

Package ‘NNMIS’

April 20, 2019

Type Package

Title Nearest Neighbor Based Multiple Imputation for Survival Data with Missing Covariates

Version 1.0.1

Date 2019-04-19

Author Di Ran, Chiu-Hsieh Hsu, Mandi Yu

Maintainer Chiu-Hsieh Hsu <pablo1639@gmail.com>

Description Imputation for both missing covariates and censored observations (optional) for survival data with missing covariates by the nearest neighbor based multiple imputation algorithm as described in Hsu et al. (2006) <doi:10.1002/sim.2452>, and Hsu and Yu (2018) <doi: 10.1177/0962280218772592>. Note that current version can only impute for a situation with one missing covariate.

License LGPL (>= 2)

Depends R (>= 2.14.0)

Imports stats, graphics, parallel, survival

LazyData TRUE

RoxygenNote 6.1.1

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2019-04-20 16:00:23 UTC

R topics documented:

coxph.pool	2
km.pool	3
nnmi	4
NNMIS	4
plotKM	7
print.nnmi	7
Index	8

 coxph.pool

Estimate Cox regression model pooling over the imputed datasets

Description

This function estimates Cox regression model, taking into account the additional uncertainty that arises due to a finite number of imputations of the missing data.

Usage

```
coxph.pool(obj, time, status, Z, forceNumeric = FALSE, setRef = NULL)
```

Arguments

obj	A 'nnmi' object, that contains a finite number of imputations of the missing data.
time	A vector contains the observed time.
status	A vector contains the event indicator.
Z	A vector or matrix that contains other covariates.
forceNumeric	Logical, if it is True, the class of imputed variable will force to be numeric. The default is FALSE.
setRef	Optional, a reference group can be set for binary or categorical variable.

Value

A data frame contains pooled estimation of Cox regression model.

Examples

```
# load required packages
library(NNMIS)
library(survival)

# load data set - stanford2 in package 'survival'
data("stanford2")
head(stanford2)
attach(stanford2)

# performance multiple imputation on missing covariate t5
imp.dat <- NNMIS(t5, xa=age, xb=age, time=time, event=status, Seed = 2016)

# this program can impute censoring time based on the imputed missing covariate
# imp.dat <- NNMIS(t5, xa=age, xb=age, time=time, event=status, imputeCT=T, Seed = 2016)
# check imputation results
# head(imp.dat$dat.NNMI) #> missing covariates
# head(imp.dat$dat.T.NNMI) #> censoring time
# head(imp.dat$dat.Id.NNMI) #> censoring indicator
```

```
# check imputation results
head(imp.dat$dat.NNMI)

# combine inference from imputed data sets by using Rubin's rules
# estimates in Cox regression
coxph.pool(imp.dat, time, status, age)
```

km.pool	<i>Perform Kaplan-Meier estimation over the multiply imputed survival data sets</i>
---------	---

Description

This function estimates Kaplan-Meier estimates based on Rubin's rules (multiple imputation algorithms) (Rubin, 2004).

Usage

```
km.pool(obj, time, status)
```

Arguments

obj	A 'nnmi' object, that contains imputed data for the missing covariate and the censored observations.
time	A vector contains the observed time.
status	A vector contains the event indicator.

Value

A data frame contains pooled Kaplan-Meier estimates.

References

Rubin DB. Multiple imputation for nonresponse in surveys. New York: John Wiley and Sons; 2004.

Examples

```
# load required packages
library(NNMI)
library(survival)

# load data set - stanford2 in package 'survival'
data("stanford2")
head(stanford2)
attach(stanford2)
```

```

# performance multiple imputation on missing covariate t5 and
# censored observations based on the imputed missing covariates
imp.dat <- NNMIS(t5, xa=age, xb=age, time=time, event=status, imputeCT=TRUE, Seed = 2016)

# check imputation results
head(imp.dat$dat.T.NNMI)

# combine inference from imputed data sets using Rubin's rules
# Kaplan-Meier estimates
kmfit <- km.pool(imp.dat, time, status)
plotKM(kmfit)

```

nnmi	<i>A nnmi class</i>
------	---------------------

Description

A nnmi class

Usage

```
nnmi(y, ...)
```

Arguments

y	a list object
...	further arguments passed to function

NNMIS	<i>Nearest Neighbor Based Multiple Imputation for Survival Data with Missing Covariates (NNMIS)</i>
-------	---

Description

This function performs the nearest neighbor based multiple imputation approach proposed by Hsu et al. (2006), Long et al. (2012), Hsu et al. (2014) and Hsu and Yu (2017, 2018) to impute for missing covariates and censored observations (optional). To perform imputation for missing covariates, the approach requires one to fit two working models: one for predicting the missing covariate values and the other for predicting the missing probabilities based on the observed data. The distribution of the working model for predicting the missing covariate values will be automatically decided by the data type of the missing covariate. A logistic regression model will be fitted to predict the missing probabilities. The estimation results of the two working models are then used to select a nearest neighborhood for each missing covariate observation. Once the nearest neighborhood is chosen, multiple imputation is then performed on the neighborhood non-parametrically. The detailed procedures can be found in Long et al. (2012), Hsu et al. (2014), and Hsu and Yu (2017, 2018).

