Package ‘seqimpute’

November 7, 2022

Type Package

Title Imputation of Missing Data in Sequence Analysis

Version 1.8

Date 2022-11-07

Description Multiple imputation of missing data present in a dataset through the prediction based on either a random forest or a multinomial regression model. Covariates and time-dependant covariates can be included in the model. The prediction of the missing values is based on the method of Halpin (2012) <https://researchrepository.ul.ie/articles/report/Multiple_imputation_for_life-course_sequence_data/19839736>.

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 7.2.0

Imports Amelia, rms, stringr, TraMineR, swfscMisc, plyr, dplyr, dffdx, mice, foreach, parallel, doRNG, doSNOW, ranger, mlr, nnet

Author Andre Berchtold [aut, cre], Anthony Guinchard [aut], Kevin Emery [aut], Kamyar Taher [aut]

Maintainer Andre Berchtold <andre.berchtold@unil.ch>

Repository CRAN

Repository/R-Forge/Project seqimpute

Repository/R-Forge/Revision 101

Repository/R-Forge/DateTimeStamp 2022-11-07 11:20:38

Date/Publication 2022-11-07 13:10:02 UTC

NeedsCompilation no

Depends R (>= 3.5.0)
R topics documented:

<table>
<thead>
<tr>
<th>Package</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO</td>
<td>Dataset containing 3 fixed covariates about the game addiction of young subjects</td>
</tr>
<tr>
<td>COt</td>
<td>Dataset containing 1 time-dependant covariate about the game addiction of young subjects</td>
</tr>
</tbody>
</table>

Description

These covariates are respectively 'Gender' (male/female), 'Age' (at T1, in years) and 'Track' (school/apprenticeship).

Usage

data(CO)

data(COt)

Format

A data frame of fixed covariates with 500 sequences and 3 columns.

Details

- 500 sequences
- 3 columns
Format

A data frame of time-dependant covariates with 500 sequences and 4 columns.

Details

• 500 sequences
• 4 columns

OD

Dataset containing variables about the game addiction of young subjects

Description

An original dataset example OD to test seqimpute.R and its aside functions.

Usage

data(OD)

Format

A data frame of factor variables with 500 sequences and 4 columns.

Details

• 500 sequences (i.e. 500 rows)
• 4 time measurements (i.e. 4 columns)
• The variables can be either 'no', 'yes' or NA

seqimpute

Imputation of missing data in sequence analysis

Description

Multiple imputation of missing data present in a dataset through the prediction based on either a multinomial or a random forest regression model. Covariates and time-dependant covariates can be included in the model. The prediction of the missing values is based on the theory of Prof. Brendan Halpin. It considers a various amount of surrounding available information to perform the prediction process. In fact, we can among others specify \( np \) (the number of past variables taken into account) and \( nf \) (the number of future information taken into account).
Usage

seqimpute(
  OD,
  regr = "multinom",
  np = 1,
  nf = 0,
  nfi = 1,
  npt = 1,
  available = TRUE,
  CO = matrix(NA, nrow = 1, ncol = 1),
  COt = matrix(NA, nrow = 1, ncol = 1),
  pastDistrib = FALSE,
  futureDistrib = FALSE,
  mi = 1,
  mice.return = FALSE,
  include = FALSE,
  noise = 0,
  ParExec = FALSE,
  ncores = NULL,
  SetRNGSeed = FALSE,
  num.trees = 10,
  min.node.size = NULL,
  max.depth = NULL,
  verbose = TRUE
)

Arguments

OD either a data frame containing sequences of a multinomial variable with missing data (coded as NA) or a state sequence object built with the TraMineR package

dregr a character specifying the imputation method. If regr="multinom", multinomial models are used, while if regr="rf", random forest models are used.
np number of previous observations in the imputation model of the internal gaps.
nf number of future observations in the imputation model of the internal gaps.
nfi number of future observations in the imputation model of the initial gaps.
npt number of previous observations in the imputation model of the terminal gaps.
available a logical value allowing the user to choose whether to consider the already imputed data in the predictive model (available = TRUE) or not (available = FALSE).
CO a data frame containing some covariates among which the user can choose in order to specify his model more accurately.
COt a data frame object containing some time-dependent covariates that help specifying the predictive model more accurately.
pastDistrib a logical indicating if the past distribution should be used as predictor in the imputation model.
futureDistrib a logical indicating if the future distribution should be used as predictor in the imputation model.

mi number of multiple imputations (default: 1).

mice.return a logical indicating whether an object of class mids, that can be directly used by the mice package, should be returned by the algorithm. By default, a data frame with the imputed datasets stacked vertically is returned.

include logical. If a dataframe is returned (mice.return = FALSE), indicates if the original dataset should be included or not. This parameter does not apply if mice.return=TRUE.

noise numeric object adding a noise on the predicted variable pred determined by the multinomial model (by introducing a variance noise for each components of the vector pred) (the user can choose any value for noise, but we recommend to choose a rather relatively small value situated in the interval [0.005-0.03]).

