Package 'BMLCimpute'

June 4, 2019

Type Package

Title BMLCimpute: Bayesian Multilevel Latent Class Models for the Multiple Imputation of Nested Categorical Data

Version 0.0.1

Date 2018-05-24

Author Davide Vidotto

Maintainer Davide Vidotto <d.vidotto@uvt.nl>

Description A package for the multiple imputation of single-level and nested categorical data by means of Bayesian Multilevel Latent Class models.

License GPL (>= 2)

LazyData true

Imports Rcpp (>= 0.12.5)

LinkingTo Rcpp

URL http://github.com/davidevdt/BMLCimpute

RoxygenNote 6.0.1

Archs x64

R topics documented:

BMLCimpute *BMLCimpute : Bayesian Multilevel Latent Class Models for the Multiple Imputation of Nested Categorical Data*

Description

A package for the multiple imputation of single-level and nested categorical data by means of Bayesian Multilevel Latent Class models.

Details

'BMLCimpute' allows researchers and users of categorical datasets with missing data to perform Multiple Imputation via Bayesian latent class models. Data can be either single- or multi-level. Model estimation and imputations are implemented via a Gibbs sampler run with the Rcpp package interface. The function `multilevelLCMI` performs the imputations. Prior to the imputation step, data should be processed with the function `convData`; the resulting list is then passed as input to the `multilevelLCMI`. Complete datasets are obtained via the `compData` function.

Functions

- `multilevelLCMI` for the imputations and model estimation (internally calls Rcpp code);
- `convData` for data preparation (preprocessing);
- `compData` for dataset completion.

Author(s)

D. Vidotto <d.vidotto@uvt.nl>

References

1 Vidotto D., Vermunt J.K., Van Deun K. (2018). 'Bayesian Multilevel Latent Class Models for the Multiple Imputation of Nested Categorical Data'. Journal of Educational and Beahvioral Statistics 43(5), 511-539.

Examples

```
## Not run:

library(BMLCimpute)

# Load data
data(simul_incomplete)

# Preprocess the Data
cd <- convData(simul_incomplete, GID = 1, UID = 2, var2 = 8:12)

# Model Selection
set.seed(1)
mmLC <- multilevelLCMI( convData = cd, L = 10, K = 10, it1 = 1000, it2 = 3000, it3 = 100, it.print = 250,
   v = 10, I = 0, pri2 = 1 / 10, pri1 = 1 / 15, priresp =  0.01, priresp2 = 0.01, random = TRUE,
    estimates = FALSE, count = TRUE, plot.loglik = FALSE, prec = 3, scale = 1.0)
```

```
# Select posterior maxima of the number of classes for the imputations
# (Other alternatives are possible, such as posterior modes or posterior quantiles)
L = max(which(mmLC[[12]] != 0))
K = max(apply(mmLC[[13]], 1, function(x) max(which( x != 0))), na.rm = TRUE)

# Perform 5 imutations on the dataset
mmLC <- multilevelLCMI( convData = cd, L = L, K = K, it1 = 2000, it2 = 4000, it3 = 100, it.print = 250,
    v = 10, I = 5, pri2 = 500, pri1 = 50, priresp = 0.01, priresp2 = 0.01, random = TRUE,
    estimates = FALSE, count = TRUE, plot.loglik = TRUE, prec = 4, scale = 1.0)

# Obtain the dataset completed with the first set of imputations (ind = 1)
complete_data = compData( convData = cd, implev1 = mmLC[[1]], implev2 = mmLC[[2]], ind = 1 )


## End(Not run)
```

`compData` *Create a completed dataset with the imputated values of BMLCimpute (compData)*

Description

Plug the imputations obtained with multilevelLCMI into the original dataset, in order to obtain a completed dataset.

A package for the multiple imputation of single-level and nested categorical data by means of Bayesian Multilevel Latent Class models.

Usage

```
compData(convData, implev1, implev2 = NULL, ind)
```

Arguments

`convData`	Ouptut list produced by the 'convData' function
`implev1`	The set of imputations for the level-1 variables provided by the 'multilevelLCMI' function. It corresponds to the first element of the list returned by 'multilevelLCMI'.
`implev2`	The set of imputations for the level-2 variables (when present) provided by the 'multilevelLCMI' function. It corresponds to the second element of the list returned by 'multilevelLCMI'.
`ind`	The imputation index; an integer value that ranges from 1 to M, where M is the number of imputations computed by `BMLCimpute`.

