Package ‘ctmcd’

January 2, 2019

Type Package

Title Estimating the Parameters of a Continuous-Time Markov Chain from Discrete-Time Data

Version 1.4.1

Date 2019-01-02

Author Marius Pfeuffer [aut,cre], Greig Smith [ctb], Goncalo dos Reis [ctb], Linda Moes-
tel [ctb], Matthias Fischer [ctb]

Maintainer Marius Pfeuffer <marius.pfeuffer@fau.de>

Description Functions for estimating Markov generator matrices from discrete-time observations. The implemented approaches comprise diagonal adjustment, weighted adjustment and quasi-optimization of matrix logarithm based candidate solutions, an expectation-maximization algorithm as well as a Gibbs sampler.

License GPL-3

Imports Rcpp (>= 0.12.17), coda, expm, numDeriv

Suggests knitr, rmarkdown, R.rsp, markovchain

LinkingTo Rcpp, RcppArmadillo

VignetteBuilder knitr, R.rsp

NeedsCompilation yes

Repository CRAN

Date/Publication 2019-01-02 13:40:23 UTC

R topics documented:

ctmcd-package ................................................................. 2
ctmcdlogLik ................................................................. 3
expmMC ................................................................. 4
gm ................................................................. 5
gm.default ................................................................. 7
gmcI ................................................................. 9
gmcI.default ......................................................... 10
gmDA ................................................................. 12
Estimating the Parameters of a Continuous-Time Markov Chain from Discrete-Time Data

Description

Functions for estimating Markov generator matrices from discrete-time observations.

Author(s)

Marius Pfeuffer [aut,cre], Greig Smith [ctb], Goncalo dos Reis [ctb], Linda Moestel [ctb], Matthias Fischer [ctb]

Maintainer: Marius Pfeuffer <marius.pfeuffer@fau.de>

References


Examples

```r
## Not run:
data(tm_abs)

## Maximum Likelihood Generator Matrix Estimate
gm0 = matrix(1, 8, 8)
diag(gm0) = 0
diag(gm0) = -rowSums(gm0)
gm0[8,] = 0

gmem = gm(tm_abs, te = 1, method = "EM", gmguess = gm0)
plot(gmem)

## Confidence Interval
ciem = gmc(gmem, alpha = 0.05)
plot(ciem)

## End (Not run)
```

description
Function for evaluating the likelihood function of a continuous-time Markov chain given discrete-time data.

Usage

```r
cmcdloglik(gm, tmabs, te)
```

Arguments

gm generator matrix of continuous-time Markov chain
tmabs matrix of absolute transition frequencies
te time elapsed in transition process

Author(s)

Marius Pfeuffer

Examples

```r
data(tm_abs)

## Initial guess for generator matrix (absorbing default state)
gm0 = matrix(1, 8, 8)
diag(gm0) = 0
```
expmMC

Matrix Exponential Function

Description

Computation of the matrix exponential and interface to the markovchain package

Usage

expmMC(gm, t, method = "PadeRBS", order = 8)

Arguments

gm generator matrix (either gm or matrix object)
t time horizon
method method to compute the matrix exponential, see expm for details
order order for Pade or Taylor method, see expm for details

Details

An interface to the markovchain package is provided so that the resulting transition matrices are returned as markovchain objects and can be further processed in the markovchain package, e.g., visualized (see example below).

Author(s)

G. A. Spedicato, M. Pfeuffer

References

G. A. Spedicato: Discrete Time Markov Chains with R. The R Journal (To Appear), 2017

See Also

expm
Generative Matrix Estimation

Description

Generic function to estimate the parameters of a continuous Markov chain

Usage

The `gm` function is used to estimate the generator matrix of a continuous Markov chain. Its usage is as follows:

```r
gm(tm, te, method, ...)  
```

Arguments

- **tm**: A matrix of either absolute transition frequencies (if method is "EM" or "GS") or relative transition frequencies (if method is "DA", "WA" or "QO").
- **te**: Time elapsed in transition process.
- **method**: A method to derive the generator matrix: "DA" - Diagonal Adjustment, "WA" - Weighted Adjustment, "QO" - Quasi-Optimization, "EM" - Expectation-Maximization Algorithm, "GS" - Gibbs Sampler.

