**NATCOR Stochastic Modelling, Lancaster University**

Computer Lab Exercises – Monday March 31st 2025

There are 4 exercises for you to attempt. Don’t worry if you don’t have time to complete all of them. We will go over the main learning points in tomorrow morning’s first lecture.

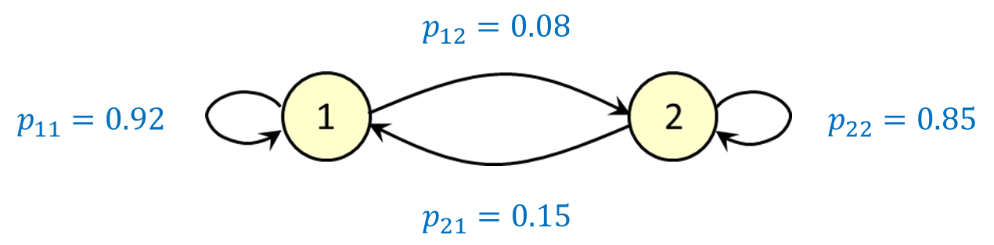
You will need to use both **Microsoft Excel** and **Python**. (You could also use R instead of Python if you want.) You can open Excel via the Windows start menu. To use Python on the university computers, we recommend that you use the university’s **AppsAnywhere** service and load **Anaconda**. Instructions for doing this are as follows:

1. Go to <https://apps.lancaster.ac.uk/>
2. Find **Anaconda** in the list of apps. Click the launch button.
3. You should see an “Anaconda Navigator” window with another list of apps. Find **Spyder** and lick the launch button.
4. In Spyder you can go to File, Open and open a Python script (we will make the scripts available to you). You can then edit the script as needed and run it by clicking the green “Run” button in the toolbar at the top of the screen.

If you run into any problems, just let us know! Also, if you have your own laptop with Python or R already installed, you may prefer to use that.

Exercise 1

This exercise is based on the “brand switching” example from the earlier lecture. This is a Markov chain with only 2 states, representing 2 different brands.



Open the “Brand switching example – spreadsheet” Excel file. The initial market shares for the two brands are shown in cells B2 and C2. Initially, both brands have a 50% market share. The transition probabilities for this example are shown in cells E2, F2, E3 and F3. As a reminder:

* **Cell E2** shows the probability that a consumer will prefer **Brand 1** next week, given that they prefer **Brand 1** in the current week
* **Cell F2** shows the probability that a consumer will prefer **Brand 2** next week, given that they prefer **Brand 1** in the current week
* **Cell E3** shows the probability that a consumer will prefer **Brand 1** next week, given that they prefer **Brand 2** in the current week
* **Cell F3** shows the probability that a consumer will prefer **Brand 2** next week, given that they prefer **Brand 2** in the current week

The formulas in cells B3 and C3 show how to calculate the expected market shares after one week.

**Task 1A:** Select the formulas in cells B3 and C3 and copy the formulas down so that the next 30 rows (at least) are filled in. [Ask a tutor if you are not sure how to do this.] How long does it take until the expected market share for Brand 1 is greater than 60%?

The answer to Task 1A should be 5 weeks.

**Task 1B:** Try changing the initial market shares in cells B2 and C2 so that the initial shares for Brands 1 and 2 are 30% and 70% respectively. How long does it take until the expected market share for Brand 1 is greater than that of Brand 2? What would the answer be if you started with initial shares of 15% and 85% instead?

You should find that the answer to the first part of Task 1B is 4 weeks, and for the second part is 5 weeks.

The rows of the spreadsheet (and also the line chart on the right-hand side) show that, regardless of the initial market shares, the Markov chain quickly converges to a “steady-state distribution”. In other words, the expected market shares converge towards constant values. These constant values depend on the transition probabilities, but they do not depend on the initial market shares. (Note that the initial market shares can be referred to collectively as an “initial probability distribution”. The distribution tells us the probability that a randomly-selected consumer will prefer a particular brand.)

** Task 1C:** By looking further down the rows of the spreadsheet, find the steady-state expected market shares for Brands 1 and 2. Write them down as proportions between 0 and 1, correct to at least 4 decimal places.

