

Print a summary of a BVCfit object

Usage

\begin{verbatim}
## S3 method for class 'BVCfit'
print(x, digits = \text{max}(3, getOption("digits") - 3), ...)
\end{verbatim}

Arguments

\begin{itemize}
\item \text{x} \quad \text{BVCfit object.}
\item \text{digits} \quad \text{significant digits in printout.}
\item \text{...} \quad \text{other print arguments}
\end{itemize}
See Also

BVCfit

print.BVCfit.pred  print a BVCfit.pred object

Description

Print a summary of a BVCfit.pred object

Usage

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

Arguments

x  BVCfit object.
digits  significant digits in printout.
...  other print arguments

See Also

predict.BVCfit

print.BVSelection  print a BVSelection object

Description

Print a summary of a BVSelection object

Usage

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

Arguments

x  BVSelection object.
digits  significant digits in printout.
...  other print arguments

See Also

BVSelection
Index

*Topic datasets
  data, 7

*Topic models
  BVCfit, 3

*Topic overview
  spinBayes-package, 2

BVCfit, 3, 3, 7–10
BVSelection, 6, 10

clin (data), 7
clin2 (data), 7
data, 7
E (data), 7
E2 (data), 7
gExp.L (data), 7
gExp.new (data), 7

predict.BVCfit, 8, 10
predict.LinOnly (predict.BVCfit), 8
predict.VarLin (predict.BVCfit), 8
predict.VarOnly (predict.BVCfit), 8
print.BVCfit, 9
print.BVCfit.pred, 10
print.BVSelection, 10

spbayes (data), 7
spinBayes-package, 2

X (data), 7
X2 (data), 7

Y (data), 7
Y2 (data), 7

Z (data), 7
Z2 (data), 7