Details

This function takes a 'convData' list, the imputations provided by 'multilevelLCMI' and the imputation index (ind in 1,..., M where M is the number of imputations) and returns the completed dataset.

'BMLCimpute' allows researchers and users of categorical datasets with missing data to perform Multiple Imputation via Bayesian latent class models. Data can be either single- or multi-level. Model estimation and imputations are implemented via a Gibbs sampler run with the Rcpp package

interface. The function multilevelLCMI performs the imputations. Prior to the imputation step, data should be processed with the function convData; the resulting list is then passed as input to the multilevelLCMI. Complete datasets are obtained via the compData function.

Value

The imputed dataset

Functions

- multilevelLCMI for the imputations and model estimation (internally calls Rcpp code);
- convData for data preparation (preprocessing);
- compData for dataset completion.

Author(s)

D. Vidotto <d.vidotto@uvt.nl>

BMLCimpute : Bayesian Multilevel Latent Class Models for the Multiple Imputation of Nested Categorical Data

Examples

```
## Not run:

library(BMLCimpute)

# Load data
data(simul_incomplete)

# Preprocess the Data
cd <- convData(simul_incomplete, GID = 1, UID = 2, var2 = 8:12)

# Model Selection
set.seed(1)
mmLC <- multilevelLCMI( convData = cd, L = 10, K = 10, it1 = 1000, it2 = 3000, it3 = 100, it.print = 250,
   v = 10, I = 0, pri2 = 1 / 10, pri1 = 1 / 15, priresp =  0.01, priresp2 = 0.01, random = TRUE,
    estimates = FALSE, count = TRUE, plot.loglik = FALSE, prec = 3, scale = 1.0)

# Select posterior maxima of the number of classes for the imputations
# (Other alternatives are possible, such as posterior modes or posterior quantiles)
L = max(which(mmLC[[12]] != 0))
K = max(apply(mmLC[[13]], 1, function(x) max(which( x != 0))), na.rm = TRUE)

# Perform 5 imutations on the dataset
mmLC <- multilevelLCMI( convData = cd, L = L, K = K, it1 = 2000, it2 = 4000, it3 = 100, it.print = 250,
    v = 10, I = 5, pri2 = 500, pri1 = 50, priresp = 0.01, priresp2 = 0.01, random = TRUE,
    estimates = FALSE, count = TRUE, plot.loglik = TRUE, prec = 4, scale = 1.0)

# Obtain the dataset completed with the first set of imputations (ind = 1)
complete_data = compData( convData = cd, implev1 = mmLC[[1]], implev2 = mmLC[[2]], ind = 1 )


## End(Not run)
```

convData *Prepare data for imputations with BMLCimpute (convData)*

Description

This function takes a categorical dataset as input (categories can be denoted by numbers) and returns a list of objects that will be used by the 'multilevelLCMI' function to perform the imputations.

A package for the multiple imputation of single-level and nested categorical data by means of Bayesian Multilevel Latent Class models.

Usage

```
convData(dat, GID = NULL, UID = NULL, var2 = NULL)
```

Arguments

dat	Raw (categorical) data frame with missing data. It can also be a data matrix. The GID and UID arguments, if passed to the function, must be in the first two columns of the dataset.
GID	Group (level-2 unit) indicator (expressed as column number corresponding to the group ID in the dataset). It can be omitted in single-level datasets.
UID	Lower-level unit indicator (expressed as column number corresponding to the unit ID in the dataset). Optional.
var2	Higher-level (group-specific) variables (expressed as a vector of column numbers in the dataset corresponding to the variables measured at the higher levels). Optional.

Details

Convert a raw categorical dataset with missing data into one ready to be imputed with the multilevelLCMI function. In particular, the function will transform factor variables into numeric ones, where numbers denote a different category. A coding list is returned along with the converted dataset.

'BMLCimpute' allows researchers and users of categorical datasets with missing data to perform Multiple Imputation via Bayesian latent class models. Data can be either single- or multi-level. Model estimation and imputations are implemented via a Gibbs sampler run with the Rcpp package interface. The function multilevelLCMI performs the imputations. Prior to the imputation step, data should be processed with the function convData; the resulting list is then passed as input to the multilevelLCMI. Complete datasets are obtained via the compData function.

