Tutorial: Studying the MC of Exercise 3-19

Almut Veraart

Autumn 2022

Solving Exercise 2-19 in R

In this tutorial we show how we can solve Exercise 2-19 using R.

Consider a discrete-time homogeneous Markov chain $(X_n)_{n\in\mathbb{N}_0}$ with state space $E = \{1,2,3,4,5,6,7,8\}$ and transition matrix given by

$$\mathbf{P} = \begin{pmatrix} 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0.5 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.25 & 0 & 0.75 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

a) Draw the transition diagram.

b) Specify the communicating classes and determine whether they are transient, null recurrent or positive recurrent. *Please note that you need to justify your answers.*

c) Find all stationary distributions.

d) For each communication class, pick a state i and find the first passage times

$$f_{ii}(n) = \mathrm{P}(X_n = i, X_{n-1} \neq i, \ldots, X_1 \neq i | X_0 = i)$$

for all $n \in \mathbb{N}$ and derive

$$f_{ii} = \sum_{n=1}^{\infty} f_{ii}(n).$$

a) Draw the transition diagram

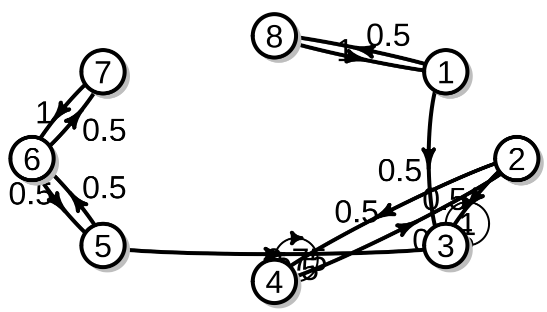
```
library(markovchain)
library(diagram)

#State space
E <- c("1","2","3", "4", "5", "6", "7", "8")


#Transition matrix
P <- matrix(data = c(0, 0, 0.5,0,0,0,0,0.5,
                     0, 0, 0.5,0.5,0,0,0,0,
                     0,0,1,0,0,0,0,0,
```

```
                      0, 0.25, 0, 0.75, 0, 0, 0,0,
                      0,0, 0.5, 0, 0, 0.5, 0, 0,
                      0, 0, 0, 0, 0.5, 0, 0.5, 0,
                      0, 0, 0,0,0,1,0,0,
                      1,0,0,0,0,0,0,0),
             byrow = TRUE, nrow = 8, dimnames = list(E,E)
)
#Create Markov chain
MC <- new("markovchain", states = E, transitionMatrix = P, name = "MC")


#Plot the transition diagramme (transpose P!)
plotmat(t(P), relsize=0.6, box.size=0.05, arr.length=0.2)
```



b) Specify the communicating classes and determine whether they are transient, null recurrent or positive recurrent.

```
summary(MC)
```

```
## MC  Markov chain that is composed by:
## Closed classes:
## 3
## Recurrent classes:
## {3}
## Transient classes:
## {1,8},{2,4},{5,6,7}
```

```
## The Markov chain is not irreducible
## The absorbing states are: 3
```

Note that the justification is as follows: Class $\{3\}$ is finite and closed, hence positive recurrent. The other classes are all not closed, hence transient.

c) Find all stationary distributions.

```r
steadyStates(MC)
```

```
##      1 2 3 4 5 6 7 8
## [1,] 0 0 1 0 0 0 0 0
```

Since we have one closed communicating class on a finite state space, we know that the stationary distribution is unique.

