Stokastiska processer lab 1

Mona Hazeem & Simon Jorstedt

2021-02-16

Summary

We shall examine how a transition probability matrix evolves over time. Does it converge? We will specifically focus on an example of a coffee-machine which can be in 6 different conditions $(0, 1, 2, 3, 4, 5)$, and whose transition probabilities are described by matrix P defined below.

Definitions

These functions will be used and referenced throughout this report in order to perform the required tasks and calculations on the given material.

```
# Denna funktion räknar ut matrisen A upphöjt till n, enligt den iterativa
# definitionen A^n = A %*% A %*% ... %*% A (n stycken A), med A^0 = I.
# Exempel: mpow(A, 3) == A %*% A %*% A
mpow <- function(A, n) {
resultat <- diag(nrow(A))
potens <- n
while (potens > 0) {
resultat <- A %*% resultat
potens <- potens - 1
}
return(resultat)
}
# Låt A vara en matris innehållandes sannolikheter. Denna funktion testar om
# raderna i A är identiska upp till de d första decimalerna. Som ett exempel,
# talet 0.12309 är lika med 0.12301 upp till den fjärde decimalen, men avrundat
# till 4 decimaler är dessa tal ej lika.
# Funktionen returnerar TRUE om raderna är identiska; FALSE annars.
rows_equal <- function(A, d = 4) {
A_new <- trunc(A * 10^d) # förstora talet och ta heltalsdelen
for (k in 2:nrow(A_new)) {
# Kolla om alla element i rad 1 är lika med motsvarande element i rad k
if (!all(A_new[1, ] == A_new[k, ])) {
# Om något element skiljer sig så är raderna ej lika
return(FALSE)
}
}
# Hamnar vi här så var alla rader lika
return(TRUE)
}
# Låt A och B vara matriser innehållandes sannolikheter. Denna funktion testar
# om elementen A är identiska, upp till de d första decimalerna, med motsvarande
```

```
# element i matrisen B.
# Funktionen returnerar TRUE om matriserna är identiska; FALSE annars.
matrices_equal <- function(A, B, d = 4) {
A_new <- trunc(A * 10^d)
B_new <- trunc(B * 10^d)
if (all(A_new == B_new)) {
return(TRUE)
} else {
return(FALSE)
}
}
```

We define the matrix P below and then use it throughout this report.

```
P <- matrix(c(0, 0, 0, 0.5, 0, 0.5,
0.1, 0.1, 0, 0.4, 0, 0.4,
0, 0.2, 0.2, 0.3, 0, 0.3,
0, 0, 0.3, 0.5, 0, 0.2,
0, 0, 0, 0.4, 0.6, 0,
0, 0, 0, 0, 0.4, 0.6),
nrow = 6,
ncol = 6,
byrow = TRUE)
```

Problem 1

Given the coffee-machine described in the summary, one might ask oneself what is the probability that the coffee-machine will be in a certain condition after a fixed time from now, given that the machine's initial condition is known. This first exercise is dedicated to answering precisely such questions. We know the machine has 6 working-conditions $(0, 1, 2, 3, 4, 5)$. Where 0 indicates that the machine is not working at all and 5 indicates that it is working perfectly.

To illustrate our case, we may begin by asking ourselves; if the coffee-machine is functioning perfectly today, i.e. in condition 5, what is the probability distribution for each of the different conditions the machine has 2 days from now? What would it be 7 days from now? Or 14 days from now? Or even 90 days from now?

These questions are easily answered using the Chapman-Kolmogorov-equations which in effect state that the n-step transition matrix can be obtained by multiplying the transition matrix P by itself n times. This can easily be implemented, and we can extract the 5-th row (corresponding to the probability distribution if we started in condition 5) for the respective days into the future as the following:

```
svar_a = mpow(P,2)[6,] #The matrix is NOT zero-indexed. P has 6 rows.
svar_b = mpow(P,7)[6,] #The CofMa has states 0-5, so row 6 corresponds to
svar_c = mpow(P,14)[6,] #condition 5 in the CofMa.
svar_d = mpow(P,90)[6,]


df <- data.frame(Time = c("In 2 days", "In 1 week",
"In 2 weeks", "In 3 months"))
df <- cbind(df, rbind(mpow(P,2)[6,], mpow(P,7)[6,], mpow(P,14)[6,], mpow(P,90)[6,]))
names(df)[-1] <- paste0("Condition ", 0:5)


knitr::kable(df, digits = 3, caption = "Figure 1: The table showing 2-, 7-, 14-, 90-step transition prol
```

Figure 1: The table showing 2-, 7-, 14-, 90-step transition probabilities for a coffee machine starting in condition 5.

