Package: CovRegRF (via r-universe)

November 13, 2024

Type Package

Title Covariance Regression with Random Forests

Version 2.0.1

Description Covariance Regression with Random Forests (CovRegRF) is a random forest method for estimating the covariance matrix of a multivariate response given a set of covariates. Random forest trees are built with a new splitting rule which is designed to maximize the distance between the sample covariance matrix estimates of the child nodes. The method is described in Alakus et al. (2023) <doi:10.1186/s12859-023-05377-y>. 'CovRegRF' uses 'randomForestSRC' package (Ishwaran and Kogalur, 2022) <https://cran.r-project.org/package=randomForestSRC> by freezing at the version 3.1.0. The custom splitting rule feature is utilised to apply the proposed splitting rule. The 'randomForestSRC' package implements 'OpenMP' by default, contingent upon the support provided by the target architecture and operating system. In this package, 'LAPACK' and 'BLAS' libraries are used for matrix decompositions.

Depends R (>= 3.6.0)

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.2.0

Imports data.table, data.tree, DiagrammeR

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation yes

Author Cansu Alakus [aut, cre], Denis Larocque [aut], Aurelie Labbe [aut], Hemant Ishwaran [ctb] (Author of included 'randomForestSRC' codes), Udaya B. Kogalur [ctb] (Author of included 'randomForestSRC' codes), Intel Corporation [cph]

(Copyright holder of included LAPACKE codes), Keita Teranishi [ctb] (Author of included cblas_dgemm.c codes)

Maintainer Cansu Alakus <cansu.alakus@hec.ca>

Repository CRAN

Date/Publication 2024-07-15 18:40:11 UTC

Config/pak/sysreqs libglpk-dev make libicu-dev libxml2-dev libx11-dev

Contents

`CovRegRF-package` *CovRegRF: A package for estimating covariance matrix of a multivariate response given a set of covariates with random forests*

Description

Covariance Regression with Random Forests (CovRegRF) is a random forest method for estimating the covariance matrix of a multivariate response given a set of covariates. Random forest trees are built with a new splitting rule which is designed to maximize the distance between the sample covariance matrix estimates of the child nodes. The method is described in Alakus et al. (2023). CovRegRF uses 'randomForestSRC' package (Ishwaran and Kogalur, 2022) by freezing at the version 3.1.0. The custom splitting rule feature is utilised to apply the proposed splitting rule.

CovRegRF functions

`covregrf` `predict.covregrf` `significance.test` `vimp.covregrf` `plot.vimp.covregrf` `print.covregrf`

References

Alakus, C., Larocque, D., and Labbe, A. (2023). Covariance regression with random forests. BMC Bioinformatics 24, 258.

Ishwaran H., Kogalur U. (2022). Fast Unified Random Forests for Survival, Regression, and Classification (RF-SRC). R package version 3.1.0, https://cran.r-project.org/package=randomForestSRC.

covregrf *Covariance Regression with Random Forests*

Description

Estimates the covariance matrix of a multivariate response given a set of covariates using a random forest framework.

Usage

```
covregrf(
  formula,
  data,
 params.rfsrc = list(ntree = 1000, mtry = ceiling(px/3), nsplit = max(round(n/50),
    10)),
 nodesize.set = round(0.5^(1:100) * sampsize)[round(0.5^(1:100) * sampsize) > py],
  importance = FALSE
)
```

Arguments

formula	Object of class formula or character describing the model to fit. Interaction terms are not supported.
data	The multivariate data set which has n observations and $px + py$ variables where px and py are the number of covariates (X) and response variables (Y), respectively. Should be a data.frame.
params.rfsrc	List of parameters that should be passed to randomForestSRC. In the default parameter set, ntree = 1000, mtry = $px/3$ (rounded up), nsplit = $max(round(n/50), 10)$. See randomForestSRC for possible parameters.
nodesize.set	The set of nodesize levels for tuning. Default set includes the power of two times the sub-sample size ($.632n$) greater than the number of response variables (py). See below for details of the nodesize tuning.
importance	Should variable importance of covariates be assessed? The default is FALSE.

