

Package: RFCCA (via r-universe)

January 21, 2025

Title Random Forest with Canonical Correlation Analysis

Version 2.0.0

Description Random Forest with Canonical Correlation Analysis (RFCCA) is a random forest method for estimating the canonical correlations between two sets of variables depending on the subject-related covariates. The trees are built with a splitting rule specifically designed to partition the data to maximize the canonical correlation heterogeneity between child nodes. The method is described in Alakus et al. (2021) <[doi:10.1093/bioinformatics/btab158](https://doi.org/10.1093/bioinformatics/btab158)>. 'RFCCA' uses 'randomForestSRC' package (Ishwaran and Kogalur, 2020) by freezing at the version 2.9.3. 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 (>= 3.5.0)

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.2.0

Imports CCA, PMA

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

URL <https://github.com/calakus/RFCCA>

BugReports <https://github.com/calakus/RFCCA/issues>

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>

Date/Publication 2024-02-09 00:10:02 UTC

Additional_repositories <https://cranhaven.r-universe.dev>

Config/pak/sysreqs make

Repository <https://cranhaven.r-universe.dev>

RemoteUrl <https://github.com/cranhaven/cranhaven.r-universe.dev>

RemoteRef package/RFCCA

RemoteSha e1d0910d3aa313c3abdee1b10ed30636aff8be3

RemoteSubdir RFCCA

Contents

| | |
|---------------------|----|
| RFCCA-package | 2 |
| data | 3 |
| global.significance | 3 |
| plot.vimp.rfcca | 6 |
| predict.rfcca | 7 |
| print.rfcca | 8 |
| rfcca | 9 |
| vimp.rfcca | 13 |

Index **15**

| | |
|---------------|--|
| RFCCA-package | <i>RFCCA: A package for computing canonical correlations depending on subject-related covariates with random forests</i> |
|---------------|--|

Description

RFCCA is a random forest method for estimating the canonical correlations between two sets of variables depending on the subject-related covariates. The trees are built with a splitting rule specifically designed to partition the data to maximize the canonical correlation heterogeneity between child nodes. RFCCA uses 'randomForestSRC' package (Ishwaran and Kogalur, 2020) by freezing at the version 2.9.3. The custom splitting rule feature is utilised to apply the proposed splitting rule. The method is described in Alakus et al. (2021).

RFCCA functions

[rfcca](#) [predict.rfcca](#) [global.significance](#) [vimp.rfcca](#) [plot.vimp.rfcca](#) [print.rfcca](#)

References

Alakus, C., Larocque, D., Jacquemont, S., Barlaam, F., Martin, C.-O., Agbogba, K., Lippe, S., and Labbe, A. (2021). Conditional canonical correlation estimation based on covariates with random forests. *Bioinformatics*, 37(17), 2714-2721.

Ishwaran, H., Kogalur, U. (2020). Fast Unified Random Forests for Survival, Regression, and Classification (RF-SRC). R package version 2.9.3, <https://cran.r-project.org/package=randomForestSRC>.

data

Generated example data

Description

A generated data set containing three sets of variables: X, Y, Z. The canonical correlation between X and Y depends on some of the Z variables. The sample size is 300. Z1-Z5 are the important variables for the varying correlation between X and Y. Z6-Z7 are the noise variables.

Usage

data

Format

A list with three elements namely X, Y, Z. Each element has 300 rows. X has 2 columns, Y has 2 columns and Z has 7 columns.

Examples

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

global.significance

Global significance test

Description

This function runs a permutation test to evaluate the global effect of subject-related covariates (Z). Returns an estimated p -value.

Usage

```
global.significance(
  X,
  Y,
  Z,
  ntree = 200,
  mtry = NULL,
  nperm = 500,
  nodesize = NULL,
  nodedepth = NULL,
  nsplit = 10,
  Xcenter = TRUE,
  Ycenter = TRUE
)
```

Arguments

| | |
|-----------|---|
| X | The first multivariate data set which has n observations and px variables. A data.frame of numeric values. |
| Y | The second multivariate data set which has n observations and py variables. A data.frame of numeric values. |
| Z | The set of subject-related covariates which has n observations and pz variables. Used in random forest growing. A data.frame with numeric values and factors. |
| ntree | Number of trees. |
| mtry | Number of z-variables randomly selected as candidates for splitting a node. The default is $pz/3$ where pz is the number of z variables. Values are always rounded up. |
| nperm | Number of permutations. |
| nodesize | Forest average number of unique data points in a terminal node. The default is the $3 * (px + py)$ where px and py are the number of x and y variables, respectively. |
| nodedepth | Maximum depth to which a tree should be grown. In the default, this parameter is ignored. |
| nsplit | Non-negative integer value for the number of random splits to consider for each candidate splitting variable. When zero or NULL, all possible splits considered. |
| Xcenter | Should the columns of X be centered? The default is TRUE. |
| Ycenter | Should the columns of Y be centered? The default is TRUE. |