Additional Arguments:

- `gmguess`: Initial guess for the generator matrix estimation procedure (if method is "EM")
- `prior`: Prior parametrization (if method is "GS")
- `burnin`: Burn-in period (if method is "GS")
- `eps`: Convergence criterion (if method is "EM")
- `conv_pvalue`, `conv_freq`: Convergence criterion (if method is "GS")
- `niter`: Maximum number of iterations (if method is "EM" or "GS")
- `sampl_func`: Optional self-written path sampling function for endpoint-conditioned Markov processes (if method is "GS")

Examples

```r
### Exemplary Transition Matrix
tm0=matrix(c(16,4,4)
  tm0=tm0/rowSums(tm0)
  tm0[4,]=c(0,0,0,1)

### Generator Matrix Estimate
gm_est=gm(tm0,te=1,method="DA")

gm_est

### Matrix Exponential and Conversion to markovchain object
### (markovchain package needs to be installed)
mc=expm(mc,gm_est)
if(require("markovchain"){
  plot(mc)
}
```
• `compmat`: matrix stating combined use of modified rejection sampling / uniformization sampling algorithms (if method is "GS")
• `sampl_method`: sampling method for deriving endpoint-conditioned Markov process path: "Unif" - Uniformization Sampling, "ModRej" - Modified Rejection Sampling (if method is "GS")
• `logmethod`: method to compute matrix logarithm (if method is "DA", "WA" or "QO", see `?logm` from `expm` package for more information)
• `expmethod`: method to compute matrix exponential (if method is "EM" or "GS", see `?expm` from `expm` package for more information)
• `verbose`: verbose mode (if method is "EM" or "GS")

Details

The methods "DA", "WA" and "QO" provide adjustments of a matrix logarithm based candidate solution, "EM" gives the maximum likelihood estimate and "GS" a posterior mean estimate in a Bayesian setting with conjugate Gamma priors.

Author(s)

Marius Pfeuffer

References

G. dos Reis, M. Pfeuffer, G. Smith: Capturing Rating Momentum in the Estimation of Probabilities of Default, With Application to Credit Rating Migrations (In Preparation), 2018

See Also

gmDA, gmWA, gmQO, gmEM, gmGS

Examples

data(tm_abs)

## Maximum Likelihood Generator Matrix Estimate
gm0=matrix(c(1,8,8,2,7,9,3,5,10),3,3)
diag(gm0)=0
diag(gm0)=-rowSums(gm0)
gm.default

```r
gm0[8,]=0
gmem=gm(tm_abs,te=1,method="EM",gmguess=gm0)
gmem
```

```r
## Quasi Optimization Estimate
tm_rel=rbind((tm_abs/rowSums(tm_abs))[,1:7],c(rep(0,7),1))
gmqo=gm(tm_rel,te=1,method="QO")
gmqo
```

---

**gm.default**  
*Generator Matrix Estimation*

---

**Description**

Default function to estimate the parameters of a continuous Markov chain

**Usage**

```r
## Default S3 method:
gm(tm, te, method, gmguess = NULL, prior = NULL, burnin = NULL,
eps = 1e-06, conv_pvalue = 0.05, conv_freq = 10, niter = 10000, sampl_func = NULL,
combmat = NULL, sampl_method = "Unif", logmethod = "Eigen", expmethod = "PadeRBS",
verbose = FALSE, ...)
```

