Package 'PoisNor'

May 14, 2019

Type Package

Title Simultaneous Generation of Multivariate Data with Poisson and Normal Marginals

Version 1.3.1

Date 2019-05-14

Author Anup Amatya, Hakan Demirtas, Ran Gao

Maintainer Ran Gao <rgao8@uic.edu>

Depends mvtnorm, corpcor, Matrix

Description Generates multivariate data with count and continuous variables with a pre-specified correlation matrix. The count and continuous variables are assumed to have Poisson and normal marginals, respectively. The data generation mechanism is a combination of the normal to anything principle and a connection between Poisson and normal correlations in the mixture.

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2019-05-14 21:30:03 UTC

R topics documented:

PoisNor-package *Simultaneous Generation of Multivariate Data with Poisson and Normal Marginals*

Description

The package implements a procedure for simultaneous generation of multivariate data with count and continuous variables with a pre-specified correlation matrix. The count and continuous variables are assumed to have Poisson and normal marginals, respectively. The data generation mechanism is a combination of the normal to anything principle and a connection between Poisson and normal correlations in the mixture. Data generation is accomplished by first calculating an intermediate correlation matrix (`cmat.star`) which is used to generate a sample from multivariate normal distribution. Then, the first few components (corresponding to number of Poisson variables) are transformed to Poisson variables via the inverse CDF method. The resulting data are composed of a mixture of Poisson and normal variables that conform with pre-specified marginal distributions and correlation structure.

The function `Valid.correlation` returns the lower and upper bounds of the correlation coefficients of Poisson-Poisson and Poisson-normal pairs given their marginal distributions, i.e. returns the range of feasible pairwise correlations. The function `Validate.correlation` checks the validity of the values of pairwise correlations. Additionally, it checks positive definiteness, symmetry and correctness of the dimensions. The engine function `genPoisNor` generates mixed data in accordance with the specified marginal and correlational quantities.

Details

Package:	PoisNor
Type:	Package
Version:	1.3.1
Date:	2019-05-14
License:	GPL

Author(s)

Anup Amatya, Hakan Demirtas, Ran Gao

Maintainer: Ran Gao <rgao8@uic.edu>

cmat.star *Computes the correlation of intermediate multivariate normal data before subsequent inverse CDF transformation*

Description

The function computes an intermediate correlation matrix which leads to the target correlation matrix after inverse CDF transformation of the samples generated from a multivariate normal distribution with the intermediate correlation matrix.

Usage

```
cmat.star(no.pois, no.norm, corMat, lamvec)
```

Arguments

no.pois	Number of Poisson variables in the data.
no.norm	Number of normal variables in the data.
corMat	A positive definite target correlation matrix whose entries are within the valid limits.
lamvec	Vector of Poisson rates (means).

Value

An intermediate correlation matrix of size $(no.pois + no.norm) \times (no.pois + no.norm)$

References

I. Yahav and G. Shmueli (2012). On generating multivariate poisson data in management science applications. *Applied Stochastic Models in Business and Industry*; 28(1):91-102.

See Also

Validate.correlation

Examples

```
## Not run:
lamvec= c(0.5,0.7,0.9)
M=c(0.352, 0.265, 0.342, 0.09, 0.141, 0.121, 0.297,
  -0.022, 0.177, 0.294, -0.044, 0.129, 0.1, 0.354, 0.386)

N=diag(6)
N[lower.tri(N)]=M
TV=N+t(N)
diag(TV)<-1

cstar = cmat.star(no.pois=3, no.norm=3, TV, lamvec)

## End(Not run)
```

`Cor.PN.Limit` *Finds the feasible correlation range for a pair of Poisson and normal variables*

Description

The function computes the lower and upper bounds of a pairwise correlation between a Poisson and a normal variable via the method of Demirtas and Hedeker (2011).

Usage

```
Cor.PN.Limit(lam)
```

Arguments

`lam`	A marginal rate for a Poisson variable of the pair.

Value

A vector of two elements. The first element is the lower bound and the second element is the upper bound.

References

Demirtas, H., Hedeker, D. (2011). A practical way for computing approximate lower and upper correlation bounds. *The American Statistician*; 65(2):104-109.

Examples

```
Cor.PN.Limit(0.05)
```

`Cor.PP.Limit` *Finds the feasible correlation range for a pair of Poisson variables*

Description

The function computes the lower and upper bounds of a pairwise correlation between a pair of Poisson variables.

Usage

```
Cor.PP.Limit(lamvec)
```

Arguments

`lamvec`	A vector of marginal rates for a pair of Poisson variables.

Value

A vector of two elements. The first element is the lower bound and the second element is the upper bound.

Examples

```
Cor.PP.Limit(c(0.05, 0.07) )
```

`genPoisNor` *Generates a data set with Poisson and normal variables*

Description

The function simulates multivariate data with Poisson and normal components with a pre-specified correlation matrix and marginal distributions.

