Package ‘NPBayesImpute’

March 25, 2015

Type Package
Title Non-Parametric Bayesian Multiple Imputation for Categorical Data
Version 0.5
Date 2015-03-22
Author Quanli Wang, Daniel Manrique-Vallier, Jerome P. Reiter and Jingchen Hu
Maintainer Quanli Wang <quanli@stat.duke.edu>
Description These routines create multiple imputations of missing at random categorical data, with or without structural zeros. Imputations are based on Dirichlet process mixtures of multinomial distributions, which is a non-parametric Bayesian modeling approach that allows for flexible joint modeling.
License GPL (>= 3)
Depends methods, Rcpp (>= 0.10.2)
LinkingTo Rcpp
RcppModules clcm
NeedsCompilation yes
Repository CRAN
Date/Publication 2015-03-23 21:17:08

R topics documented:

NPBayesImpute-package .................................................. 2
CreateModel ............................................................... 3
GetDataFrame .............................................................. 4
GetMCZ ................................................................. 5
Lcm ................................................................. 6
MCZ ................................................................. 6
Rcpp_Lcm .............................................................. 6
Rcpp_Lcm-class ........................................................ 7
X ................................................................. 9

Index 10
NPBayesImpute-package  Bayesian Multiple Imputation for Large-Scale Categorical Data with Structural Zeros

Description

This package implements a fully Bayesian, joint modeling approach to multiple imputation for categorical data based on latent class models with structural zeros. The idea is to model the implied contingency table of the categorical variables as a mixture of independent multinomial distributions, estimating the mixture distributions nonparametrically with Dirichlet process prior distributions. Mixtures of multinomials can describe arbitrarily complex dependencies and are computationally expedient, so that they are effective general purpose multiple imputation engines. In contrast to other approaches based on loglinear models or chained equations, the mixture models avoid the need to specify (potentially many) models, which can be a very time-consuming task with no guarantee of a theoretically coherent set of models. The package is designed to include for structural zeros, i.e., certain combinations of variables are not possible a priori.

Details

<table>
<thead>
<tr>
<th>Package</th>
<th>NPBayesImpute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Package</td>
</tr>
<tr>
<td>Version</td>
<td>0.1</td>
</tr>
<tr>
<td>Date</td>
<td>2014-04-05</td>
</tr>
<tr>
<td>License</td>
<td>GPL(&gt;=3)</td>
</tr>
</tbody>
</table>

Author(s)

Quanli Wang, Daniel Manrique-Vallier, Jerome P. Reiter and Jingchen Hu

Maintainer: Quanli Wang<quanli@stat.duke.edu>

References


Manrique-Vallier, D. and Reiter, J.P. (2014), "Bayesian Multiple Imputation for Large-Scale Categorical Data with Structural Zeros", Survey Methodology.
CreateModel

Examples

```r
require(NPBayesImpute)
# Please use NYexample data set for a more realistic example
data('NYMockexample')

#create the model
model <- CreateModel(X,MCZ,10,10000,0.25,0.25)

# run 1 burnins, 2 mcmc iterations and thin every 2 iterations
model$Run(1,2,2)

# retrieve parameters from the final iteration
result <- model$snapshot

# convert ImputedX matrix to dataframe, using proper factors/names etc.
ImputedX <- GetDataFrame(result$ImputedX,X)
# View(ImputedX)

# Most exhaustive examples can be found in the demo below
# demo(example_short)
# demo(example)
```

CreateModel

Create and initialize the Repp_Lcm model object

Description

CreateModel creates and initializes an Repp_Lcm Repp_Lcm-class object for non-parametric multiple imputation of discrete multivariate categorical data with or without structural zeros.

Usage

CreateModel(X, MCZ, K, Nmax, aalpha, balpha)

Arguments

- **X**: a data frame with the dataset with missing values. All variables must be unordered factors.
- **MCZ**: a dataframe with the definition of the structural zeros. Placeholder components are represented with NAs. Variables in MCZ must be factors with the same levels as X. Rows do not need to define disjoint regions of the contingency table. See Manrique-Vallier and Reiter (2014) for details of the definition of structural zeros. MCZ should be set to NULL when there are no structure zeros.
- **K**: the maximum number of mixture components.
- **Nmax**: An upper truncation limit for the augmented sample size. This parameter will be ignored(set to 0) when there is no structural zeros.
- **aalpha**: the hyper parameter ‘a’ for alpha in stick-breaking prior distribution.
- **balpha**: the hyper parameter ‘b’ for alpha in stick-breaking prior distribution.
Details

This should be the first function one should call to use the library. The returned model object will be referenced in all subsequent calls.

Value

CreateModel returns an Rcpp_lcm object. The returned model object will be referenced in all subsequent calls.

References


Manrique-Vallier, D. and Reiter, J.P. (2014), "Bayesian Multiple Imputation for Large-Scale Categorical Data with Structural Zeros", Survey Methodology.

Examples

```r
require(NPBayesImpute)
#Please use NYexample data set for a more realistic example
data(NYMockexample')

#create the model
model <- CreateModel(X,MCZ,10,10000,0.25,0.25)

#run 1 burnins, 2 mcmc iterations and thin every 2 iterations
model$Run(1,2,2)

#retrieve parameters from the final iteration
result <- model$snapshot

#convert ImputedX matrix to dataframe, using proper factors/ names etc.
ImputedX <- GetDataFrame(result$ImputedX,X)
#View(ImputedX)
```

---

**DataFrame**

Convert imputed data to a dataframe, using the same setting from original input data.