Value

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

| | |
|--------|---|
| call | The original call to global.significance. |
| pvalue | p -value, see below for details. |
| n | Sample size of the data (NA's are omitted). |

| | |
|-----------------------------|---|
| <code>ntree</code> | Number of trees grown. |
| <code>nperm</code> | Number of permutations. |
| <code>mtry</code> | Number of variables randomly selected for splitting at each node. |
| <code>nodesize</code> | Minimum forest average number of unique data points in a terminal node. |
| <code>nodedepth</code> | Maximum depth to which a tree is allowed to be grown. |
| <code>nsplit</code> | Number of randomly selected split points. |
| <code>xvar</code> | Data frame of x-variables. |
| <code>xvar.names</code> | A character vector of the x-variable names. |
| <code>yvar</code> | Data frame of y-variables. |
| <code>yvar.names</code> | A character vector of the y-variable names. |
| <code>zvar</code> | Data frame of z-variables. |
| <code>zvar.names</code> | A character vector of the z-variable names. |
| <code>predicted.oob</code> | OOB predicted canonical correlations for training observations based on the selected final canonical correlation estimation method. |
| <code>predicted.perm</code> | Predicted canonical correlations for the permutations. A matrix of predictions with observations on the rows and permutations on the columns. |

Details

We perform a hypothesis test to evaluate the global effect of the subject-related covariates on distinguishing between canonical correlations. Define the unconditional canonical correlation between X and Y as $\rho_{CCA}(X, Y)$ which is found by computing CCA with all X and Y , and the conditional canonical correlation between X and Y given Z as $\rho(X, Y|Z)$ which is found by `rfcca()`. If there is a global effect of Z on correlations between X and Y , $\rho(X, Y|Z)$ should be significantly different from $\rho_{CCA}(X, Y)$. We conduct a permutation test for the null hypothesis

$$H_0 : \rho(X, Y|Z) = \rho_{CCA}(X, Y)$$

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.

See Also

[rfcca](#) [predict.rfcca](#) [print.rfcca](#)

Examples

```
## load generated example data
data(data, package = "RFCCA")
set.seed(2345)

global.significance(X = data$X, Y = data$Y, Z = data$Z, ntree = 40,
  nperm = 5)
```

plot.vimp.rfcca *Plot variable importance measures for rfcca objects*

Description

Plots variable importance measures (VIMP) for subject-related z-variables for training data.

Usage

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

Arguments

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

Value

Invisibly, the variable importance measures that were plotted.

See Also

[vimp.rfcca](#)

Examples

```
## load generated example data  
data(data, package = "RFCCA")  
set.seed(2345)  
  
## train rfcca  
rfcca.obj <- rfcca(X = data$X, Y = data$Y, Z = data$Z, ntree = 100,  
  importance = TRUE)  
  
## plot vimp  
plot.vimp(rfcca.obj)
```

| | |
|---------------|---|
| predict.rfcca | <i>Predict method for rfcca objects</i> |
|---------------|---|

Description

Obtain predicted canonical correlations using a rfcca forest for training or new data.

Usage

```
## S3 method for class 'rfcca'
predict(
  object,
  newdata,
  membership = FALSE,
  finalcca = c("cca", "scca", "rcca"),
  ...
)
```

Arguments

| | |
|------------|---|
| object | An object of class (rfcca, grow) created by the function rfcca. |
| newdata | Test data of the set of subject-related covariates (Z). A data.frame with numeric values and factors. If missing, the out-of-bag predictions in object is returned. |
| membership | Should terminal node membership information be returned? |
| finalcca | Which CCA should be used for final canonical correlation estimation? Choices are cca, scca and rcca, see rfcca for details. The default is cca. |
| ... | Optional arguments to be passed to other methods. |

Value

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

| | |
|------------|--|
| call | The original grow call to rfcca. |
| n | Sample size of the test data (NA's are omitted). If newdata is missing, sample size of the training set. |
| ntree | Number of trees grown. |
| xvar | Data frame of x-variables. |
| xvar.names | A character vector of the x-variable names. |
| yvar | Data frame of y-variables. |
| yvar.names | A character vector of the y-variable names. |
| zvar | Data frame of test z-variables. If newdata is missing, data frame of training z-variables. |
| zvar.names | A character vector of the z-variable names. |

| | |
|----------------|--|
| forest | The (rfcca, grow) forest. |
| membership | A matrix recording terminal node membership for the test data where each cell represents the node number that an observation falls in for that tree. |
| predicted | Test set predicted canonical correlations based on the selected final canonical correlation estimation method. If newdata is missing, OOB predictions for training observations. |
| predicted.coef | Predicted canonical weight vectors for x- and y- variables. |
| finalcca | The selected CCA used for final canonical correlation estimations. |

