Applied Stochastic Processes (30515) - Assignment 3

Andrea Lisci

May 2024

1 Part I

The setting of the exercise is a hospital unit, where patients arrive at a rate of 1 client every 48 hours, according to a Poisson process. Once at the hospital, the patients get treated and are dismissed according to an exponential distribution with a mean of 72 hours. The hospital can accomodate up to 4 patients, if a new patient arrive he gets turned to other hospitals.

1.1 Description of $\{X_t\}$

The process is a Continuous time Markov Chain since:

1. Its state space S is discrete: $S = \{0, 1, 2, 3, 4\}$.
2. The number of people at the hospital at a point t only depends on the previous number of people, so: $P(X_{t+s} = j|X_s = i, X_u \text{for} u \in [0, s]) = P(X_{t+s} = j|X_s = i)$ for all $i, j \in S$, $s, t \geq 0$ (Markov property).

This Continuous time Markov chain also evolves according to this dynamic:

1. The time needed for the patients to be treated and dismissed T_i behaves like this: $T_i \sim Exp(v_i)$, where for $i \in S$.
2. The probability of having j patients, starting at a number i is P_{ij} with $i, j \in S$.

So, $\{X_t\}_{t\geq 0}$ is time-homogeneous and in order to specify the continuous time Markov Chain, we just need a vector v_i, with $i \in S$ of transition rates and a transition probability matrix $(P_{ij})_{i,j\in S}$.

1.1.1 Transition rates

In order to obtain a vector v_i of transition rates for our continuous time Markov chain we can start by analyzing the single states. In simple terms, if we start with 0 patients the only thing that can happen is the arrival of a new patient, we know that the arrivals are modelled using a Exponential distribution with a generic λ (rate of one patient every 48 hours in our case), so the transition rate from state 0 only depends on λ, specifically:

$$v_0 = \lambda$$

Now, moving on to state 1, we have two possible situations, the patient gets dismissed or a new patient arrive before the first one is dismissed. So the transition rate doesn't only depend on λ, but also on the time needed to dismiss a patient, we know this is exponentially distributed with parameter mu (in our case we know that its mean is 72 hours). In particular, we are interested in the minimum between the two events, the departure and the arrival, so:

$$T_1 \sim min(Exp(\lambda), Exp(\mu))$$

And for the properties of the minimum of exponentials:

$$T_1 \sim Exp(\lambda + \mu)$$

So, the transition rate is:

$$v_1 = \lambda + \mu$$

With two patients already in the hospital, the situation is slightly different than the previous one, in particular we don't have only one patient that might get dismissed, we have to. So, we have a double minimum problem between exponential:

$$T_2 \sim min(min[Exp(\mu), Exp(\mu)], Exp(\lambda))$$

In other words, we first have to see which one of the two patients takes less time to get dismissed, then the minimum between the first dismissed and the arrival:

$$T_2 \sim Exp((\mu + \mu) + \lambda$$

So, the transition rate is:

$$v_2 = 2\mu + \lambda$$

For the third state the situation is basically the same as the one in state 2, the only difference is that we have 3 people that could be dismissed,so:

$$T_3 \sim min(min(Exp(\mu), Exp(\mu), Exp(\mu)), Exp(\lambda)$$

So:

$$v_3 = 3\mu + \lambda$$

For the fourth and last state possible, we can say that if the hospital unit has 4 patients then its full, new patients will be turned to other hospitals, so the only possible outcome is that one of the 4 patients gets dismissed, so:

$$T_4 \sim min(Exp(\mu), Exp(\mu), Exp(\mu), Exp(\mu))$$

And so:

$$v_4 = 4\mu$$

1.1.2 Transition probability matrix

The transition probability matrix can be derived by defining the various "jumps" as the minimum between the two events: a new patient arrives, a patient leaves, in other words:

$$P_{j,j-1} = P(Exp(\mu_j) < Exp(\lambda))$$

Where $j \in S$ and μ_j is the j-th state's mu value. By using the result of the exponential inequality[1]:

