Package 'RfEmpImp'

June 25, 2020

Type Package

Title Multiple Imputation using Chained Random Forests

Version 2.1.5

Maintainer Shangzhi Hong <shangzhi-hong@hotmail.com>

Description An R package for methods of multiple imputation using chained random forests. Implemented methods can handle missing data in mixed types of 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 continuous variables. And the method based on predicted probabilities is provided for categorical variables. For node-based imputation, the method based on the conditional distribution formed by the predicting nodes of random forests, and the method based on proximity measures 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 (>= 3.9.0), 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 2020-06-25 05:00:02 UTC

R topics documented:

`conv.factor` *Convert variables to factors*

Description

Convert variables to factors

Usage

```
conv.factor(data, convNames = NULL, exceptNames = NULL, uniqueNum = 5)
```

Arguments

`data`	Input data frame.
`convNames`	Names of variable to convert, the default is `convNames = NULL`.
`exceptNames`	Names of variables to be excluded from conversion, the default is `convNames = NULL`.
`uniqueNum`	Variables of less than or equal to a specific number of unique values in the to be converted to factors, the default is `uniqueNum = 5`.

Value

A data frame of converted variables.

Examples

```
nhanes.fix <- conv.factor(data = nhanes, convNames = c("age", "hyp"))
```

gen.mcar — *Generate missing (completely at random) cells in a data set*

Description

Generate missing (completely at random) cells in a data set

Usage

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

Arguments

df	Input data frame or matrix.
prop.na	Proportion of generated missing cells. The default is prop.na = 0.2.
warn.empty.row	Show a warning if empty rows were present in the output data set.
...	Other parameters (will be ignored).

Value

A data frame or matrix containing generated missing cells.

Author(s)

Shangzhi Hong

Examples

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

imp.rfemp — *Perform multiple imputation using the empirical error distributions and predicted probabilities of random forests*

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, etc.

RfPred.Emp is used for continuous variables, and RfPred.Cate is used for categorical variables.

Usage

```
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.
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

Hong, Shangzhi, et al. "Multiple imputation using chained random forests." Preprint, submitted April 30, 2020. https://arxiv.org/abs/2004.14823.

Zhang, Haozhe, et al. "Random Forest Prediction Intervals." The American Statistician (2019): 1-20.

Shah, Anoop D., et al. "Comparison of random forest and parametric imputation models for imputing missing data using MICE: a CALIBER study." American journal of epidemiology 179.6 (2014): 764-774.

Malley, James D., et al. "Probability machines." Methods of information in medicine 51.01 (2012): 74-81.

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
reg.ests(pool)
```

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

Description

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

Usage

```
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 `NA`s.
`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

During imputation using `imp.rfnode.cond`, for missing observations, the candidate non-missing observations will be found by the predicting nodes of random trees in the random forest model. Only the in-bag observations for each random tree will be used for imputation.

Value

An object of S3 class `mids`.

Author(s)

Shangzhi Hong

References

Hong, Shangzhi, et al. "Multiple imputation using chained random forests." Preprint, submitted April 30, 2020. https://arxiv.org/abs/2004.14823.

Zhang, Haozhe, et al. "Random Forest Prediction Intervals." The American Statistician (2019): 1-20.

Shah, Anoop D., et al. "Comparison of random forest and parametric imputation models for imputing missing data using MICE: a CALIBER study." American journal of epidemiology 179.6 (2014): 764-774.

Malley, James D., et al. "Probability machines." Methods of information in medicine 51.01 (2012): 74-81.

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.cond
imp <- imp.rfnode.cond(nhanes.fix)
# Do repeated analyses
anl <- with(imp, lm(chl ~ bmi + hyp))
# Pool the results
pool <- pool(anl)
# Get pooled estimates
reg.ests(pool)
```

---

imp.rfnode.prox *Perform multiple imputation based on the conditional distribution formed using node proximity*

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 used for imputation).