Value

An object of class (covregrf, grow) which is a list with the following components:

predicted.oob	OOB predicted covariance matrices for training observations.
importance	Variable importance measures (VIMP) for covariates.
best.nodesize	Best nodesize value selected with the proposed tuning method.
params.rfsrc	List of parameters that was used to fit random forest with randomForestSRC.
n	Sample size of the data (NA's are omitted).
xvar.names	A character vector of the covariate names.

yvar.names	A character vector of the response variable names.
xvar	Data frame of covariates.
yvar	Data frame of responses.
rf.grow	Fitted random forest object. This object is used for prediction with training or new data.

Details

For mean regression problems, random forests search for the optimal level of the nodesize parameter by using out-of-bag (OOB) prediction errors computed as the difference between the true responses and OOB predictions. The nodesize value having the smallest OOB prediction error is chosen. However, the covariance regression problem is unsupervised by nature. Therefore, we tune nodesize parameter with a heuristic method. We use OOB covariance matrix estimates. The general idea of the proposed tuning method is to find the nodesize level where the OOB covariance matrix predictions converge. The steps are as follows. Firstly, we train separate random forests for a set of nodesize values. Secondly, we compute the OOB covariance matrix estimates for each random forest. Next, we compute the mean absolute difference (MAD) between the upper triangular OOB covariance matrix estimates of two consecutive nodesize levels over all observations. Finally, we take the pair of nodesize levels having the smallest MAD. Among these two nodesize levels, we select the smaller since in general deeper trees are desired in random forests.

See Also

predict.covregrf significance.test vimp.covregrf print.covregrf

Examples

```
options(rf.cores=2, mc.cores=2)

## load generated example data
data(data, package = "CovRegRF")
xvar.names <- colnames(data$X)
yvar.names <- colnames(data$Y)
data1 <- data.frame(data$X, data$Y)

## define train/test split
set.seed(2345)
smp <- sample(1:nrow(data1), size = round(nrow(data1)*0.6), replace = FALSE)
traindata <- data1[smp,,drop=FALSE]
testdata <- data1[-smp, xvar.names, drop=FALSE]

## formula object
formula <- as.formula(paste(paste(yvar.names, collapse="+"), ".", sep=" ~ "))

## train covregrf
covregrf.obj <- covregrf(formula, traindata, params.rfsrc = list(ntree = 50),
  importance = TRUE)

## get the OOB predictions
pred.oob <- covregrf.obj$predicted.oob
```

```
## predict with new test data
pred.obj <- predict(covregrf.obj, newdata = testdata)
pred <- pred.obj$predicted

## get the variable importance measures
vimp <- covregrf.obj$importance
```

`data` *Generated example data*

Description

A generated data set containing two multivariate data sets: X and Y, which represent the set of covariates and responses, respectively. The covariance matrix of Y has a compound symmetry structure with heterogeneous variances. Both variances and correlations are functions of the covariates. X variables are generated from the standard normal distribution. The correlations are generated with a logit model and the variances are functions of these generated correlations. The sample size is 200. There are 3 covariates and 3 response variables. x1 and x2 are the importantvariables for the varying covariance matrix of Y. x3 is the noise variable.

Usage

```
data
```

Format

A list with two elements namely X and Y. Each element has 200 rows. X has 3 columns and Y has 3 columns.

Examples

```
## load generated example data
data(data, package = "CovRegRF")
```

`plot.vimp.covregrf` *Plot variable importance measures for covregrf objects*

Description

Plots variable importance measures (VIMP) for covariates for training data.

Usage

```
## S3 method for class 'covregrf'
plot.vimp(x, sort = TRUE, ndisp = NULL, ...)
```

Arguments

x	An object of class (covregrf, grow) or (covregrf, vimp).
sort	Should the covariates be sorted according to their variable importance measures in the plot? The default is TRUE.
ndisp	Number of covariates to display in the plot. If sort= TRUE, the most important ndisp covariates will be plotted. Otherwise, the first ndisp covariates in the original call will be plotted. The default value is NULL which will plot all covariates.
...	Optional arguments to be passed to other methods.

Value

Invisibly, the variable importance measures that were plotted.

See Also

vimp.covregrf

Examples

```
options(rf.cores=2, mc.cores=2)

## load generated example data
data(data, package = "CovRegRF")
xvar.names <- colnames(data$X)
yvar.names <- colnames(data$Y)
data1 <- data.frame(data$X, data$Y)

## define train/test split
set.seed(2345)
smp <- sample(1:nrow(data1), size = round(nrow(data1)*0.6), replace = FALSE)
traindata <- data1[smp,,drop=FALSE]
testdata <- data1[-smp, xvar.names, drop=FALSE]

## formula object
formula <- as.formula(paste(paste(yvar.names, collapse="+"), ".", sep=" ~ "))

## train covregrf
covregrf.obj <- covregrf(formula, traindata, params.rfsrc = list(ntree = 50),
  importance = TRUE)

## plot vimp
plot.vimp(covregrf.obj)
```

predict.covregrf *Predict method for covregrf objects*

Description

Obtain predicted covariance matrices using a covregrf forest for training or new data.