Description

This is a utility function to convert the imputed data matrix to a dataframe. This function will be implemented as a RCPP internal function later on.
GetMCZ

Usage

GetDataFrame(dest, from, cols = 1:NCOL(from))

Arguments

dest
from
cols

the imputed output data matrix.
the original input dataframe.
optional. Always use default for now.

Value

The returned dataframe object for imputed data.

Examples

require(NPBayesImpute)
# Please use NYexample data set for a more realistic example
data("NYMockexample")

# create the model
model <- CreateModel(x,MCZ,10,10000,0.25,0.25)

# run 1 burnins, 2 mcmc iterations and thin every 2 iterations
model$Run(1,2,2)

# retrieve parameters from the final iteration
result <- model$snapshot

# convert ImputedX matrix to dataframe, using proper factors/names etc.
ImputedX <- GetDataFrame(result$ImputedX,X)
# View(ImputedX)

GetMCZ

Convert disjointed structural zeros to a dataframe, using the same setting from original structural zero data.

Description

This is a utility function to convert the disjointed structural zero matrix to a dataframe. This function will be implemented as a RCPP internal function later on.

Usage

GetMCZ(dest, from, mcz, cols = 1:NCOL(from))
Rcpp_Lcm

Arguments

dest the output data matrix for disjointed structural zeros.
from the original input dataframe.
mcz the original input dataframe for structural zeros.
cols optional. Always use default for now.

Value
The returned dataframe object for disjointed structural zeros.

References


Manrique-Vallier, D. and Reiter, J.P. (2014), "Bayesian Multiple Imputation for Large-Scale Categorical Data with Structural Zeros", Survey Methodology.

Lcm RCPP implementation of the library

Description

Rcpp_Lcm-class

MCZ Example dataframe for structural zeros.

Description

Example dataframe for structural zeros.

Rcpp_Lcm RCPP implementation of the library

Description

Rcpp_Lcm-class
Rcpp_Lcm-class

Class “Rcpp_Lcm”

Description

This class implements the MCMC sampler for non-parametric imputation of discrete multivariate data described in Manrique-Vallier and Reiter (2014). It provides methods for updating and monitoring the sampler.

Details

Rcpp_lcm objects should be created with CreateModel. Please see the examples in the demo folder for more detailed explanation on model fitting and parameter tracing.

Extends

Class "C++Object", directly.

All reference classes extend and inherit methods from "envRefClass".

Fields

CurrentIteration: the total number of iterations that have been run so far.
EnableTracer: to check tracer status or to enable/disable the tracer.
MCZ: the disjointed structural zero matrix.
snapshot: retrieve a list with the current state of all the parameters in the sampler, including the imputed sample. A call the the "snapshot" method returns a list with the following components:
  alpha: the concentration parameter of the stick breaking prior.
  k_star: the effective number number of latent classes (mixture components)
  Nmis: the size of the augmented sample.
  nu: a vector with the mixture weights
  z: a matrix with the current latent class assignment of each member of the sample
  ImputedX: the current raw imputed dataset. Use GetDataFrame to convert the raw data to a data frame of factors as defined in the input data set.
  psi: The conditional multinomial probabilities. A Lmax * K * J array, where Lmax is the maximum number of levels of all discrete factors in the dataset, J is the number of factors in the dataset, and K is the number of latent classes. Since variables might have different numbers of levels, unused entries in the first dimension are filled with NAs to complete Lmax.
  traceable: list of model parameters that can be traced by the tracer.
  traced: list of model parameters that are traced.
Methods

SetTrace(paralist, num_of_iterations): set parameters to be traced.
  - **paralist**: a list of parameters to be traced.
  - **num_of_iterations**: the maximum number of traced iterations.

Run(burnin, iter, thinning): run MCMC iterations.
  - **burnin**: number of burn in iterations.
  - **iter**: number of MCMC iterations.
  - **thinning**: thinning parameter.

Resume(): resume from an interrupted call to run method.

Parameters(paralist): retrieve a selected list of model parameters from last MCMC iteration.
  - **paralist**: a list of parameters to be traced.

GetTrace(): retrieve all traced iterations. Returns a list with all the parameters set using the method SetTrace(). See description of snapshotreference method for a description of the parameters.

References


Manrique-Vallier, D. and Reiter, J.P. (2014), "Bayesian Multiple Imputation for Large-Scale Categorical Data with Structural Zeros", Survey Methodology.

Examples

```
require(npbayesImpute)
#Please use NYexample data set for a more realistic example
data("NYMockexample")

#create the model
model <- CreateModel(X,MCZ,10,10000,0.25,0.25)

#run 1 burnins, 2 mcmc iterations and thin every 2 iterations
model$Run(1,2,2)

#retrieve parameters from the final iteration
result <- model$snapshot

#convert imputedX matrix to dataframe, using proper factors/names etc.
ImputedX <- GetDataFrame(result$ImputedX,X)
#View(ImputedX)
```
Example dataframe for input categorical data with missing values.
Index

*Topic classes
  Rcpp_Lcm-class, 7
*Topic package
  NPBayesImpute-package, 2

C++Object, 7
CreateModel, 3, 7

envRefClass, 7
GetDataFrame, 4, 7
GetMCZ, 5

Lcm, 6
MCZ, 6

NPBayesImpute (NPBayesImpute-package), 2
NPBayesImpute-package, 2

Rcpp_Lcm, 6
Rcpp_Lcm-class, 3, 6, 7

X, 9