**Arguments**

- `tm`: matrix of either absolute transition frequencies (if method is "EM" or "GS") or relative transition frequencies (if method is "DA", "WA" or "QO")
- `te`: time elapsed in transition process
- `method`: method to derive generator matrix: "DA" - Diagonal Adjustment, "WA" - Weighted Adjustment, "QO" - Quasi-Optimization, "EM" - Expectation-Maximization Algorithm, "GS" - Gibbs Sampler
- `gmguess`: initial guess for generator matrix estimation procedure (if method is "EM")
- `prior`: prior parametrization (if method is "GS")
- `burnin`: burn-in period (if method is "GS")
- `eps`: convergence criterion (if method is "EM" or "GS")
- `conv_pvalue`: convergence criterion: stop, if Heidelberger and Welch’s diagnostic assumes convergence (see coda package)
- `conv_freq`: convergence criterion: absolute frequency of convergence evaluations
- `niter`: maximum number of iterations (if method is "EM" or "GS")
- `sampl_func`: optional self-written path sampling function for endpoint-conditioned Markov processes (if method is "GS")
combmat = matrix stating combined use of modified rejection sampling / uniformization sampling algorithms (if method is "GS")
sampl_method = sampling method for deriving endpoint-conditioned Markov process path: "Unif" - Uniformization Sampling, "ModRej" - Modified Rejection Sampling (if method is "GS")
logmethod = method to compute matrix logarithm (if method is "DA", "WA" or "QO", see ?logm from expm package for more information)
expmethod = method to compute matrix exponential (if method is "EM" or "GS", see ?expm from expm package for more information)
verbose = verbose mode (if method is "EM" or "GS")
... additional arguments

Details

The methods "DA", "WA" and "QO" provide adjustments of a matrix logarithm based candidate solution, "EM" gives the maximum likelihood estimate and "GS" a posterior mean estimate in a Bayesian setting with conjugate Gamma priors.

Author(s)

Marius Pfeuffer

References


See Also

gmDA, gmWA, gmQO, gmEM, gmGS

Examples

data(tm_abs)

## Maximum Likelihood Generator Matrix Estimate
gm0=matrix(1,8,8)
diag(gm0)=0
diag(gm0)=-rowSums(gm0)
gmci

Description

Generic function to derive confidence / credibility intervals for "EM" or "GS" based generator matrix objects

Usage

gmci(gm, alpha, ...)

Arguments

gm a "EM" or "GS" generator matrix object
alpha significance level
... additional arguments:
  • eps: threshold for which generator matrix parameters are assumed to be fixed at zero (if "EM" object)
  • cimethod: "Direct" and "SdR" use analytical expressions of the Fisher information matrix, "BS" employs the numerical approach of Bladt and Sørensen, 2009 (if "EM" object)
  • expmethod: method to compute matrix exponentials (see ?expm from expm package for more information)

Details

If gm is based on the "EM" method (expectation-maximization algorithm), the function computes a Wald confidence interval based on the method of Oakes, 1999. If gm is based on the "GS" method (Gibbs sampler), the function computes an equal-tailed credibility interval.

Author(s)

Marius Pfeuffer
References


G. Smith and G. dos Reis. Robust and Consistent Estimation of Generators in Credit Risk. Quantitative Finance 18(6):983-1001, 2018

G. dos Reis, M. Pfeuffer, G. Smith: Capturing Rating Momentum in the Estimation of Probabilities of Default, With Application to Credit Rating Migrations (In Preparation), 2018

Examples

```r
## Not run:
data(tm_abs)

## Maximum Likelihood Generator Matrix Estimate
gm0=matrix(1,8,8)
diag(gm0)=0
diag(gm0)=-rowSums(gm0)
gm0[8,]=0
gmem=gm(tm_abs,te=1,method="EM",gmguess=gm0)

## Oakes Confidence Interval
ciem=gmci(gmem,alpha=0.05)
ciem

## End(Not run)
```

---

### gmci.default

Confidence / Credibility Intervals for Generator Matrix Objects

Description

Default function to derive confidence / credibility intervals for "EM" or "GS" based generator matrix objects

Usage

```
## Default S3 method:
gmci(gm, alpha, eps = 1e-04, cimethod="Direct", expmethod = "PadeRBS", ...)
```

Arguments

- `gm` a "EM" or "GS" generator matrix object
- `alpha` significance level
eps threshold for which generator matrix parameters are assumed to be fixed at zero (if "EM" object)
cimethod "Direct" or "SdR" use analytical expressions of the Fisher information matrix, "BS" employ the numerical expressions of Bladt and Soerensen, 2009 (if "EM" object)
expmethod method to compute matrix exponentials (see ?expm from expm package for more information)
... additional arguments

Details

If gm is based on the "EM" method (expectation-maximization algorithm), the function computes a Wald confidence interval based on the method of Oakes, 1999. If gm is based on the "GS" method (Gibbs sampler), the function computes an equal-tailed credibility interval.

