Package ‘ar.matrix’

Type Package
Title Simulate Auto Regressive Data from Precision Matrices
Version 0.1.0
Author Neal Marquez [aut, cre, cph]
Maintainer Neal Marquez <nmarquez@uw.edu>
Description Using sparse precision matrices and Choleski factorization simulates data that is auto-regressive.
Depends R (>= 3.3.0)
Imports MASS, Matrix, sparseMVN, sp
Suggests ggplot2, leaflet
License GPL (>= 2)
BugReports https://github.com/nmmarquez/ar.matrix/issues
Encoding UTF-8
LazyData true
RoxygenNote 6.1.1
NeedsCompilation no
Repository CRAN
Date/Publication 2018-12-02 17:30:06 UTC

R topics documented:

Q.AR1 ............................................. 2
Q.iid ............................................ 3
Q.ICAR ........................................... 4
Q.mBYM .......................................... 5
Q.pCAR .......................................... 6
sim.AR .......................................... 7
US.df ........................................... 8
US.graph ................................. 9

Index 10
Description

Functions for creating precision matrices and observations of an AR1 process.

Usage

\texttt{Q.ARI}(M, sigma, rho, sparse=FALSE, vcov=FALSE)

\texttt{r.ARI}(n, M, sigma, rho)

Arguments

\texttt{M} \hspace{1cm} \text{int} > 0, \text{number of elements in the AR1 process.}

\texttt{sigma} \hspace{1cm} \text{float} > 0, \text{pairwise observation standard deviation.}

\texttt{rho} \hspace{1cm} \text{float} \geq 0 \& < 1, \text{how correlated pairwise observations are. The function will still run with values outside of the range [0,1) however the stability of the simulation results are not guaranteed.}

\texttt{sparse} \hspace{1cm} \text{bool} Should the matrix be of class 'dsCMatrix'

\texttt{vcov} \hspace{1cm} \text{bool} If the vcov matrix should be returned instead of the precision matrix.

\texttt{n} \hspace{1cm} \text{int} > 0, \text{number of observations to simulate from the GMRF.}

Value

\texttt{Q.ARI} returns either a precision or variance-covariance function with a AR1 structure.

\texttt{r.ARI} returns a matrix with \texttt{n} rows which are the \texttt{n} observations of a Gaussian Markov random field AR1 process.

Examples

\texttt{require("ggplot2")}
\texttt{# simulate AR1 GMRF}
\texttt{obs <- r.ARI(100, M=30, sigma=1, rho=.98)}
\texttt{# resulting matrix is n x M}
\texttt{dim(obs)}
\texttt{# subtract off the first time point to more easily observe correlation}
\texttt{obs_adj <- obs - obs[1,]}
\texttt{# move objects to a data frame}
\texttt{ar1_df <- data.frame(obs=c(t(obs_adj)), realization=rep(1:100, each=30), time=rep(1:30, 100))}
\texttt{# plot each realization}
\texttt{ggplot(data=ar1_df, aes(time, obs, group=realization, color=realization)) + geom_line()}
Q.iid

**Precision matrix for a IID process**

**Description**

Functions for creating precision matrices and observations of a independent identically distributed GMRF process.

**Usage**

```r
Q.iid(M, sigma, sparse=FALSE, vcov=FALSE)

r.iid(n, M, sigma)
```

**Arguments**

- `M` int > 0, number of elements in the process.
- `sigma` float > 0, standard deviation.
- `sparse` bool Should the matrix be of class 'dsCMatrix'
- `vcov` bool If the vcov matrix should be returned instead of the precision matrix.
- `n` int > 0, number of observations to simulate from the GMRF.

**Value**

Q.iid returns either a precision or variance-covariance function with iid structure.

r.iid returns a matrix with n rows which are the n observations of a Gaussian Markov random field iid process.