See Also

[rfcca](#) [vimp.rfcca](#) [print.rfcca](#)

Examples

```
## load generated example data
data(data, package = "RFCCA")
set.seed(2345)

## define train/test split
smp <- sample(1:nrow(data$X), size = round(nrow(data$X) * 0.7),
  replace = FALSE)
train.data <- lapply(data, function(x) {x[smp, ]})
test.Z <- data$Z[-smp, ]

## train rfcca
rfcca.obj <- rfcca(X = train.data$X, Y = train.data$Y, Z = train.data$Z,
  ntree = 100)

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

## predict with new test data
pred.obj2 <- predict(rfcca.obj, newdata = test.Z)
pred <- pred.obj2$predicted

## print predict objects
print(pred.obj)
print(pred.obj2)
```

print.rfcca

Print summary output of a RFCCA analysis

Description

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

Usage

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

Arguments

x An object of class (rfcca,grow), (rfcca,predict) or (rfcca,globalsignificance).
... Optional arguments to be passed to other methods.

Value

No return value, called for side effects.

Examples

```
## load generated example data  
data(data, package = "RFCCA")  
set.seed(2345)  
  
## train rfcca  
rfcca.obj <- rfcca(X = data$X, Y = data$Y, Z = data$Z, ntree = 100,  
                  importance = TRUE)  
  
## print the grow object  
print(rfcca.obj)
```

rfcca

Random Forest with Canonical Correlation Analysis

Description

Estimates the canonical correlations between two sets of variables depending on the subject-related covariates.

Usage

```
rfcca(  
  X,  
  Y,  
  Z,  
  ntree = 200,  
  mtry = NULL,  
  nodesize = NULL,  
  nodedepth = NULL,  
  nsplit = 10,  
  importance = FALSE,  
  finalcca = c("cca", "scca", "rfcca"),
```

```

bootstrap = TRUE,
samptype = c("swor", "swr"),
sampsize = if (samptype == "swor") function(x) {
  x * 0.632
} else function(x)
{
  x
},
forest = TRUE,
membership = FALSE,
bop = TRUE,
Xcenter = TRUE,
Ycenter = TRUE,
...
)

```

Arguments

| | |
|------------|---|
| X | The first multivariate data set which has n observations and px variables. A data.frame of numeric values. |
| Y | The second multivariate data set which has n observations and py variables. A data.frame of numeric values. |
| Z | The set of subject-related covariates which has n observations and pz variables. Used in random forest growing. A data.frame with numeric values and factors. |
| ntree | Number of trees. |
| mtry | Number of z-variables randomly selected as candidates for splitting a node. The default is $pz/3$ where pz is the number of z variables. Values are always rounded up. |
| nodesize | Forest average number of unique data points in a terminal node. The default is the $3 * (px + py)$ where px and py are the number of x and y variables, respectively. |
| nodedepth | Maximum depth to which a tree should be grown. In the default, this parameter is ignored. |
| nsplit | Non-negative integer value for the number of random splits to consider for each candidate splitting variable. When zero or NULL, all possible splits considered. |
| importance | Should variable importance of z-variables be assessed? The default is FALSE. |
| finalcca | Which CCA should be used for final canonical correlation estimation? Choices are cca, scca and rcca, see below for details. The default is cca. |
| bootstrap | Should the data be bootstrapped? The default value is TRUE which bootstraps the data by sampling without replacement. If FALSE is chosen, the data is not bootstrapped. It is not possible to return OOB predictions and variable importance measures if FALSE is chosen. |
| samptype | Type of bootstrap. Choices are swor (sampling without replacement/sub-sampling) and swr (sampling with replacement/ bootstrapping). The default action here (as in randomForestSRC) is sampling without replacement. |

| | |
|------------|--|
| sampsize | Size of sample to draw. For sampling without replacement, by default it is .632 times the sample size. For sampling with replacement, it is the sample size. |
| forest | Should the forest object be returned? It is used for prediction on new data. The default is TRUE. |
| membership | Should terminal node membership and inbag information be returned? |
| bop | Should the Bag of Observations for Prediction (BOP) for training observations be returned? The default is TRUE. |
| Xcenter | Should the columns of X be centered? The default is TRUE. |
| Ycenter | Should the columns of Y be centered? The default is TRUE. |
| ... | Optional arguments to be passed to other methods. |