Similarly, to perform imputation for censored observations, one has to fit two working models first: one for predicting the survival time and the other for predicting the censoring time. These two working models are derived using Cox regression. The estimation results of the two working models are then used to select a nearest neighborhood for each censored observation. Once the nearest neighborhood is chosen, multiple imputation is then performed on the neighborhood non-parametrically. The detailed procedures can be found in Hsu et al. (2006).

Note that the current version can only perform imputation for a situation with only one missing covariate. Before you use this package, please check the input covariates matrix to see if there is more than one missing covariate.

Usage

```
NNMIS(y, xa = NULL, xb = NULL, time, event, MI = 10, NN = 5,
      w1 = 0.8, w2 = 0.2, Seed = NA, imputeCT = FALSE, NN.t = 10,
      mc.cores = 1, verbose = TRUE)
```

Arguments

y	Can be any vector of covariate, which contains missing values to be imputed. Missing values are coded as NA.
xa	Can be any vector or matrix, which will be used as the covariates along with the estimated cumulative baseline hazard and the observed censoring indicator for the working model of predicting the missing covariate values. Note that no missing values are allowed for this.
xb	Can be any vector or matrix, which will be used as the covariates along with the estimated cumulative baseline hazard and the observed censoring indicator for the working model of predicting the missing probabilities. Note that no missing values are allowed for this.
time	This is the observed time.
event	This is the censoring indicator, i.e. 0:censored; 1: event.
MI	Number of imputation. The default is MI=10.
NN	Size of the nearest neighborhood considered for imputing missing covariate. Default is NN=5.
w1	Weight will be used in the working model of predicting the missing covariate values. The default is w1=0.8.
w2	Weight will be used in the working model of predicting the missing probabilities. The default is w1=0.2.
Seed	An integer that is used as argument by the set.seed() for offsetting the random number generator. Default is to leave the random number generator alone.
imputeCT	Logical. If TRUE, survival times for censored observations will be imputed and exported as part of output. (optional)
NN.t	Size of the nearest neighborhood considered for imputing survival times for each censored observation. Default is NN.t=10.
mc.cores	Number of cpu cores to be used. This option depends on package "parallel". The default is mc.core=1.
verbose	If True, print messages.

Value

An object of class "nnmi" is a list containing parameters used in multiple imputation and all outputs.

N	Number of observations.
MI	Number of imputation.
NN	Size of the nearest neighborhood considered for imputing missing covariate.
w1	Weight in the working model for predicting the missing covariate values/survival times.
w2	Weight in the working model for predicting the missing probabilities/censoring times.
mfamily	Distribution family used in the working model for predicting the missing covariate values.
imputeCT	Logical, whether to impute survival times for censored observations or not.
dat.NNMI	data frame containing imputed missing covariate values.
dat.T.NNMI	data frame containing imputed survival times.
dat.Id.NNMI	data frame containing censoring indicator.

References

Hsu CH, Taylor JM, Murray S, Commenges D. Survival analysis using auxiliary variables via non-parametric multiple imputation. *Statistics in Medicine* 2006; 25: 3503-17.

Hsu CH, Long Q, Li Y, Jacobs E. A Nonparametric Multiple Imputation Approach for Data with Missing Covariate Values with Application to Colorectal Adenoma Data. *Journal of Biopharmaceutical Statistics* 2014; 24: 634-648.

Hsu CH, Yu M. Cox regression analysis with missing covariates via nonparametric multiple imputation. *arXiv* 2017; 1710.04721.

Hsu CH, Yu M. Cox regression analysis with missing covariates via nonparametric multiple imputation. *Statistical Methods in Medical Research* 2018; doi: 10.1177/0962280218772592.

Long Q, Hsu CH, Li Y. Doubly robust nonparametric multiple imputation for ignorable missing data. *Statistica Sinica* 2012; 22: 149-172.

Examples

```
# load required packages
library(NNMIS)
library(survival)

# load data set - stanford2 in package 'survival'
data("stanford2")
head(stanford2)
attach(stanford2)

# performance multiple imputation on missing covariate t5
imp.dat <- NNMIS(t5, xa=age, xb=age, time=time, event=status, Seed = 2016, mc.core=1)

# check imputation results
```

```

head(imp.dat$dat.NNMI)

# this program can impute survival times for censored observations based on
# the imputed missing covariate values
# imp.dat <- NNMIS(t5, xa=age, xb=age, time=time, event=status, imputeCT=TRUE, Seed = 2016)
# check imputation results
# head(imp.dat$dat.NNMI) # imputed missing covariate values
# head(imp.dat$dat.T.NNMI) # imputed survival times
# head(imp.dat$dat.Id.NNMI) # censoring indicator

```

plotKM	<i>Plot function for pooled Kaplan-Meier estimates</i>
--------	--

Description

A plot of survival curves is produced.

Usage

```
plotKM(x)
```

Arguments

x	a data.frame contains pooled estimates of survival function generated from function 'km.pool'.
---	--

See Also

km.pool

print.nnmi	<i>Print function for object of 'nnmi' class.</i>
------------	---

Description

Print function for object of 'nnmi' class.

Usage

```
## S3 method for class 'nnmi'
print(x, ...)
```

Arguments

x	a 'nnmi' object
...	further arguments passed to function

Index

`coxph.pool`, 2

`km.pool`, 3

`nnmi`, 4

`NNMIS`, 4

`plotKM`, 7

`print.nnmi`, 7