**Examples**

```r
require("leaflet")
require("sp")

# simulate iid data and attach to spatial polygons data frame
US.df@data$data <- c(r.iid(1, M=nrow(US.graph), sigma=1))

# color palette of data
pal <- colorNumeric(palette="YlGnBu", domain=US.df$data)

# see map
map1 <- leaflet()
    %>%
    addProviderTiles("CartoDBPOSITRON")
    %>%
    addPolygons(data=US.df, fillColor=pal(data), color="#b2aeae",
                fillOpacity=0.7, weight=0.3, smoothFactor=0.2)
    %>%
    addLegend("bottomright", pal=pal, values=US.df$data, title="", opacity=1)
map1
```
Q.lCAR  

*Precision matrix for a pCAR process*

**Description**

Functions for creating precision matrices and observations of a Leroux CAR(lCAR) process as defined in MacNab 2011. The matrix defines the precision of estimates when observations share connections which are conditionally auto-regressive(CAR).

**Usage**

\[ Q.lCAR(\text{graph}, \sigma, \rho, \text{sparse} = \text{FALSE}, \text{vcov} = \text{FALSE}) \]

\[ r.lCAR(n, \text{graph}, \sigma, \rho) \]

**Arguments**

- **graph**: matrix, square matrix indicating where two observations are connected (and therefore conditionally auto-regressive).
- **sigma**: float > 0, process standard deviation see MacNab 2011.
- **rho**: float >= 0 & < 1, how correlated neighbors are. The function will still run with values outside of the range [0,1) however the stability of the simulation results are not guaranteed see MacNab 2011.
- **sparse**: bool Should the matrix be of class 'dsCMatrix'
- **vcov**: bool If the vcov matrix should be returned instead of the precision matrix.
- **n**: int > 0, number of observations to simulate from the GMRF.

**Value**

Q.lCAR returns either a precision or variance-covariance function with a lCAR structure.

r.lCAR returns a matrix with n rows which are the n observations of a Gaussian Markov random field lCAR process.

**References**


**Examples**

```r
require("leaflet")
require("sp")

# simulate lCAR data and attach to spatial polygons data frame
US.df@data$data <- c(r.lCAR(1, graph = US.graph, sigma = 1, rho = .99))
```
# color palette of data
pal <- colorNumeric(palette="YlGnBu", domain=US.df$data$data)

# see map
map1<-leaflet() %>%
  addProviderTiles("CartoDB.Positron") %>%
  addPolygons(data=US.df, fillColor=pal(data), color="#b2aeae",
  fillOpacity=0.7, weight=0.3, smoothFactor=0.2) %>%
  addLegend("bottomright", pal=pal, values=US.df$data, title="", opacity=1)
map1

---

**Q.mBYM**

**Modified Precision matrix for a BYM process**

**Description**

EXPIREMENTAL. Functions for creating precision matrices and observations of a modified BYM process as defined in MacNab 2011. The matrix defines the precision of estimates when observations share connections which are conditionally auto-regressive (CAR). Because the precision matrix is not symmetric the process is not a true GMRF.

**Usage**

Q.mBYM(graph, sigma, rho, vcov=FALSE)

r.mBYM(n, graph, sigma, rho)

**Arguments**

- **graph**: matrix, square matrix indicating where two observations are connected (and therefore conditionally auto-regressive).
- **sigma**: float > 0, process standard deviation see MacNab 2011.
- **rho**: float >= 0 & < 1, how correlated neighbors are. The function will still run with values outside of the range [0,1) however the stability of the simulation results are not guaranteed. see MacNab 2011.
- **vcov**: bool If the vcov matrix should be returned instead of the precision matrix.
- **n**: int > 0, number of observations to simulate from the GMRF.

**Value**

Q.mBYM returns either a precision or variance-covariance function with a modified BYM structure. r.mBYM returns a matrix with n rows which are the n observations of a pseudo Gaussian Markov random field of a modified BYM process.
References


Examples

```r
## Not run:
require("leaflet")
require("sp")

# simulate mBYM data and attach to spatial polygons data frame
US.df@data$data <- c(r.mBYM(1, graph=US.graph, sigma=1, rho=.99))

# color palette of data
pal <- colorNumeric(palette="YlGnBu", domain=US.df$data)

# see map
map <- leaflet()
  %>% addProviderTiles("CartoDBPOSITRON")
  %>% addPolygons(data=US.df, fillColor=pal(data), color="#2a9d8a",
                 fillOpacity=0.7, weight=0.3, smoothFactor=0.2)
  %>% addLegend("bottomright", pal=pal, values=US.df$data, title="", opacity=1)
map

## End(Not run)
```

**Q.pCAR**

*Precision matrix for a pCAR process*

**Description**

Functions for creating precision matricies and observations of a proper CAR(pCAR) process as defined in MacNab 2011. The matrix defines the precision of estimates when observations share connections which are conditionally auto-regressive(CAR).

