Package ‘systemicrisk’

March 20, 2015

Type Package
Title A Toolbox for Systemic Risk
Version 0.2
Date 2015-03-19
Author Axel Gandy and Luitgard A.M. Veraart
Maintainer Axel Gandy <a.gandy@imperial.ac.uk>
Description A toolbox for systemic risk based on liabilities matrices. Contains a Gibbs sampler for liabilities matrices where only row and column sums of the liabilities matrix are observed.
License GPL-3
Imports Rcpp (&gt;= 0.11.2), lpSolve
LinkingTo Rcpp
Suggests testthat, knitr
VignetteBuilder knitr
NeedsCompilation yes
Repository CRAN
Date/Publication 2015-03-20 00:02:28

R topics documented:

cloneMatrix ................................................................. 2
default ................................................................. 2
default_cascade ......................................................... 3
default_clearing ........................................................ 4
ERE_step_cycle ......................................................... 5
getfeasibleMatr ........................................................ 6
GibbsSteps_kcycle ...................................................... 6
sample_ERE .............................................................. 7
steps_ERE ............................................................... 8

Index 10
cloneMatrix  

_Crates a deep copy of a matrix_

**Description**

Useful when calling `ERE_step_cycle` or `GibbsSteps_kcycle` to ensure that there are no side effects for the return values.

**Usage**

```r
clonematrix(M)
```

**Arguments**

- `M`  
  A matrix

**Value**

A deep copy of the matrix.

**Examples**

```r
lambda <- matrix(0.5,nrow=2,ncol=2)
p <- matrix(0.7, nrow=2,ncol=2)
L <- matrix(rexp(4),nrow=2);
L
Lold <- L
Lcopy <- cloneMatrix(L)
ERE_step_cycle(r=c(0,1),c=c(0,1),L=L,lambda=lambda,p=p)

L ## new value
Lold ## equal to L !!!
Lcopy ## still has the original value
```

---

default  

_Default of Banks_

**Description**

Computes bank defaults based on a liabilities matrix and external assets and liabilities.

**Usage**

```r
default(L, ea, el = 0, method = c("clearing", "cascade"), ...)
```
Arguments

L liability matrix
ea vector of external assets
el vector of external liabilities.
method the method to be used. See Details.
... Additional information for the various methods. See Details.

Value

A list with at least one element "defaultind", which is a vector indicating which banks default (1=default, 0= no default). Depending on the method, other results such as the clearing vector may also be reported.

See Also

default_cascade, default_clearing.

Examples

ea <- c(1/2,5/8,3/4)
el <- c(3/2,1/2,1/2)
x <- 0.5
L <- matrix(c(0,x,1-x,1-x,0,x,x,1-x,0),nrow=3)
default(L,ea,el)
default(L,ea,el,"cascade")

default_cascade  Default Cascade

Description

Computes bank defaults via the default cascade algorithm.

Usage

default_cascade(L, ea, el = 0, recoveryrate = 0)

Arguments

L liability matrix
ea vector of external assets
el vector of external liabilities (default 0)
recoveryrate recovery rate in [0,1] (defaults to 0)
Value

vector indicating which banks default (1=default, 0= no default)

Examples

```r
ea <- c(1/2, 5/8, 3/4)
el <- c(3/2, 1/2, 1/2)
x <- 0.5
L <- matrix(c(0, x, 1-x, 1-x, 0, x, x, 1-x, 0), nrow=3)
default_cascade(L, ea, el)
```

default_cascade `Clearing Vector with Bankruptcy Costs`

Description

Computes bank defaults for the clearing vector approach without and with bankruptcy costs (Eisenberg and Noe, 2001), (Rogers and Veraart, 2013).

Usage

```r
default_cashing(L, ea, el = 0, alpha = 1, beta = 1)
```

Arguments

- `L`: Liabilities matrix
- `ea`: Vector of external assets
- `el`: Vector of external liabilities (default 0)
- `alpha`: 1-proportional default costs on external assets in [0, 1] (default to 1).
- `beta`: 1-proportional default costs on interbank assets in [0, 1] (defaults to 1).

Details

Without bankruptcy costs the approach of Eisenberg and Noe (2001) is used using a linear programme. With bankruptcy costs, the implementation is based on the Greatest Clearing Vector Algorithm (GA), see Definition 3.6, Rogers & Veraart (2013).

Value

A list consisting of a vector indicating which banks default (1=default, 0= no default) and the greatest clearing vector.

References


**Description**

Does one Gibbs Step on a cycle

**Usage**

```r
ERE_step_cycle(r, c, L, lambda, p, eps = 1e-10)
```

**Arguments**

- `r`: Row indices of cycle, starting at 0 (vector of length k)
- `c`: Column indices of cycle, starting at 0 (vector of length k)
- `L`: nxn matrix with nonnegative values (will be modified)
- `lambda`: nxn matrix of intensities
- `p`: nxn matrix of probabilities (must be in [0,1] and 0 on diagonal)
- `eps`: Threshold for values to be interpreted as equal to 0 (default = 1e-10)

**Value**

no return value

**Examples**

```r
L = matrix(rexp(9), nrow=3)
lambda <- matrix(0.5, nrow=3, ncol=3)
p <- matrix(0.7, nrow=3, ncol=3)
ERE_step_cycle(r=c(0,1), c=c(1,2), L=L, lambda=lambda, p=p)
ERE_step_cycle(r=c(0,1,2), c=c(0,1,2), L=L, lambda=lambda, p=p)
ERE_step_cycle(r=c(0,1,2), c=c(2,1,0), L=L, lambda=lambda, p=p)
```
**getfeasibleMatr** *Creates a feasible starting matrix*

**Description**

Creates a matrix with nonnegative entries, given row and column sums and 0 on the diagonal.