$$P_{j,j-1} = P(Exp(\mu_j) < Exp(\lambda)) = \frac{\mu_j}{\lambda + \mu_j}$$

And:

$$P_{j,j+1} = P(Exp(\lambda) < Exp(\mu_j)) = \frac{\lambda}{\lambda_j + \mu_j}$$

We know have the definitions of the different "jumps", we also have the λ for every state, we found them in the previous point, in the transition rates, so:

$$\text{P} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ \frac{\mu}{\mu+\lambda} & 0 & \frac{\lambda}{\mu+\lambda} & 0 & 0 \\ 0 & \frac{2\mu}{2\mu+\lambda} & 0 & \frac{\lambda}{2\mu+\lambda} & 0 \\ 0 & 0 & \frac{3\mu}{3\mu+\lambda} & 0 & \frac{\lambda}{3\mu+\lambda} \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} (1)$$

1.2 Is the process a birth and death process? If so, describe it in terms of arrival and departure rates

A birth and death process $(X_t)_{t\geq 0}$ with arrival rates $(\lambda_n)_{n\in\mathcal{N}}$ is a continuous time Markov chain with:

1. $S = \mathcal{N}$,
2. Transition rates $v_n = \lambda_n + \mu_n$ for $n \in \mathcal{N}$,
3. Transition probabilities:

$$P = \begin{cases} \frac{\lambda_i}{\lambda_i+\mu_i} & \text{if } j = i+1 \\ \frac{\mu_i}{\lambda_i+\mu_i} & \text{if} j = i-1 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

[1]Let $X_1 \sim Exp(\lambda_1)$ and $X_2 \sim Exp(\lambda_2)$ be independent r.v.'s. Then: $P(X_1 < X_2) = \frac{\lambda_1}{\lambda_1+\lambda_2}$

In order to check if the our specific continuous time Markov Chain is a birth and death process, we just have to check if the previous points apply to it. Starting from the first one, we know that our continuous time Markov Chain has a discrete state space, in particular $S = \{0, 1, 2, 3, 4\}$, so the first one checks. For the second and third point we just have to look at the results we got in the two previous subsections, the transition rates and the transition probabilities, in particular:

1. The transition rates can be written in the form $\lambda_n + \mu_n$ for $n \in \mathcal{N}_{\prime}$, by saying that $\mu_n = i\mu$ with $i = 0, 1, 2, 3, 4$.
2. The transition probabilities are the same we got in the subsection on the transition matrix and, by considering $\mu_n = i\mu$ with $i = 0, 1, 2, 3, 4$, we obtain our transition matrix.

So our CTMC is a birth and death process. We can now describe it in terms of arrival and departure rates: we know that the arrival rate doesn't depend on the number of patients present at the hospital, because of this, its the same for every state except 4 (if we have 4 patients than we can only decrease), so the arrival rate is:

$$\lambda_i = \lambda \forall i \in S/\{4\}$$

The departure rates depend on the number of patients present in the hospital unit, in particular:

$$\mu_i = i\mu \forall i \in S$$

.

1.3 In the long run, what is the proportion of time the hospital unit is empty?

To answer this question we have to find the limiting distribution of the continuous time Markov chain. For CTMC we can use the Chapman-Kolmogorov differential equation:

$$P_{ij}^{'} = \sum_{k \neq j} q_{kj} P_{ik}(t) - v_j P_{ij}(t)$$

From this differential equation the balance equation can be derived. We know that the limiting probabilities $P_j = \lim_{t \to \infty} P_{ij}(t)$, if they exist, must satisfy the balance equation $v_j P_j = \sum_{k \neq j} q_{kj} P_k$ for all $j \in S$ [2]. Now, since this is a birth and death process we can set up a system of differential equations:

$$\begin{cases} P_0 \lambda_0 = P_1 \mu_1 \\ P_1(\lambda_1 + \mu_1) = P_0 \lambda_0 + P_2 \mu_2 \\ P_2(\lambda_2 + \mu_2) = P_1 \lambda_1 + P_3 \mu_3 \\ P_3(\lambda_3 + \mu_3) = P_2 \lambda_2 + P_4 \mu_4 \\ P_4(\lambda_4 + \mu_4) = P_3 \lambda_3 \end{cases} \tag{3}$$

