Package ‘RfEmpImp’

May 16, 2020

Type Package
Title Multiple Imputation using Chained Random Forests
Version 2.0.3
Maintainer Shangzhi Hong <shangzhi-hong@hotmail.com>
Description Functions for methods for multiple imputation using chained random forests. Implemented algorithms can handle missing data in both continuous and categorical variables by using prediction-based or node-based conditional distributions constructed using random forests. For prediction-based imputation, the method based on the empirical distribution of out-of-bag prediction errors of random forests and the method based on normality assumption are provided. For node-based imputation, the method based on the conditional distribution formed by predicting nodes of random forests and the method based on measures of proximities of random forests are provided. More details of the statistical methods can be found in Hong et al. (2020) <arXiv:2004.14823>.
License GPL-3
Encoding UTF-8
LazyData true
RoxygenNote 7.1.0
Depends R (>= 3.5.0), mice (>= 3.8.0), ranger (>= 0.12.1)
Suggests testthat (>= 2.1.0), knitr, rmarkdown
NeedsCompilation no
URL https://github.com/shangzhi-hong/RfEmpImp
BugReports https://github.com/shangzhi-hong/RfEmpImp/issues
VignetteBuilder knitr
Author Shangzhi Hong [aut, cre], Henry S. Lynn [ths]
Repository CRAN
Date/Publication 2020-05-16 09:30:11 UTC
R topics documented:

- `gen.mcar` ......................................................... 2
- `imp.rfemp` .......................................................... 3
- `imp.rfnode.cond` .................................................. 5
- `imp.rfnode.prox` .................................................. 6
- `mice.impute.rfemp` ............................................... 8
- `mice.impute.rfnode` .............................................. 11
- `mice.impute.rfpred.emp` ......................................... 14
- `mice.impute.rfpred.norm` ........................................ 16
- `reg.ests` ............................................................... 17

Index

---

<table>
<thead>
<tr>
<th>gen.mcar</th>
<th>Generate missing (completely at random) cells in the dataset</th>
</tr>
</thead>
</table>

Description

Generate missing (completely at random) cells in the dataset

Usage

```
gen.mcar(df, prop.na = 0.2, warn.empty.row = TRUE, ...)
```

Arguments

- `df` Input data frame.
- `prop.na` Proportion of missing cells, default = 0.2.
- `warn.empty.row` Show warnings if empty rows were present.
- `...` Other parameters (will be ignored).

Value

A data frame containing generated missing values.

Examples

```
data("mtcars")
mtcars.mcar <- gen.mcar(mtcars, warn.empty.row = FALSE)
```
**Description**

RfEmp multiple imputation method is for mixed types of variables, and calls corresponding functions based on variable types. Categorical variables should be of type factor or logical. RfPred.Emp is used for continuous variables, and RfPred.Cate is used for categorical variables.

**Usage**

```r
imp.rfemp(
  data, 
  num.imp = 5, 
  max.iter = 5, 
  num.trees = 10, 
  alpha.emp = 0, 
  sym.dist = TRUE, 
  pre.boot = TRUE, 
  num.trees.cont = NULL, 
  num.trees.cate = NULL, 
  num.threads = NULL, 
  print.flag = FALSE, 
  ...
)
```

**Arguments**

- **data**: A data frame or a matrix containing the incomplete data. Missing values should be coded as NAs.
- **num.imp**: Number of multiple imputations. The default is `num.imp = 5`.
- **max.iter**: Number of iterations. The default is `max.iter = 5`.
- **num.trees**: Number of trees to build. The default is `num.trees = 10`.
- **alpha.emp**: The "significance level" for the empirical distribution of out-of-bag prediction errors, can be used for prevention for outliers (helpful for highly skewed variables). For example, set `alpha = 0.05` to use 95% confidence level. The default is `alpha.emp = 0.0`, and the empirical distribution of out-of-bag prediction errors will be kept intact.
- **sym.dist**: If TRUE, the empirical distribution of out-of-bag prediction errors will be assumed to be symmetric; if FALSE, the empirical distribution will be kept intact. The default is `sym.dist = TRUE`.
- **pre.boot**: If TRUE, bootstrapping prior to imputation will be performed to perform 'proper' multiple imputation, for accommodating sampling variation in estimating population regression parameters (refer to Shah et al. 2014). It should be noted that if TRUE, this option is valid even if the number of trees is set to one.
imp.rfemp

num.trees.cont Number of trees to build for continuous variables. The default is num.trees.cont = NULL and the value of num.trees will be used.