Time	Condition 0	Condition 1	Condition 2	Condition 3	Condition 4	Condition 5
In 2 days	0.000	0.000	0.000	0.160	0.480	0.360
In 1 week	0.002	0.027	0.124	0.323	0.261	0.262
In 2 weeks	0.003	0.026	0.116	0.311	0.272	0.272
In 3 months	0.003	0.026	0.117	0.311	0.272	0.272

Similarly, if the coffee-machine initial state is condition 3, we can calculate the probability distributions for the different states and extract the 4-th row (corresponding to condition 3) as follows:

```
svar_a = mpow(P,3)[4,]
svar_b = mpow(P,7)[4,]
svar_c = mpow(P,14)[4,]
svar_d = mpow(P,90)[4,]

df2 <- data.frame(Time = c("In 3 days", "In 1 week",
"In 2 weeks", "In 3 months"))
df2 <- cbind(df2, rbind(mpow(P,3)[4,], mpow(P,7)[4,], mpow(P,14)[4,], mpow(P,90)[4,]))
names(df2)[-1] <- paste0("Condition ", 0:5)

knitr::kable(df2, digits = 3, caption = "Figure 2: The table showing 3-, 7-, 14-, 90-step transition pr
```

Figure 2: The table showing 3-, 7-, 14-, 90-step transition probabilities for a coffee machine starting in condition 3.

Time	Condition 0	Condition 1	Condition 2	Condition 3	Condition 4	Condition 5
In 3 days	0.006	0.048	0.144	0.289	0.172	0.341
In 1 week	0.002	0.024	0.111	0.306	0.285	0.272
In 2 weeks	0.003	0.026	0.117	0.311	0.272	0.272
In 3 months	0.003	0.026	0.117	0.311	0.272	0.272

Studying figure 1 and 2 we notice that rows further down in the tables seem to converge on something, and finally the last rows of both tables are equivalent. This is interesting because it hints that the long-term transition probabilities are independent of what condition we are starting in.

Problem 2

2.a

We are going to examine whether it is true that the probabilities in the matrix P converge when we multiply the matrix with itself n times, and at which n the convergence take place in that case. It can be assumed that convergence occurs when the difference between the elements in matrix P raised to the power n and $n+1$ is sufficiently small, and when the difference between the rows in P is sufficiently small. As advised in the lab instructions the following while-loop uses the functions matrices_equal and rows_equal. Note that we are here considering two elements to be equal if they are equal up to four decimals.

```
n <- 1
while ((matrices_equal(mpow(P,n),mpow(P,n+1),d=4) == FALSE) && (rows_equal(mpow(P,n),d=4) == FALSE)) {
  n <- n+1
```

```
}
n
```

```
## [1] 20
```

We observe that the conditions are met for P^{20}, and a row from this matrix has the values:

$(0.0025, 0.0259, 0.1165, 0.3108, 0.2720, 0.2720)$

2.b

In this task we are going to calculate the stationary distribution by solving the following system of equations:

$$\pi_j = \sum_i \pi_i \cdot P_{i,j}$$

$$\sum_j \pi_j = 1$$

We write them down in matrix form to maintain the following:

$$\begin{pmatrix} P_{0,0} & P_{1,0} & P_{2,0} & P_{3,0} & P_{4,0} & P_{5,0} \\ P_{0,1} & P_{1,1} & P_{2,1} & P_{3,1} & P_{4,1} & P_{5,1} \\ P_{0,2} & P_{1,2} & P_{2,2} & P_{3,2} & P_{4,2} & P_{5,2} \\ P_{0,3} & P_{1,3} & P_{2,3} & P_{3,3} & P_{4,3} & P_{5,3} \\ P_{0,4} & P_{1,4} & P_{2,4} & P_{3,4} & P_{4,4} & P_{5,4} \\ P_{0,5} & P_{1,5} & P_{2,5} & P_{3,5} & P_{4,5} & P_{5,5} \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \pi_0 \\ \pi_1 \\ \pi_2 \\ \pi_3 \\ \pi_4 \\ \pi_5 \end{pmatrix} = \begin{pmatrix} \pi_0 \\ \pi_1 \\ \pi_2 \\ \pi_3 \\ \pi_4 \\ \pi_5 \\ 1 \end{pmatrix}$$

