Package ‘openVA’

May 4, 2020

Type Package

Title Automated Method for Verbal Autopsy

Version 1.0.11

Date 2020-05-01

Author Zehang Richard Li, Tyler McCormick, Sam Clark

Maintainer Zehang Richard Li <lizehang@gmail.com>

Depends R (>= 3.1)

Imports InterVA5 (>= 1.0.1), InSilicoVA (>= 1.1.3), InterVA4 (>= 1.7.3), Tariff (>= 1.0.1), ggplot2, crayon, cli, methods

Suggests nbc4va, testthat


License GPL-2

URL https://github.com/verbal-autopsy-software/openVA

BugReports https://github.com/verbal-autopsy-software/openVA/issues

RoxygenNote 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2020-05-04 10:00:02 UTC
R topics documented:

- codeVA
- ConvertData
- ConvertData.phmrc
- getCSMF
- getCSMF_accuracy
- getIndivProb
- getPHMRC_url
- getTopCOD
- interVA.train
- openVA_status
- openVA_update
- plotVA
- stackplotVA

Index

CodeVA

Running automated method on VA data

Description

Running automated method on VA data

Usage

codeVA(data, data.type = c("WHO2012", "WHO2016", "PHMRC", "customize"))[1], data.train = NULL, causes.train = NULL, causes.table = NULL, model = c("InSilicoVA", "InterVA", "Tariff", "NBC")[1], Nchain = 1, Nsim = 10000, version = c("4.02", "4.03", "5"))[2], HIV = "h", Malaria = "h", phmrc.type = c("adult", "child", "neonate")[1], convert.type = c("quantile", "fixed", "empirical")[1], ...

Arguments

data Input VA data, see data.type below for more information about the format.
data.type There are four data input types currently supported by codeVA function as below.

- WHO2012: InterVA-4 input format using WHO 2012 questionnaire. For example see data(RandomVA1). The first column should be death ID.
- WHO2016: InterVA-5 input format using WHO 2016 questionnaire. For example see data(RandomVA5). The first column should be death ID.
- PHMRC: PHMRC data format. For example see ConvertData.phmrc
- customized: Any dichotomized dataset with “Y” denote “presence”, “” denote “absence”, and “.” denote “missing”. The first column should be death ID.
Training data with the same columns as data, except for an additional column specifying cause-of-death label. It is not used if data.type is “WHO” and model is “InterVA” or “InSilicoVA”. The first column also has to be death ID for “WHO” and “customized” types.

causes.train

column name of the cause-of-death assignment label in training data.

causes.table

list of causes to consider in the training data. Default to be NULL, which uses all the causes present in the training data.

model

Currently supports four models: “InSilicoVA”, “InterVA”, “Tariff”, and “NBC”.

Nchain

Parameter specific to “InSilicoVA” model. Currently not used.

Nsim

Parameter specific to “InSilicoVA” model. Number of iterations to run the sampler.

version

Parameter specific to “InterVA” model. Currently supports “4.02”, “4.03”, and “5”. For InterVA-4, “4.03” is strongly recommended as it fixes several major bugs in “4.02” version. “4.02” is only included for backward compatibility. “5” version implements the InterVA-5 model, which requires different data input format.

HIV

Parameter specific to “InterVA” model. HIV prevalence level, can take values “h” (high), “l” (low), and “v” (very low).

Malaria

HIV Parameter specific to “InterVA” model. Malaria prevalence level, can take values “h” (high), “l” (low), and “v” (very low).

phmrc.type

Which PHMRC data format is used. Currently supports only “adult” and “child”, “neonate” will be supported in the next release.

convert.type

type of data conversion when calculating conditional probability (probability of each symptom given each cause of death) for InterVA and InSilicoVA models. Both “quantile” and “fixed” usually give similar results empirically.

• quantile: the rankings of the P(S|C) are obtained by matching the same quantile distributions in the default InterVA P(S|C)
• fixed: P(S|C) are matched to the closest values in the default InterVA P(S|C) table.
• empirical: no ranking is calculated, but use the empirical conditional probabilities directly, which will force updateCondProb to be FALSE for InSilicoVA algorithm.