Author(s)

Marius Pfeuffer

References

G. dos Reis, M. Pfeuffer, G. Smith: Capturing Rating Momentum in the Estimation of Probabilities of Default, With Application to Credit Rating Migrations (In Preparation), 2018

Examples

```r
## Not run:
data(tm_abs)

## Maximum Likelihood Generator Matrix Estimate
gm0=matrix(1,8,8)
diag(gm0)=0
diag(gm0)=-rowSums(gm0)
gm0[8,]=0
gmem=gm(tm_abs,te=1,method="EM",gmguess=gm0)

## Oakes Confidence Interval
ciem=gmci(gmem,alpha=0.05)
ciem

## End(Not run)
```
Diagonal Adjustment

Description

Function for deriving a Markov generator matrix estimate based on the diagonal adjustment method of Israel et al., 2001

Usage

\texttt{gmda(tmrel, te, logmethod = "Eigen")}

Arguments

- \texttt{tmrel}: matrix of relative transition frequencies
- \texttt{te}: time elapsed in transition process
- \texttt{logmethod}: method for computation of matrix logarithm, by default eigendecomposition is chosen (see \texttt{?logm} from \texttt{expm} package for more information)

Details

A candidate solution is derived by the matrix logarithm and then adjusted in order to fulfill the properties of a Markov generator matrix.

Author(s)

Marius Pfeuffer

References


Examples

```r
# Derive matrix of relative transition frequencies
data(tm_abs)
 tm_rel=rbind((tm_abs/rowSums(tm_abs))[1:7,],c(rep(0,7),1))

# Derive diagonal adjustment generator matrix estimate
gmda=gmda(tm_rel,)
gmda
```
Description

Function for deriving a Markov generator matrix estimate by an instance of the expectation-maximization algorithm (described by Bladt and Soerensen, 2005)

Usage

gmEM(tmabs, te, gmguess, eps = 1e-06, niter = 1000, expmethod = "PadeRBS", verbose = FALSE)

Arguments

tmabs
matrix of absolute transition frequencies

te
time elapsed in transition process

gmguess
initial guess (for generator matrix)

eps
stop criterion: stop, if relative change in log-likelihood is smaller than eps

niter
stop criterion: maximum number of iterations

expmethod
method for computation of matrix exponential, by default "PadeRBS" is chosen (see ?expm from expm package for more information)

verbose
verbose mode

Details

A maximum likelihood generator matrix estimate is derived by an instance of the expectation-maximization algorithm.

Author(s)

Marius Pfeuffer

References


Examples

data(tm_abs)

## Initial guess for generator matrix (absorbing default state)
gm0=matrix(1,8,8)
diag(gm0)=0
diag(gm0)=-rowSums(gm0)
## Description

Function for deriving a Markov generator matrix estimate by Gibbs sampling (described by Bladt and Soerensen, 2005)

## Usage

```r
gmGS(tmabs, te, prior, burnin, conv_pvalue = 0, conv_freq = 10, niter = 10000, sampl_method = "Unif", expmethod = "PadeRBS", verbose = FALSE, combmat=NULL, sampl_func = NULL)
```

## Arguments

- **tmabs**: matrix of absolute transition frequencies
- **te**: time elapsed in transition process
- **prior**: list of prior parameters (Gamma prior)
- **burnin**: number of burn-in iterations
- **conv_pvalue**: convergence criterion: stop, if Heidelberger and Welch’s diagnostic assumes convergence (see coda package), convergence check is only employed if conv_pvalue > 0
- **conv_freq**: convergence criterion: absolute frequency of convergence evaluations
- **niter**: stop criterion: stop, if maximum number of iterations is exceeded
- **sampl_method**: method for sampling paths from endpoint-conditioned Markov processes. options: "Unif" - Uniformization sampling, "ModRej" - Modified Rejection Sampling
- **expmethod**: method for computation of matrix exponential, by default "PadeRBS" is chosen (see ?expm from expm package for more information)
- **verbose**: verbose mode
- **combmat**: matrix specifying the combined use of sampling methods: "U" - uniformization sampling, "M" - modified rejection sampling
- **sampl_func**: interface for own endpoint-conditioned Markov process sampling function