num.trees.cate Number of trees to build for categorical variables. The default is num.trees.cate = NULL and the value of num.trees will be used.

num.threads Number of threads for parallel computing. The default is num.threads = NULL and all the processors available can be used.

print.flag If TRUE, details will be sent to console. The default is print.flag = FALSE.

... Other arguments to pass down.

Details

For continuous variables, mice.impute.rfpred.emp is called, performing imputation based on the empirical distribution of out-of-bag prediction errors of random forests.

For categorical variables, mice.impute.rfpred.cate is called, performing imputation based on predicted probabilities.

Value

An object of S3 class mids.

Author(s)

Shangzhi Hong

References


Examples

# Prepare data: convert categorical variables to factors
nhanes.fix <- nhanes
nhanes.fix[, c("age", "hyp")] <- lapply(nhanes[, c("age", "hyp")], as.factor)

# Perform imputation using imp.rfemp
imp <- imp.rfemp(nhanes.fix)

# Do repeated analyses
anl <- with(imp, lm(chl ~ bmi + hyp))

# Pool the results
pool <- pool(anl)

# Get pooled estimates
imp.rfnode.cond

reg.ests(pool)

---

imp.rfnode.cond | Perform multiple imputation based on the conditional distribution formed by prediction nodes of random forests

**Description**

RfEmpImp multiple imputation method for mixed types of variables, using conditional distribution formed by predicting nodes of random forest (out-of-bag observations will be excluded).

**Usage**

```r
imp.rfnode.cond(
  data,
  num.imp = 5,
  max.iter = 5,
  num.trees = 10,
  pre.boot = TRUE,
  print.flag = FALSE,
  ...
)
```

**Arguments**

- `data` A data frame or a matrix containing the incomplete data. Missing values should be coded as NAs.
- `num.imp` Number of multiple imputations. The default is `num.imp = 5`.
- `max.iter` Number of iterations. The default is `max.iter = 5`.
- `num.trees` Number of trees to build. The default is `num.trees = 10`.
- `pre.boot` If TRUE, bootstrapping prior to imputation will be performed to perform 'proper' multiple imputation, for accommodating sampling variation in estimating population regression parameters (see Shah et al. 2014). It should be noted that if TRUE, this option is in effect even if the number of trees is set to one.
- `print.flag` If TRUE, details will be sent to console. The default is `print.flag = FALSE`.
- `...` Other arguments to pass down.

**Details**

imp.rfnode.cond multiple imputation, for missing observations, the non-missing observations used for imputation will be found by the observations included in the predicting nodes in the random trees.

**Value**

An object of S3 class mids.
imp.rfnode.prox

Multiple imputation using chained random forests and node proximities

Description

RfNodeProx multiple imputation method is for mixed types of variables, using conditional distributions formed by proximity measures of random forests (both in-bag and out-of-bag observations will be included).

Usage

imp.rfnode.prox(
  data,
  num.imp = 5,
  max.iter = 5,
  num.trees = 10,
Arguments

- **data**: A data frame or a matrix containing the incomplete data. Missing values should be coded as NAs.
- **num.imp**: Number of multiple imputations. The default is `num.imp = 5`.
- **max.iter**: Number of iterations. The default is `max.iter = 5`.
- **num.trees**: Number of trees to build. The default is `num.trees = 10`.
- **pre.boot**: If `TRUE`, bootstrapping prior to imputation will be performed to perform 'proper' multiple imputation, for accommodating sampling variation in estimating population regression parameters (see Shah et al. 2014). It should be noted that if `TRUE`, this option is in effect even if the number of trees is set to one.
- **print.flag**: If `TRUE`, details will be sent to console. The default is `print.flag = FALSE`.
- **...**: Other arguments to pass down.