Value

a fitted object

References


http://www.interva.net/


See Also

insilico, InterVA, interVA.train, tariff, and nbc function in the nbc4va package.

Examples

data(RandomVA3)
test <- RandomVA3[1:200, ]
train <- RandomVA3[201:400, ]
fit1 <- codeVA(data = test, data.type = "customize", model = "InSilicoVA", data.train = train, causes.train = "cause", Nsim=1000, auto.length = FALSE)

fit2 <- codeVA(data = test, data.type = "customize", model = "InterVA", data.train = train, causes.train = "cause", write=FALSE, version = "4.02", HIV = "h", Malaria = "l")

fit3 <- codeVA(data = test, data.type = "customize", model = "Tariff", data.train = train, causes.train = "cause", nboot.sig = 100)

ConvertData

Converting Input data with different coding scheme to standard format

Description

Converting Input data with different coding scheme to standard format

Usage

ConvertData(input, yesLabel = NULL, noLabel = NULL, missLabel = NULL)
Arguments

input  matrix input, the first column is ID, the rest of the columns each represent one symptom
yesLabel  The value(s) coding "Yes" in the input matrix.
noLabel  The value(s) coding "No" in the input matrix.
missLabel  The value(s) coding "Missing" in the input matrix.

Value

a matrix with coding scheme as follows: "Y" for yes, "" for No, and "." for missing.

Examples

# make up a fake 2 by 3 dataset with 2 deaths and 3 symptoms
id <- c("d1", "d2")
x <- matrix(c("Yes", "No", "Don't know", "Yes", "Refused to answer", "No"),
byrow = TRUE, nrow = 2, ncol = 3)
x <- cbind(id, x)
colnames(x) <- c("ID", "S1", "S2", "S3")
# see possible raw data (or existing data created for other purpose)
x
new <- ConvertData(x, yesLabel = "Yes", noLabel = "No",
missLabel = c("Don't know", "Refused to answer"))
new

ConvertData.phmrc  Convert standard PHMRC data into binary indicator format

Description

This function converts the symptoms into binary indicators of three levels: Yes, No, and Missing. The health care experience (HCE) and free-text columns, i.e., columns named "word_****", are not considered in the current version of data conversion.

Usage

ConvertData.phmrc(input, input.test = NULL, cause = NULL,
phmrc.type = c("adult", "child", "neonate")[[1]], cutoff = c("default",
"adapt")[[1]], ...)
Arguments

- **input**: standard PHMRC data format
- **input.test**: standard PHMRC data format to be transformed in the same way as input
- **cause**: the column name for the cause-of-death variable to use. For example, "va34", "va46", or "va55". It is used if adaptive cut-offs are to be calculated for continuous variables. See below for details.
- **phmrc.type**: which data input format it is. The three data formats currently available are "adult", "child", and "neonate".
- **cutoff**: This determines how the cut-off values are to be set for continuous variables. "default" sets the cut-off values proposed in the original paper published with the dataset. "adapt" sets the cut-off values using the rules described in the original paper, which calculates the cut-off as being two median absolute deviations above the median of the mean durations across causes. However, we are not able to replicate the default cut-offs following this rule. So we suggest users to use this feature with caution.
- **...**: not used

Value

converted dataset with only ID and binary symptoms. Notice that when applying this function to the raw PHMRC data, the returned ID variable corresponds to the row index of the raw PHMRC data (i.e., cleaned data with ID = 10 correspond to the 10th row of the raw dataset), and does not correspond to the "newid" column in the PHMRC data.