Value

A convData object, a list containing the following items:

convDat	The converted dataset
codLev1	List containing the new (and original) scores which will be used for the imputations (Level-1 variables).
codLev1	Vector containing the number of categories observed for each variable (Level-1 variables).

nCatLev1	Vector containing the number of categories observed for each variable (Level-1 variables).
codLev2	List containing the new (and original) scores which will be used for the imputations (Level-2 variables).
nCatLev2	List containing the new (and original) scores which will be used for the imputations (Level-2 variables).
GroupIDs	Matrix containing original and new Group ID's.
GID	The column Group ID number (as entered by the user).
UID	The column Unit ID number (as entered by the user).
var2	The column numbers for level-2 variables (as entered in the input).
doVar2	Boolean. Shall the BMLC model impute variables at level-2? (Result of `!is.null(var2)`).
namesLev1	Vector of variable names (level-1 variables).
namesLev2	Vector of variable names (level-2 variables).
GroupName	Group ID variable name.
CaseName	Unit ID variable name.
caseID	Unit ID vector (re-permuted).
sort_	Vector containing the original permutation of the dataset rows.

Functions

- `multilevelLCMI` for the imputations and model estimation (internally calls Rcpp code);
- `convData` for data preparation (preprocessing);
- `compData` for dataset completion.

Author(s)

D. Vidotto <d.vidotto@uvt.nl>

BMLCimpute : Bayesian Multilevel Latent Class Models for the Multiple Imputation of Nested Categorical Data

Examples

```
## Not run:

library(BMLCimpute)

# Load data
data(simul_incomplete)

# Preprocess the Data
cd <- convData(simul_incomplete, GID = 1, UID = 2, var2 = 8:12)

# Model Selection
set.seed(1)
mmLC <- multilevelLCMI( convData = cd, L = 10, K = 10, it1 = 1000, it2 = 3000, it3 = 100, it.print = 250,
   v = 10, I = 0, pri2 = 1 / 10, pri1 = 1 / 15, priresp =  0.01, priresp2 = 0.01, random = TRUE,
    estimates = FALSE, count = TRUE, plot.loglik = FALSE, prec = 3, scale = 1.0)

# Select posterior maxima of the number of classes for the imputations
# (Other alternatives are possible, such as posterior modes or posterior quantiles)
```

```
L = max(which(mmLC[[12]] != 0))
K = max(apply(mmLC[[13]], 1, function(x) max(which( x != 0))), na.rm = TRUE)

# Perform 5 imutations on the dataset
mmLC <- multilevelLCMI( convData = cd, L = L, K = K, it1 = 2000, it2 = 4000, it3 = 100, it.print = 250,
    v = 10, I = 5, pri2 = 500, pri1 = 50, priresp = 0.01, priresp2 = 0.01, random = TRUE,
    estimates = FALSE, count = TRUE, plot.loglik = TRUE, prec = 4, scale = 1.0)

# Obtain the dataset completed with the first set of imputations (ind = 1)
complete_data = compData( convData = cd, implev1 = mmLC[[1]], implev2 = mmLC[[2]], ind = 1 )


## End(Not run)
```

`multilevelLCMI` *Multilevel Latent Class models for the Multiple Imputation of Categorical Data (multilevelLCMI).*

Description

Perform single- and multi-level multiple imputation of categorical data through single/multi level Bayesian Latent Class models.

