Package ‘binomialRF’

July 22, 2025

Type Package

Title Binomial Random Forest Feature Selection

Version 0.1.0

URL https://www.biorxiv.org/content/10.1101/681973v1.abstract

Description The 'binomialRF' is a new feature selection technique for decision trees that aims at providing an alternative approach to identify significant feature subsets using binomial distributional assumptions (Rachid Zaim, S., et al. (2019)) <doi:10.1101/681973>. Treating each splitting variable selection as a set of exchangeable correlated Bernoulli trials, 'binomialRF' then tests whether a feature is selected more often than by random chance.

License GPL-2

Encoding UTF-8

LazyData true

biocViews Software, GenePrediction, StatisticalMethod, DecisionTree, DimensionReduction, ExperimentalDesign

Imports randomForest, data.table, stats, rlist

Suggests foreach, knitr, rmarkdown, correlbinom

RoxygenNote 7.0.2

VignetteBuilder knitr

NeedsCompilation no

Author Samir Rachid Zaim [aut, cre]

Maintainer Samir Rachid Zaim <samirrachidzaim@math.arizona.edu>

Repository CRAN

Date/Publication 2020-03-26 17:00:13 UTC

Contents

`.cv_binomialRF` *random forest feature selection based on binomial exact test*

Description

`cv.binomialRF` is the cross-validated form of the `binomialRF`, where K-fold crossvalidation is conducted to assess the feature's significance. Using the `cvFolds`=K parameter, will result in a K-fold cross-validation where the data is 'chunked' into K-equally sized groups and then the averaged result is returned.

Usage

```
.cv_binomialRF(X, y, cvFolds = 5, fdr.threshold = 0.05,
  fdr.method = "BY", ntrees = 2000, keep.both = FALSE)
```

Arguments

`X`	design matrix
`y`	class label
`cvFolds`	how many times should we perform cross-validation
`fdr.threshold`	fdr.threshold for determining which set of features are significant
`fdr.method`	how should we adjust for multiple comparisons (i.e., `p.adjust.methods` =c("holm", "hochberg", "hommel", "bonferroni", "BH", "BY","fdr", "none"))
`ntrees`	how many trees should be used to grow the `randomForest`? (Defaults to 5000)
`keep.both`	should we keep the naive binomialRF as well as the correlated adjustment

Value

a data.frame with 4 columns: Feature Name, cross-validated average for Frequency Selected, CV Median (Probability of Selecting it randomly), CV Median(Adjusted P-value based on `fdr.method`), and averaged number of times selected as signficant.

References

Zaim, SZ; Kenost, C.; Lussier, YA; Zhang, HH. binomialRF: Scalable Feature Selection and Screening for Random Forests to Identify Biomarkers and Their Interactions, bioRxiv, 2019.

Examples

```
set.seed(324)

###############################
### Generate simulation data
###############################

X = matrix(rnorm(1000), ncol=10)
trueBeta= c(rep(10,5), rep(0,5))
z = 1 + X %*% trueBeta
pr = 1/(1+exp(-z))
y = as.factor(rbinom(100,1,pr))

###############################
### Run cross-validation
###############################
```

`binomialRF` *random forest feature selection based on binomial exact test*

Description

`binomialRF` is the R implementation of the feature selection algorithm by (Zaim 2019)

Usage

```
binomialRF(X,y, fdr.threshold = .05,fdr.method = 'BY',
                        ntrees = 2000, percent_features = .5,
                        keep.both=FALSE, user_cbinom_dist=NULL,
                        sampsize=round(nrow(X)*.63))
```

Arguments

`X`	design matrix
`y`	class label
`fdr.threshold`	fdr.threshold for determining which set of features are significant
`fdr.method`	how should we adjust for multiple comparisons (i.e., `p.adjust.methods` =c("holm", "hochberg", "hommel", "bonferroni", "BH", "BY","fdr", "none"))
`ntrees`	how many trees should be used to grow the `randomForest`?
`percent_features`	what percentage of L do we subsample at each tree? Should be a proportion between (0,1)
`keep.both`	should we keep the naive binomialRF as well as the correlated adjustment
`user_cbinom_dist`	insert either a pre-specified correlated binomial distribution or calculate one via the R package `correlbinom`.
`sampsize`	how many samples should be included in each tree in the randomForest

Value

a data.frame with 4 columns: Feature Name, Frequency Selected, Probability of Selecting it randomly, Adjusted P-value based on `fdr.method`

References

Zaim, SZ; Kenost, C.; Lussier, YA; Zhang, HH. binomialRF: Scalable Feature Selection and Screening for Random Forests to Identify Biomarkers and Their Interactions, bioRxiv, 2019.

Examples

```
set.seed(324)

###############################
### Generate simulation data
###############################

X = matrix(rnorm(1000), ncol=10)
trueBeta= c(rep(10,5), rep(0,5))
z = 1 + X %*% trueBeta
pr = 1/(1+exp(-z))
y = as.factor(rbinom(100,1,pr))

###############################
### Run binomialRF
###############################
require(correlbinom)

rho = 0.33
ntrees = 250
cbinom = correlbinom(rho, successprob =  calculateBinomialP(10, .5), trials = ntrees,
                               precision = 1024, model = 'kuk')

binom.rf <-binomialRF(X,y, fdr.threshold = .05,fdr.method = 'BY',
                      ntrees = ntrees,percent_features = .5,
                      keep.both=FALSE, user_cbinom_dist=cbinom,
                      sampsize=round(nrow(X)*rho))

print(binom.rf)
```

`calculateBinomialP` *calculate the probability, p, to conduct a binomial exact test*

Description

`calculateBinomialP` returns a probability of randomly selecting a feature as the root node in a decision tree. This is a generic function that is called internally in `binomialRF` but that may also be called directly if needed. The arguments ... should be, L= Total number of features in X, and percent_features= what percent of L is subsampled in the `randomForest` call.

