Package 'spinBayes'

February 10, 2025

Type Package

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

Version 0.2.2

Maintainer Jie Ren <renjie0910@gmail.com>

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. (2020) <doi:10.1002/sim.8434>). 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 (>= 4.0.0)

License GPL-2

Encoding UTF-8

LazyData true

LinkingTo Rcpp, RcppArmadillo, testthat

Imports Rcpp, splines, glmnet, utils, stats, ggplot2

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

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

RoxygenNote 7.3.2

Suggests testthat (>= 3.0.0), covr

Config/testthat/edition 3

NeedsCompilation yes

Author Jie Ren [aut, cre],
Fei Zhou [aut],
Xiaoxi Li [aut],
Cen Wu [aut],
Yu Jiang [aut]

Repository CRAN

Date/Publication 2025-02-10 07:20:02 UTC

Contents

`spinBayes-package` *spinBayes: Semi-Parametric Gene-Environment Interaction via Bayesian Variable Selection*

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. (2020) doi:10.1002/sim.8434). 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++.

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(), plot() 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.

Author(s)

Maintainer: Jie Ren <renjie0910@gmail.com>

Authors:

- Fei Zhou
- Xiaoxi Li
- Cen Wu
- Yu Jiang

References

Ren, J., Zhou, F., Li, X., Chen, Q., Zhang, H., Ma, S., Jiang, Y., Wu, C. (2020). Semiparametric Bayesian variable selection for gene-environment interactions. *Statistics in Medicine*, 39(5): 617–638. doi:10.1002/sim.8434.

Zhou, F., Ren, J., Lu, X., Ma, S., and Wu, C. (2021). Gene-Environment Interaction: A Variable Selection Perspective. *Methods in Molecular Biology*, 2212:191-223. doi:10.1007/978107160947-7_13. PMID: 33733358.

Wu, C., Jiang, Y., Ren, J., Cui, Y., 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., 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/sagmb20170008.

Jiang, Y., Huang, Y., Du, Y., Zhao, Y., Ren, J., Ma, S., Wu, C. (2017). Identification of prognostic genes and pathways in lung adenocarcinoma using a Bayesian Approach. *Cancer Informatics*, 1(7).

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., 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.

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., and Cui, Y. (2013). Boosting signals in gene–based association studies via efficient SNP selection. *Briefings in Bioinformatics*, 15(2):279–291. doi:10.1093/bib/bbs087.

Wu, C., and Cui, Y. (2013). A novel method for identifying nonlinear gene–environment interactions in case–control association studies. *Human Genetics*, 132(12):1413–1425. doi:10.1007/s004390131350z.

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

Wu, C., Li, S., and Cui, Y. (2012). Genetic Association Studies: An Information Content Perspective. *Current Genomics*, 13(7), 566–573. doi:10.2174/138920212803251382.

See Also

Useful links:

- https://github.com/jrhub/spinBayes
- Report bugs at https://github.com/jrhub/spinBayes/issues

BVCfit

`BVCfit` *fit a Semi-parametric Bayesian variable selection*

Description

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

Usage

```
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.
Y	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 constant 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 information 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. γ_{j1} and $(\gamma_{j2}, \ldots, \gamma_{jq})^\top$ correspond to the constant and varying parts of the coefficient functional, respectively. q=kn+degree+1 is the number of basis

functions. By default, kn=degree=2. User can change the values of kn and degree to any other positive integers. If E is provided, the linear interactions between E and X will be added modeled as pairwise-products:

$$Y = \beta_0(Z) + \sum \beta_j(Z)X_j + \zeta_0 E + \sum \zeta_j E X_j + \epsilon$$

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 λ_c, λ_v and λ_e, respectively.
- b.c, b.v, b.e: rate parameters of the Gamma priors on λ_c, λ_v and λ_e, respectively.
- r.c, r.v, r.e: shape parameters of the Beta priors ($\pi^{r-1}(1-\pi)^{w-1}$) on π_c, π_v and π_e, respectively.
- w.c, w.v, w.e: shape parameters of the Beta priors on π_c, π_v and π_e, respectively.
- s: shape parameters of the Inverse-gamma prior on σ^2.
- h: scale parameters of the Inverse-gamma prior on σ^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

Ren, J., Zhou, F., Li, X., Chen, Q., Zhang, H., Ma, S., Jiang, Y., Wu, C. (2020) Semiparametric Bayesian variable selection for gene-environment interactions. *Statistics in Medicine*, 39(5): 617–638 doi:10.1002/sim.8434

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
```

`BVSelection` *Variable selection for a BVCfit object*

Description

Variable selection for a BVCfit object

Usage

```
BVSelection(obj, ...)

## S3 method for class 'BVCNonSparse'
BVSelection(obj, burn.in = obj$burn.in, prob = 0.95, ...)

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

Arguments

`obj`	BVCfit object.
`...`	other BVSelection 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 'BVCNonSparse', variable selection is based on 95% credible interval. Please check the references for more details about the variable selection.

Value

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

- method: method used for identifying important effects
- indices: a list of indices and names of selected variables
- summary: a summary of selected variables

References

Ren, J., Zhou, F., Li, X., Chen, Q., Zhang, H., Ma, S., Jiang, Y., Wu, C. (2020) Semiparametric Bayesian variable selection for gene-environment interactions. *Statistics in Medicine*, 39(5): 617–638 doi:10.1002/sim.8434

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

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 $\epsilon \sim N(0,1)$, $\beta_0 = 2\sin(0.2\pi * Z)$, $\beta_1 = 2\exp(0.2Z - 1)$ and $\beta_2 = -0.6Z(1 - 0.1Z)$

See Also

BVCfit

Examples

```
data(gExp)
dim(X)

data(gExp.L)
dim(X)
```

---

plot.BVCfit *plot a BVCfit object*

Description

plot the identified varying effects

Usage

```
## S3 method for class 'BVCfit'
plot(x, prob=0.95, ...)
```

Arguments

x	BVCfit object.
prob	probability for credible interval, between 0 and 1. e.g. prob=0.95 leads to 95% credible interval
...	other plot arguments

See Also

BVCfit

Examples

```
data(gExp)
spbayes=BVCfit(X, Y, Z, E, clin)
plot(spbayes)
```

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

Value

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

`pmse`	predictions mean squared error. pmse is NULL is Y.new=NULL.
`y.pred`	predicted values of the new observations.

See Also

BVCfit

Examples

```
data(gExp)
spbayes=BVCfit(X, Y, Z, E, clin)
spbayes

data(gExp.new)
pred = predict(spbayes, X.new, Z.new, E.new, clin.new, Y.new)
pred$pmse
# pred$y.pred
```

---

print.BVCfit *print a BVCfit object*

Description

Print a summary of a BVCfit object

Usage

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

Arguments

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

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