Details

`imp.rfnode.prox` multiple imputation, for missing observations, the non-missing observations used for imputation will be found by whether two observations can be retrieved from the same predicting node, the observations used for imputation may not be necessarily be contained in the node.

Value

An object of S3 class `mids`.

Author(s)

Shangzhi Hong

References

Examples

# Prepare data: convert categorical variables to factors
nhanes.fix <- nhanes
nhanes.fix[, c("age", "hyp")]
  <- lapply(nhanes[, c("age", "hyp")], as.factor)
# Perform imputation using imp.rfnode.prox
imp <- imp.rfnode.prox(nhanes.fix)
# Do repeated analyses
anl <- with(imp, lm(chl ~ bmi + hyp))
# Pool the results
pool <- pool(anl)
# Get pooled estimates
reg.ests(pool)

mice.impute.rfemp

Multiple imputation for categorical variables based on predictions of random forest

Description

Please note that functions with names starting with "mice.impute" are exported to be visible for the mice sampler functions. Please do not call these functions directly unless you know exactly what you are doing.

Part of project RfEmpImp, the function mice.impute.rfpred.cate is for mixed categorical variables, performing imputation based on predicted probabilities.

Please note that functions with names starting with "mice.impute" are exported to be visible for the mice sampler functions. Please do not call these functions directly unless you know exactly what you are doing.

RfEmpImp multiple imputation method, adapter for mice samplers. These functions can be called by the mice sampler function. In the mice() function, set method = "rfemp" to call it.

mice.impute.rfemp is for mixed types of variables, and calls corresponding functions based on variable types. Categorical variables should be of type factor or logical.

For continuous variables, mice.impute.rfpred.emp is called, performing imputation based on the empirical distribution of out-of-bag prediction errors of random forests.

For categorical variables, mice.impute.rfpred.cate is called, performing imputation based on predicted probabilities.

Usage

mice.impute.rfpred.cate(
  y,
  ry,
  x,
  wy = NULL,
  num.trees.cate = 10,
mice.impute.rfemp

```r
use.pred.prob.cate = TRUE,
pre.boot = TRUE,
num.threads = NULL,
... )
```

```r
mice.impute.rfemp(
  y,
  ry,
  x,
  wy = NULL,
  num.trees = 10,
  alpha.emp = 0,
  sym.dist = TRUE,
  pre.boot = TRUE,
  num.trees.cont = NULL,
  num.trees.cate = NULL,
  ... )
```

**Arguments**

`y` Vector to be imputed.

`ry` Logical vector of length `length(y)` indicating the subset `y[ry]` of elements in `y` to which the imputation model is fitted. The `ry` generally distinguishes the observed (TRUE) and missing values (FALSE) in `y`.

`x` Numeric design matrix with `length(y)` rows with predictors for `y`. Matrix `x` may have no missing values.

`wy` Logical vector of length `length(y)`. A TRUE value indicates locations in `y` for which imputations are created.

`num.trees.cate` Number of trees to build for categorical variables, default to NULL to use the value of `num.trees`.

`use.pred.prob.cate` Logical, TRUE for assigning categories based on predicted probabilities, FALSE for imputation based on majority votes, default to TRUE.

`pre.boot` Perform bootstrap prior to imputation to get 'proper' multiple imputation, i.e. accommodating sampling variation in estimating population regression parameters (see Shah et al. 2014). It should be noted that if TRUE, this option is in effect even if the number of trees is set to one.

`num.threads` Number of threads for parallel computing. The default is `num.threads = NULL` and all the processors available can be used.

`...` Other arguments to pass down.

`num.trees` Number of trees to build, default to 10.

