Michael Batista
Math 6397

Homework 1

Problem. 1 Show that in a pure birth process $\sum_{n=0}^{\infty} P_n(t) = 1$ for all t if and only if $\sum_{n=0}^{\infty} 1/\lambda_n = \infty$. What is an example of an "explosive" process, that is a birth process which cannot be normalized for all finite times.

Assume that $\sum_{n=0}^{\infty} P_n(t) = 1$ for all t. This implies that there is probability 1 that at time t that there is a finite population. Logically, that is the same as $\sum_{n=0}^{\infty} 1/\lambda_n = \infty$ which states that the expected time to reach infinite population is infinity.

Assume that $\sum_{n=0}^{\infty} 1/\lambda_n = \infty$ is true. This implies that the expected time to reach infinite population is infinity. Therefore, the probability to reach infinite population at time t is zero and thus $1 - \sum_{n=0}^{\infty} P_n(t) = 0$ which implies $\sum_{n=0}^{\infty} P_n(t) = 1$.

For the second part of the question, consider the example where $\lambda_n = (1/3)^n$. With this example, we see that

$$\sum_{n=0}^{\infty} \frac{1}{\lambda_n} = \sum_{n=0}^{\infty} \left(\frac{1}{3}\right)^n = \frac{3}{2} < \infty$$

and thus it will approach infinite agents in roughly 1.5 seconds. In my simulation, we see that on average at roughly time 0.75 python identifies the expected time until a new birth as zero seconds. Therefore, it will continue identifying births occurring in zero seconds and never reach any time after the given time. Therefore, we will eventually have infinite births within a finite time frame. Below is the code

```
In [42]: from numpy import random

         timelaps=0
         total=0
         for trials in range(10000):
             count=1
             time=0
             while time<100:
                 lam=pow(1/3,count)
                 birth=random.exponential(scale=count*lam, size=(1, 1))
                 if lam==0:
                     break
                 count=count+1
                 time=time+birth
             total=total+count
             timelaps=timelaps+time
         print(total/(trials+1))
         print(timelaps/(trials+1))

         679.0
         [[0.7465625]]
```

FIGURE 1. Code showing the example where $\lambda_n = (1/3)^n$

Problem. 2 n the simple birth and death process $\lambda_n = \lambda n$ and $\mu_n = \mu n$ for some constants λ and μ. We can add immigration to this process, by assuming that during each interval of length h, there an individual will move into the population with probability νh. Here ν is constant independent of the population size. The deterministic model of the system has the form

$$\frac{dn}{dt} = (\lambda - \mu)n + \nu$$

Show that the mean of the of the corresponding stochastic model, $\overline{n}$, equals the solution of this differential equation, with an appropriate initial condition. Give an example to show that this is not true for more general birth and death processes with immigration, and illustrate the example numerically.

We can see that the general solution to the differential equation is

$$n(t) = Ce^{\lambda t - \mu t} - \frac{c}{\lambda - \mu}$$

I do not know how to calculate the mean from this differential equation. We see that in the simulation that we can identify the mean for any λ, μ and ν over any length of time. For a specific simulation we see that

```
from numpy import random

time=0
mu=.6
lam=.6
nu=.1
total=0
for trials in range(10000):
    count=1
    time=0
    while time<100:
        death=random.exponential(scale=count*mu, size=(1, 1))
        birth=random.exponential(scale=count*lam, size=(1, 1))
        immigration=random.exponential(scale=nu, size=(1, 1))
        if birth<death:
            count=count+1
            time=time+birth
        elif count>0:
            count=count-1
            time=time+death
        ran=random.uniform(0,1)
        if(ran<nu):
            count=count+1
    total=total+count
print(total/(trials+1))
```

```
11.145314531453145
```

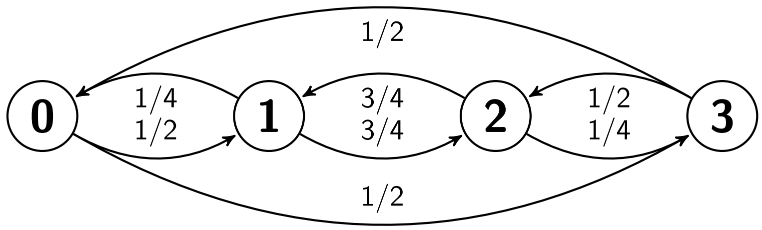
FIGURE 2. Solving for the mean with $\mu = .6$, $= .6$, and $\nu = .1$.

Problem. 3 The transition matrix for a four-state Markov Chain is

$$P = \begin{bmatrix} 0 & 1/2 & 0 & 1/2 \\ 1/4 & 0 & 3/4 & 0 \\ 0 & 3/4 & 0 & 1/4 \\ 1/2 & 0 & 1/2 & 0 \end{bmatrix}$$

(a) Draw the directed graph of the chain
(b) Show that the chain is irreducible, positive recurrent, and periodic. What is the period?
(c) Find the unique stationary probability distribution.

