Package 'MixedDataImpute'

February 7, 2016

Type Package

Title Missing Data Imputation for Continuous and Categorical Data using Nonparametric Bayesian Joint Models

Version 0.1

Date 2016-02-02

Author Jared S. Murray

Maintainer Jared S. Murray <jsmurray@stat.cmu.edu>

Description Missing data imputation for continuous and categorical data, using nonparametric Bayesian joint models (specifically the hierarchically coupled mixture model with local dependence described in Murray and Reiter (2015); see 'citation(``MixedDataImpute'')' or http://arxiv.org/abs/1410.0438). See '?hcmm_impute' for example usage.

License GPL-3

Imports methods

Depends gdata, Rcpp (>= 0.11), R (>= 3.1.0)

LinkingTo Rcpp, RcppArmadillo, BH

RcppModules HCMMLD

LazyData true

NeedsCompilation yes

Repository CRAN

Date/Publication 2016-02-07 09:20:13

R topics documented:

hcmm_hyperpar *Generate a list of hyperparameters*

Description

Generates a list of hyperparameters for use in hcmm_impute. Specifying only hcmmdat or q AND cx will generate default values (see citation).

Usage

```
hcmm_hyperpar(hcmmdat = NULL, q = ncol(hcmmdat$Y), cx = hcmmdat$cx,
  alpha_a = 0.5, alpha_b = 0.5, beta_x_a = 0.5, beta_x_b = 0.5,
  beta_y_a = 0.5, beta_y_b = 0.5, tau_a = 0.5, tau_b = 0.5, v = q + 1,
  w = q + 2, Sigma0 = diag(1, q)/v, gamma = 1/cx, sigma2_0beta = 10)
```

Arguments

hcmmdat	An hcmm_data object
q	The number of continuous variables
cx	A length p vector (where p is the number of categorical variables). cx[j] is the number of distinct values taken by X[,j]
alpha_a,alpha_b	Gamma prior on top-level concentration parameter, where E(alpha) = alpha_a/alpha_b
beta_x_a,beta_x_b	Gamma prior on X model concentration parameter
beta_y_a,beta_y_b	Gamma prior on Y model concentration parameter
tau_a,tau_b	Gamma prior on coefficient precision parameters
v,w	Degree of freedom parameters in the hierarchical inverse-Wishart/Wishart prior
Sigma0	Centering matrix in the hierarchical inverse-Wishart/Wishart prior
gamma	Parameter of the symmetric Dirichlet priors in the product multinomial kernel. (Should be a length p vector.)
sigma2_0beta	Variance of the prior on B0

Value

A list of hyperparameters

hcmm_impute *Generate multiply imputed datasets*

Description

Imputations are generated using nonparametric Bayesian joint models (specifically the hierarchically coupled mixture model with local dependence described in Murray and Reiter (2015); see citation(MixedDataImpute) or http://arxiv.org/abs/1410.0438).

Usage

```
hcmm_impute(X, Y, kz, kx, ky, hyperpar = NULL, num.impute, num.burnin,
  num.skip, thin.trace = -1, status = 50)
```

Arguments

X	A data frame of categorical variables (as factors)
Y	A matrix or data frame of continuous variables
kz	Number of top-level clusters
kx	Number of X-model clusters
ky	Number of Y-model clusters
hyperpar	A list of hyperparameter values (see hcmm_hyperpar)
num.impute	Number of imputations
num.burnin	Number of MCMC burn-in iterations
num.skip	Number of MCMC iterations between saved imputations
thin.trace	If negative, only save the num.impute datasets. If positive, save summaries of the model state at every thin.trace iterations for diagnostic purposes.
status	Interval at which to print status messages

Value

A list with three elements:

imputations A list of length num.impute. Each element is an imputed dataset.

