Package 'roben'

June 1, 2020

Type Package

Title Robust Bayesian Variable Selection for Gene-Environment Interactions

Version 0.1.0

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

Maintainer Jie Ren <jieren@ksu.edu>

Description Gene-environment (G×E) interactions have important implications to elucidate the etiology of complex diseases beyond the main genetic and environmental effects. Outliers and data contamination in disease phenotypes of G×E studies have been commonly encountered, leading to the development of a broad spectrum of robust penalization methods. Nevertheless, within the Bayesian framework, the issue has not been taken care of in existing studies. We develop a robust Bayesian variable selection method for G×E interaction studies. The proposed Bayesian method can effectively accommodate heavy-tailed errors and outliers in the response variable while conducting variable selection by accounting for structural sparsity. In particular, the spike-and-slab priors have been imposed on both individual and group levels to identify important main and interaction effects. An efficient Gibbs sampler has been developed to facilitate fast computation. 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, glmnet, stats

RoxygenNote 7.1.0

Suggests testthat (>= 2.1.0), covr

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

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

NeedsCompilation yes

Repository CRAN

Date/Publication 2020-05-11 12:30:03 UTC

Archs i386, x64

R topics documented:

`roben-package` *Robust Bayesian Variable Selection for Gene-Environment Interactions*

Description

In this package, we provide a set of robust Bayesian variable selection methods tailored for interaction analysis. A Bayesian formulation of the least absolute deviation (LAD) regression has been adopted to accommodate data contamination and long-tailed distributions in the response/ phenotype. The default method (the proposed method) conducts variable selection by accounting for structural sparsity. In particular, the spike–and–slab priors are imposed on both individual and group levels to identify important main and interaction effects (bi-level/ sparse-group selection).

In addition to the default method, users can also choose different selection structures (group-level-only or individual-level-only), methods without spike–and–slab priors and non-robust methods. In total, *roben* provides 12 different methods (6 robust and 6 non-robust). Among them, robust methods with spike–and–slab priors and the robust method for bi-level selection have been developed for the first time. Please read the Details below for how to configure the method used.

Details

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

robust:	whether to use robust methods.
sparse:	whether to use the spike-and-slab priors to create sparsity.
structure:	structural identification. Three choices are available: "sparsegroup", "group" and “individual”.

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

References

Ren, J., Zhou, F., Li, X., Ma, S., Jiang, Y. and Wu, C. (2020). Robust Bayesian variable selection for gene-environment interactions.

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

Zhou, F., Ren, J., Lu, X., Ma, S. and Wu, C. (2020). Gene–Environment Interaction: a Variable Selection Perspective. Epistasis. Methods in Molecular Biology. *Humana Press* (Accepted) https://arxiv.org/abs/2003.02930

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 https://doi.org/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 https://doi.org/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 https://doi.org/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 https://doi.org/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) https://doi.org/10.1515/sagmb-2017-0008

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

See Also

roben

`data` *simulated data for demonstrating the features of roben*

Description

Simulated gene expression data for demonstrating the features of roben.

Usage

```
data("GxE_small")
data("GxE_large")
```

Format

GxE_small consists of five components: X, Y, E, clin and coeff. coeff contains the true values of parameters used for generating Y.

GxE_large contains larger datasets: X2, Y2, E2 and clin2

Details

The data model for generating Y

Use subscript i to denote the ith subject. Let $(X_i, Y_i, E_i, Clin_i)$, $(i = 1, \ldots, n)$ be independent and identically distributed random vectors. Y_i is a continuous response variable representing the disease phenotype. X_i is the p–dimensional vector of G factors. The environmental factors and clinical covariates are denoted as the k-dimensional vector E_i and the q-dimensional vector $Clin_i$, respectively. The ϵ follows some heavy-tailed distribution. Considering the following model:

$$Y_i = \alpha_0 + \sum_{t=1}^{q} \alpha_t Clin_{it} + \sum_{m=1}^{k} \theta_m E_{im} + \sum_{j=1}^{p} \gamma_j X_{ij} + \sum_{j=1}^{p} \sum_{m=1}^{k} \zeta_{jm} E_{im} X_{ij} + \epsilon_i,$$

where α_0 is the intercept; α_t's, θ_m's, γ_j's and ζ_{jm}'s are the regression coefficients for the clinical covariates, environmental factors, genetic factors and G$\times$E interactions, respectively.