Usage

```
## S3 method for class 'covregrf'
predict(object, newdata, ...)
```

Arguments

object	An object of class (covregrf, grow) created by the function covregrf.
newdata	Test data of the set of covariates. A data.frame with numeric values and factors. If missing, the out-of-bag predictions in object is returned.
...	Optional arguments to be passed to other methods.

Value

An object of class (covregrf, predict) which is a list with the following components:

predicted	Predicted covariance matrices for test data. If newdata is missing, OOB predictions for training observations.
bop	Bag of Observations for Prediction. An nxn matrix of counts.
n	Sample size of the test data (NA's are omitted). If newdata is missing, sample size of the training set.
xvar.names	A character vector of the covariate names.
yvar.names	A character vector of the response variable names.

See Also

covregrf vimp.covregrf print.covregrf

Examples

```
options(rf.cores=2, mc.cores=2)

## load generated example data
data(data, package = "CovRegRF")
xvar.names <- colnames(data$X)
yvar.names <- colnames(data$Y)
data1 <- data.frame(data$X, data$Y)

## define train/test split
set.seed(2345)
```

```
smp <- sample(1:nrow(data1), size = round(nrow(data1)*0.6), replace = FALSE)
traindata <- data1[smp,,drop=FALSE]
testdata <- data1[-smp, xvar.names, drop=FALSE]

## formula object
formula <- as.formula(paste(paste(yvar.names, collapse="+"), ".", sep=" ~ "))

## train covregrf
covregrf.obj <- covregrf(formula, traindata, params.rfsrc = list(ntree = 50))

## predict without new data (OOB predictions will be returned)
pred.obj <- predict(covregrf.obj)
pred.oob <- pred.obj$predicted

## predict with new test data
pred.obj2 <- predict(covregrf.obj, newdata = testdata)
pred <- pred.obj2$predicted
```

`print.covregrf` *Print summary output of a CovRegRF analysis*

Description

Print summary output of a CovRegRF analysis. This is the default print method for the package.

Usage

```
## S3 method for class 'covregrf'
print(x, ...)
```

Arguments

`x`	An object of class (`covregrf`, `grow`), (`covregrf`, `predict`) or (`covregrf`, `significancetest`).
`...`	Optional arguments to be passed to other methods.

Value

Returns a `character` string for the summary of CovRegRF analysis.

Examples

```
options(rf.cores=2, mc.cores=2)

## load generated example data
data(data, package = "CovRegRF")
xvar.names <- colnames(data$X)
```

```
yvar.names <- colnames(data$Y)
data1 <- data.frame(data$X, data$Y)

## define train/test split
set.seed(2345)
smp <- sample(1:nrow(data1), size = round(nrow(data1)*0.6), replace = FALSE)
traindata <- data1[smp,,drop=FALSE]
testdata <- data1[-smp, xvar.names, drop=FALSE]

## formula object
formula <- as.formula(paste(paste(yvar.names, collapse="+"), ".", sep=" ~ "))

## train covregrf
covregrf.obj <- covregrf(formula, traindata, params.rfsrc = list(ntree = 50))

## print the grow object
print(covregrf.obj)

## predict with new test data
pred.obj <- predict(covregrf.obj, newdata = testdata)

## print the predict object
print(pred.obj)
```

`significance.test` *Significance test*

Description

This function runs a permutation test to evaluate the effect of a subset of covariates on the covariance matrix estimates. Returns an estimated p-value.