References


Examples

```r
# read the raw data files from PHMRC website
# notice reading directly from internet could be time consuming
# so we only read 100 rows here.
# in practice, it is much easier and faster to download the file first,
# and read all at once.
raw <- read.csv(getPHMRC_url("adult"), nrow = 100)
head(raw[, 1:20])
# default way of conversion
clean <- ConvertData.phmrc(raw, phmrc.type = "adult")
head(clean$output[, 1:20])
# using cut-offs calculated from the data (caution)
clean2 <- ConvertData.phmrc(raw, phmrc.type = "adult",
cause = "va55", cutoff = "adapt")
head(clean2$output[, 1:20])
```
## Now using the first 100 rows of data as training dataset
## And the next 100 as testing dataset
```
test <- read.csv(getPHMRC_url("adult"), nrow = 200)
test <- test[-(1:100),]
```

## For the default transformation it does matter
```
clean <- ConvertData.phmrc(raw, test, phmrc.type = "adult")
head(clean$output[, 1:20])
head(clean$output.test[, 1:20])
```

## For adaptive transformation, need to make sure both files use the same cutoff
```
clean2 <- ConvertData.phmrc(raw, test, phmrc.type = "adult",
cause = "va55", cutoff = "adapt")
head(clean2$output[, 1:20])
head(clean2$output.test[, 1:20])
```

---

**getCSMF**

*Obtain CSMF from fitted model*

### Description
Obtain CSMF from fitted model

### Usage
```
getCSMF(x, CI = 0.95, interVA.rule = TRUE)
```

### Arguments
- **x**: a fitted object from codeVA.
- **CI**: For insilico object only, specifying the credible interval to return. Default value to be 0.95.
- **interVA.rule**: Logical indicator for interVA object only. If TRUE, it means only up to top 3 causes for each death are used to calculate CSMF and the rest are categorized as "undetermined"

### Value
a vector or matrix of CSMF for all causes.

### Examples
```
## Not run:
library(InSilicoVA)
data(RandomVA1)
# for illustration, only use interVA on 100 deaths
fit <- codeVA(RandomVA1[1:100,], data.type = "WHO2012", model = "InterVA",
              version = "4.03", HIV = "h", Malaria = "l", write=FALSE)
getCSMF(fit)
```
library(InterVA5)
data(RandomVA5)
fit <- codeVA(RandomVA5[1:100, ], data.type = "WHO2016", model = "InterVA",
            version = "5", HIV = "h", Malaria = "l", write = FALSE)
getCSMF(fit)

## End(Not run)

getCSMF_accuracy

*Calculate CSMF accuracy*

**Description**

Calculate CSMF accuracy

**Usage**

```r
getCSMF_accuracy(csmf, truth, undet = NULL)
```

**Arguments**

- `csmf`: a CSMF vector from `getCSMF` or a `InSilicoVA` fitted object.
- `truth`: a CSMF vector of the true CSMF.
- `undet`: name of the category denoting undetermined causes. Default to be NULL.

**Value**

A number (or vector if input is `InSilicoVA` fitted object) of CSMF accuracy as 1 - \(\sum(\text{abs}(\text{CSMF} - \text{CSMF_true})) / (2 \times (1 - \min(\text{CSMF_true})))\).

**Examples**

```r
csmf1 <- c(0.2, 0.3, 0.5)
csmf0 <- c(0.3, 0.3, 0.4)
acc <- getCSMF_accuracy(csmf1, csmf0)
```
getIndivProb

Extract individual distribution of cause of death

Description
Extract individual distribution of cause of death

Usage
getIndivProb(x, CI = NULL, ...)

Arguments
- x: a fitted object from codeVA.
- CI: Credible interval for posterior estimates. If CI is set to TRUE, a list is returned instead of a data frame.
- ...: additional arguments that can be passed to get.indiv from InSilicoVA package.

Value
a data frame of COD distribution for each individual specified by row names.