Usage

```
imp.rfnode.prox(
  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

During imputation using imp.rfnode.prox, for missing observations, the candidate non-missing observations will be found by whether two observations can be retrieved from the same predicting node during prediction. The observations used for imputation may not be necessarily be contained in the terminal node of random forest model.

Value

An object of S3 class mids.

Author(s)

Shangzhi Hong

References

Hong, Shangzhi, et al. "Multiple imputation using chained random forests." Preprint, submitted April 30, 2020. https://arxiv.org/abs/2004.14823.

Zhang, Haozhe, et al. "Random Forest Prediction Intervals." The American Statistician (2019): 1-20.

Shah, Anoop D., et al. "Comparison of random forest and parametric imputation models for imputing missing data using MICE: a CALIBER study." American journal of epidemiology 179.6 (2014): 764-774.

Malley, James D., et al. "Probability machines." Methods of information in medicine 51.01 (2012): 74-81.

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` *Univariate sampler function for mixed types of variables for prediction-based imputation, using empirical distribution of out-of-bag prediction errors and predicted probabilities 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.

`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 use the `RfEmp` method.

`mice.impute.rfemp` is for mixed types of variables, and it calls corresponding functions according to variable types. Categorical variables should be of type `factor` or `logical` etc.

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.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 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	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 0.0, 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.
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.trees.cont	Number of trees to build for continuous variables, default to NULL to use the value of num.trees.
num.trees.cate	Number of trees to build for categorical variables, default to NULL to use the value of num.trees.
...	Other arguments to pass down.

Details

RfEmpImp imputation sampler, the mice.impute.rfemp calls mice.impute.rfpred.emp if the variable is.numeric is TRUE, otherwise it calls mice.impute.rfpred.cate.

Value

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

Author(s)

Shangzhi Hong

References

Hong, Shangzhi, et al. "Multiple imputation using chained random forests." Preprint, submitted April 30, 2020. https://arxiv.org/abs/2004.14823.

Zhang, Haozhe, et al. "Random Forest Prediction Intervals." The American Statistician (2019): 1-20.

Shah, Anoop D., et al. "Comparison of random forest and parametric imputation models for imputing missing data using MICE: a CALIBER study." American journal of epidemiology 179.6 (2014): 764-774.

Malley, James D., et al. "Probability machines." Methods of information in medicine 51.01 (2012): 74-81.

Examples

```
# Prepare data: convert categorical variables to factors
nhanes.fix <- conv.factor(nhanes, c("age", "hyp"))

# 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 = 0,
remove.collinear = FALSE, remove.constant = FALSE,
printFlag = FALSE
)
```

`mice.impute.rfnode` *Univariate sampler function for mixed types of variables for node-based imputation, using 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

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

mice.impute.rfnode.prox(
  y,
  ry,
  x,
  wy = 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 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 only, 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.
...	Other arguments to pass down.
num.trees	Number of trees to build, default to 10.

Details

Advanced users can get more flexibility from mice.impute.rfnode function, as it provides more options than mice.impute.rfnode.cond or mice.impute.rfnode.prox.

Value

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

Author(s)

Shangzhi Hong

References

Hong, Shangzhi, et al. "Multiple imputation using chained random forests." Preprint, submitted April 30, 2020. https://arxiv.org/abs/2004.14823.

Doove, Lisa L., Stef Van Buuren, and Elise Dusseldorp. "Recursive partitioning for missing data imputation in the presence of interaction effects." Computational Statistics & Data Analysis 72 (2014): 92-104.

Examples

```
# Prepare data: convert categorical variables to factors
nhanes.fix <- conv.factor(nhanes, c("age", "hyp"))

# Using "rfnode.cond" or "rfnode"
impRfNodeCond <- mice(nhanes.fix, method = "rfnode.cond", m = 5,
maxit = 5, maxcor = 1.0, eps = 0, printFlag = FALSE)

# Using "rfnode.prox"
```

```
impRfNodeProx <- mice(nhanes.fix, method = "rfnode.prox", m = 5,
maxit = 5, maxcor = 1.0, eps = 0,
remove.collinear = FALSE, remove.constant = FALSE,
printFlag = FALSE)
```

`mice.impute.rfpred.cate`

Univariate sampler function for categorical variables for prediction-based imputation, using predicted probabilities 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.

For categorical variables only.

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

Usage

```
mice.impute.rfpred.cate(
  y,
  ry,
  x,
  wy = NULL,
  num.trees.cate = 10,
  use.pred.prob.cate = TRUE,
  forest.vote.cate = FALSE,
  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.
`num.trees.cate`	Number of trees to build for categorical variables, default to `10`.

`use.pred.prob.cate`
: Logical, `TRUE` for assigning categories based on predicted probabilities, `FALSE` for imputation based on random draws from predictions of classification trees, default to `TRUE`. Note that if `forest.vote.cate = TRUE`, then this option is invalid.

`forest.vote.cate`
: Logical, `TRUE` for assigning categories based on majority votes of random forests, `FALSE` for imputation based on control of option `use.pred.prob.cate`, default to `FALSE`.

`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.

Details

`RfEmpImp` Imputation sampler for: categorical variables based on predicted probabilities.

Value

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

Author(s)

Shangzhi Hong

References

Hong, Shangzhi, et al. "Multiple imputation using chained random forests." Preprint, submitted April 30, 2020. https://arxiv.org/abs/2004.14823.

Shah, Anoop D., et al. "Comparison of random forest and parametric imputation models for imputing missing data using MICE: a CALIBER study." American journal of epidemiology 179.6 (2014): 764-774.

Malley, James D., et al. "Probability machines." Methods of information in medicine 51.01 (2012): 74-81.