**Usage**

\[ \text{getfeasibleMatr}(L, A) \]

**Arguments**

- **L**: Vector of row sums
- **A**: Vector of column sums

**Value**

A matrix with nonnegative entries and given row/column sums and 0 on the diagonal.

**Examples**

- getfeasibleMatr(c(0.5, 1, 0), c(0.5, 0, 1))
- getfeasibleMatr(rep(1, 4), rep(1, 4))
- getfeasibleMatr(2^c(1:3), 2^c(3:1))
- getfeasibleMatr(1:5, 1:5)
- getfeasibleMatr(1:5, 5:1)

---

**GibbsSteps_kcycle** *Gibbs sampling step of a matrix in the ERE model*

**Description**

The sampling is conditional on row and column sums and uses k-cycle steps. Then dimensions of L, lambda and p must match.

**Usage**

\[ \text{GibbsSteps_kcycle}(L, \lambda, p, \text{it} = 1000, \text{eps} = 1e-10, \text{debug} = 0) \]
sample_ERE

Arguments

- \textbf{L} \hspace{1cm} \text{Starting matrix - will be modified to contain the results.}
- \textbf{lambda} \hspace{1cm} \text{Matrix of intensities}
- \textbf{p} \hspace{1cm} \text{Matrix of probabilities (must be in \([0,1]\))}
- \textbf{it} \hspace{1cm} \text{Number of iterations (default=1000)}
- \textbf{eps} \hspace{1cm} \text{Threshold for values to be interpreted as equal to 0 (default = 1e-10)}
- \textbf{debug} \hspace{1cm} \text{Should additional debug information be printed? (0 no output, 1 output debug information)}

Value

\text{no return value}

Examples

\begin{verbatim}
L <- matrix(c(1,2,3,4,5,6,7,8,9),nrow=3)
diag(L) <- 0
lambda <- matrix(0.5,nrow=3,ncol=3)
p <- matrix(0.7, nrow=3, ncol=3)
diag(p) <- 0
GibbsSteps_kcycle(L=L,lambda=lambda,p=p)
L
L <- matrix(1:16,nrow=4)
diag(L) <- 0
lambda <- matrix(0.5,nrow=4,ncol=4)
p <- matrix(0.25, nrow=4, ncol=4)
diag(p) <- 0
GibbsSteps_kcycle(L=L,lambda=lambda,p=p)
L
\end{verbatim}

Description

Samples from the Erdos Reny model with Exponential weights and known marginals. Runs a Gibbs sampler to do this. A starting liabilities is generated via \texttt{getfeasibleMatr} before \texttt{steps_ERE} is called.

Usage

\texttt{sample_ERE(l, a, p, lambda, nsamples = 10000, thin = 1000, burnin = 10000)}


**Arguments**

- `l` vector of interbank liabilities
- `a` vector of interbank assets
- `p` Probability of existence of a link (either a numerical value or a matrix with `diag(p)=0`)
- `lambda` (either a numerical value or a matrix with `diag(lambda)=0`)
- `nsamples` Number of samples to return.
- `thin` Frequency at which samples should be generated (default=1, every step)
- `burnin` Number of initial steps to discard.

**Value**

List of simulation results

**Examples**

```r
l <- c(1, 2.5, 3)
a <- c(0.7, 2.7, 3.1)
L <- sample_ERE(l, a, p=0.5, lambda=0.25, nsamples=5, thin=20, burnin=10)
L
```

---

**steps_ERE**

*Perform Steps of the Gibbs Sampler of the ERE model*

**Description**

Runs a Gibbs sampler in the Erdos Renyi model with Exponential weights (ERE model) and fixed marginals. The algorithm starts from a given matrix.

**Usage**

```r
steps_ERE(L, p, lambda, nsamples = 10000, thin = 1000, burnin = 10000)
```

**Arguments**

- `L` Starting matrix for the Gibbs sampler. Implicitly defines the fixed marginals.
- `p` A matrix with entries in $[0,1]$.
- `lambda` A matrix with nonnegative entries.
- `nsamples` Number of samples to return.
- `thin` Frequency at which samples should be generated (default=1, every step)
- `burnin` Number of initial steps to discard.

**Value**

List of simulation results
steps_ERE

See Also

sample_ERE

Examples

L <- matrix(rexp(4*4),nrow=4,ncol=4); diag(L)=0;
p <- matrix(0.5,nrow=4,ncol=4); diag(p) <-0;
lambda <- matrix(1,nrow=4,ncol=4); diag(lambda)<-0;

L <- steps_ERE(L=L,p=p,lambda=lambda,nsamples=5,thin=50,burnin=20)
L
Index

cloneMatrix, 2

default, 2
default_cascade, 3, 3
default_clearing, 3, 4

ERE_step_cycle, 2, 5

getfeasiibleMatr, 6, 7
GibbsSteps_kcycle, 2, 6

sample_ERE, 7, 9
steps_ERE, 7, 8