This system is overdetermined, but we can get it in the right form by deleting an arbitrary row in the left matrix and in the right-hand side vector. In this case we let it be the first row, which gives us the following:

$$\begin{pmatrix} P_{0,1} & P_{1,1} & P_{2,1} & P_{3,1} & P_{4,1} & P_{5,1} \\ P_{0,2} & P_{1,2} & P_{2,2} & P_{3,2} & P_{4,2} & P_{5,2} \\ P_{0,3} & P_{1,3} & P_{2,3} & P_{3,3} & P_{4,3} & P_{5,3} \\ P_{0,4} & P_{1,4} & P_{2,4} & P_{3,4} & P_{4,4} & P_{5,4} \\ P_{0,5} & P_{1,5} & P_{2,5} & P_{3,5} & P_{4,5} & P_{5,5} \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \pi_0 \\ \pi_1 \\ \pi_2 \\ \pi_3 \\ \pi_4 \\ \pi_5 \end{pmatrix} = \begin{pmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \\ \pi_4 \\ \pi_5 \\ 1 \end{pmatrix}$$

In order of removing of the unknown π's in the solution, we may set them all to 0 in the right hand side vector. To maintain our correct system of equations, we must subtract 1 from each of the entries representing the transition probabilities. This gives us:

$$\begin{pmatrix} P_{0,1} & P_{1,1}-1 & P_{2,1} & P_{3,1} & P_{4,1} & P_{5,1} \\ P_{0,2} & P_{1,2} & P_{2,2}-1 & P_{3,2} & P_{4,2} & P_{5,2} \\ P_{0,3} & P_{1,3} & P_{2,3} & P_{3,3}-1 & P_{4,3} & P_{5,3} \\ P_{0,4} & P_{1,4} & P_{2,4} & P_{3,4} & P_{4,4}-1 & P_{5,4} \\ P_{0,5} & P_{1,5} & P_{2,5} & P_{3,5} & P_{4,5} & P_{5,5}-1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \pi_0 \\ \pi_1 \\ \pi_2 \\ \pi_3 \\ \pi_4 \\ \pi_5 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

We use the Solve built-in function in R to solve the system of equations right after we take the transpose matrix and apply the mentioned changes to it to maintain:

```
P2 <-matrix(c(0,-0.9,0.2,0,0,0, 0,0,-0.8,0.3,0,0, 0.5,0.4,0.3,-0.5,0.4,0, 0,0,0,0,-0.4,0.4, 0.5,0.4,0.3
            nrow = 6,
            ncol = 6,
```

```r
            byrow = TRUE)

solve(P2,c(0,0,0,0,0,1))
```

```
## [1] 0.002590674 0.025906736 0.116580311 0.310880829 0.272020725 0.272020725
```

We observe that our answer here is matching with the answer from the previous task 2.a.

Problem 3

If we keep statistics on the machine's condition throughout the year, would the distribution of its conditions then be similar to the stationary distribution? In order to answer this question we will be using some of the simulation functions found in the document "Simulering av diskreta Markovkedjor".

We concluded from the previous task that the pattern for stationary distributions appears after 20 steps. Leading us to assume that it is going to apply in this case as well. To strengthen this assumption we perform a stimulation with 1000 experiments and then illustrate the results using a bar-chart.

```r
gen_sim <- function(x, P) {
u <-runif(1)
y <- 0
test <- P[x + 1, 1]
while (u > test) {
y <- y + 1
test <- test + P[x + 1, y + 1]
}
y
}

set.seed(1)
simulate_chain <- function(x, n, P) {
results <-numeric(n+1)
results[1] <- x
for (i in 2:n) {
results[i] <-gen_sim(results[i - 1], P)
}
results
}

resultat <- simulate_chain(5,1000,P)
```

```r
barplot(table(resultat),
xlab = "The Condition",
ylab = "The Number",
main = "1000 simulations of a Markov chain")
```

In figure 3 the stacks of the histogram seem to coincide well with what we expect from the stationary distribution, since by multiplying the long-term probabilities by 1000, we get that:

Condition 0:	$0.0026 \cdot 1000 = 2.6$
Condition 1:	$0.026 \cdot 1000 = 26$
Condition 2:	$0.12 \cdot 1000 = 120$
Condition 3:	$0.31 \cdot 1000 = 310$
Condition 4:	$0.27 \cdot 1000 = 270$
Condition 5:	$0.27 \cdot 1000 = 270$

