Package 'PhenomeImpute'

September 14, 2014

Type Package

Title PhenomeImpute

Version 1.0

Date 2014-09-13

Depends
R (>= 2.15.0), foreach, polycor, psych, ltm, scrime, sensitivity, cluster, stats, mlogit, missForest

Author Serena Liao, George C. Tseng

Maintainer Serena Liao <liaoge.serena@gmail.com>

Description This package contains functions for Phenome data imputation algorithm based on KNN and others.

License GPL (>= 2)

R topics documented:

PhenomeImpute-package *Missing data imputation in High dimensional Phenome data*

Description

In this package, we provide a complete pipeline for missing value problem in large Phenome data based on Self Train Selection (STS) Scheme. Six methods, including KNN-V, KNN-S, KNN-H, KNN-A, MeanImp, and MissForest (proposed in a recent paper), are trained based on the given missing data. Based on the performance of different methods on a second layer of artificial missing values, optimized methods are selected for different types of variables. We also provide meaningful imputability measures(IM) for researchers' convenience to exclude un-imputable values if there're any.

Details

Package:	PhenomeImpute
Type:	Package
Version:	1.0
Date:	2013-10-26
License:	Unversity of Pittsburgh

PhenomeImpute

Author(s)

Serena Liao Maintainer: Serena Liao <liaoge.serena@gmail.com>

References

Missing value imputation in high-dimensional phenomic data: Imputable or not? And how? Serena G. Liao1,*, Yan Lin1,*, Dongwan D. Kang, Naftali Kaminski, Frank C. Sciurba, George C. Tseng.

COPD *test data for PhenomeImpute package*

Description

sample phenomic data.

Usage

```
data(COPD)
```

Format

The data is a list of 2 items: element 1 is the data matrix; (subject on rows and variables on columns) element 2 is the vector of variable types.

`mixedDist` *Calculate distance matrix of mixted types of variables*

Description

Calculate distance matrix of mixted types of variables.

Usage

```
mixedDist(dat, types)
```

Arguments

`dat`	A data matrix with rows being subjects and column being variables
`types`	vector of variable types in Data. "con" for continuous variable; "nom" for nominal variables(including binary and multi-level variables); "ord" for ordinal variables.

Value

a symmetric distance matrix. (defined as 1 minus absolute pairwise correlation)

References

Multivariate correlation models with mixed disrete and continuous variables. Olkin I, Tate RF. The Annals of Mathematical Statistics 1961, 32(2):448-465.

Measures of Nominal-Ordinal Association. Agresti A. Journal of the American Statistical Association 1981, 76(375):524-529.

The polyserial correlation coefficient. Ulf Olsson FD, Neil J. Dorans. Psychometrika 1982, 47(3):337-347.

Maximum likelihood estimation of the polychoric correlation coefficient. Olsson U. Psychometrika 1979, 44(4):443-460.

Determination of the coefficient of correlation. Science 1909, 29:823-824.

Mathematical contributions to the theory of evolution. VII. On the correlation of characters not quantitatively measurable. Pearson K. Philos Trans R Soc Lond Ser A Math Phys Eng Sci 1900, 195:1-47.

Examples

```
## Example 1
## Generate a data matrix with mixed variables
set.seed(1234)
Data = cbind(rnorm(100,0,4),rnorm(100,0,7),rnorm(100,5,4),rnorm(100,-3,4),rnorm(100,0,4),
sample(1:4,100,replace=TRUE),sample(1:2,100,replace=TRUE),sample(1:3,100,replace=TRUE),
sample(1:20,100,replace=TRUE),sample(1:10,100,replace=TRUE))
Data = cbind(Data,Data)
colnames(Data) = paste("var",1:20,sep="")
row.names(Data) = paste("sub",1:100,sep="")
Type = c(rep("con",5),rep("nom",3),rep("ord",2))
Type = c(Type,Type)

Dist.mat=mixedDist(Data,Type)
```

PhenomeImpute *Self-Train Imputation for High dimensional Phenome data*

Description

Take in any phenome data with missing values to self train best methods for different types of variables.

Usage

```
PhenomeImpute(Data, Type, k, n.re)
```

Arguments

Data	A data matrix with rows being subjects and column being variables
Type	vector of variable types in Data. "con" for continuous variable; "nom" for nominal variables(including binary and multi-level variables); "ord" for ordinal variables.
k	Number of neighbors prefered in KNN methods
n.re	Number of second layer missing values being generated

Value

a list with first element being the imputed dataset; the second element being methods selected by each type of variables; third element is the unimputable cells within the original missing data(represented by row, col and variable type of missing value)

References

MissForest - nonparametric missing value imputation for mixed-type data. Daniel J. Stekhoven, Peter Buhlmann. Bioinformatics 2011, 28:113-118.

Missing value imputation in high-dimensional phenomic data: Imputable or not? And how? Serena G. Liao1,*, Yan Lin1,*, Dongwan D. Kang, Naftali Kaminski, Frank C. Sciurba, George C. Tseng.

Examples

```
## Example 1
## Generate a data matrix with mixed variables
set.seed(1234)
Data = cbind(rnorm(100,0,4),rnorm(100,0,7),rnorm(100,5,4),rnorm(100,-3,4),rnorm(100,0,4),
sample(1:4,100,replace=TRUE),sample(1:2,100,replace=TRUE),sample(1:3,100,replace=TRUE),
sample(1:20,100,replace=TRUE),sample(1:10,100,replace=TRUE))
Data = cbind(Data,Data)
colnames(Data) = paste("var",1:20,sep="")
row.names(Data) = paste("sub",1:100,sep="")
Type = c(rep("con",5),rep("nom",3),rep("ord",2))
Type = c(Type,Type)

for(i in 1:ncol(Data)){
  Data[sample(1:nrow(Data),5),i] = NA
}
Example.1 = PhenomeImpute(Data,Type,5,5)


## Example 2
## Load masked COPD data (a list with first element data matrix and second element variable types)
data(COPD)
set.seed(12345)

for(i in 1:ncol(COPD[[1]])){
COPD[[1]][sample(1:nrow(COPD[[1]]),5),i] = NA
}
Example.2 = PhenomeImpute(COPD[[1]],COPD[[2]],5,2)
## to save time, we set n.re=2. In reality, n.re should be >=5
```

Index