`alpha.emp` The "significance level" for empirical distribution of prediction errors, can be used for prevention for outliers (useful for highly skewed variables). For example, set `alpha = 0.05` to use 95% confidence level for empirical distribution.
of prediction errors. Default is 0, and the empirical error distribution is kept intact.

**sym.dist**
If TRUE, the empirical distribution of out-of-bag prediction errors will be assumed to be symmetric; if FALSE, the empirical distribution will be kept intact. The default is sym.dist = TRUE. This option is invalid when emp.err.cont is set to FALSE.

**num.trees.cont**
Number of trees to build for continuous variables, default to NULL to use the value of num.trees.

**Details**
RfEmpImp Imputation sampler for: continuous variables based on probability machines
RfEmpImp imputation sampler, the mice.impute.rfemp calls mice.impute.rfpred.emp if the variable is numeric, otherwise it calls mice.impute.rfpred.cate.

**Value**
Vector with imputed data, same type as y, and of length sum(wy)
Vector with imputed data, same type as y, and of length sum(wy).

**Author(s)**
Shangzhi Hong
Shangzhi Hong

**References**
Examples

```r
# Prepare data
mtcars.catmcar <- mtcars
mtcars.catmcar[, c("gear", "carb")]
  <- gen.mcar(mtcars.catmcar[, c("gear", "carb")], warn.empty.row = FALSE)
mtcars.catmcar[, c("gear", "carb")]
  <- lapply(mtcars.catmcar[, c("gear", "carb")], as.factor)
# Perform imputation
impObj <- mice(mtcars.catmcar, method = "rfpred.cate", m = 5,
  maxit = 5, maxcor = 1.0,
  eps = .Machine$double.xmin,
  printFlag = FALSE)
# Prepare data: convert categorical variables to factors
nhanes.fix <- nhanes
nhanes.fix[, c("age", "hyp")]
  <- lapply(nhanes[, c("age", "hyp")], as.factor)
# This function is exported to be visible to the mice sampler functions, and
# users can set method = "rfemp" in call to mice to use this function.
# Users are recommended to use the imp.rfemp function instead:
impObj <- mice(nhanes.fix, method = "rfemp", m = 5,
  maxit = 5, maxcor = 1.0, eps = .Machine$double.xmin, printFlag = FALSE)
```

mice.impute.rfnode

Sampling function for multiple imputation based on predicting nodes
of random forests

Description

Please note that functions with names starting with "mice.impute" are exported to be visible for the
mice sampler functions. Please do not call these functions directly unless you know exactly what
you are doing.

RfNode imputation methods, adapter for mice samplers. These functions can be called by the mice
sampler functions.
mice.impute.rfnode.cond is for imputation using the conditional formed by the predicting nodes
of random forests. To use this function, set method = "rfnode.cond" in mice function.
mice.impute.rfnode.prox is for imputation based on proximity measures from random forests,
and provides functionality similar to mice.impute.rf. To use this function, set method = "rfnode.prox"
in mice function.
mice.impute.rfnode is the main function for performing imputation, and both mice.impute.rfnode.cond
and mice.impute.rfnode.prox call this function. By default, mice.impute.rfnode works like
mice.impute.rfnode.cond.

Usage

```r
mice.impute.rfnode(
  y,
  ry,
```
mice.impute.rfnode

x,
w y = NULL,
num.trees.node = 10,
 pre.boot = TRUE,
 use.node.cond.dist = TRUE,
 obs.eq.prob = FALSE,
 do.sample = TRUE,
 num.threads = NULL,
...
}

mice.impute.rfnode.cond(
y,
ry,
x,
w y = NULL,
num.trees = 10,
 pre.boot = TRUE,
 obs.eq.prob = FALSE,
...
)

mice.impute.rfnode.prox(
y,
ry,
x,
w y = NULL,
num.trees = 10,
 pre.boot = TRUE,
 obs.eq.prob = FALSE,
...
)