Usage

```
genPoisNor(n, no.norm, no.pois, cmat.star, lamvec, sd.vec, mean.vec)
```

Arguments

`n`	Number of rows
`no.pois`	Number of Poisson variables.
`no.norm`	Number of normal variables.
`cmat.star`	The intermediate correlation matrix obtained from `cmat.star` function.
`lamvec`	A vector of marginal rates for Poisson variables.
`mean.vec`	A vector of means for the normal variables.
`sd.vec`	A vector of standard deviations for the normal variables.

Value

A matrix of size $n \times (no.pois + no.norm)$, of which first `no.pois` are Poisson variables.

Examples

```
## Not run:
  lamvec= c(0.05,0.07,0.09)
  M=c(0.352, 0.265, 0.342, 0.09, 0.141, 0.121, 0.297,
-0.022, 0.177, 0.294, -0.044, 0.129, 0.1, 0.354, 0.386)
  N=diag(6)
  N[lower.tri(N)]=M
  TV=N+t(N)
  diag(TV)<-1

  cstar = cmat.star(no.pois=3, no.norm=3, TV, lamvec)
```

```
 mydata=genPoisNor(n=200, no.norm=3,  no.pois=3, cmat.star=cstar,
lamvec, sd.vec=c(1,1,1), mean.vec=c(0,0,0))

## End(Not run)
```

`Valid.correlation` *Computes the lower and upper correlation bounds in the form of two matrices*

Description

The function computes the lower and upper bounds for the target correlations based on the marginal rates.

Usage

```
Valid.correlation(no.pois, no.norm, lamvec)
```

Arguments

`no.pois`	Number of Poisson variables.
`no.norm`	Number of normal variables.
`lamvec`	A vector of marginal rates for Poisson variables.

Details

The function returns a list of two matrices. The `min` contains the lower bounds and the `max` contains the upper bounds of the feasible correlations.

Examples

```
lamvec= c(0.05,0.07,0.09)
Valid.correlation(no.pois=3, no.norm=3,lamvec)
```

`Validate.correlation` *Checks the target correlation matrix*

Description

The function checks the validity of the values of pairwise correlations. Additionally, it checks positive definiteness, symmetry and correctness of the dimensions.

Usage

```
Validate.correlation(no.pois, no.norm, corMat, lamvec)
```

Arguments

`no.pois`	Number of Poisson variables.
`no.norm`	Number of normal variables.
`corMat`	The target correlation matrix which must be positive definite and within the valid limits.
`lamvec`	A vector of marginal rates for Poisson variables.

Details

In addition to being positive definite and symmetric, the values of pairwise correlations in the target correlation matrix must also fall within the limits imposed by the marginal distributions in the system. The function ensures that the supplied correlation matrix is valid for simulation. If a violation occurs, an error message is displayed that identifies the violation. The function returns a logical value `TRUE` when no such violation occurs.

Examples

```
## Not run:
# An example with a valid target correlation matrix.

lamvec= c(0.05,0.07,0.09)
M=c(0.352, 0.265, 0.342, 0.09, 0.141, 0.121, 0.297,
-0.022, 0.177, 0.294, -0.044, 0.129, 0.1, 0.354, 0.386)

N=diag(6)
N[lower.tri(N)]=M
TV=N+t(N)
diag(TV)<-1

Validate.correlation(no.pois=3, no.norm=3, corMat=TV, lamvec)


# An example with an invalid target correlation matrix (bound violation).

lamvec= c(0.05,0.07,0.09)
```

```
M=c(-0.151, -0.085, -0.11, 0.29, 0.6, 0.132, 0.161, 0.139,
-0.088, 0.075, -0.025, -0.293, -0.67, -0.03, 0.61)
N=diag(6)
N[lower.tri(N)]=M
TV1=N+t(N)
diag(TV1)<-1
Validate.correlation(no.pois=3, no.norm=3, corMat=TV1, lamvec)

# Examples with an incorrect dimension specification.

lamvec= c(0.05,0.07,0.09)
Validate.correlation(no.pois=3, no.norm=2, corMat=TV, lamvec)
Validate.correlation(no.pois=2, no.norm=3, corMat=TV, lamvec)

# An example with a non-positive definite correlation matrix.

TV1=TV
TV1[5,1]=TV1[1,5] = 1.5
Validate.correlation(no.pois=3, no.norm=3, corMat=TV1, lamvec)

# An example with a non-symmetric correlation matrix.

TV1=TV
TV1[5,1] = 0.1
Validate.correlation(no.pois=3, no.norm=3, corMat=TV1, lamvec)

# An example with an invalid diagonal element in the correlation matrix.

TV1=TV
TV1[5,5] = 2
Validate.correlation(no.pois=3, no.norm=3, corMat=TV1, lamvec)

## End(Not run)
```

Index