trace MCMC output (currently the component sizes for the three mixture indices)

model An interface to the C++ object containing the current state

Examples

```
## Not run:
library(MixedDataImpute)
library(mice) # For the functions implementing combining rules

data(sipp08)
```

```
set.seed(1)
n = 1000
s = sample(1:nrow(sipp08), n)

Y = sipp08[s,1:2]
Y[,1] = log(Y[,1]+1)
X = sipp08[s,-c(1:2,9)] # Also removes occ code, which has ~23 levels

# MCAR with probability 0.2, for illustration purposes (not matching the paper)

Y[runif(n)<0.2,1] = NA
Y[runif(n)<0.2,2] = NA
for(j in 1:ncol(X)) X[runif(n)<0.2,j] = NA

kz = 15
ky = 60
kx = 90

num.impute = 5
num.burnin = 10000
num.skip = 1000
thin.trace = 10

imp = hcmm_impute(X, Y, kz=kz, kx=kx, ky=ky,
                  num.impute=num.impute, num.burnin=num.burnin,
                  num.skip=num.skip, thin.trace=thin.trace)

# Example of getting MI estimates for a regression, using the
# pooling functions in mice
form = total_earnings~age+I(age^2) + sex*I(own_kid!=0)

fits = lapply(imp$imputations, function(dat) lm(form, data=dat))
pooled_ests = pool(as.mira(fits))
summary(pooled_ests)

# original, complete data estimates for comparison
comdat = sipp08[s,]
comdat[,1] = log(comdat[,1]+10)
summary(lm(form, data=comdat))

# true population values for comparison
pop = sipp08
pop[,1] = log(pop[,1]+10)
summary(lm(form, data=pop))


## End(Not run)
```

`MixedDataImpute` *Missing data imputation for continuous and categorical data using nonparametric Bayesian joint models*

Description

Missing data imputation for continuous and categorical data, using nonparametric Bayesian joint models (specifically the hierarchcially coupled mixture model with local dependence described in Murray and Reiter (2015); see citation(NPBayesMixedDataImpute)).

---

prepare_data *Prepare a dataset for imputation*

Description

Prepares a dataset for imputation by mapping factor levels to integers and scaling Y. Primarily used by `hcmm_impute` internally

Usage

```
prepare_data(X, Y, init = TRUE)
```

Arguments

`X`	A data frame of categorical variables (as factors)
`Y`	A matrix or data frame of continuous variables
`init`	If `TRUE`, initialize missing values (from a marginal bootstrap of observed values)

Value

An object of class `hcmm_data`

---

remap_imputations *Map raw imputations back to original scale/factor labels.*

Description

Map raw imputations back to original scale (for continuous data) or factor labels. Most users can ignore this function, which is primarily used by `hcmm_impute` internally.

Usage

```
remap_imputations(Ximp, Yimp, hcmmdat)
```

Arguments

`Ximp`	A raw imputed X matrix
`Yimp`	A raw imputed Y matrix
`hcmmdat`	An `hcmm_data` object, used to recode/rescale the raw imputations

Value

A list with elements `X` and `Y` (transformed imputations).

`sipp08` *Housholder earnings from the SIPP*

Description

A dataset extracted from the first wave of the 2008 Survey of Income and Program Participation, constructed by retaining all heads of household with positive earnings from employment and complete cases. This is useful as a realistic population for studying imputation routines.

Usage

```
sipp08
```

Format

A data frame with 30507 rows and 13 variables:

total_earnings Total monthly earnings from employment (USD)

age Age (years)

sex Sex

race Single race/ethnicity

marital_status Marital Status

born_us Born in or outside the United States

own_kid Number of respondent's own children living in the household

edu_level Level of education

occ Occupation code for primary job (recoded from variable TJBOCC1 in the SIPP)

worker_class Private/Non-profit/Government worker

union Union member

hourly Primary job is paid hourly

hrs Usual hours worked per week

Source

Extracted from the SIPP public use files using Anthony Damico's scripts http://www.asdfree.com/

Index