## Details

A posterior mean generator matrix estimate is derived by Gibbs Sampling. The gamma distribution is used as prior.
gmQO

Author(s)
Marius Pfeuffer

References

See Also
rnijTRiT_ModRej, rnijTRiT_Unif

Examples

data(tm_abs)

## Example prior parametrization (absorbing default state)
pr=list()
pr[[1]]=matrix(1,8,8)
pr[[1]][8,]=0
pr[[2]]=c(rep(5,7),Inf)

## Derive Gibbs sampling generator matrix estimate
## Not run:
gmgs=gmGS(tmabs=tm_abs,te=1,sampl_method="Unif",prior=pr,burnin=10,niter=100,verbose=TRUE)
gmgs

## End(Not run)

---

gmQO Quasi-Optimization

Description
Function for deriving a Markov generator matrix estimate based on the quasi-optimization procedure of Kreinin and Sidelnikova, 2001

Usage
gmQO(tmrel, te, logmethod = "Eigen")

Arguments
- **tmrel**: matrix of relative transition frequencies
- **te**: time elapsed in transition process
- **logmethod**: method for computation of matrix logarithm, by default eigendecomposition is chosen (see ?logm from expm package for more information)
Details
From the set of possible Markov generator matrices, the one is chosen which is closest to a matrix logarithm based candidate solution in terms of sum of squared deviations.

Author(s)
Marius Pfeuffer

References

Examples
data(tm_abs)
## Derive matrix of relative transition frequencies
data(tm_abs)
tm_rel=rbind((tm_abs/rowSums(tm_abs))[1:7,],c(rep(0,7),1))

## Derive quasi optimization generator matrix estimate
gmqo=gmQ0(tm_rel,1)
gmqo

---

**gmWA**  
*Weighted Adjustment*

Description
Function for deriving a Markov generator matrix estimate based on the weighted adjustment method of Israel et al., 2001

Usage
gmWA(tmrel, te, logmethod = "Eigen")

Arguments
- `tmrel`  
  matrix of relative transition frequencies
- `te`  
  time elapsed in transition process
- `logmethod`  
  method for computation of matrix logarithm, by default eigendecomposition is chosen (see ?logm from expm package for more information)

Details
A candidate solution is derived by the matrix logarithm and then adjusted in order to fulfil the properties of a Markov generator matrix.
plot.gm

Author(s)

Marius Pfeuffer

References


Examples

```r
## Derive matrix of relative transition frequencies
data(tm_abs)
tm_rel=rbind((tm_abs/rowSums(tm_abs))[,1:7],c(rep(0,7),1))

## Derive weighted adjustment generator matrix estimate
gmwa=gmWA(tm_rel,1)
gmwa
```

### plot.gm

Plot Function for Generator Matrix Estimation Objects

**Description**

Function for visualizing the output of a generator matrix estimation procedure.

**Usage**

```r
## S3 method for class 'gm'
plot(x, mattext = NULL, col = c("grey", "red"), main = x$method, las = 1, xlab = "To", ylab = "From", xnames, ynames, cex = 1, fig = 3, opacity_factor, ...)
```

**Arguments**

- `x`: a generator matrix estimation object
- `mattext`: optional; matrix of strings replacing the parameter estimates
- `col`: two element vector of basis colors for positive and negative parameter estimate entries
- `main`: optional; plot title
- `las`: orientation of x and y axis elements
- `xlab`: x axis name
- `ylab`: y axis name
- `xnames`: description of x axis elements
- `ynames`: description of y axis elements
- `cex`: font size
- `fig`: number of significant figure to be plotted
opacity_factor  two element vector for specification of opacity for positive and negative parameter entry highlighting (must be greater than zero)
...  additional arguments

Author(s)
Marius Pfeuffer

See Also
print.gm, summary.gm, plotM

Examples

data(tm_abs)