Examples
```r
data(RandomVA1)
# for illustration, only use interVA on 100 deaths
fit <- codeVA(RandomVA1[1:100, ], data.type = "WHO", model = "InterVA",
version = "4.02", HIV = "h", Malaria = "l", write=FALSE)
probs <- getIndivProb(fit)
```

getPHMRC_url

Get the URL to the PHMRC dataset

Description
Get the URL to the PHMRC dataset

Usage
getchMRC_url(type)

Arguments
- type: adult, child, or neonate
getTopCOD

Extract the most likely cause of death

Description

Extract the most likely cause of death

Usage

getTopCOD(x, interVA.rule = TRUE)

Arguments

x

a fitted object from codeVA.

interVA.rule

Logical indicator for interVA object only. If TRUE, only the InterVA reported first cause is extracted.

Value

a data frame of ID and most likely cause assignment.

Examples

data(RandomVA1)
# for illustration, only use interVA on 100 deaths
fit <- codeVA(RandomVA1[1:100, ], data.type = "WHO", model = "InterVA",
version = "4.02", HIV = "h", Malaria = "l", write=FALSE)
getTopCOD(fit)
**Extended InterVA method for non-standard input**

**Description**

Extended InterVA method for non-standard input

**Usage**

```r
interVA.train(data, train, causes.train, causes.table = NULL,
       thre = 0.95, type = c("quantile", "fixed", "empirical")[1],
       prior = c("uniform", "train")[1], ...)
```

**Arguments**

- **data**
  A matrix input, or data read from csv files. Sample input is included as `data(RandomVA3)`.

- **train**
  A matrix input, or data read from csv files in the same format as `data`, but with an additional column specifying cause-of-death. Sample input is included as `data(RandomVA3)`.

- **causes.train**
  the column name of the cause-of-death assignment label in training data.

- **causes.table**
  list of causes to consider in the training data. Default to be NULL, which uses all the causes present in the training data.

- **thre**
  numerical number between 0 and 1. Symptoms with missing rate higher than `thre` in the training data will be dropped from both training and testing data.

- **type**
  type of data conversion when calculating conditional probability (probability of each symptom given each cause of death) for InterVA and InSilicoVA models. Both “quantile” and “fixed” usually give similar results empirically.

  - **quantile**: the rankings of the P(S|C) are obtained by matching the same quantile distributions in the default InterVA P(S|C)
  - **fixed**: P(S|C) are matched to the closest values in the default InterVA P(S|C) table.
  - **empirical**: no ranking is calculated, but use the empirical conditional probabilities directly.

- **prior**
  The prior distribution of CSMF. “uniform” uses no prior information, i.e., 1/C for all C causes and “train” uses the CSMF in the training data as prior distribution of CSMF.

- **...**
  not used

**Value**

fitted `interVA` object
openVA_status

References


http://www.interva.net/

Examples

data(RandomVA3)
test <- RandomVA3[1:200, ]
train <- RandomVA3[201:400, ]
out <- interVA.train(data = test, train = train, causes.train = "cause",
prior = "train", type = "quantile")

openVA_status

Check openVA packages status

Description

This will print the current versions of all openVA packages (and optionally, their dependencies) are up-to-date, and will install after an interactive confirmation.

Usage

openVA_status()

Examples

## Not run:
openVA_status()

## End(Not run)
openVA_update

Description
This will check to see if all openVA packages (and optionally, their dependencies) are up-to-date, and will install after an interactive confirmation.

Usage
openVA_update()

Examples
## Not run:
openVA_update()
## End(Not run)

plotVA

Plot top CSMF for a fitted model

Description
Plot top CSMF for a fitted model

Usage
plotVA(object, top = 10, title = NULL, ...)

Arguments
- object: a fitted object using `codeVA`
- top: number of top causes to plot
- title: title of the plot
- ...: additional arguments passed to `plot.insilico`, `plot.tariff`, `CSMF`, or `plot.nbc` function in the `nbc4va` package.