(a) The directed graph of the chain is



(b) We can show that P is irreducible by showing that every position communicates with every other position. To show this, we simply can see the path in which 0 goes to 1, 1 goes to 2, 2 goes to 3, and finally 3 goes back to 0. Each transition in this path has probability greater than zero and contains every position. Therefore, every position can follow this path to make it to any other position with probability greater than 0, thus showing the matrix is irreducible.

It is clear that the transition matrix has period 2. In order to show this, we need to show that every position can only return to itself in some multiple of 2 steps. We can look at direct communication(states that can communicate with other states in one step). We see that 0 directly communicates with 1 and 3, that 1 directly communicates to 0 and 2, that 2 directly communicates with 1 and 3, and that 3 directly communicates with 0 and 2. We see that 0 and 2 directly communicate with 1 and 3, while 1 and 3 also directly communicate with 0 and 2. This means it takes on step to go from either 1 or 3 to either 0 or 2 and one step to go from 1 or 3 to either 0 or 2. Therefore, it takes exactly two steps to go from either 0 or 2 to either 1 or 3, and therefore all paths returning to the state you started will take an even number of steps and thus has period 2.

Finally, we need to show that the chain is positive recurrent. This is quite easy we can see that the probability that state 0 will never return to itself is the same as the probability of return state 1 and 3 always going back to state 2. The probability of that state 3 returning to state 2 occurring n times is $(1/2)^n$, and the probability of state 1 going to state 2 n times is $(3/4)^n$. It is the case that $\lim_{n\to\infty}(1/2)^n = 0$ and $\lim_{n\to\infty}(3/4)^n = 0$ thus there is a probability of zero that state 0 will never return to itself. This same argument can be applied to each state since the transition matrix is of period 2.

(c) We can see that the stationary distribution is equal to

$$\pi = \left[\begin{array}{cccc} 1/6 & 1/3 & 1/3 & 1/6 \end{array}\right]$$

since it follows the formula such that $\pi P = \pi$. However, despite this being a stationary distribution all initial states do not reach this distribution since the period is 2.

Problem. **4** Assume that the arrival of proteins to a promoter can be described as a Poisson process with rate λ. If the promoter is not occupied, it will bind the protein. The time the proteins remain bound are independent random variables with mean μ. A protein that arrives when the promoter is occupied diffuses away. Show that the long time fraction of time that the promoter is unoccupied is $1/(1+\mu\lambda)$.

Through simulations we are able to show that the unoccupied time is equal to $1/(1+\mu\lambda)$ for $mu = 1$, however, this trend does not continue for mu of other values. I am unsure of why this occurs, maybe it it that the random variable for the protein process isn't a Poisson process, since it doesn't explicitly say that it is, or some other issue that I cannot think of. For reference, we can see that the code looks like this and when comparing the expected results with the simulations they are very close

```
mu=1
lam=5
total=0
length=1000
for trials in range(1000):
    time=0
    unoccupied=0
    bound=0
    while time<length:
        protiens=random.exponential(scale=lam, size=(1, 1))
        if bound ==0:
            unoccupied=unoccupied+protiens
            time=time+protiens
            bound=1
            bound_time=random.exponential(scale=mu, size=(1, 1))
        elif bound_time<protiens:
            time=time+protiens
            unoccupied=unoccupied+protiens-bound_time
            bound_time=random.exponential(scale=mu, size=(1, 1))
        else:
            time=time+protiens
            bound_time=bound_time-protiens
    total=total+unoccupied
print(total/(trials+1))
print(length*(1/(1+(1/lam)*(1/mu))))

[[837.68004157]]
833.3333333333334
```

FIGURE 3. Showing that for $\mu = 1$ and $\lambda = 5$ that our code matches expected results

Problem. 5 Assume that the sequence of action potentials (APs) fired by a neuron can be described as a Poisson process. For a fixed time t, let $T(t)$ be the time to the nearest AP in time. This could be an AP preceding or subsequent to the time t. What is the mean of $T(t)$? What is the probability density function of $T(t)$?

We see when simulating this process that $T(t)$ has a similar sojourn time as that of a standard Poisson process for a $\lambda/2$ when using λ as our rate. This means that the mean is half of what the standard Poisson process for rate λ and the pdf is going to likely be that of $\lambda/2$ if I was to take a guess. We have successfully simulated the model, but it is hard to identify if the statements above are true.

```
lam=10
total=0
t=1000
avgmean=0
length=1000
for trials in range(length):
    time=0
    new=0
    old=1000
    mean=0
    while time<t:
        new=random.exponential(scale=lam, size=(1, 1))
        if new<old:
            time=time+new
            mean=new*new/2
            old=0
        else:
            mean=new/2*(new)/4
            time=time+new
            mean=pow((new)/2,2)/2
            old=0

    avgmean=(avgmean+mean)
avgmean=avgmean/(t*length)
print(avgmean)
```

```
[[0.07527258]]
```

FIGURE 4. The main trends we see is that λ increasing makes the time also increase