[2]The first term is the rate at which X_t leaves j in balance, the second ther is the rate at which X_t enters j in balance

But we also have the condition that $\sum_{i \in S} P_i = 1$. We know that $\lambda_i = \lambda$ with $i = 1, 2, 3, 4$ and $\mu_1 = \mu, \mu_2 = 2\mu, \mu_3 = 3\mu$ and $\mu_4 = 4\mu$. So, by using the recursive fact that $P_i\lambda = P_{i+1}(i+1)\mu$, we can simplify the system to:

$$\begin{cases} P_0\lambda_0 = P_1\mu \\ P_1\mu = P_0\lambda \\ P_2 2\mu = P_1\lambda \\ P_3 3\mu = P_2\lambda \\ P_4 4\mu = P_3\lambda \\ \sum_{i \in S} P_i = 1 \end{cases} \tag{4}$$

We can now write every P_i with $i = 1, 2, 3, 4$ as a function of P_0:

$$\begin{cases} P_1 = P_0\frac{\lambda}{\mu} \\ P_2 = P_0\frac{(\lambda/\mu)^2}{2}\lambda \\ P_3 = P_0\frac{(\lambda/\mu)^3}{6}\lambda \\ P_4 = P_0\frac{(\lambda/\mu)^4}{24}\lambda \\ \sum_{i \in S} P_i = 1 \end{cases} \tag{5}$$

So $P_i = P_0\frac{(\lambda/\mu)^i}{i!}$ for all $i \in S$. In this case we know that $\lambda = 1/48$ and $\mu = 1/72$. By using the condition that $\sum_{i \in S} P_i = 1$ for all $i \in S$:

$$P_0 = \frac{128}{563} = 0.23$$

1.4 In the long run, what is the average number of patients in the hospital unit?

To calculate the average number of patients in the long run we can use the proportion of time spent in each state, so the limiting probabilities of the process and the vector of the states, so:

$$S \doteq P = P_0(0 + \lambda/\mu + (\lambda/\mu)^2 + (\lambda/\mu)^3/2 + (\lambda/\mu)^4/6) = 1.43$$

1.4.1 In the long run, what is the proportion of patients that have to be turned to other hospitals?

The proportion of patients that have to be turned to other hospitals is the number of patients that arrive when the hospital unit is full, so we can use the proportion of time when the hospital is full (P_4) as a function of P_0:

$$P_0(\frac{\lambda}{\mu})^4/4! = 0.048$$

2 Part II

For this part I used Python. We are now interested in simulating the behaviour of our process, by first creating a cycle that repeats it 100 times.

2.1 Simulate the process (X_t) for $N = 100$ jumps. Store the inter-arrival times T_i for $i = 1, \ldots, N$, the jumping times J_i for $i = 1, \ldots, N$ and the state of the process at the jumping time J

In order to store all of this information I created 3 lists, one for the inter-arrival times, one for the jumping times and one for the state of the process, since 2 separate lists for inter-arrival and jumping time might not be very useful, I decided to create another list called T, in order to store both the inter-arrival and jumping times 1

```
N = 100
for i in range(0,N):
    if X[i]==0:
        tup = np.random.exponential(48)
        X.append(1)
        T.append(tup)
    elif X[i] == 1:
        tup = np.random.exponential(48)
        tdown = np.random.exponential(72)
        if tup < tdown:
            X.append(2)
            T.append(tup)
            arrivals = arrivals +1
            time_in_1 = time_in_1 + tup
        else:
            X.append(0)
            T.append(tdown)
            time_in_1 = time_in_1 + tdown
    elif X[i] == 2:
        tup = np.random.exponential(48)
        tdown = np.random.exponential(72/2)
        if tup < tdown:
            X.append(3)
            T.append(tup)
            arrivals = arrivals +1
            time_in_2 = time_in_2 + tup
        else:
            X.append(1)
            T.append(tdown)
            time_in_2 = time_in_2 + tdown
    elif X[i] == 3:
        tup = np.random.exponential(48)
        tdown = np.random.exponential(72/3)
        if tup < tdown:
            X.append(4)
            T.append(tup)
            arrivals = arrivals +1
            time_in_3 = time_in_3 + tup
        else:
            X.append(2)
            T.append(tdown)
            time_in_3 = time_in_3 + tdown
    elif X[i]==4:
        tdown = np.random.exponential(72/4)
        X.append(3)
        j.append(tdown)
        T.append(tdown)
```

