Package ‘niaidMI’

February 11, 2022

Title    Markov Model Multiple Imputation for NIAID OS
Version  1.1.0
Date     2022-02-11
Description The implementation of Markov Model Multiple Imputation
            with the application to COVID-19 scale, NIAID OS.
License   MIT + file LICENSE
Depends  R (>= 3.5.0)
Suggests testthat
Imports   Rcpp (>= 1.0.3)
LinkingTo Rcpp
RoxygenNote 7.1.2
NeedsCompilation yes
Author    Nathan Morris [aut, cre],
           Chaoran Hu [aut],
           Eli Lilly and Company [cph]
Maintainer Nathan Morris <morris_nathan@lilly.com>
Repository CRAN
Date/Publication 2022-02-11 19:20:02 UTC

R topics documented:

niaidMI-package ........................................... 2
bootstrap_param_est ..................................... 2
get_emission ............................................. 3
impute .................................................... 4
impute_ref .............................................. 6
sim_data ............................................... 7

Index 9
Description

The implementation of Markov Model Multiple Imputation with the application to COVID-19 scale, NIAID OS.

Author(s)

- *maintainer, author* Nathan Morris <morris_nathan@lilly.com>
- *author* Chaoran Hu <hu_chaoran@lilly.com>

Usage

```
bootstrap_param_est(
    wide,          
    b,             
    days = paste0("D", 1:28),
    bin = rep(1, length(days) - 1),
    Em = get_emission(wide, days),
    tol = 1e-06,
    maxiter = 200,
    silent = FALSE
)
```

Arguments

- **wide** Data in wide format (i.e., each day is a column). See details.
- **b** Number of bootstrap samples to take.
- **days** Names of the columns that contain the score for each day.
- **bin** The assigned bin for pooling together information across transitions. Must be a numeric vector of length=(length(days)-1). By default all transitions are pooled together.
- **Em** Emission probabilities. Default should be used unless user is advanced.
- **tol** Tolerance for relative reduction the log-likelihood to determine convergence of the Baum-Welch algorithm.
- **maxiter** Maximum iterations before stopping the EM algorithm.
- **silent** Allows silencing some messages.
Details

States for each patient/day in 'wide' may be the following:

- Not missing: An integer from 1 to 8.
- Missing: NA
- Partially Missing: range which may be code as a characters string such as '[1,7]' or '[1,2]'. Such a character string indicates that while the actual value is unknown, it is known that the value falls within the specified range.

Generally the user will not need to call this function directly because it is called by the 'impute' function.

Value

A list object with the following components:

- **fit** Contains results of the primary model fit
- **boot** Contains results from the bootstrap model fit.
- **bin** The input.
- **s** Ignor. May be used in future.
- **days** From input.
- **Em** From input.

See Also

- impute

Examples

```r
test <- sim_data(100)
bs <- bootstrap_param_est(wide=test,b=2)
```

get_emission

**Emission probability evaluation.**

Description

Creates emission probabilities from data. Generally for internal niaidMI package use only.

Usage

```r
get_emission(wide, days)
```

Arguments

- **wide** Data in wide format (i.e., each day is a column).
- **days** Names of the columns that contain the score for each day.
Details

States for each patient/day in 'wide' may be the following:

- Not missing: An integer from 1 to 8.
- Missing: NA
- Partially Missing: Range which may be code as a characters string such as '[1,7]' or '[1,2]'. Such a character string indicates that while the actual value is unknown, it is known that the value falls within the specified range.

Generally the user will not need to call this function directly because it is called by the 'impute' function.

Value

Creates a 3 dimensional array that is "number of patients" x "number of days" x "8 NIAID stats." This array contains only 1 or 0 for each entry indicating if the state for a given day and individual is consistent with the data.

See Also

impute

Examples

test <- sim_data(200)
Em <- get_emission(wide=test, days=paste0("D", 1:28))

impute

Multiple Imputation for NIAID-OS.

Description

Imputes NIAID OS data using a Markov model.

Usage

impute(
  wide,
  m,
  by = NULL,
  days = paste0("D", 1:28),
  bin = rep(1, length(days) - 1),
  Em = get_emission(wide, days),
  listFormatOut = FALSE,
  tol = 1e-06,
  maxiter = 200,
  silent = FALSE
)


Arguments

- **wide**: Data in wide format (i.e., each day is a column).
- **m**: Number of imputations.
- **by**: Character vector with column names. Data will be broken up and imputed separately for every combination of values for specified columns in the data.
- **days**: Names of the columns that contain the score for each day. Columns should be in sequential order.
- **bin**: The assigned bin for pooling together information across transitions. Must be a numeric vector of length=(length(days)-1). By default all transitions are pooled together.
- **Em**: Emission probabilities. Generally the default should not be changed.
- **listFormatOut**: Return each imputed dataset in a list or combine into a single dataset.
- **tol**: Tolerance for relative reduction the log-likelihood to determine convergence of the Baum-Welch algorithm.
- **maxiter**: Maximum iterations before stopping the EM algorithm.
- **silent**: Allows silencing some messages.

Details

States for each patient/day in 'wide' may be the following:

- Not missing: An integer from 1 to 8.
- Missing: NA
- Partially Missing: Range which may be code as a characters string such as '[1,7]' or '[1,2]'. Such a character string indicates that while the actual value is unknown, it is known that the value falls within the specified range.

Value

If listFormatOut = TRUE, then a list will be returned with each element being an imputed data set. If listFormatOut = FALSE, then a single data.frame will be returned where IMP_ID column is created.