## Maximum Likelihood Generator Matrix Estimate
gm0 = matrix(1, 8, 8)
diag(gm0) = 0
diag(gm0) = -rowSums(gm0)
gm0[8, ] = 0

gmem = gm(tm_abs, te=1, method = "EM", gmguess = gm0)
plot(gmem)
plotM

xlab x axis name
ylab y axis name
xnames description of x axis elements
ynames description of y axis elements
cex font size
fig number of significant figures to be plotted
opacity_factor two element vector for specification of opacity for positive and negative parameter entry highlighting (must be greater than zero)
... additional arguments

Author(s)

Marius Pfeuffer

See Also

print.gmci, plotM

Examples

## Not run:
data(tm_abs)

## Maximum Likelihood Generator Matrix Estimate
gm0=matrix(1,8,8)
diag(gm0)=0
diag(gm0)=-rowSums(gm0)
gm0[8,]=0
gmem=gmem(tm_abs,te=1,method="EM",gmguees=gm0)
plot(gmem)

## Confidence Interval
ciem=gmci(gmem,alpha=0.05)
plot(ciem)

## End(Not run)

plotM Matrix Plot Function

Description

Function to visualize matrices
Usage

```
plotM(mat, mattext, col = c("grey", "red"), main, las = 1, xlab = "To",
ylab = "From", xnames, ynames, cex = min(1, nrow(mat)/8), fig = 3, opacity_factor)
```

Arguments

- `mat` a matrix
- `mattext` optional: matrix of strings replacing the original matrix entries
- `col` two element vector of basis colors for positive and negative matrix entries
- `main` optional: plot title
- `las` orientation of x and y axis elements
- `xlab` x axis name
- `ylab` y axis name
- `xnames` description of x axis elements
- `ynames` description of y axis elements
- `cex` font size
- `fig` number of significant figures to be plotted
- `opacity_factor` two element vector for specification of opacity for positive and negative parameter entry highlighting (must be greater than zero)

Author(s)

Marius Pfeuffer

See Also

`plotNgm`, `plotNgmci`

Examples

```
gm0 = matrix(1,8,8)
diag(gm0) = 0
diag(gm0) = -rowSums(gm0)
gm0[8,] = 0

plotM(gm0)
```
print.gm

Print Method for Generator Matrix Estimation Objects

Description

Function for printing the results of a generator matrix estimation

Usage

"""S3 method for class 'gm'
print(x, ...)
"""

Arguments

x a generator matrix estimation object
...
additional arguments

See Also

summary.gm, plot.gm

print.gmci

Print Method for Generator Matrix Confidence / Credibility Interval Objects

Description

Function for printing the boundaries of a generator matrix confidence / credibility interval

Usage

"""S3 method for class 'gmc1'
print(x, ...)
"""

Arguments

x a generator matrix confidence / credibility interval
...
additional arguments

See Also

plot.gmci
Description

Function for generating initial and endpoint-conditioned Markov process sampling paths for a given
discrete-time transition matrix

Usage

```
rNijTRiT_ModRej(tmabs, te, gm)
```

Arguments

- `tmabs`: matrix of absolute transition frequencies
- `te`: time elapsed in transition process
- `gm`: generator matrix

Details

Function for the simulation of paths from an endpoint-conditioned Markov process. Returns number
of transitions NijT and cumulative holding times RiT.

Author(s)

Jon Fintzi, Marius Pfeuffer

References

A. Hobolth and E. A. Stone: Simulation from Endpoint-Conditioned, Continuous-Time Markov
Chains on a Finite State Space, with Applications to Molecular Evolution. Annals of Applied
Statistics 3(3):1204-1231, 2009

Examples

```
data(tm_abs)

## Initial guess for generator matrix (absorbing default state)

gm=matrix(1,8,8)
diag(gm)=0
diag(gm)=-rowSums(gm)
gm[8,]=0

rNijTRiT_ModRej(tm_abs,1,gm)
```
**rNijTRiT_Unif**

---

**C++ Based Uniformization Sampling**

---

### Description

Function for generating initial and endpoint-conditioned Markov process sampling paths for a given discrete-time transition matrix.