Figure 1: Process simulated 100 times

2.2 Plot

I then used the list X and the list $Tcumulated$ to plot the simulated process (figure 2):

Provide the plot of a typical trajectory of the process

```
T_cumulated = np.cumsum(T)
```

```
plt.plot(T_cum,X)
```

```
[<matplotlib.lines.Line2D at 0x292544d2190>]
```

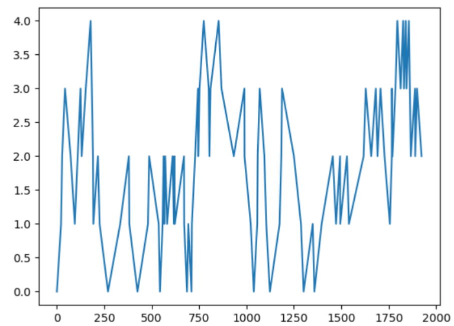

Figure 2: Plot for the process

2.3 Now set a higher (e.g.) and use the simulated process to solve (again) the following:

1. Estimate the proportion of time the hospital unit is empty
2. Estimate the average number of patients in the hospital unit
3. Estimate the proportion of patients that have to be turned to other hospital

In order to calculate this 3 values, I used the cycle built for the previous section (figure 1), but I added a counter for the number of time spent in each state, a counter for the number of arrivals and one for the number of people that get send to other hospitals: figure 3

```python
X = [0]
t = [0]
j = [0]
T = [0]
time_in_0 = 0
time_in_1 = 0
time_in_2 = 0
time_in_3=0
time_in_4 = 0
counter = 0
arrivals = 0
```

Figure 3: New lists and counters

Another thing I had to add to the original cycle is a while cycle into the state number 4, in order to include a particular situation, in which there are 4 patients, the arrival time is less than the departure time and more than one patient arrives before the departure time (figure 4):

```python
if X[i]==4:
    tdown = np.random.exponential(72/4)
    tup = np.random.exponential(48)
    X.append(3)
    j.append(tdown)
    T.append(tdown)
    time_in_4 = time_in_4 + tdown
    wait_bef_dep = tup
    while wait_bef_dep < tdown:
        tup_2 = np.random.exponential(48)
        counter = counter +1
        arrivals = arrivals +1
        wait_bef_dep = wait_bef_dep + tup_2
```

Figure 4: Addition to state 4

2.3.1 Estimate the proportion of time the hospital unit is empty

Now, in order to obtain the proportion of time spent in state 0 I used the list "time in 0" created before the cycle and the total time using the list T (figure 5):

Estimate the proportion of time the hospital unit is empty

```
empty = time_in_0/sum(T)
empty
```

```
0.23069405312385824
```

Figure 5: Proportion of time spent on state 0

2.3.2 Estimate the average number of patients in the hospital unit

For this I used the list that contained the time spent on each state and, like the previous point, the list T (figure 6):

Estimate the average number of patients in the hospital unit

```
states = [0,1,2,3,4]
times_in = [time_in_0,time_in_1,time_in_2,time_in_3,time_in_4]
prop = 0
for i in states:
    prop=prop + i*times_in[i]/sum(T)
prop
```

```
1.4108101402981075
```

Figure 6: Average number of patients in the hospital

2.3.3 Estimate the proportion of patients that have to be turned to other hospital

In order to compute this last value I just used the two counters I initialized before the cycle, so (figure 7):

Estimate the proportion of patients that have to be turned to other hospital

```
turned = counter/arrivals
turned
```

```
0.047798598415600246
```

Figure 7: Proportion of patients that got turned to other hospitals