Examples

```
# Prepare data
mtcars.catmcar <- mtcars
mtcars.catmcar[, c("gear", "carb")] <-
    gen.mcar(mtcars.catmcar[, c("gear", "carb")], warn.empty.row = FALSE)
mtcars.catmcar <- conv.factor(mtcars.catmcar, c("gear", "carb"))
# Perform imputation
impObj <- mice(mtcars.catmcar, method = "rfpred.cate", m = 5, maxit = 5,
maxcor = 1.0, eps = 0,
```

```
remove.collinear = FALSE, remove.constant = FALSE,
printFlag = FALSE)
```

`mice.impute.rfpred.emp`

Univariate sampler function for continuous variables using the empirical error distributions

---

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,
  alpha.emp = 0,
  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.

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

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.

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.

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

Hong, Shangzhi, et al. "Multiple imputation using chained random forests." Preprint, submitted April 30, 2020. https://arxiv.org/abs/2004.14823.

Zhang, Haozhe, et al. "Random Forest Prediction Intervals." The American Statistician (2019): 1-20.

Shah, Anoop D., et al. "Comparison of random forest and parametric imputation models for imputing missing data using MICE: a CALIBER study." American journal of epidemiology 179.6 (2014): 764-774.

Malley, James D., et al. "Probability machines." Methods of information in medicine 51.01 (2012): 74-81.

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 = 0,
remove.collinear = FALSE, remove.constant = FALSE,
printFlag = FALSE)
```

`mice.impute.rfpred.norm` *Univariate sampler function for continuous variables for prediction-based imputation, assuming normality for prediction errors 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.

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 the variance.

Usage

```
mice.impute.rfpred.norm(
  y,
  ry,
  x,
  wy = NULL,
  num.trees.cont = 10,
  norm.err.cont = TRUE,
  alpha.oob = 0,
  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.
num.trees.cont	Number of trees to build for continuous variables. The default is num.trees = 10.
norm.err.cont	Use normality assumption for prediction errors of random forests. The default is norm.err.cont = TRUE, and 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). If FALSE, then the predictions of random forest are used.
alpha.oob	The "significance level" for individual out-of-bag prediction errors used for the calculation for out-of-bag mean squared error, useful when presence of extreme values. For example, set alpha = 0.05 to use 95% confidence level. The default is alpha.oob = 0.0, and all the individual out-of-bag prediction errors will be kept intact.
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(wy).

Author(s)

Shangzhi Hong

References

Shah, Anoop D., et al. "Comparison of random forest and parametric imputation models for imputing missing data using MICE: a CALIBER study." American journal of epidemiology 179.6 (2014): 764-774.

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 = 0,
```

```
remove.collinear = FALSE, remove.constant = FALSE,
printFlag = FALSE)
```

---

query.rf.pred.idx *Identify corresponding observations indexes under the terminal nodes for a random forest model by* ranger

---

Description

The observation indexes (row numbers) constituting the terminal node associated with each observation are queried using the ranger object and the training data. The parameter keep.inbag = TRUE should be applied to call to ranger.

Usage

```
query.rf.pred.idx(obj, data, id.name = FALSE, unique.by.id = FALSE, ...)
```

Arguments

obj	An R object of class ranger.
data	Input for training data.
id.name	Use the IDs of the terminal nodes as names for the lists.
unique.by.id	Only return results of unique terminal node IDs.
...	Other parameters (will be ignored).

Details

The observations are found based on terminal node IDs. It should be noted that the out-of-bag observations are not present in the indexes.

Value

A nested list of length num.trees.

Author(s)

Shangzhi Hong

Examples

```
data(iris)
rfObj <- ranger(
    Sepal.Length ~ Sepal.Width + Petal.Length + Petal.Width + Species,
    data = iris, num.trees = 5, keep.inbag = TRUE)
outList <- query.rf.pred.idx(rfObj, iris)
```

query.rf.pred.val *Identify corresponding observed values for the response variable under the terminal nodes for a random forest model by* ranger

Description

The observed values (for the response variable) constituting the terminal node associated with each observation are queried using the ranger object and the training data. The parameter keep.inbag = TRUE should be applied to call to ranger.

Usage

```
query.rf.pred.val(obj, data, id.name = FALSE, unique.by.id = FALSE, ...)
```

Arguments

obj	An R object of class ranger.
data	Input for training data.
id.name	Use the IDs of the terminal nodes as names for the lists.
unique.by.id	Only return results of unique terminal node IDs.
...	Other parameters (will be ignored).

Details

The observations are found based on terminal node IDs. It should be noted that the out-of-bag observations are not present in the indexes.

Value

A nested list of length num.trees.

Author(s)

Shangzhi Hong

Examples

```
data(iris)
rfObj <- ranger(
    Sepal.Length ~ Sepal.Width + Petal.Length + Petal.Width + Species,
    data = iris, num.trees = 5, keep.inbag = TRUE)
outList <- query.rf.pred.val(rfObj, iris)
```

rangerCallerSafe *Remove unnecessary arguments for* ranger *function*

Description

This function serves as an workaround for ranger function.

Usage

```
rangerCallerSafe(...)
```

Arguments

...	Parameters to pass down.

Value

Constructed ranger object.

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