Define $\beta_j = (\gamma_j, \zeta_{j1}, \ldots, \zeta_{jk})^\top \equiv (\beta_{j1}, \ldots, \beta_{jL})^\top$ and $U_{ij} = (X_{ij}, X_{ij}E_{i1} \ldots, X_{ij}E_{ik})^\top \equiv (U_{ij1}, \ldots, U_{ijL})^\top$, where $L = k + 1$. The model can be written as

$$Y_i = \alpha_0 + \sum_{t=1}^{q} \alpha_t Clin_{it} + \sum_{m=1}^{k} \theta_m E_{im} + \sum_{j=1}^{p} \left(U_{ij}^\top \beta_j\right) + \epsilon_i,$$

where the coefficient vector β_j represents all the main and interaction effects corresponding to the jth genetic measurement.

The object **coeff** in GxE_small is a list of four components, corresponding to α_0, α_t's, θ_m's and β_j's.

See Also

roben

Examples

```
data(GxE_small)
dim(X)
print(coeff)

data(GxE_large)
dim(X)
print(coeff)
```

`GxESelection` *Variable selection for a roben object*

Description

Variable selection for a roben object

Usage

```
GxESelection(obj, ...)

## S3 method for class 'Sparse'
GxESelection(obj, burn.in = obj$burn.in, ...)

## S3 method for class 'NonSparse'
GxESelection(obj, burn.in = obj$burn.in, prob = 0.95, ...)
```

Arguments

obj	roben object.
...	other GxESelection 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 '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 'GxESelection' is returned, which is a list with components:

method	method used for identifying important effects.
effects	a list of names of selected effects.
summary	a summary of selected effects.
indicator	a matrix of indicators of selected effects.

References

Ren, J., Zhou, F., Li, X., Ma, S., Jiang, Y. and Wu, C. (2020). Robust Bayesian variable selection for gene-environment interactions.

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

See Also

roben

Examples

```
data(GxE_small)
iter=5000
## sparse
fit=roben(X, Y, E, clin, iterations=iter)
selected=GxESelection(fit)
selected
```

```
## non-sparse
fit=roben(X, Y, E, clin, iterations=iter, sparse=FALSE)
selected=GxESelection(fit)
selected
```

`predict.roben` *make predictions from a roben object*

Description

make predictions from a roben object

Usage

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

Arguments

`object`	roben object.
`X.new`	a matrix of new values for X 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 error will be computed based on Y.new.
`...`	other predict arguments

Details

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

If Y.new is provided, the prediction error will be computed. For robust methods, the prediction mean absolute deviations (PMAD) will be computed. For non-robust methods, the prediction mean squared error (PMSE) will be computed.

Value

an object of class 'roben.pred' is returned, which is a list with components:

`error`	prediction error. error is NULL is Y.new=NULL.
`y.pred`	predicted values of the new observations.

See Also

`roben`

Examples

```
data(GxE_small)
test=sample((1:nrow(X)), floor(nrow(X)/5))
fit=roben(X[-test,], Y[-test,], E[-test,], clin[-test,], iterations=5000)
predict(fit, X[test,], E[test,], clin[test,], Y[test,])
```

---

print.GxESelection *print a GxESelection object*

Description

Print a summary of a GxESelection object

Usage

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

Arguments

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

See Also

GxESelection

---

print.roben *print a roben object*

Description

Print a summary of a roben object

Usage

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

Arguments

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

See Also

roben

print.roben.pred *print a roben.pred object*

Description

Print a summary of a roben.pred object

Usage

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

Arguments

x	roben.pred object.
digits	significant digits in printout.
...	other print arguments

See Also

predict.roben

roben *fit a robust Bayesian variable selection*

Description

fit a robust Bayesian variable selection model for G×E interactions.