Usage

```
calculateBinomialP(L, percent_features)
```

Arguments

`L`	the total number of features in X. Should be a positive integer >1
`percent_features`	what percentage of L do we subsample at each tree? Should be a proportion between (0,1)

Value

If L is an integeter returns a probability value for selecting predictor Xj randomly

Examples

```
calculateBinomialP(110, .4)
calculateBinomialP(13200, .5)
```

`calculateBinomialP_Interaction` *calculate the probability, p, to conduct a binomial exact test*

Description

`calculateBinomialP_Interaction` returns a probability of randomly selecting a feature as the root node in a decision tree. This is a generic function that is called internally in `binomialRF` but that may also be called directly if needed. The arguments `...` should be, L= Total number of features in X, and percent_features= what percent of L is subsampled in the `randomForest` call.

Usage

```
calculateBinomialP_Interaction(L, percent_features, K = 2)
```

Arguments

`L`	the total number of features in X. Should be a positive integer >1
`percent_features`	what percentage of L do we subsample at each tree? Should be a proportion between (0,1)
`K`	interaction level

Value

If L is an integeter returns a probability value for selecting predictor Xj randomly

Examples

```
calculateBinomialP_Interaction(110, .4,2 )
```

`geneset_binomialRF` *random forest feature selection based on binomial exact test*

Description

`binomialRF` is the R implementation of the feature selection algorithm by (Zaim 2019)

Usage

```
geneset_binomialRF(binomialRF_object, gene_ontology, cutoff = 0.2)
```

Arguments

`binomialRF_object`	the binomialRF object output
`gene_ontology`	a two- or three-column representation of a gene ontology with gene and geneset names
`cutoff`	a real-valued number between 0 and 1, used as a p-value threshold

Value

a data.frame with 4 columns: Geneset Name, P-value, Adjusted P-value based on `fdr.method`

References

Zaim, SZ; Kenost, C.; Lussier, YA; Zhang, HH. binomialRF: Scalable Feature Selection and Screening for Random Forests to Identify Biomarkers and Their Interactions, bioRxiv, 2019.

`k_binomialRF` *random forest feature selection based on binomial exact test*

Description

`k_binomialRF` is the R implementation of the interaction feature selection algorithm by (Zaim 2019). `k_binomialRF` extends the `binomialRF` algorithm by searching for k-way interactions.

Usage

```
k_binomialRF(X, y, fdr.threshold = 0.05, fdr.method = "BY",
  ntrees = 2000, percent_features = 0.3, K = 2, cbinom_dist = NULL,
  sampsize = nrow(X) * 0.4)
```

Arguments

`X`	design matrix
`y`	class label
`fdr.threshold`	fdr.threshold for determining which set of features are significant
`fdr.method`	how should we adjust for multiple comparisons (i.e., `p.adjust.methods` =c("holm", "hochberg", "hommel", "bonferroni", "BH", "BY","fdr", "none"))
`ntrees`	how many trees should be used to grow the `randomForest`? (Defaults to 5000)
`percent_features`	what percentage of L do we subsample at each tree? Should be a proportion between (0,1)
`K`	for multi-way interactions, how deep should the interactions be?
`cbinom_dist`	user-supplied correlated binomial distribution
`sampsize`	user-supplied sample size for random forest

Value

a data.frame with 4 columns: Feature Name, Frequency Selected, Probability of Selecting it randomly, Adjusted P-value based on `fdr.method`

References

Zaim, SZ; Kenost, C.; Lussier, YA; Zhang, HH. binomialRF: Scalable Feature Selection and Screening for Random Forests to Identify Biomarkers and Their Interactions, bioRxiv, 2019.

Examples

```
set.seed(324)

###############################
### Generate simulation data
###############################

X = matrix(rnorm(1000), ncol=10)
trueBeta= c(rep(10,5), rep(0,5))
z = 1 + X %*% trueBeta
pr = 1/(1+exp(-z))
y = rbinom(100,1,pr)

###############################
### Run interaction model
###############################

require(correlbinom)

rho = 0.33
ntrees = 250
cbinom = correlbinom(rho, successprob =  calculateBinomialP_Interaction(10, .5,2),
                               trials = ntrees, precision = 1024, model = 'kuk')
```

```
k.binom.rf <-k_binomialRF(X,y, fdr.threshold = .05,fdr.method = 'BY',
                      ntrees = ntrees,percent_features = .5,
                      cbinom_dist=cbinom,
                      sampsize=round(nrow(X)*rho))
```

`pmf_list` *A prebuilt distribution for correlated binary data*

Description

This data contains probability mass functions (pmf's) for correlated binary data for various parameters. The sum of correlated exchangeable binary data is a generalization of the binomial distribution that deals with correlated trials. The correlation in decision trees occurs as the subsampling and bootstrapping step in random forests touch the same data, creating a co-dependency. This data contains some pre-calculated distributions for random forests with 500, 1000, and 2000 trees with 10, 100, and 1000 features. For more distributions, they can be calculated via the correlbinom R package.

Usage

```
pmf_list
```

Format

A list of lists

References

Witt, Gary. "A Simple Distribution for the Sum of Correlated, Exchangeable Binary Data." Communications in Statistics-Theory and Methods 43, no. 20 (2014): 4265-4280.

Index