ParExec logical. If TRUE, the multiple imputations are run in parallel. This allows faster run time depending of how many core the processor has.

ncores integer. Number of cores to be used for the parallel computation. If no value is set for this parameter, the number of cores will be set to the maximum number of CPU cores minus 1.

SetRNGSeed an integer that is used to set the seed in the case of parallel computation. Note that setting set.seed() alone before the seqimpute function won’t work in case of parallel computation.

num.trees random forest parameter setting the number of trees of each random forest model.

min.node.size random forest parameter setting the minimum node size for each random forest model.

max.depth random forest parameter setting the maximal depth tree for each random forest model.

verbose logical. If TRUE, seqimpute will print history and warnings on console. Use verbose=FALSE for silent computation.

Details
The imputation process is divided into several steps. According to the location of the gaps of NA among the original dataset, we have defined 5 types of gaps:
- Internal Gaps (simple usual gaps)
- Initial Gaps (gaps situated at the very beginning of a sequence)
- Terminal Gaps (gaps situated at the very end of a sequence)
- Left-hand side SLG (Specially Located Gaps) (gaps of which the beginning location is included in the interval [0, np] but the ending location is not included in the interval [ncol(OD)-nf, ncol(OD)])
- Right-hand side SLG (Specially Located Gaps) (gaps of which the ending location is included in the interval [ncol(OD)-nf, ncol(OD)] but the beginning location is not included in the interval [0, np])
- Both-hand side SLG (Specially Located Gaps) (gaps of which the beginning location is included in the interval [0, np] and the ending location is included in the interval [ncol(OD)-nf, ncol(OD)])

Value

Returns either an S3 object of class mids if mice.return = TRUE or a dataframe, where the imputed dataset are stacked vertically. In the second case, two columns are added: .imp integer that refers to the imputation number (0 corresponding to the original dataset if include=TRUE) and .id character corresponding to the rownames of the dataset to impute.

Author(s)

Andre Berchtold <andre.berchtold@unil.ch> Kevin Emery Anthony Guinchard Kamyar Taher

References


HALPIN, Brendan (2013). Imputing sequence data: Extensions to initial and terminal gaps, Stata’s. Working Paper WP2013-01, Department of Sociology, University of Limerick. http://hdl.handle.net/10344/3620

Examples

# Default single imputation
RESULT <- seqimpute(OD=OD, np=1, nf=1, nfi=1, npt=1, mi=1)

# Seqimpute used with parallelisation
## Not run:
RESULT <- seqimpute(OD=OD, np=1, nf=1, nfi=1, npt=1, mi=2, ParExec=TRUE, SetRNGSeed=17, nc=2)

## End(Not run)

seqQuickLook

Numbering NAs and types of gaps among a dataset

Description

seqQuickLook.R is a function aimed at providing an overview of the number and size of the different types of gaps spread in the original dataset OD.

Usage

seqQuickLook(OD, np = 1, nf = 0)
seqTrans

Arguments

OD matrix object containing sequences of a variable with missing data (coded as NA).

np numeric object corresponding to the number of previous observations in the imputation model of the internal gaps (default 1).

nf numeric object corresponding to the number of future observations in the imputation model of the internal gaps (default 0).

Value

It returns a data.frame object that summarizes for each type of gaps (Internal Gaps, Initial Gaps, Terminal Gaps, LEFT-hand side SLG, RIGHT-hand side SLG, Both-hand side SLG), the minimum length, the maximum length, the total number of gaps and the total number of NAs induced.

Author(s)

Andre Berchtold <andre.berchtold@unil.ch>

Examples

data(OD)

seqQuickLook(OD=OD, np=1, nf=0)

seqTrans

Computing and spotting transitions among a dataset

Description

The purpose of seqTrans.R is to spot transitions in a dataset.

Usage

seqTrans(OD, trans)

Arguments

OD matrix object containing sequences of a variable with missing data (coded as NA).

trans character vector gathering the impossible transitions. For example: trans <- c("1->3","1->4","2->1","4->1","4->3")

Value

It returns a matrix whose rows each are the indices of an impossible transition.
Author(s)
Andre Berchtold <andre.berchtold@unil.ch>, Kevin Emery

Examples

data(OD)

seqTransList <- seqTrans(OD=OD, trans=c("yes->no"))
Index

* datasets
  CO, 2
  Cot, 2
  OD, 3
CO, 2
Cot, 2
OD, 3
seqimpute, 3
seqQuickLook, 6
seqTrans, 7