Package ‘ISR3’

October 14, 2016

Type Package

Title Iterative Sequential Regression

Version 0.98

Date 2016-10-28

BugReports https://github.com/jlisic/isr3/issues

Maintainer Jonathan Lisic <jonathan.lisic@nass.usda.gov>

Description Performs multivariate normal imputation through iterative sequential regression. Conditional dependency structure between imputed variables can be specified a priori to accelerate imputation.

Depends R (>= 3.0.2)

License Unlimited

RoxygenNote 5.0.1

NeedsCompilation yes

Author Jonathan Lisic [aut, cre]

Repository CRAN

Date/Publication 2016-10-14 00:21:04

R topics documented:

isr *Iterative Sequential Regression*

Description

`isr` performs imputation of missing values based on an optionally specified model. Missingness is assumed to be missing at random (MAR).

Usage

```
isr(X, M, Xinit, mi = 1, burnIn = 100, thinning = 20, intercept = T)
```

Arguments

`X`	A matrix of points to be imputed or used for covariates by isr. `NA` values are considered missing. If column names are used, duplicate column names are not allowed.
`M`	A boolean valued optional matrix specifying the factorized pdf of the joint multivariate normal distribution of the variables requiring imputation. A description of the factorized pdf is provided in the details. The column names of `M` must match the column names of `X`, and the rows names of `M` must be a subset of the column names in `X`, in the same order as in `X`. Variables requiring imputation are each associated with a row in `M`; the conditional relationship to variables in `X` is indicated by the boolean valued elements of each row vector. A value of `TRUE` indicates conditional dependence, likewise a value of `FALSE` indicates conditional independence. Because this is a factorized pdf, the variable in the first row of `M` cannot specify a conditional dependence with a variable in a later row of `M`. If `M` is missing, dependence is assumed between all variables being imputed. No missing values are allowed.
`Xinit`	An optional matrix with the same dimensions of `X`, with no missing values. All values of `Xinit` should match those of `X`, with the exception of missing values. Values of `Xinit` that share an index with a missing value in `X` are treated as initial imputations. If Xinit is not specified, variable means are used as initial imputations.
`mi`	A scalar indicating the number of imputations to return
`burnIn`	A scalar indicating the number of iterations to burn in before returning imputations. Note, that burnIn is the total number of iterations, no thinning is performed until multiple imputation generation starts.
`thinning`	A scalar that represents the amount of thinning for the MCMC routine. A value of one implies no thinning.
`intercept`	A logical value identifying if the imputation model should have an intercept.

Details

The ISR algorithm performs Bayesian multivariate normal imputation. This imputation follows two steps, an imputation step and a prediction step. In the imputation step, the missing values are imputed from a Normal-Inverse-Wishart model with non-informative priors. In the prediction step, the parameters are estimated using both the observed and imputed values.

Imputation of parameters are done through the conditional factoring of the joint pdf. A conditional factoring is an expansion of the joint pdf of all the dependent variables in `X`. e.g. Pr(X|Z) = Pr(X1,X2,X3|Z) = Pr(X1,Z) Pr(X2|X1,Z) Pr(X3|X1,X2,Z), where the right hand side is the fully conditional specification for the dependent variables X1-X3 and independent variable Z.

Value

This function returns a list with two elements: `param` a three dimensional array of parameter estimates of the factored pdf. The last dimension is an index for the multiple imputations. `imputed` a three dimensional array of `X` with imputed values, the last dimension is an index for the multiple imputations.

References

Robbins, M. W., & White, T. K. (2011). Farm commodity payments and imputation in the Agricultural Resource Management Survey. American journal of agricultural economics, DOI: 10.1093/ajae/aaq166.

Examples

```
# simulation parameters
set.seed(100)
n <- 30
p <- 5
missing <- 10

# generate a covar matrix
covarMatrix <- rWishart(1,p+1,diag(p))[,,1]

# simulation of variables under the variable relationships
U <- chol(covarMatrix)

X <- matrix(rnorm(n*p), nrow=n) %*% U

# make some data missing
X[sample(1:(n*p),size=missing)] <- NA

# specify relationships
fitMatrix <- matrix( c(
  #  Covar2    CoVar1   Var1     Var2     Var3
     # 1. Var1
       TRUE,    TRUE,   FALSE,   FALSE,   FALSE,
     # 2. Var2
       TRUE,    TRUE,   FALSE,    FALSE,   FALSE,
     # 3. Var3
       TRUE,    TRUE,   TRUE,    TRUE,    FALSE
```

```
 ),nrow=3,byrow=TRUE)

covarList <- c('Covar2', 'CoVar1', 'Var1', 'Var2','Var3')

# setup names
colnames(fitMatrix) <- covarList
rownames(fitMatrix) <- covarList[-1:-2]
colnames(X) <- covarList

XImputed <- isr(X,fitMatrix)
```

RSWP *Reverse Sweep Function*

Description

RSWP performs the reverse sweep operator.

Usage

```
RSWP(V, b)
```

Arguments

V	A symmetric matrix to be reverse swept; this matrix cannot contain missing data or infinite values.
b	An array of integers or column names to reverse sweep.

Details

This program applies the reverse sweep operator as defined in (Dempster 1969).

Value

The reverse swept matrix V. Reverse sweeping will not occur if the column being swept has a zero-valued diagonal element.

References

Dempster, A.P. (1969). *Elements of continuous multivariate analysis*. Reading, MA: Addison-Wesley.

Examples

```
set.seed(100)
# generate symmetric positive definite matrix
Sigma <- rWishart(1,4,diag(3))[,,1]
# sweep all the columns to produce the inverse
# and then reverse sweep them all back to Sigma
Sigma2 <- RSWP(SWP(Sigma,1:3),1:3)
```

SWP *Sweep Function*

Description

SWP performs the sweep operator.

Usage

```
SWP(V, b)
```

Arguments

V	A symmetric matrix to be swept; this matrix cannot contain missing data or infinite values.
b	An array of integers or column names to sweep.

Details

This program applies the sweep operator as defined in (Dempster 1969).

Value

The swept matrix V. Sweeping will not occur if the column being swept has a zero-valued diagonal element.

References

Dempster, A.P. (1969). *Elements of continuous multivariate analysis*. Reading, MA: Addison-Wesley.

Examples

```
set.seed(100)
# generate a symmetric positive definite matrix
Sigma <- rWishart(1,4,diag(3))[,,1]
# sweep all the columns to produce the inverse
Sigma.inv <- SWP(Sigma,1:3)
```

Index