Usage

```
multilevelLCMI(convData, L, K, it1, it2, it3, it.print, v, I = 5, pri2 = 1,
   pri1 = 1, priresp = 1, priresp2 = 1, random = TRUE, estimates = TRUE,
   count = FALSE, plot.loglik = FALSE, prec = 3, scale = 1)

## Default S3 method:
multilevelLCMI(convData, L, K, it1, it2, it3, it.print, v, I = 5, pri2 = 1,
     pri1 = 1, priresp = 1, priresp2 = 1, random = TRUE, estimates = TRUE,
     count = FALSE, plot.loglik = FALSE, prec = 3, scale = 1)

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

Arguments

`convData`	Dataset produced as output by the `convData` function.
`L`	Number of higher-level mixture components. When L=1, single-level Latent Class multiple imputation is performed.
`K`	Number of Latent Classes at the lower-level.
`it1`	Number of Gibbs sampler iterations for the burn-in (must be larger than 0).
`it2`	Number of Gibbs sampler iterations for the imputations.
`it3`	Every `it3` iterations, the sampler stores new parameter estimates for the calculation of psoterior estimates. Meaningful only when `estimates=TRUE`.
`it.print`	Every `it.print` iterations, the state of the Gibbs sampler is screen-printed.

v	The Gibbs sampler will produce the first set of imputations at the iteration number (it1+V), where V~Unif(1,v). Subsequent imputations are automatically spaced from each others across the remaining iterations, so that the last imputation (imputation I) occurs at the iteration number (it1+it2).
I	Number of imputations to be performed.
pri2	Hyperparameter value for the higher-level mixture probabilities. Default to 1.
pri1	Hyperparameter value for the lower-level mixture probabilities. Default to 1.
priresp	Hyperparameter value for the lower-level conditional response probabilities. Default to 1.
priresp2	Hyperparameter value for the higher-level conditional response probabilities. Default to 1.
random	Logical. Should the model parameters be initialized at random values? If TRUE, parameters are initialized through draws from uniform Dirichlet distributions. If FALSE, parameters are initialized to be equal to 1/D, with D the number of categories in the (observed/latent) variable of interest.
estimates	Logical. If TRUE, the function returns the posterior means of the model parameters. Default to TRUE.
count	Logical. Should the output include the posterior distribution of L and K? (only K if L=1)
plot.loglik	Logical. Should the output include a traceplot of the log-likelihood ratios obtained through the Gibbs sampler iterations? Helpful for assessing convergence.
prec	When estimates=TRUE, prec defines the number of digits with which the estimates are returned.
scale	Re-scale the log-likelihood value by a factor equal to scale; this parameter is useful to avoid underflow in the calculation of the log-likelihood (which can occur in large datasets) and consequently to prevent error messages for the visualization of the log-likelihood traceplot. This parameter is meaningful only when plot.loglik is set to TRUE. Default value equal to 1.0.
x	A multilevelLCMI object (print method).
...	Not used.

Details

Function for performing Multiple Imputation with the Bayesian Multilevel Latent Class Model. The model takes the list produced by the 'convData' function as input, in which the dataset converted and prepared for the imputations is present, along with other parameters specified by the user (e.g., number of latent classes and specification of the prior distribution hyperparameters). The function can also offer (when the corresponding boolean parameter is activated) a graphical representation of the posterior distribution of the number of occupied classes during the Gibbs sampler iterations. In this way, multilevelLCMI' can also perform model selection in a pre-imputation stage. For model selection, set count=TRUE. Symmetric Dirichlet priors are used.

Value

A multilevelLCMI object, a list containing:

imp	Set of imputations for the level-1 variables and units.
imp2	Set of imputations for the level-2 variables and units.
piL	posterior means of the level-2 class probabilities. Calculated only if estimates = TRUE.

piLses	Posterior standard deviations of the level-2 class probabilities. Calculated only if estimates = TRUE.
piK	Posterior means of the level-1 class probabilities. Calculated only if estimates = TRUE.
piKses	Posterior standard deviations of the level-2 class probabilities. Calculated only if estimates = TRUE.
picondlev1	Posterior means of the level-1 conditional probabilities. Calculated only if estimates = TRUE.
picondlev1ses	Posterior standard deviations of the level-1 conditional probabilities. Calculated only if estimates = TRUE.
picondlev2	Posterior means of the level-2 conditional probabilities. Calculated only if estimates = TRUE.
picondlev1ses	Posterior standard deviations of the level-2 conditional probabilities. Calculated only if estimates = TRUE.
DIC	DIC index for the BMLC model. Calculated only if estimates = TRUE.
freqL	Posterior distribution of the number of latent classes at level-2. Calculated only if count is set to TRUE.
freqK	Posterior distribution of the number of latent classes at level-1. Calculated only if count is set to TRUE.
time	Running time of the Gibbs sampler iterations.

Author(s)

D. Vidotto <d.vidotto@uvt.nl>

References

1 Vidotto D., Vermunt J.K., Van Deun K. (2018). 'Bayesian Multilevel Latent Class Models for the Multiple Imputation of Nested Categorical Data'. Journal of Educational and Beahvioral Statistics 43(5), 511-539.

Examples

```
## Not run:

library(BMLCimpute)