### Usage

```r
rNijTRiT_Unif(tmabs, te, gm, tpm)
```

### Arguments

- `tmabs`: matrix of absolute transition frequencies
- `te`: time elapsed in transition process
- `gm`: generator matrix
- `tpm`: discrete-time transition probability matrix, matrix exponential of `gm`

### Details

Function for the simulation of paths from an endpoint-conditioned Markov process. Returns number of transitions NijT and cumulative holding times RiT.

### Author(s)

Jon Fintzi, Marius Pfeuffer

### References


### Examples

```r
data(tm_abs)

## Generator Matrix
gm=matrix(c(1,8,8),
diag(gm)=0
diag(gm)=rowSums(gm)
gm[8,]=0

## Transition Probability Matrix
library(expm)
```
tmci

Description

Function for providing results and extended output of a generator matrix estimation procedure.

Usage

```r
# S3 method for class 'gm'
summary(object, ...)```

Arguments

- `object`: a generator matrix estimation object
- `...`: additional arguments

See Also

`print.gm`, `plot.gm`

tmci

Delta Method Confidence Intervals for Matrix Exponential Transformations of Generator Matrix Objects

Description

Generic function to derive delta method based confidence intervals for matrix exponential transformations of "EM" based generator matrix objects

Usage

`tmc(gmem, alpha, te, eps = 1e-04, expmethod = "PadeRBS")`

Arguments

- `gmem`: an "EM" generator matrix object
- `alpha`: significance level
- `te`: discrete time horizon for which the interval is supposed to be computed
- `eps`: threshold for which generator matrix parameters are assumed to be fixed at zero
- `expmethod`: method to compute matrix exponentials (see `expm` from `expm` package for more information)
Details
Confidence intervals for discrete-time transition matrix predictions given generator matrix estimates are computed by using the delta method for matrix exponential transformations.

References
G. dos Reis, M. Pfeuffer, G. Smith: Capturing Rating Momentum in the Estimation of Probabilities of Default, With Application to Credit Rating Migrations (In Preparation), 2018

Examples
```r
## Not run:
data(tm_abs)

## Maximum Likelihood Generator Matrix Estimate
gm0=matrix(1,8,8)
diag(gm0)=0
diag(gm0)=-rowSums(gm0)
gm0[8,1]=0
gmem=gm(tm_abs,te=1,method="EM",gmguess=gm0)

## 2.5 Year Transition Matrix Confidence Interval
citm=tmci(gmem,alpha=0.05,te=2.5)
citm

## End(Not run)
```

---

**tm_abs**

*Single Year Corporate Credit Rating Transitions*

**Description**
Matrix of Standard and Poor's Global Corporate Rating Transition Frequencies 2000 (NR Removed)

**Usage**

data("tm_abs")

**Format**
The format is: num [1:8, 1:8] 17 2 0 0 0 0 0 1 455 ... - attr(*, "dimnames")=List of 2 ..$ : chr [1:8] "AAA" "AA" "A" "BBB" ...

**References**
European Securities and Markets Authority, 2016
Examples

data(tm_abs)

## Matrix of relative transition frequencies
tm_rel=rbind((tm_abs/rowSums(tm_abs))[1:7,],c(rep(0,7),1))
tm_rel
Index

*Topic datasets
  tm_abs, 25

*Topic package
  ctmcd-package, 2

  ciEMBS (gmci.default), 10
  ciEMDM (gmci.default), 10
  ciEMoFI (gmci.default), 10
  ciEMsDR (gmci.default), 10
  ciGS (gmci.default), 10
  ctmcd (ctmcd-package), 2
  ctmcd-package, 2
  ctmcdlogLik, 3

  expm, 4
  expmMC, 4

  gm, 5
  gm.default, 7
  gmci, 9
  gmci.default, 10
  gmDA, 6, 8, 12
  gmEM, 6, 8, 13
  gmGS, 6, 8, 14
  gmQQ, 6, 8, 15
  gmWA, 6, 8, 16

  plot.gm, 17, 20, 21, 24
  plot.gmci, 18, 20, 21
  plotM, 18, 19, 19
  print.gm, 18, 21, 24
  print.gmci, 19, 21

  rNijTRiT_ModRej, 15, 22
  rNijTRiT_Unif, 15, 23

  summary.gm, 18, 21, 24

  tm_abs, 25
  tmci, 24