See Also

- `bootstrap_param_est`

Examples

```r
test <- sim_data(100)
bs <- impute(wide=test,m=2, by="strata", silent=TRUE)
```
impute_ref  

*Multiple Imputation for NIAID-OS using a reference.*

**Description**

Imputes NIAID OS data using a Markov model.

**Usage**

```r
impute_ref(
  wide,
  m,
  ref = NULL,
  by = NULL,
  days = paste0("D", 1:28),
  bin = rep(1, length(days) - 1),
  Em = get_emission(wide, days),
  listFormatOut = FALSE,
  tol = 1e-06,
  maxiter = 200,
  silent = FALSE
)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>wide</code></td>
<td>Data in wide format (i.e., each day is a column).</td>
</tr>
<tr>
<td><code>m</code></td>
<td>Number of imputations.</td>
</tr>
<tr>
<td><code>ref</code></td>
<td>Character vector with 1 column names. Specifies the reference group column. The reference group column must be logical, and there must be some (&gt;=15) members of the reference group in each strata.</td>
</tr>
<tr>
<td><code>by</code></td>
<td>Character vector with column names. Data will be broken up and imputed separately for every combination of values for specified columns in the data.</td>
</tr>
<tr>
<td><code>days</code></td>
<td>Names of the columns that contain the score for each day. Columns should be in sequential order.</td>
</tr>
<tr>
<td><code>bin</code></td>
<td>The assigned bin for pooling together information across transitions. Must be a numeric vector of length=(length(days)-1). By default all transitions are pooled together.</td>
</tr>
<tr>
<td><code>Em</code></td>
<td>Emission probabilities. Generally the default should not be changed.</td>
</tr>
<tr>
<td><code>listFormatOut</code></td>
<td>Return each imputed dataset in a list or combine into a single dataset.</td>
</tr>
<tr>
<td><code>tol</code></td>
<td>Tolerance for relative reduction the log-likelihood to determine convergence of the Baum-Welch algorithm.</td>
</tr>
<tr>
<td><code>maxiter</code></td>
<td>Maximum iterations before stopping the EM algorithm.</td>
</tr>
<tr>
<td><code>silent</code></td>
<td>Allows silencing some messages.</td>
</tr>
</tbody>
</table>
**Details**

States for each patient/day in `wide` may be the following:

- Not missing: An integer from 1 to 8.
- Missing: NA
- Partially Missing: Range which may be code as a characters string such as `[1,7]` or `[1,2]`. Such a character string indicates that while the actual value is unknown, it is known that the value falls within the specified range.

The reference based imputation uses a simple modification to the standard procedure. First, within each strata, the model fit and bootstrapping procedure is performed only using the patients that are in the reference group. Second, the imputation for all of the patients in that strata (both reference and treated patients) is performed using the parameters as estimated/simulated based on the patients in the reference group.

**Value**

If listFormatOut = TRUE, then a list will be returned with each element being an imputed data set. If listFormatOut = FALSE, then a single data.frame will be returned where IMP_ID column is created.

**See Also**

`bootstrap_param_est`

**Examples**

```r

test <- sim_data(300)
test$PBO = sample(c(TRUE, FALSE), size=nrow(test), replace = TRUE)
bs <- impute_ref(wide=test, ref="PBO", m=2, by="strata", silent=TRUE)
```

---

**Simulate data.**

Simulate data for the purpose of testing the package.

**Usage**

```r
sim_data(
  n,
  fit = list(Pri = list(c(0, 0, 0, 0.5, 0.25, 0.25, 0, 0)),
              Tran = list(matrix(c(0.74, 0.16, 0.04, 0.01, 0, 0, 0, 0.19, 0.63, 0.15, 0.04, 0.01, 0, 0, 0.05, 0.16, 0.61, 0.15, 0.04, 0.01, 0, 0, 0.02, 0.04, 0.15, 0.6, 0.15, 0.04, 0.01, 0, 0, 0.01, 0.04, 0.15, 0.6, 0.15, 0.04, 0, 0, 0.01, 0.04, 0.15, 0.63, 0, 0, 0, 0.01, 0.04, 0.16, 1), 8, 8))),
```
strata = c(s1 = 0.2, s2 = 0.8),
days = paste0("D", 1:28),
bin = rep(1, length(days) - 1),
drop_out_rate = 0.01,
sporatic_rate = 0.05,
miss_not_dead_rate = 0.2
)

Arguments

n: Number of samples.
fit: Contains transition and prior parameters.
strata: Named vector with probabilities to simulate strata.
days: Names of the output columns that contain the score for each day.
bin: The assigned bin for pooling together information across transitions. Must be a numeric vector of length=(length(days)-1). By default, all transitions are pooled together.
drop_out_rate: Parameter for simulating dropout. Rate is per day.
sporatic_rate: Parameter for simulating missing.
miss_not_dead_rate: Parameter to control missing but not dead rate.

Details

This simulation function is used to generated data for examples and testing of the package.

Value

Returns wide format data. Possible niad os values may be:

- Not missing: An integer from 1 to 8.
- Missing: NA
- Partially Missing: range which may be code as a characters string such as '[1,7]' or '[1,2]'. Such a character string indicates that while the actual value is unknown, it is known that the value falls within the specified range.

Examples

test <- sim_data(200)
Index

bootstrap_param_est, 2, 5, 7
get_emission, 3
impute, 3, 4, 4
impute_ref, 6
niaidMI-package, 2
sim_data, 7