You should be able to see from the spreadsheet that the steady-state expected market shares for Brands 1 and 2 (to 2dp) are 0.6522 and 0.3478 respectively.

The answers to Tasks 1A-1C have been obtained by using Excel and inspecting spreadsheet cells. We can also do things using Python. In fact, the steady-state distribution of a Markov chain can be calculated directly by using linear algebra formulas, which can save time. (In tomorrow’s lecture we will see how to find the steady-state distribution by solving a system of linear equations.) It’s easier to use these formulas using Python rather than Excel because Python can more easily handle computations using vectors, matrices etc.

** Task 1D:** Open the Python script “DTMC.py”. This script can be used to find the steady-state distribution of a Markov chain, given the transition probability matrix. In our case, we are interested in using it to calculate the steady-state expected market shares for the two brands. By adjusting line 49 of the script, find the steady-state distribution in each of the following cases:

1. p=np.matrix([[0.96,0.04],[0.07,0.93]])
2. p=np.matrix([[0.75,0.25],[0.22,0.78]])
3. p=np.matrix([[0.99,0.01],[0.02,0.98]])

Note that “p” above represents the transition probability matrix and the numbers in square brackets are the transition probabilities. You should be able to directly copy and paste these lines into Python.

The answers to Task 1D should be:

1. [0.6363, 0.3636]
2. [0.4681, 0.5319]
3. [0.6667, 0.3333]

Are you surprised by how big the differences are between the expected market shares for the two brands in part c), given how similar the transition probabilities are?

** Task 1E: [This is an extra challenge based on the result of Task 1D, part c) above. Feel free to skip it and go straight to Exercise 2. If you want to answer the question but are not sure how, you may find it useful to look at the lecture slides for tomorrow’s lecture.]**

Consider a Markov chain with two states, denoted 1 and 2, and the following transition probability matrix:

where is a small positive constant (assume ). Show that the steady-state probability distribution is

Exercise 2

Consider the example of a shop in which a maximum of 2 customers can be present at any time (see lecture slides). The “state” of the system is the number of customers in the shop. We can model this as a continuous-time Markov chain (CTMC) where:

* If the state is 0 or 1, then the time until the next customer enters the shop is exponentially distributed with a rate of .
* If the state is 1 then the time until the next customer is served is exponentially distributed with a rate of If the state is 2, then the time until the next customer is served is exponentially distributed with a rate of .

Note that in Monday’s lecture slides it was assumed that and .

The generator matrix of the CTMC is:

We will use Python for this exercise. Open the Python script “CTMC.py”. Note that you can change the values of and by changing lines 50 and 51. The program will output the steady-state distribution of the CTMC (which tells you the proportion of time spent in each state), and the long-run average number of customers in the shop.

** Task 2A:** Use the Python script to find the long-run average number of customers in the shop in each of the following cases:

1. and are equal
2. is twice as big as
3. is twice as big as

Note that the answers don’t depend on the exact values of and – only the ratio between and . Which of the 3 cases gives the largest average? Does this make sense?

** Task 2B:** For this task, set and assume must be an integer. Then:

1. Find how large needs to be in order for the average number of customers in the shop to be greater than 1.9.
2. Find how small needs to be in order for the average number of customers in the shop to be smaller than 0.1.

** Task 2C: [This is an extra challenge that will require you to use calculus. Feel free to skip it and go straight to Exercise 3.]** Let denote the long-run proportion of time spent in state , for Show that cannot be greater than (Hint: first calculate an expression for in terms of and using the method described in the Tuesday lecture slides, then use differentiation to find the maximum possible value of .)

Exercise 3

This exercise involves a “Markov chain with rewards”. It is helpful for understanding Markov decision processes, which will be covered later this week.