d) Find the first passage probabilities and return probabilities

In the following we compute the

$$f_{ij}(n) = \mathrm{P}(X_n = j, X_{n-1} \neq j, \dots, X_1 \neq j | X_0 = i)$$

for $n \in \{1, \dots, 10\}$ for $i, j \in E$.

```r
#Computing f_ij(n) for n=10 and j\in\{1, \dots, 8\}
for(i in 1:8){
  my_state <- c(i)
  cat("State i=", my_state, ": f_ij(n) for n =1,...10 \n")
  print(firstPassage(object = MC, state = my_state, n = 10))
  cat("\n")

  cat("I.e. for state i=", my_state, ", we have: f_ii(n) for n =1,...10 \n")
  print(firstPassage(object = MC, state = my_state, n = 10)[,i])
  cat("\n \n")
}
```

```
## State i= 1 : f_ij(n) for n =1,...10
##      1 2       3 4 5 6 7   8
## 1  0.0 0 0.50000 0 0 0 0 0.5
## 2  0.5 0 0.00000 0 0 0 0 0.0
## 3  0.0 0 0.25000 0 0 0 0 0.0
## 4  0.0 0 0.00000 0 0 0 0 0.0
## 5  0.0 0 0.12500 0 0 0 0 0.0
## 6  0.0 0 0.00000 0 0 0 0 0.0
## 7  0.0 0 0.06250 0 0 0 0 0.0
## 8  0.0 0 0.00000 0 0 0 0 0.0
## 9  0.0 0 0.03125 0 0 0 0 0.0
## 10 0.0 0 0.00000 0 0 0 0 0.0
##
## I.e. for state i= 1 , we have: f_ii(n) for n =1,...10
##   1   2   3   4   5   6   7   8   9  10
## 0.0 0.5 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
##
##
## State i= 2 : f_ij(n) for n =1,...10
##    1           2           3   4 5 6 7 8
```

```
## 1  0 0.00000000 0.50000000 0.5 0 0 0 0
## 2  0 0.12500000 0.00000000 0.0 0 0 0 0
## 3  0 0.09375000 0.06250000 0.0 0 0 0 0
## 4  0 0.07031250 0.04687500 0.0 0 0 0 0
## 5  0 0.05273438 0.04296875 0.0 0 0 0 0
## 6  0 0.03955078 0.03808594 0.0 0 0 0 0
## 7  0 0.02966309 0.03393555 0.0 0 0 0 0
## 8  0 0.02224731 0.03021240 0.0 0 0 0 0
## 9  0 0.01668549 0.02690125 0.0 0 0 0 0
## 10 0 0.01251411 0.02395248 0.0 0 0 0 0
##
## I.e. for state i= 2 , we have: f_ii(n) for n =1,...10
##          1          2          3          4          5          6          7
## 0.00000000 0.12500000 0.09375000 0.07031250 0.05273438 0.03955078 0.02966309
##          8          9         10
## 0.02224731 0.01668549 0.01251411
##
##
## State i= 3 : f_ij(n) for n =1,...10
##    1 2 3 4 5 6 7 8
## 1  0 0 1 0 0 0 0 0
## 2  0 0 0 0 0 0 0 0
## 3  0 0 0 0 0 0 0 0
## 4  0 0 0 0 0 0 0 0
## 5  0 0 0 0 0 0 0 0
## 6  0 0 0 0 0 0 0 0
## 7  0 0 0 0 0 0 0 0
## 8  0 0 0 0 0 0 0 0
## 9  0 0 0 0 0 0 0 0
## 10 0 0 0 0 0 0 0 0
##
## I.e. for state i= 3 , we have: f_ii(n) for n =1,...10
##  1  2  3  4  5  6  7  8  9 10
##  1  0  0  0  0  0  0  0  0  0
##
##
## State i= 4 : f_ij(n) for n =1,...10
##    1          2          3     4 5 6 7 8
## 1  0 0.25000000 0.00000000 0.750 0 0 0 0
## 2  0 0.18750000 0.12500000 0.125 0 0 0 0
## 3  0 0.14062500 0.09375000 0.000 0 0 0 0
## 4  0 0.10546875 0.08593750 0.000 0 0 0 0
## 5  0 0.07910156 0.07617188 0.000 0 0 0 0
## 6  0 0.05932617 0.06787109 0.000 0 0 0 0
## 7  0 0.04449463 0.06042480 0.000 0 0 0 0
## 8  0 0.03337097 0.05380249 0.000 0 0 0 0
## 9  0 0.02502823 0.04790497 0.000 0 0 0 0
## 10 0 0.01877117 0.04265404 0.000 0 0 0 0
##
## I.e. for state i= 4 , we have: f_ii(n) for n =1,...10
##     1     2     3     4     5     6     7     8     9    10
## 0.750 0.125 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
##
##
```

```
## State i= 5 : f_ij(n) for n =1,...10
##    1 2          3 4        5   6            7 8
## 1  0 0 0.50000000 0 0.000000 0.5 0.0000000000 0
## 2  0 0 0.00000000 0 0.250000 0.0 0.2500000000 0
## 3  0 0 0.12500000 0 0.000000 0.0 0.0000000000 0
## 4  0 0 0.00000000 0 0.125000 0.0 0.0625000000 0
## 5  0 0 0.09375000 0 0.000000 0.0 0.0000000000 0
## 6  0 0 0.00000000 0 0.062500 0.0 0.0156250000 0
## 7  0 0 0.07031250 0 0.000000 0.0 0.0000000000 0
## 8  0 0 0.00000000 0 0.031250 0.0 0.0039062500 0
## 9  0 0 0.05273438 0 0.000000 0.0 0.0000000000 0
## 10 0 0 0.00000000 0 0.015625 0.0 0.0009765625 0
##
## I.e. for state i= 5 , we have: f_ii(n) for n =1,...10
##        1        2        3        4        5        6        7        8
## 0.000000 0.250000 0.000000 0.125000 0.000000 0.062500 0.000000 0.031250
##        9       10
## 0.000000 0.015625
##
##
## State i= 6 : f_ij(n) for n =1,...10
##    1 2          3 4       5    6           7 8
## 1  0 0 0.00000000 0 0.50000 0.00 0.500000000 0
## 2  0 0 0.25000000 0 0.00000 0.75 0.000000000 0
## 3  0 0 0.00000000 0 0.25000 0.00 0.125000000 0
## 4  0 0 0.18750000 0 0.00000 0.00 0.000000000 0
## 5  0 0 0.00000000 0 0.12500 0.00 0.031250000 0
## 6  0 0 0.14062500 0 0.00000 0.00 0.000000000 0
## 7  0 0 0.00000000 0 0.06250 0.00 0.007812500 0
## 8  0 0 0.10546875 0 0.00000 0.00 0.000000000 0
## 9  0 0 0.00000000 0 0.03125 0.00 0.001953125 0
## 10 0 0 0.07910156 0 0.00000 0.00 0.000000000 0
##
## I.e. for state i= 6 , we have: f_ii(n) for n =1,...10
##    1    2    3    4    5    6    7    8    9   10
## 0.00 0.75 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
##
##
## State i= 7 : f_ij(n) for n =1,...10
##    1 2         3 4       5 6           7 8
## 1  0 0 0.0000000 0 0.00000 1 0.000000000 0
## 2  0 0 0.0000000 0 0.50000 0 0.500000000 0
## 3  0 0 0.2500000 0 0.00000 0 0.000000000 0
## 4  0 0 0.0000000 0 0.25000 0 0.125000000 0
## 5  0 0 0.1875000 0 0.00000 0 0.000000000 0
## 6  0 0 0.0000000 0 0.12500 0 0.031250000 0
## 7  0 0 0.1406250 0 0.00000 0 0.000000000 0
## 8  0 0 0.0000000 0 0.06250 0 0.007812500 0
## 9  0 0 0.1054688 0 0.00000 0 0.000000000 0
## 10 0 0 0.0000000 0 0.03125 0 0.001953125 0
##
## I.e. for state i= 7 , we have: f_ii(n) for n =1,...10
##           1           2           3           4           5           6
## 0.000000000 0.500000000 0.000000000 0.125000000 0.000000000 0.031250000
```

```
##           7           8           9          10
## 0.000000000 0.007812500 0.000000000 0.001953125
##
##
## State i= 8 : f_ij(n) for n =1,...10
##    1 2       3 4 5 6 7   8
## 1  1 0 0.00000 0 0 0 0 0.0
## 2  0 0 0.50000 0 0 0 0 0.5
## 3  0 0 0.00000 0 0 0 0 0.0
## 4  0 0 0.25000 0 0 0 0 0.0
## 5  0 0 0.00000 0 0 0 0 0.0
## 6  0 0 0.12500 0 0 0 0 0.0
## 7  0 0 0.00000 0 0 0 0 0.0
## 8  0 0 0.06250 0 0 0 0 0.0
## 9  0 0 0.00000 0 0 0 0 0.0
## 10 0 0 0.03125 0 0 0 0 0.0
##
## I.e. for state i= 8 , we have: f_ii(n) for n =1,...10
##   1   2   3   4   5   6   7   8   9  10
## 0.0 0.5 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
##
##
```