**Usage**

```r
Q.pCAR(graph, sigma, rho, sparse=FALSE, vcov=FALSE)
r.pCAR(n, graph, sigma, rho)
```

**Arguments**

- `graph` matrix, square matrix indicating where two observations are connected (and therefore conditionally auto-regressive).
- `sigma` float > 0, process standard derivation see MacNab 2011.
**sim.AR**

Simulate correlated data from a precision matrix.

**Description**

Takes in a square precision matrix, which ideally should be sparse and using Choleski factorization simulates data from a mean 0 process where the inverse of the precision matrix represents the variance-covariance of the points in the process. The resulting simulants represent samples of a Gaussian Markov random field (GMRF).

**Value**

Q.pCAR returns either a precision or variance-covariance function with a pCAR structure.
r.pCAR returns a matrix with n rows which are the n observations of a Gaussian Markov random field pCAR process.

**References**


**Examples**

```r
require("leaflet")
require("sp")

# simulate pCAR data and attach to spatial polygons data frame
US.df@data$data <- c(r.pCAR(1, graph=US.graph, sigma=1, rho=.99))

# color palette of data
pal <- colorNumeric(palette="YlGnBu", domain=US.df@data$data)

# see map
map1 <- leaflet() %>%
  addProviderTiles("CartoDB.Positron") %>%
  addPolygons(data=US.df, fillColor=pal(data), color="#b2aeae",
  fillOpacity=0.7, weight=0.3, smoothFactor=0.2) %>%
  addLegend("bottomright", pal=pal, values=US.df$data, title="", opacity=1)
map1
```
Usage

\texttt{sim.Nar(n, Q)}

Arguments

\begin{itemize}
  \item \texttt{n} \quad \text{int > 0, number of observations to simulate from the GMRF.}
  \item \texttt{Q} \quad \text{matrix, a square precision matrix.}
\end{itemize}

Value

Matrix object, matrix where each row is a single observation from a GMRF with covariance structure $Q^{-1}$.

Examples

```r
require("ggplot2")

# simulate 2D ar1 process
# pairwise correlation
rho <- .95
# pairwise variance
sigma <- .5

# 2 dimensions of simulations
years <- 20
ages <- 10

# kronnecker product to get joint covariance
Q2D <- kronecker(Q.AR1(M=years, sigma, rho), Q.AR1(M=ages, sigma, rho))

# simulate the data and place it in a data frame
Q2D.df <- data.frame(obs=c(sim.AR1(1, Q2D)), age=rep(1:ages, years),
                     year=rep(1:years, each=ages))

# graph results
ggplot(data=Q2D.df, aes(year, obs, group=age, color=age)) + geom_line()
```

---

\underline{US.df}

\textit{Spatial Polygons Data Frame of Counties for Several States}

Description

Spatial Polygons data frame with 475 counties from the US states Louisiana, Texas, Mississippi, & Arkansas. FIPS codes for the state and county are provided in the data frame.
US.graph  

Matrix of Shared Boundaries Between US.df Counties

**Description**

A 475x475 matrix where the index corresponds to a row in the US.df Spatial Polygons data frame and the index of the matrix at row $i$ column $j$ is 1 when US.df[$i$,] and US.df[$j$,] share a border and 0 when they do not.
Index

*Topic data
  US.df, 8
  US.graph, 9

Q.AR1, 2
Q.iid, 3
Q.1CAR, 4
Q.mBYM, 5
Q.pCAR, 6

r.AR1 (Q.AR1), 2
r.iid (Q.iid), 3
r.1CAR (Q.1CAR), 4
r.mBYM (Q.mBYM), 5
r.pCAR (Q.pCAR), 6

sim.AR, 7

US.df, 8
US.graph, 9