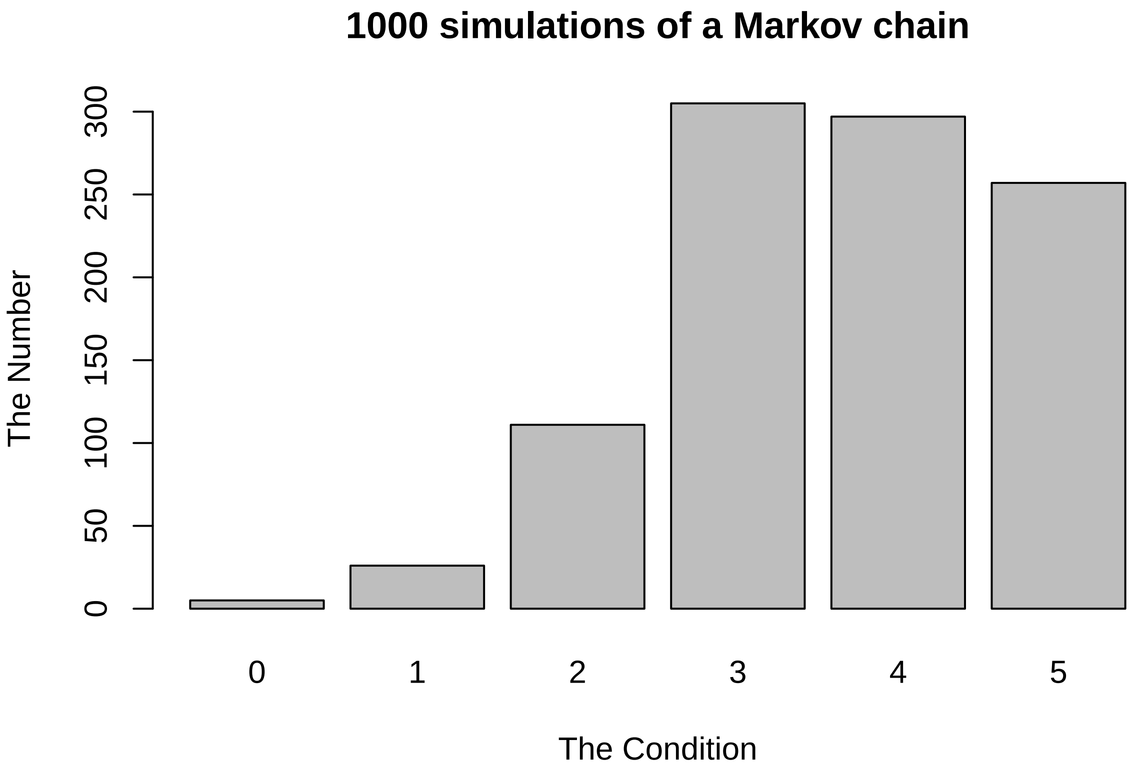

Figure 3: The bar-plot representing the condition of the machine.

The numbers are approximately these frequencies that we can read from the bar-plot graph in figure 3.

Problem 4

If the coffee-machine is in condition 5 one day, what is the probability that it was in condition 1 the day before? This is an interesting problem that can be examined and solved in more than one way. We will be focusing on two different methods using an empirical approach as well as a theoretical approach.

4.a

We shall start empirically by returning to problem 3. Starting at an arbitrary condition we can simulate a large number of transitions. We examine the portion of times condition 5 was visited via condition 1 and then look at all the times condition 5 was visited via any condition (including condition 1). This should coincide well with the probability we are looking after. In other words, if we go through the result-vector from problem 3, we can count the number of times state 5 was visited via condition 1, and divide that by the total number of times condition 5 was visited.

```
last_state = 7
via_1_count = 0
via_some_count = 0

for (state in resultat) {
  if (state == 5) {
    via_some_count <- via_some_count +1
  }
  if (state == 5 && last_state == 1) {
    via_1_count <- via_1_count +1
```

```
  }

  last_state <- state
}

print(via_1_count/via_some_count)
```

```
## [1] 0.03501946
```

Our estimated probability that the last state was 1 if we are in state 5 now, is thus 0.0350195.

4.b

Our second approach to the problem is theoretical. We are going to calculate the probability that yesterday the machine was in condition 1 with help of Bayes theorem for conditional probability. Recall the Bayes formula is:

$$P(A \mid B) = \frac{P(A) \cdot P(B \mid A)}{P(B)}$$

Plugging in our data for this task in the formula yields:

$$P(X_n = 1 \mid X_{n+1} = 5) = \frac{P(X_n = 1) \cdot P(X_{n+1} = 5 \mid X_n = 1)}{P(X_{n+1} = 5)}$$

We know from problem 2 that $P(X_n = 1) = 0.02590674$ and that $P(X_{n+1} = 5) = 0.2720207$. From the matrix P we get: $P(X_{n+1} = 5 \mid X_n = 1) = 0.4$. We Plug in these probabilities in our formula to obtain:

$$P(X_n = 1 \mid X_{n+1} = 5) = \frac{0.02590674 \cdot 0.4}{0.2720207} = 0.0380952$$

We observe that the two results from $4.a$ and $4.b$ are very close ($0.035 \approx 0.038$). More specifically the difference is 0.00307574.