A broadband internet provider offers two types of subscription: “gold” and “regular”. The gold subscription is more expensive, but gives greater benefits (e.g. faster download speeds). Each household in a particular city can be categorised as either:

* Gold subscriber
* Regular subscriber
* Non-subscriber (they don’t use the company’s services)

The company earns an income of £30 per month for every gold subscriber, and £20 per month for every regular subscriber. We can think of modelling this as a Markov chain where the “state” is the subscription status of an individual household (randomly chosen from the city’s population), and the transition probabilities tell us how households change their subscription preferences from one month to the next. Let the states be written as “G”, “R” and “N”. Suppose the transition probabilities are:

This means that if a household is a gold subscriber in one month, there is a probability of 0.84 that they will still be a gold subscriber in the next month, but a probability of 0.05 that they will become a regular subscriber, and so on.

If we assume that each household makes subscription decisions independently of other households, then it makes sense to calculate the company’s average total income per month from subscription fees as:

where is the steady-state probability of a household being in state , for

** Task 3A:** Use the Python script “DTMC.py” to calculate the company’s average total income per month, assuming the number of households is 30,000. You will need to enter the values of the transition probabilities (given above) in order to find the steady-state probability distribution, then use an additional line of code to calculate the average income using the formula above (or just work this out using a calculator).

Now we will consider a decision-making problem. Suppose the company can improve its subscription rates by advertising (using flyers/leaflets for example). They can also use *targeted advertising*, which means that for each individual household they can decide whether or not to advertise based on the household’s current subscription status. The cost of advertising is £10 per month per household. The table below shows how the transition probabilities are affected if the company uses advertising:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Gold | Regular | Non-subscriber |
| Gold | 0.84 / **0.89** | 0.05 / **0.04** | 0.11 / **0.07** |
| Regular | 0.08 / **0.09** | 0.77 / **0.83** | 0.15 / **0.08** |
| Non-subscriber | 0.02 / **0.07** | 0.06 / **0.12** | 0.92 / **0.81** |

In each of the cells above, the first number (in regular font) shows the transition probability if the company doesn’t advertise – these numbers are the same as shown before. The second number (in bold font) shows the transition probability if advertising is used. For example, the probability that a non-subscriber will change to a gold subscriber is 0.02 if no advertising is used, but 0.07 if advertising is used.

In each month, the company effectively has 3 decisions to make: whether they will advertise to gold subscribers, whether they will advertise to regular subscribers and whether they will advertise to non-subscribers. Since the transition probabilities are the same every month, it makes sense that they will make the same decisions each month, but a different advertising decision will be made for each of the 3 types of household. For each , let if advertising is used and if no advertising is used. The steady-state probability will now depend on the vector , so we will write it as . Now we can calculate the average net profit per month, taking into account advertising costs, as follows:

** Task 3B:** Find the optimal advertising policy which specifies whether or not the company should advertise to each of the 3 household types, and also calculate the average monthly net profit under this policy, assuming that there are 30,000 households. To do this you will need to experiment with different policies, modify the transition probabilities accordingly and calculate the net profit using the formula above.

Exercise 4

In this exercise we will consider a queueing system formulated as a continuous-time Markov chain (CTMC) and calculate its steady-state distribution. Several of the lectures later in the week will discuss queues of various different types.

Consider a queueing system with a single server (meaning that only one customer can be served at a time) and a finite capacity of , where is a positive integer. Customers are served in first-come-first-served order and, after being served, they exit from the system. This can be formulated as a CTMC in which the system state, , is the number of customers in the system (including the customer being served, if there is one).

A diagram of a path to a person walking

AI-generated content may be incorrect.

If then the time until the next customer arrives is exponentially distributed with a rate of . Also, if , then this means a customer is in service and the time until the customer finishes service is exponentially distributed with a rate of .

** Task 4A:** Choose your own values of , and , then think about what the generator matrix for the CTMC should look like and specify the appropriate transition rates in the Python script “CTMC.py”. Run the script in order to find the steady-state distribution. Check that the steady-state probabilities satisfy the formula

where (this is known as the traffic intensity rate of the system). Note that the formula above is only valid if , otherwise we will get a division by zero. Can you work out what the formula should be in the case where ?

** Task 4B:** Work out a formula for the long-run average number of customers in the system in terms of and , and use the Python script to check that your formula is correct.