Usage

```
roben(
  X,
  Y,
  E,
  clin = NULL,
  iterations = 10000,
  burn.in = NULL,
  robust = TRUE,
  sparse = TRUE,
  structure = c("sparsegroup", "group", "individual"),
  hyper = NULL,
  debugging = FALSE
)
```

Arguments

X	the matrix of predictors (genetic factors) without intercept. Each row should be an observation vector. X will be centered and a column of 1 will be added to the X matrix as the intercept.
Y	the response variable. The current version of roben only supports continuous response.
E	a matrix of environmental factors. E will be centered. The interaction terms between X (G factors) and E will be automatically created and included in the model.
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.
robust	logical flag. If TRUE, robust methods will be used.
sparse	logical flag. If TRUE, spike-and-slab priors will be used to shrink coefficients of irrelevant covariates to zero exactly.
structure	three choices are available. "sparsegroup" for sparse-group selection, which is a bi-level selection on both group-level and individual-level. "group" for selection on group-level only. "individual" for selection on individual-level only.
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

Consider the data model described in "data":

$$Y_i = \alpha_0 + \sum_{t=1}^{q} \alpha_t Clin_{it} + \sum_{m=1}^{k} \theta_m E_{im} + \sum_{j=1}^{p} \left(U_{ij}^{\top} \beta_j\right) + \epsilon_i,$$

where the main and interaction effects of the jth genetic variant is corresponding to the coefficient vector $\beta_j = (\beta_{j1}, \beta_{j2}, \ldots, \beta_{jL})^{\top}$.

When structure="sparsegroup" (default setting), selection will be conducted on both individual and group levels (bi-level selection):

- **Group-level selection:** by determining whether $||\beta_j||_2 = 0$, we can know if the jth genetic variant has any effect at all.
- **Individual-level selection:** investigate whether the jth genetic variant has main effect, G×E interaction or both, by determining which components in β_j has non-zero values.

If structure="group", only group-level selection will be conducted on $||\beta_j||_2$. If structure="individual", only individual-level selection will be conducted on each β_{jl}, $(l = 1, \ldots, L)$.

When sparse=TRUE (default), spike–and–slab priors are imposed on individual and/or group levels to identify important main and interaction effects. Otherwise, Laplacian shrinkage will be used.

When robust=TRUE (default), the distribution of ϵ_i is defined as a Laplace distribution with density $f(\epsilon_i|\nu) = \frac{\nu}{2} \exp\{-\nu|\epsilon_i|\}$, $(i = 1, \ldots, n)$, which leads to a Bayesian formulation of LAD regression. If robust=FALSE, ϵ_i follows a normal distribution.

Both X and E will be centered before the generation of interaction terms, in order to prevent the multicollinearity between main effects and interaction terms.

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.
- a1, b1: shape parameters of the Beta priors $(\pi^{a_1-1}(1-\pi)^{b_1-1})$ on π_1.
- c1, c2: the shape parameter and the rate parameter of the Gamma prior on ν.
- d1, d2: the shape parameter and the rate parameter of the Gamma priors on η.

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

See Also

GxESelection

Examples

```
data(GxE_small)

## default method
iter=5000
fit=roben(X, Y, E, clin, iterations = iter)
fit$coefficient

## Ture values of parameters of mian G effects and interactions
coeff$GE

## Compute TP and FP
sel=GxESelection(fit)
pos=which(sel$indicator != 0)
tp=length(intersect(which(coeff$GE != 0), pos))
fp=length(pos)-tp
list(tp=tp, fp=fp)


## alternative: robust group selection
fit=roben(X, Y, E, clin, iterations=iter, structure="g")
fit$coefficient

## alternative: non-robust sparse group selection
fit=roben(X, Y, E, clin, iterations=iter, robust=FALSE)
fit$coefficient
```

Index