Package ‘missCforest’

January 17, 2023

Title  Ensemble Conditional Trees for Missing Data Imputation
Version 0.0.8
License GPL (>= 3)
Encoding UTF-8

URL https://github.com/ielbadisy/missCforest
BugReports https://github.com/ielbadisy/missCforest/issues
Depends partykit
Imports stats
Suggests testthat
RoxygenNote 7.2.3
NeedsCompilation no
Author Imad El Badisy [aut, cre],
Roch Giorgi [ctb]
Maintainer Imad El Badisy <elbadisyimad@gmail.com>
Repository CRAN
Date/Publication 2023-01-17 17:30:05 UTC

R topics documented:

- generateNA ........................................ 2
- missCforest ...................................... 2

Index 5
**generateNA**  
*Generate Missing Values Completely at Random*

**Description**
Introducing a proportion of missing values given a `data.frame` under Missing Completely at Random mechanism (MCAR).

**Usage**
```r
generateNA(dat, pmiss = 0.2, seed = 123)
```

**Arguments**
- `dat`: complete `data.frame`.
- `pmiss`: proportion of NA.
- `seed`: seed value to ensure reproducibility.

**Value**
data.frame with the desired proportion of missing values.

**Details**
This function is made only for experimental purpose. Without specifying the columns (i.e. variables), missing values are introduced to all columns of the dataset.

**Examples**
```r
data(iris)
# introduce 30% of NA
irisNA <- generateNA(iris, 0.3)
# check the proportion of NA
mean(is.na(irisNA))
```

**missCforest**  
*Ensemble Conditional Trees for Missing Data Imputation*

**Description**
Single imputation based on the Ensemble Conditional Trees Cforest algorithm.
**Usage**

```r
missCforest(
  dat,  
  formula = . ~ .,
  ntree = 100L,
  minsplit = 20L,
  minbucket = 7L,
  alpha = 0.05,
  cores = 1
)
```

**Arguments**

- `dat`: data.frame containing continuous and/or categorical variables to be imputed.
- `formula`: formula description of the imputation model. Details about imputation model specification are provided below.
- `ntree`: number of trees to grow for the forest.
- `minsplit`: minimum sum of weights in a node in order to be considered for splitting in a single tree.
- `minbucket`: minimum sum of weights in a terminal node of a single tree.
- `alpha`: statistical significance level (alpha).
- `cores`: number of cores to use or in most cases how many child processes will be run simultaneously. This option is initialized at 4 to ensure fast execution.

**Value**

complete (i.e. imputed) data.frame.

**Imputation model specification**

Formula for defining the imputation model is of the form

```
[imputed_variables ~ predictors]
```

The variables to be imputed are specified on the left-side and the predictors to be used for imputation are specified on the right-side of the formula. The user can specify a customized imputation model using the formula argument. By default, latter is set to `[, ~ .]` which corresponds to the situation where all variables that contain missing values will be imputed by the rest of variables.

**Details**

missCforest can be used for numerical, categorical, or mixed-type data imputation. Missing values are imputed through ensemble prediction using Conditional Inference Trees (Ctree) as base learners (Hothorn, Hornik, and Zeileis 2006). Ctree is a non-parametric class of regression and classification trees embedding recursive partitioning into the theory of conditional inference (Strasser and Weber 1999). The missCforest algorithm redefines the imputation problem as a prediction one using single imputation approach. Iteratively, missing values are predicted based on the the complete cases set updated at each iteration. No stopping criterion is pre-defined, the imputation process ends when
the missing data are all imputed. This algorithm is robust to outliers and gives a particular attention to the association structure between covariates (i.e. variables used for imputation) and the outcome (i.e. variable to be imputed) since the recursive partitioning of Conditional Trees is based on the multiple tests procedures.

References


Examples

```r
library(missCforest)

# import the iris dataset
data(iris)

# introduce randomly 30% of NA to variables
irisNA <- generateNA(iris, 0.3)
summary(irisNA)

# impute all the missing values using all the possible combinations of the imputation model formula
irisImp <- missCforest(irisNA, .~.)
summary(irisImp)
```
Index

generateNA, 2
missCforest, 2