Usage

```
significance.test(
  formula,
  data,
 params.rfsrc = list(ntree = 1000, mtry = ceiling(px/3), nsplit = max(round(n/50),
    10)),
 nodesize.set = round(0.5^(1:100) * round(0.632 * n))[round(0.5^(1:100) * round(0.632
    * n)) > py],
  nperm = 500,
  test.vars = NULL
)
```

Arguments

`formula`	Object of class `formula` or `character` describing the model to fit. Interaction terms are not supported.

data	The multivariate data set which has n observations and $px + py$ variables where px and py are the number of covariates (X) and response variables (Y), respectively. Should be a data.frame.
params.rfsrc	List of parameters that should be passed to randomForestSRC. In the default parameter set, ntree = 1000, mtry = $px/3$ (rounded up), nsplit = $max(round(n/50), 10)$. See randomForestSRC for possible parameters.
nodesize.set	The set of nodesize levels for tuning. Default set includes the power of two times the sub-sample size ($.632n$) greater than the number of response variables (py).
nperm	Number of permutations.
test.vars	Subset of covariates whose effect on the covariance matrix estimates will be evaluated. A character vector defining the names of the covariates. The default is NULL, which tests for the global effect of the whole set of covariates.

Value

An object of class (covregrf, significancetest) which is a list with the following components:

pvalue	Estimated *p*-value, see below for details.
best.nodesize	Best nodesize value selected with the proposed tuning method using all covariates including the test.vars.
best.nodesize.control	Best nodesize value selected with the proposed tuning method using only the set of controlling covariates. If test.vars is NULL, returns NULL.
test.vars	Covariates whose effect on the covariance matrix estimates is evaluated.
control.vars	Controlling set of covariates.
predicted.oob	OOB predicted covariance matrices for training observations using all covariates including the test.vars.
predicted.perm	Predicted covariance matrices for the permutations using all covariates including the test.vars. A list of predictions for each permutation.
predicted.oob.control	OOB predicted covariance matrices for training observations using only the set of controlling covariates. If test.vars is NULL, returns NULL.
predicted.perm.control	Predicted covariance matrices for the permutations using only the set of controlling covariates. If test.vars is NULL, returns NULL.

Details

We perform a hypothesis test to evaluate the effect of a subset of covariates on the covariance matrix estimates, while controlling for the rest of the covariates. Define the conditional covariance matrix of Y given all X variables as Σ_X, and the conditional covariance matrix of Y given only the set of controlling X variables as Σ_X^c. If a subset of covariates has an effect on the covariance matrix estimates obtained with the proposed method, then Σ_X should be significantly different from Σ_X^c. We conduct a permutation test for the null hypothesis

$$H_0 : \Sigma_X = \Sigma_X^c$$

We estimate a p-value with the permutation test. If the p-value is less than the pre-specified significance level α, we reject the null hypothesis.

Testing the global effect of the covariates on the conditional covariance estimates is a particular case of the proposed significance test. Define the unconditional covariance matrix estimate of Y as Σ_{root} which is computed as the sample covariance matrix of Y, and the conditional covariance matrix of Y given X as Σ_X which is obtained with `covregrf()`. If there is a global effect of X on the covariance matrix estimates, the Σ_X should be significantly different from Σ_{root}. The null hypothesis for this particular case is

$$H_0 : \Sigma_X = \Sigma_{root}$$

See Also

`covregrf` `predict.covregrf` `print.covregrf`

`vimp.covregrf` *Variable importance for covregrf objects*

Description

Calculates variable importance measures (VIMP) for covariates for training data.

Usage

```
## S3 method for class 'covregrf'
vimp(object, ...)
```

Arguments

`object`	An object of class (covregrf, grow).
`...`	Optional arguments to be passed to other methods.

Value

An object of class `(covregrf, vimp)` which is a list with the following component:

`importance`	Variable importance measures (VIMP) for covariates.

See Also

`plot.vimp.covregrf`

Examples

```
options(rf.cores=2, mc.cores=2)

## load generated example data
data(data, package = "CovRegRF")
xvar.names <- colnames(data$X)
yvar.names <- colnames(data$Y)
data1 <- data.frame(data$X, data$Y)

## define train/test split
set.seed(2345)
smp <- sample(1:nrow(data1), size = round(nrow(data1)*0.6), replace = FALSE)
traindata <- data1[smp,,drop=FALSE]
testdata <- data1[-smp, xvar.names, drop=FALSE]

## formula object
formula <- as.formula(paste(paste(yvar.names, collapse="+"), ".", sep=" ~ "))

## train covregrf
covregrf.obj <- covregrf(formula, traindata, params.rfsrc = list(ntree = 50),
  importance = TRUE)

## get the variable importance measures
vimp <- covregrf.obj$importance
vimp2 <- vimp(covregrf.obj)$importance
```

Index