Package 'synMicrodata'

February 6, 2023

Type Package

Title Synthetic Microdata Generator

Version 1.0.0

Date 2023-02-04

Maintainer Hang J. Kim <hangkim0@gmail.com>

Description This tool fits a non-parametric Bayesian model called ``hierarchically coupled mixture model (HCMM)'' to the original microdata in order to generate synthetic microdata for privacy protection. The non-parametric feature of the adopted model is useful for catching the joint distribution of the original data in a highly flexible manner, leading to the generation of synthetic data very similar to the original data. Important reference papers on this method include Murray & Reiter (2016) <doi:10.1080/01621459.2016.1174132>.

License GPL (>= 3)

Imports methods, Rcpp

LinkingTo Rcpp, RcppArmadillo

RcppModules IO_module

NeedsCompilation yes

Author Hang J. Kim [aut, cre],
Juhee Lee [aut],
Young-Min Kim [aut]

Repository CRAN

Date/Publication 2023-02-06 11:00:02 UTC

R topics documented:

createModel *Create model object*

Description

Create model object for multipleSyn.

Usage

```
createModel(data_obj, max_R_S_K = c(30, 50, 20))
```

Arguments

data_obj	a data object passed readData
max_R_S_K	a maximum value of the number of mixture component index (r, s, k).

Value

createModel returns a Rcpp_modelobject

See Also

multipleSyn, readData

Examples

```
## preparing to generate synthetic datasets
dat_obj <- readData(iris[,1:4], iris[,5])
mod_obj <- createModel(dat_obj)

## generating synthetic datasets
syn_results <- multipleSyn(dat_obj, mod_obj,
    n_burnin=100, m=5, interval_btw_Syn = 50, show_iter=FALSE)

head(syn_results$Synt_Y_list[[1]])
head(syn_results$Synt_X_list[[1]])

## table of mixture component index in each synthetic datasets
apply(syn_results$r_i_cube, MARGIN=1, table)
apply(syn_results$s_i_cube, MARGIN=1, table)
apply(syn_results$k_i_cube, MARGIN=1, table)
```

modelobject *RCPP Implementation of the Library*

Description

Rcpp_modelobject-class

Value

No return value

multipleSyn *Generate synthetic micro datasets*

Description

Generate synthetic micro datasets using hierarchically coupled mixture model with local dependence (HCMM-LC).

Usage

```
multipleSyn(data_obj, model_obj, n_burnin, m, interval_btw_Syn, show_iter = TRUE)
```

Arguments

data_obj	a data object passed readData
model_obj	a model object passed createModel
n_burnin	the size of burn-in
m	number of synthetic micro data to be generated
interval_btw_Syn	the size of interval between synthetic micro datasets
show_iter	a logical value. If TRUE, multipleSyn will print history of (r,s,k) components on console.

Value

multipleSyn returns a list containing the following conmponents:

Synt_Y_list	a list of m continuous synthetic micro dataset(s).
Synt_X_list	a list of m categorical synthetic micro dataset(s).
r_i_cube	a matrix of the mixture component index for continuous variables.
s_i_cube	a matrix of the mixture component index for categorical variables.
k_i_cube	a matrix of the mixture component index for continuous & categorical structures.

References

Murray, J. S. and Reiter, J. P. (2016). Multiple imputation of missing categorical and continuous values via Bayesian mixture models with local dependence. *Journal of the American Statistical Association*, **111(516)**, pp.1466-1479.

See Also

`readData`, `createModel`

Examples

```
## preparing to generate synthetic datasets
dat_obj <- readData(iris[,1:4], iris[,5])
mod_obj <- createModel(dat_obj)

## generating synthetic datasets
syn_results <- multipleSyn(dat_obj, mod_obj,
    n_burnin=100, m=5, interval_btw_Syn = 50, show_iter=FALSE)

head(syn_results$Synt_Y_list[[1]])
head(syn_results$Synt_X_list[[1]])

## table of mixture component index in each synthetic datasets
apply(syn_results$r_i_cube, MARGIN=1, table)
apply(syn_results$s_i_cube, MARGIN=1, table)
apply(syn_results$k_i_cube, MARGIN=1, table)
```

Rcpp_modelobject-class

Class "Rcpp_modelobject"

Description

This class implements the MCMC sampler for a joint modeling approach to multiple edit-imputation for continuous data. It provides methods for updating and monitoring the sampler.

Details

Rcpp_modelobject objects should be created with `createModel`. Please see the example below.

Extends

Class "`C++Object`", directly.

Fields

- `data_obj`: input dataset generated from `readData`.

Methods

- `multipleSyn`: Generate synthetic micro datasets.

References

Hang J. Kim, Lawrence H. Cox, Alan F. Karr, Jerome P. Reiter and Quanli Wang (2015). "Simultaneous Edit-Imputation for Continuous Microdata", Journal of the American Statistical Association, DOI: 10.1080/01621459.2015.1040881.

Examples

```
## preparing to generate synthetic datasets
dat_obj <- readData(iris[,1:4], iris[,5])
mod_obj <- createModel(dat_obj)

## generating synthetic datasets
syn_results <- multipleSyn(dat_obj, mod_obj,
    n_burnin=100, m=5, interval_btw_Syn = 50, show_iter=FALSE)

head(syn_results$Synt_Y_list[[1]])
head(syn_results$Synt_X_list[[1]])

## table of mixture component index in each synthetic datasets
apply(syn_results$r_i_cube, MARGIN=1, table)
apply(syn_results$s_i_cube, MARGIN=1, table)
apply(syn_results$k_i_cube, MARGIN=1, table)
```

`readData` *Read the original datasets*

Description

Read the original datasets to be generated.

Usage

```
readData(Y_input, X_input, RandomSeed = 99)
```

Arguments

`Y_input`	a data.frame consisting of continuous variables of the original data. It should consist only of numerical values.
`X_input`	a data.frame consisting of categorical variables of the original data. It should consist only of factor-type classes.
`RandomSeed`	a random seed number

Value

`readData` returns an object of "`readData_passed`" class.

An object of class "`readData_passed`" is a list containing the following components:

`n_sample`	the size of the original datasets.
`p_Y`	the number of continuous variables.
`Y_mat_std`	a matrix that standardized `Y_input` with mean 0 and standard deviation 1.
`mean_Y_input`	the mean vectors of original `Y_input`.
`sd_Y_input`	the standard deviation vectors of original `Y_input`.
`NA_Y_mat`	an indicator matrix indicating a missing value (continuous).
`p_X`	the number of categorical variables
`D_l_vec`	the maximum number of levels each categorical variables.
`X_mat_std`	a numerical matrix recoded from categorical dataset.
`levels_X_input`	a list containing levels of categorical variables in `X_input`
`NA_X_mat`	an indicator matrix indicating a missing value (categorical).
`var_names`	a list containing variable names of `X_input` and `Y_input`.

See Also

`multipleSyn`, `createModel`

Index