See Also
`plot.insilico`, `plot.tariff`, `CSMF`
Examples

```r
data(RandomVA3)
test <- RandomVA3[1:200, ]
train <- RandomVA3[201:400, ]
fit1 <- codeVA(data = test, data.type = "customize", model = "InSilicoVA",
data.train = train, causes.train = "cause",
Nsim=1000, auto.length = FALSE)

fit2 <- codeVA(data = test, data.type = "customize", model = "InterVA",
data.train = train, causes.train = "cause",
version = "4.02", HIV = "h", Malaria = "l")

fit3 <- codeVA(data = test, data.type = "customize", model = "Tariff",
data.train = train, causes.train = "cause",
nboot.sig = 100)

plotVA(fit1)
plotVA(fit2)
plotVA(fit3)
```

Description

Produce bar plot of the CSMFs for a fitted "insilico" object in broader groups.

Usage

```r
stackplotVA(x, grouping = NULL, type = c("stack", "dodge")[1],
group_order = NULL, err = TRUE, CI = 0.95,
sample.size.print = FALSE, xlab = "Group", ylab = "CSMF",
ylim = NULL, title = "CSMF by broader cause categories",
horiz = FALSE, angle = 60, err.width = 0.4, err.size = 0.6,
border = "black", bw = FALSE, filter.legend = FALSE, ...)
```

Arguments

- **x**: one or a list of fitted object from codeVA function
- **grouping**: C by 2 matrix of grouping rule. If set to NULL, make it default.
- **type**: type of the plot to make
- **group_order**: list of grouped categories. If set to NULL, make it default.
- **err**: indicator of inclusion of error bars
- **CI**: Level of posterior credible intervals.
sample_size_print
Logical indicator for printing also the sample size for each sub-population labels.

xlab
Labels for the causes.

ylab
Labels for the CSMF values.

ylim
Range of y-axis.

title
Title of the plot.

horiz
Logical indicator indicating if the bars are plotted horizontally.

angle
Angle of rotation for the texts on x axis when horiz is set to FALSE

err_width
Size of the error bars.

err_size
Thickness of the error bar lines.

border
The color for the border of the bars.

bw
Logical indicator for setting the theme of the plots to be black and white.

filter_legend
Logical indicator for including all broad causes in the plot legend (default: FALSE) or filtering to only the broad causes in the data being plotted

Author(s)
Zehang Li, Tyler McCormick, Sam Clark

Maintainer: Zehang Li <lizehang@uw.edu>

Examples

data(RandomVA3)
test <- RandomVA3[1:200, ]
train <- RandomVA3[201:400, ]
fit1 <- codeVA(data = test, data.type = "customize", model = "InSilicoVA",
data.train = train, causes.train = "cause",
Nsim=1000, auto.length = FALSE)

fit2 <- codeVA(data = test, data.type = "customize", model = "InterVA",
data.train = train, causes.train = "cause", write=FALSE,
version = "4.02", HIV = "h", Malaria = "l")

fit3 <- codeVA(data = test, data.type = "customize", model = "Tariff",
data.train = train, causes.train = "cause",
nboot.sig = 100)

data(SampleCategory)
stackplotVA(fit1, grouping = SampleCategory, type ="dodge",
ylim = c(0, 1), title = "InSilicoVA")
stackplotVA(fit2, grouping = SampleCategory, type = "dodge",
ylim = c(0, 1), title = "InterVA4.02")
stackplotVA(fit3, grouping = SampleCategory, type = "dodge",
ylim = c(0, 1), title = "Tariff")
Index

*Topic **InSilicoVA**
  codeVA, 2
*Topic **InterVA4**
  codeVA, 2
  interVA.train, 11
*Topic **NBC4VA**
  codeVA, 2
*Topic **Tariff**
  codeVA, 2

codeVA, 2, 13
ConvertData, 4
ConvertData.phmrc, 2, 5
CSMF, 13

getCSMF, 7
getCSMF_accuracy, 8
getIndivProb, 9
getPHMRC_url, 9
getTopCOD, 10

insilico, 3, 4
InterVA, 3, 4
interVA.train, 3, 4, 11

openVA_status, 12
openVA_update, 13

plot.insilico, 13
plot.tariff, 13
plotVA, 13

stackplotVA, 14

tariff, 3, 4