Arguments

y Vector to be imputed

ry Logical vector of length length(y) indicating the subset y[ry] of elements in y to which the imputation model is fitted. The ry generally distinguishes the observed (TRUE) and missing values (FALSE) in y.

x Numeric design matrix with length(y) rows with predictors for y. Matrix x may have no missing values.

wy Logical vector of length length(y). A TRUE value indicates locations in y for which imputations are created.

num.trees.node Number of trees to build, default to 10. For function mice.impute.rfnode only.

pre.boot Perform bootstrap prior to imputation to get 'proper' imputation, i.e. accommodating sampling variation in estimating population regression parameters (see Shah et al. 2014).
use.node.cond.dist
If TRUE, use conditional distribution formed by predicting nodes of random forest (out-of-bag observations were excluded); if FALSE, use proximity-based imputation.

obs.eq.prob
If TRUE, the candidate observations will be sampled with equal probability.

do.sample
If TRUE, draw samples for missing observations; if FALSE, the corresponding observations numbers will be returned. For testing purposes, and WILL CAUSE ERRORS for the mice sampler function.

num.threads
Number of threads for parallel computing. The default is num.threads = NULL and all the processors available can be used.

num.trees
Number of trees to build, default to 10.

Details
Users can get more flexibility from mice.impute.rfnode function.

Value
Vector with imputed data, same type as y, and of length sum(wy)

Author(s)
Shangzhi Hong

References


Examples
# Prepare data: convert categorical variables to factors
nhanes.fix <- nhanes
nhanes.fix[, c("age", "hyp")]
<- lapply(nhanes[, c("age", "hyp")], as.factor)

# Using "rfnode.cond" or "rfnode"
impRfNodeCond <- mice(nhanes.fix, method = "rfnode.cond", m = 5,
maxit = 5, maxcor = 1.0, printFlag = FALSE)

# Using "rfnode.prox"
impRfNodeProx <- mice(nhanes.fix, method = "rfnode.prox", m = 5,
maxit = 5, maxcor = 1.0, printFlag = FALSE)
mice.impute.rfpred.emp

Multiple imputation using chained random forests: RfPred.Emp

Description

Please note that functions with names starting with "mice.impute" are exported to be visible for the mice sampler functions. Please do not call these functions directly unless you know exactly what you are doing.

For continuous variables only.

This function is for RfPred.Emp multiple imputation method, adapter for mice samplers. In the mice() function, set method = "rfpred.emp" to call it.

The function performs multiple imputation based on the empirical distribution of out-of-bag prediction errors of random forests.

Usage

mice.impute.rfpred.emp(
  y,
  ry,
  x,
  wy = NULL,
  num.trees.cont = 10,
  sym.dist = TRUE,
  emp.err.cont = TRUE,
  alpha.emp = 0,
  pre.boot = TRUE,
  num.threads = NULL,
  ...
)

Arguments

y Vector to be imputed.
ry Logical vector of length \text{length}(y) indicating the the subset y[\text{ry}] of elements in y to which the imputation model is fitted. The ry generally distinguishes the observed (TRUE) and missing values (FALSE) in y.

x Numeric design matrix with \text{length}(y) rows with predictors for y. Matrix x may have no missing values.

wy Logical vector of length \text{length}(y). A TRUE value indicates locations in y for which imputations are created.

num.trees.cont Number of trees to build for continuous variables. The default is \text{num.trees} = 10.
sym.dist  If TRUE, the empirical distribution of out-of-bag prediction errors will be assumed to be symmetric; if FALSE, the empirical distribution will be kept intact. The default is sym.dist = TRUE. This option is invalid when emp.err.cont = FALSE.

desc(emp.err.cont)  If TRUE, the empirical distribution of out-of-bag prediction errors will be used for constructing conditional distributions; if FALSE, for normality will be assumed for the distribution for the prediction errors, the variance estimate equals to overall out-of-bag prediction error, i.e. out-of-bag mean squared error (see Shah et al. 2014).

