Package 'crassmat'

July 2, 2019

Type Package

Title Conditional Random Sampling Sparse Matrices

Version 0.0.6

Date 2019-06-28

Author Nick Kunz

Maintainer Nick Kunz <nick.kunz@columbia.edu>

Description Conducts conditional random sampling on observed values in sparse matrices. Useful for training and test set splitting sparse matrices prior to model fitting in cross-validation procedures and estimating the predictive accuracy of data imputation methods, such as matrix factorization or singular value decomposition (SVD). Although designed for applications with sparse matrices, CRASSMAT can also be applied to complete matrices, as well as to those containing missing values.

License GPL-3

Depends svMisc

Suggests NMF, recommenderlab

Encoding UTF-8

RoxygenNote 6.1.1

ByteCompile TRUE

LazyData TRUE

NeedsCompilation no

Repository CRAN

Date/Publication 2019-07-02 09:20:04 UTC

R topics documented:

A *Sparse Matrix A*

Description

Data for implementing the example given for CRASSMAT.

Usage

```
data(A)
```

Format

A sparse matrix containing 15 columns and 3000 observations

Author(s)

Nick Kunz <<nick.kunz@columbia.edu>>

crassmat *Conditional Random Sampling Sparse Matrices*

Description

Conducts conditional random sampling on observed values in sparse matrices. Useful for training and test set splitting sparse matrices prior to model fitting in cross-validation procedures and estimating the predictive accuracy of data imputation methods, such as matrix factorization or singular value decomposition (SVD). Although designed for applications with sparse matrices, CRASSMAT can also be applied to complete matrices, as well as to those containing missing values.

Usage

```
crassmat(data, sample_thres, conditional)
```

Arguments

`data`	a matrix (supports sparsity, missing values, and complete matrices)
`sample_thres`	a non-negative decimal specifying the percentage of observed values sampled out
`conditional`	a non-negative integer specifying the number of observed values to remain per row

Details

Takes a matrix A*ij* and samples out a single *jth* value on the condition that the number of *jth* values within the *ith* observation is greater than the specified conditional (minimum number of values to remain per *ith* observation). This process repeats itself until the specified sampling threshold is met.

Value

Returns a matrix object with observed values removed according to the specified `sample_thres` and `conditional`.

Author(s)

Nick Kunz <<nick.kunz@columbia.edu>>

References

Kunz, N. (2019). *Unsupervised Learning for Submarket Modeling: A Proxy for Neighborhood Change* (Master's Thesis). Columbia University, New York, NY.

Examples

```
## test set
A_test <- A

## training set
A_train <- crassmat(data = A,            # matrix
                    sample_thres = 0.20, # remove 20% of observed values
                    conditional = 1)     # keep > 1 observed values per row
```

Index