# Load data
data(simul_incomplete)

# Preprocess the Data
cd <- convData(simul_incomplete, GID = 1, UID = 2, var2 = 8:12)

# Model Selection
set.seed(1)
mmLC <- multilevelLCMI( convData = cd, L = 10, K = 10, it1 = 1000, it2 = 3000, it3 = 100, it.print = 250,
   v = 10, I = 0, pri2 = 1 / 10, pri1 = 1 / 15, priresp =  0.01, priresp2 = 0.01, random = TRUE,
    estimates = FALSE, count = TRUE, plot.loglik = FALSE, prec = 3, scale = 1.0)

# Select posterior maxima of the number of classes for the imputations
# (Other alternatives are possible, such as posterior modes or posterior quantiles)
L = max(which(mmLC[[12]] != 0))
K = max(apply(mmLC[[13]], 1, function(x) max(which( x != 0))), na.rm = TRUE)

# Perform 5 imutations on the dataset
```

```
mmLC <- multilevelLCMI( convData = cd, L = L, K = K, it1 = 2000, it2 = 4000, it3 = 100, it.print = 250,
    v = 10, I = 5, pri2 = 500, pri1 = 50, priresp = 0.01, priresp2 = 0.01, random = TRUE,
    estimates = FALSE, count = TRUE, plot.loglik = TRUE, prec = 4, scale = 1.0)

# Obtain the dataset completed with the first set of imputations (ind = 1)
complete_data = compData( convData = cd, implev1 = mmLC[[1]], implev2 = mmLC[[2]], ind = 1 )


## End(Not run)
```

`simul` *Toy multilevel categorical dataset (simul)*

Description

The dataset contains 1000 level-1 artificial observations grouped into 200 level-2 groups. The dataset consists of: one identifier for the group, one identifier for the level-1 units, six level-1 binary variables (5 predictors and 1 response), five level-2 binary variables. The dataset was generated following the simulation study settings in Vidotto, Vermunt, van Deun (2018). The dataset can be used for performance evaluation of multilevel imputation methods applied on the `simul_incomplete` dataset.

Usage

```
data(simul)
```

Format

A data frame with 13 columns and 1,000 rows (level-1 units) from 200 groups (level-2 units):

[,1]	GroupID	numeric	Level-2 unit Identifier
[,2]	UnitID	numeric	Level-1 unit Identifier
[,3]	X1,...,X5	binary	Level-1 predictors
[,4]	Z1,...,Z5	binary	Level-2 predictors
[,5]	Y	binary	Response Variable

References

1 Vidotto D., Vermunt J.K., Van Deun K. (2018). 'Bayesian Multilevel Latent Class Models for the Multiple Imputation of Nested Categorical Data'. Journal of Educational and Beahvioral Statistics 43(5), 511-539.

Examples

```
## Not run:

library(BMLCimpute)

data(simul)
```

```
## End(Not run)
```

`simul_incomplete` *Toy multilevel categorical dataset with missing entries (simul_incomplete)*

Description

The dataset contains 1000 level-1 artificial observations grouped into 200 level-2 groups. The dataset consists of: one identifier for the group, one identifier for the level-1 units, six level-1 binary variables (5 predictors and 1 response), five level-2 binary variables. The level-1 variables X1, X2, X5 and the level-2 variables Z1 and Z3 have missing observations, generated through a MAR mechanism. The dataset and the missing data were generated following the simulation study settings in Vidotto, Vermunt, van Deun (2018). Dataset used for testing of multilevel imputation models; model performance can be assessed by using the `simul` dataset.

Usage

```
data(simul_incomplete)
```

Format

A data frame with 13 columns and 1,000 rows (level-1 units) from 200 groups (level-2 units):

[,1]	GroupID	numeric	Level-2 unit Identifier
[,2]	UnitID	numeric	Level-1 unit Identifier
[,3]	X1,...,X5	binary	Level-1 predictors
[,4]	Z1,...,Z5	binary	Level-2 predictors
[,5]	Y	binary	Response Variable

References

1 Vidotto D., Vermunt J.K., Van Deun K. (2018). 'Bayesian Multilevel Latent Class Models for the Multiple Imputation of Nested Categorical Data'. Journal of Educational and Beahvioral Statistics 43(5), 511-539.

Examples

```
## Not run:

library(BMLCimpute)

data(simul_incomplete)


## End(Not run)
```

Index