desc(alpha.emp)  The "significance level" for the empirical distribution of out-of-bag prediction errors, can be used for prevention for outliers (useful for highly skewed variables). For example, set alpha = 0.05 to use 95% confidence level. The default is alpha.emp = 0.0, and the empirical distribution of out-of-bag prediction errors will be kept intact. This option is invalid when emp.err.cont = FALSE.

desc(pre.boot)  If TRUE, bootstrapping prior to imputation will be performed to perform 'proper' multiple imputation, for accommodating sampling variation in estimating population regression parameters (see Shah et al. 2014). It should be noted that if TRUE, this option is in effect even if the number of trees is set to one.

desc(num.threads)  Number of threads for parallel computing. The default is num.threads = NULL and all the processors available can be used.

desc(...)  Other arguments to pass down.

desc(num.trees)  Number of trees to build. The default is num.trees = 10.

Details

RfPred.Emp imputation sampler.

Value

Vector with imputed data, same type as y, and of length sum(wy).

Author(s)

Shangzhi Hong

References


Examples

# Users can set method = "rfpred.emp" in call to mice to use this method
data("airquality")
impObj <- mice(airquality, method = "rfpred.emp", m = 5,
maxit = 5, maxcor = 1.0, eps = .Machine$double.xmin, printFlag = FALSE)

mice.impute.rfpred.norm

Multiple imputation using chained random forests: RfPred.Norm

Description

Please note that functions with names starting with "mice.impute" are exported to be visible for the
mice sampler functions. Please do not call these functions directly unless you know exactly what
you are doing.

For continuous variables only.

This function is for RfPred.Norm multiple imputation method, adapter for mice samplers. In the
mice() function, set method = "rfpred.norm" to call it.

The function performs multiple imputation based on normality assumption using out-of-bag mean
squared error as the estimate for variance.

Usage

mice.impute.rfpred.norm(
  y,
  ry,
  x,
  wy = NULL,
  num.trees.cont = 10,
  pre.boot = TRUE,
  num.threads = NULL,
  ...
)

Arguments

y Vector to be imputed.
ry Logical vector of length length(y) indicating the the subset y[ry] of elements
    in y to which the imputation model is fitted. The ry generally distinguishes the
    observed (TRUE) and missing values (FALSE) in y.

x Numeric design matrix with length(y) rows with predictors for y. Matrix x
    may have no missing values.

wy Logical vector of length length(y). A TRUE value indicates locations in y for
    which imputations are created.
reg.ests

num.trees.cont  Number of trees to build for continuous variables. The default is num.trees = 10.

pre.boot  If TRUE, bootstrapping prior to imputation will be performed to perform 'proper' multiple imputation, for accommodating sampling variation in estimating population regression parameters (see Shah et al. 2014). It should be noted that if TRUE, this option is in effect even if the number of trees is set to one.

num.threads  Number of threads for parallel computing. The default is num.threads = NULL and all the processors available can be used.

...  Other arguments to pass down.

Details

RfPred.Norm imputation sampler.

Value

Vector with imputed data, same type as y, and of length \( \sum(\text{wy}) \).

Author(s)

Shangzhi Hong

References


Examples

# Users can set method = "rfpred.norm" in call to mice to use this method
data("airquality")
impObj <- mice(airquality, method = "rfpred.norm", m = 5,
maxit = 5, maxcor = 1.0, eps = .Machine$double.xmin, printFlag = FALSE)

reg.ests  Get regression estimates for pooled object

Description

Get the estimates with corresponding confidence intervals after pooling.

Usage

reg.ests(obj, ...)
Arguments

obj Pooled object.

Other parameters to pass down.

Value

A data frame containing estimates and confidence intervals.
Index

gen.mcar, 2
imp.rfemp, 3
imp.rfnode.cond, 5
imp.rfnode.prox, 6
mice.impute_rfemp, 8
mice.impute_rfnode, 11
mice.impute_rfpred.cate
  (mice.impute_rfemp), 8
mice.impute_rfpred.emp, 14
mice.impute_rfpred.norm, 16
reg.ests, 17