

# Crash Introduction to markovchain R package

Giorgio Alfredo Spedicato, Ph.D C.Stat ACAS

2016-08-29





# Intro

- ▶ The markovchain package (Spedicato 2016) will be introduced.
- ▶ The package is intended to provide S4 classes to perform probabilistic and statistical analysis of Discrete Time Markov Chains (DTMC). See (Brémaud 1999) for a theoretical review of the mathematics underlying the DTMC models.
- ▶ The vignette will show: how to load the package and create a DTMC, how to manage a DTMC, how to perform basic probabilistic analysis, how to fit a DTMC.

- ▶ The package is on Cran since Summer 2013.
- ▶ It requires a recent version of R ( $\geq 3.0$ ). Since version 0.2 parts of code have been moved to Rcpp (Eddelbuettel 2013).
- ▶ The package won a slot in Google Summer of Code 2015 for optimizing internals and expanding functionalities.

# First moves into the markovchain package

## Loading the package

- ▶ The package is loaded using

```
#load the package  
library(markovchain)
```

## Creating a DTMC

- ▶ DTMC can be easily create following standard S4 classes syntax. The show method displays it.

```
tmA <- matrix(c(0,0.5,0.5,.5,0,.5,.5,.5,0),nrow = 3,  
             byrow = TRUE) #define the transition matrix  
dtmcA <- new("markovchain",transitionMatrix=tmA,  
            states=c("a","b","c"),  
            name="MarkovChain A") #create the DTMC  
dtmcA
```

```
## MarkovChain A  
## A 3 - dimensional discrete Markov Chain defined by the  
## a, b, c  
## The transition matrix (by rows) is defined as follows  
##   a   b   c  
## a 0.0 0.5 0.5  
## b 0.5 0.0 0.5  
## c 0.5 0.5 0.0
```

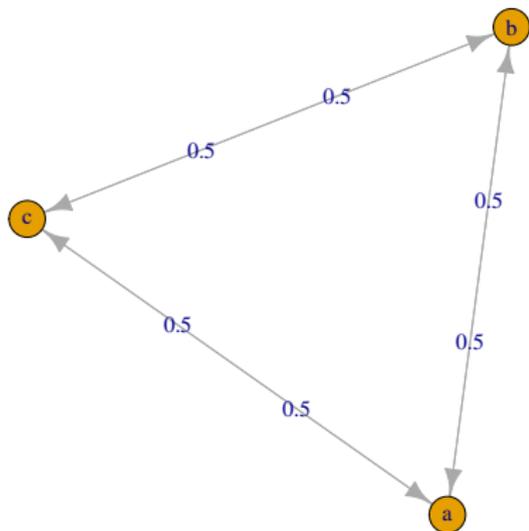
- ▶ Otherwise, it can also be created directly coercing a matrix.

```
dtmcA2<-as(tmA, "markovchain") #using coerce from matrix  
states(dtmcA2) #note default names assigned to states
```

```
## [1] "s1" "s2" "s3"
```

- ▶ It is also possible to display a DTMC, using igraph package (Csardi and Nepusz 2006) capabilities

```
plot(dtmcA)
```



# Probabilistic analysis

## The basic

- ▶ It is possible to access transition probabilities and to perform basic operations.
- ▶ Similarly, it is possible to access the conditional distribution of states,  $Pr(X_{t+1}|X_t = s)$

```
dtmcA[2,3] #using [ method
```

```
## [1] 0.5
```

```
transitionProbability(dtmcA,  
                      "b","c") #using specific S4 method
```

```
## [1] 0.5
```

```
conditionalDistribution(dtmcA,"b")
```

- ▶ It is possible to simulate states distribution after n-steps

```
initialState<-c(0,1,0)
steps<-4
finalState<-initialState*dtmcA^steps #using power operator
finalState
```

```
##           a      b      c
## [1,] 0.3125 0.375 0.3125
```

- ▶ As well as steady states distribution

```
steadyStates(dtmcA) #S4 method
```

```
##           a           b           c  
## [1,] 0.3333333 0.3333333 0.3333333
```

## Advanced

- ▶ We use an example found on Mathematica Web page, (Wolfram Research 2013)

```
E <- matrix(0, nrow = 4, ncol = 4)
E[1, 2] <- 1; E[2, 1] <- 1/3; E[2, 3] <- 2/3
E[3, 2] <- 1/4; E[3, 4] <- 3/4; E[4, 3] <- 1
mcMathematica <- new("markovchain", states = c("a", "b", "c", "d"),
                    transitionMatrix = E, name = "Mathematica")
```

- ▶ The summary method shows the proprieties of the DTCM

```
summary(mcMathematica)
```

```
## Mathematica Markov chain that is composed by:  
## Closed classes:  
## a b c d  
## Recurrent classes:  
## {a,b,c,d}  
## Transient classes:  
## NONE  
## The Markov chain is irreducible  
## The absorbing states are: NONE
```

## Estimation and simulation

The package permits to fit a DTMC estimating the transition matrix from a sequence of data. - `createSequenceMatrix` returns a function showing previous vs actual states from the pairs in a given sequence.

```
#using Alofi rainfall dataset  
data(rain)  
mysequence<-rain$rain  
createSequenceMatrix(mysequence)
```

```
##           0 1-5  6+  
## 0       362 126  60  
## 1-5    136  90  68  
## 6+     50  79 124
```

- ▶ markovchainFit function allows to obtain the estimated transition matrix and the confidence levels (using elliptic MLE hypothesis).

```
myFit<-markovchainFit(data=mysequence,confidencelevel = .9)
myFit
```

```
## $estimate
## MLE Fit
## A 3 - dimensional discrete Markov Chain defined by the
## 0, 1-5, 6+
## The transition matrix (by rows) is defined as follows
##           0           1-5           6+
## 0    0.6605839  0.2299270  0.1094891
## 1-5  0.4625850  0.3061224  0.2312925
## 6+   0.1976285  0.3122530  0.4901186
##
##
## $standardError
##           0           1-5           6+
##           0.0000000  0.0000000  0.0000000
```

- ▶ See the vignettes for further fitting methods as well as for functionalities targeted on non - homogeneous Markov chains.

```
alofiMc<-myFit$estimate  
alofiMc
```

```
## MLE Fit  
## A 3 - dimensional discrete Markov Chain defined by the  
## 0, 1-5, 6+  
## The transition matrix (by rows) is defined as follows  
##           0           1-5           6+  
## 0    0.6605839 0.2299270 0.1094891  
## 1-5  0.4625850 0.3061224 0.2312925  
## 6+   0.1976285 0.3122530 0.4901186
```

# Bibliography I

Brémaud, Pierre. 1999. “Discrete-Time Markov Models.” In *Markov Chains*, 53–93. Springer.

Csardi, Gabor, and Tamas Nepusz. 2006. “The Igraph Software Package for Complex Network Research.” *InterJournal Complex Systems*: 1695. <http://igraph.sf.net>.

Eddelbuettel, Dirk. 2013. *Seamless R and C++ Integration with Rcpp*. New York: Springer-Verlag.

Spedicato, Giorgio Alfredo. 2016. *Markovchain: An R Package to Easily Handle Discrete Markov Chains*.

Wolfram Research, Inc. 2013. *Mathematica*. Ninth. Wolfram Research, Inc.