Value

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

| | |
|----------------|--|
| call | The original call to rfcca. |
| n | Sample size of the data (NA's are omitted). |
| ntree | Number of trees grown. |
| mtry | Number of variables randomly selected for splitting at each node. |
| nodesize | Minimum forest average number of unique data points in a terminal node. |
| nodedepth | Maximum depth to which a tree is allowed to be grown. |
| nsplit | Number of randomly selected split points. |
| xvar | Data frame of x-variables. |
| xvar.names | A character vector of the x-variable names. |
| yvar | Data frame of y-variables. |
| yvar.names | A character vector of the y-variable names. |
| zvar | Data frame of z-variables. |
| zvar.names | A character vector of the z-variable names. |
| leaf.count | Number of terminal nodes for each tree in the forest. Vector of length ntree. |
| bootstrap | Was the data bootstrapped? |
| forest | If forest=TRUE, the rfcca forest object is returned. This object is used for prediction with new data. |
| membership | A matrix recording terminal node membership where each cell represents the node number that an observations falls in for that tree. |
| importance | Variable importance measures (VIMP) for each z-variable. |
| inbag | A matrix recording inbag membership where each cell represents whether the observation is in the bootstrap sample in the corresponding tree. |
| predicted.oob | OOB predicted canonical correlations for training observations based on the selected final canonical correlation estimation method. |
| predicted.coef | Predicted canonical weight vectors for x- and y- variables. |
| bop | If bop=TRUE, a list containing BOP for each training observation is returned. |
| finalcca | The selected CCA used for final canonical correlation estimations. |
| rfsrc.grow | An object of class (rfsrc, grow) is returned. This object is used for prediction with training or new data. |

Details

Final canonical correlation estimation: Final canonical correlation can be computed with CCA (Hotelling, 1936), Sparse CCA (Witten et al., 2009) or Regularized CCA (Vinod, 1976; Leurgans et al., 1993). If Regularized CCA will be used, λ_1 and λ_2 should be specified.

References

- Hotelling, H. (1936). Relations between two sets of variates. *Biometrika*, 28(3/4), 321–377.
- Leurgans, S. E., Moyeed, R. A., & Silverman, B. W. (1993). Canonical correlation analysis when the data are curves. *Journal of the Royal Statistical Society: Series B (Methodological)*, 55(3), 725-740.
- Vinod, H.D. (1976). Canonical ridge and econometrics of joint production. *Journal of econometrics*, 4(2), 147–166.
- Witten, D. M., Tibshirani, R., & Hastie, T. (2009). A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3), 515-534.

See Also

[predict.rfcca](#) [global.significance](#) [vimp.rfcca](#) [print.rfcca](#)

Examples

```
## load generated example data
data(data, package = "RFCCA")
set.seed(2345)

## define train/test split
smp <- sample(1:nrow(data$X), size = round(nrow(data$X) * 0.7),
  replace = FALSE)
train.data <- lapply(data, function(x) {x[smp, ]})
test.Z <- data$Z[-smp, ]

## train rfcca
rfcca.obj <- rfcca(X = train.data$X, Y = train.data$Y, Z = train.data$Z,
  ntree = 100, importance = TRUE)

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

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

## predict with new test data
pred.obj <- predict(rfcca.obj, newdata = test.Z)
pred <- pred.obj$predicted

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

```
## train rfcca and estimate the final canonical correlations with "scca"
rfcca.obj2 <- rfcca(X = train.data$X, Y = train.data$Y, Z = train.data$Z,
  ntree = 100, finalcca = "scca")
```

| | |
|------------|--|
| vimp.rfcca | <i>Variable importance for rfcca objects</i> |
|------------|--|

Description

Calculates variable importance measures (VIMP) for subject-related z-variables for training data.

Usage

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

Arguments

| | |
|--------|---|
| object | An object of class (rfcca,grow). |
| ... | Optional arguments to be passed to other methods. |

Value

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

| | |
|---------------|---|
| call | The original grow call to rfcca. |
| n | Sample size of the data (NA's are omitted). |
| ntree | Number of trees grown. |
| zvar | Data frame of z-variables. |
| zvar.names | A character vector of the z-variable names. |
| predicted.oob | OOB predicted canonical correlations for training observations based on the selected final canonical correlation estimation method. |
| finalcca | The selected CCA used for final canonical correlation estimations. |
| importance | Variable importance measures (VIMP) for each z-variable. |

See Also

[plot.vimp.rfcca](#)

Examples

```
## load generated example data
data(data, package = "RFCCA")
set.seed(2345)

## train rfcca
rfcca.obj <- rfcca(X = data$X, Y = data$Y, Z = data$Z, ntree = 100)

## get variable importance measures
vimp.obj <- vimp(rfcca.obj)
vimp.z <- vimp.obj$importance
```

Index

* datasets

data, 3

data, 3

global.significance, 2, 3, 12

plot.vimp (plot.vimp.rfcca), 6

plot.vimp.rfcca, 2, 6, 13

predict.rfcca, 2, 5, 7, 12

print.rfcca, 2, 5, 8, 8, 12

rfcca, 2, 5, 7, 8, 9

RFCCA-package, 2

vimp (vimp.rfcca), 13

vimp.rfcca, 2, 6, 8, 12, 13