Next we approximate the return probability by

$$f_{ii} \approx \sum_{n=1}^{1000} f_{ii}(n).$$

```
f11n <- firstPassage(object = MC, state = "1", n = 1000)
cat("f_11=", sum(f11n[,1]), "\n")
```

```
## f_11= 0.5
```

```
f88n <- firstPassage(object = MC, state = "8", n = 1000)
cat("f_88=", sum(f88n[,8]), "\n")
```

```
## f_88= 0.5
```

```
f22n <- firstPassage(object = MC, state = "2", n = 1000)
cat("f_22=", sum(f22n[,2]), "\n")
```

```
## f_22= 0.5
```

```
f44n <- firstPassage(object = MC, state = "4", n = 1000)
cat("f_44=", sum(f44n[,4]), "\n")
```

```
## f_44= 0.875
```

```
f55n <- firstPassage(object = MC, state = "5", n = 1000)
cat("f_55=", sum(f55n[,5]), "\n")
```

```
## f_55= 0.5
```

```
f66n <- firstPassage(object = MC, state = "6", n = 1000)
cat("f_661=", sum(f66n[,6]), "\n")
```

```
## f_661= 0.75
```

```r
f77n <- firstPassage(object = MC, state = "7", n = 1000)
cat("f_77=", sum(f77n[,7]), "\n")
```

```
## f_77= 0.6666667
```

```r
f33n <- firstPassage(object = MC, state = "3", n = 1000)
cat("f_33=", sum(f33n[,3]), "\n")
```

```
## f_33= 1
```

We note that $f_{33} = 1$, hence state 3 is recurrent and for all the other states the return probability $f_{ii} < 1$ for $i \in \{1, 2, 4, 5, 6, 7, 8\}$, hence the other states are all transient.

Now compare the results you have derived yourself with the results you obtained here!