AMATH 536 Problem Set 2

Cade Ballew #2120804

April 25, 2022

1 Problem 1

1.1 Part a

Let X be a r.v. that denotes the number of individuals in the normal population and let Y be a r.v. that denotes the number of individuals in the mutant population. Assume that $X(0) = a, Y(0) = 0$, both X and Y have growth rate λ, and division of normal bacteria produces a mutant with probability p. We compute

$$P\{X(t+\Delta t)-X(t) = 1, Y(t+\Delta t)-Y(t) = 0|X(t), Y(t)\} = (1-p)\lambda\Delta tX(t)+\mathrm{o}(\Delta t)$$

and

$$P\{X(t+\Delta t)-X(t) = 0, Y(t+\Delta t)-Y(t) = 1|X(t), Y(t)\} = p\lambda\Delta tX(t)+\lambda\Delta tY(t)+\mathrm{o}(\Delta t).$$

Dividing through by Δt, we see by definition that

$$f_{10}(X, Y) = \lambda(1-p)X(t) + \mathrm{o}(1)$$

and

$$f_{01}(X, Y) = \lambda(pX(t) + Y(t)) + \mathrm{o}(1).$$

Now, note that the probability of more than one event occurring in a small time interval Δt is $\mathrm{o}(\Delta t)$, meaning that $f_{ij}(X, Y) = \mathrm{o}(1)$ for $i, j \geq 1$. Sending $\Delta t \to 0$ allows us to drop these asymptotic terms, so we can write

$$\begin{cases} f_{10}(X, Y) = \lambda(1-p)X(t) \\ f_{01}(X, Y) = \lambda(pX(t) + Y(t)) \\ f_{ij}(X, Y) = 0, \quad i, j \geq 1. \end{cases}$$

1.2 Part b

Now, we use this this to write the PDE

$$
\begin{aligned}
\frac{\partial M(\theta,\phi,t)}{\partial t} &= {\sum_{j,k}}' (e^{j\theta+k\phi}-1) f_{jk}\left(\frac{\partial}{\partial\theta},\frac{\partial}{\partial\phi}\right) M(\theta,\phi,t) \\
&= (e^{\theta}-1) f_{10}\left(\frac{\partial}{\partial\theta},\frac{\partial}{\partial\phi}\right) M(\theta,\phi,t) + (e^{\phi}-1) f_{01}\left(\frac{\partial}{\partial\theta},\frac{\partial}{\partial\phi}\right) M(\theta,\phi,t) \\
&= (1-p)\lambda(e^{\theta}-1)\frac{\partial M(\theta,\phi,t)}{\partial\theta} + \lambda(e^{\phi}-1)\left(p\frac{\partial M(\theta,\phi,t)}{\partial\theta} + \frac{\partial M(\theta,\phi,t)}{\partial\phi}\right).
\end{aligned}
$$

Note that an initial condition is given by

$$
M(\theta,\phi,0) = P(e^{\theta},e^{\phi},0) = \sum_{m,n} p_{mn}(0) e^{m\theta+n\phi} = e^{a\theta}.
$$

1.3 Part c

To derive the PDE for the cumulant-generating function, we note that $K = \log M$, so

$$
\frac{\partial K}{\partial t} = \frac{\partial K}{\partial M}\frac{\partial M}{\partial t} = \frac{1}{M}\frac{\partial M}{\partial t}.
$$

Note that we get a similar result by replacing t with θ or ϕ. Then, we can rewrite our PDE as

$$
M\frac{\partial K(\theta,\phi,t)}{\partial t} = M(1-p)\lambda(e^{\theta}-1)\frac{\partial K(\theta,\phi,t)}{\partial\theta} + M\lambda(e^{\phi}-1)\left(p\frac{\partial K(\theta,\phi,t)}{\partial\theta} + \frac{\partial K(\theta,\phi,t)}{\partial\phi}\right),
$$

so

$$
\frac{\partial K(\theta,\phi,t)}{\partial t} = (1-p)\lambda(e^{\theta}-1)\frac{\partial K(\theta,\phi,t)}{\partial\theta} + \lambda(e^{\phi}-1)\left(p\frac{\partial K(\theta,\phi,t)}{\partial\theta} + \frac{\partial K(\theta,\phi,t)}{\partial\phi}\right)
$$

with initial condition

$$
K(\theta,\phi,0) = \log M(\theta,\phi,0) = \log e^{a\theta} = a\theta.
$$

1.4 Part d

Now, let

$$
K(\theta,\phi,t) = {\sum_{j,k}}' \frac{k_{jk}(t)\theta^{j}\phi^{k}}{j!k!}.
$$

Then, we compute

$$
\frac{\partial K}{\partial t} = {\sum_{j,k}}' \frac{k'_{jk}(t)\theta^{j}\phi^{k}}{j!k!},
$$

$$
\frac{\partial K}{\partial \theta} = \sum_{j=1}^{\infty}\sum_{k=0}^{\infty} \frac{k_{jk}(t)\theta^{j-1}\phi^{k}}{(j-1)!k!},
$$

$$\frac{\partial K}{\partial \phi} = \sum_{j=0}^{\infty}\sum_{k=1}^{\infty} \frac{k_{jk}(t)\theta^j \phi^{k-1}}{j!(k-1)!}.$$

Substituting this into part c,

$$\sum_{j,k}{}' \frac{k'_{jk}(t)\theta^j\phi^k}{j!k!} = \lambda\left((1-p)\sum_{\ell=1}^{\infty}\frac{\theta^\ell}{\ell!} + p\sum_{\ell=1}^{\infty}\frac{\phi^\ell}{\ell!}\right)\sum_{j=1}^{\infty}\sum_{k=0}^{\infty}\frac{k_{jk}(t)\theta^{j-1}\phi^k}{(j-1)!k!}$$
$$+\lambda\sum_{\ell=1}^{\infty}\frac{\phi^\ell}{\ell!}\sum_{j=0}^{\infty}\sum_{k=1}^{\infty}\frac{k_{jk}(t)\theta^j\phi^{k-1}}{j!(k-1)!}.$$

1.5 Part e

Now, we use Mathematica[1] to equate coefficients of θ, ϕ, θ^2, $\theta\phi$ and ϕ^2 on both sides of this expression. This yields the system of ODEs

$$\begin{aligned}
k'_{10}(t) &= \lambda(1-p)k_{10}(t)\\
k'_{01}(t) &= \lambda(k_{01}(t) + pk_{10}(t))\\
k'_{11}(t) &= \lambda((2-p)k_{11}(t) + pk_{20}(t))\\
k'_{20}(t) &= \lambda(1-p)(k_{10}(t) + 2k_{20}(t))\\
k'_{02}(t) &= \lambda(k_{01}(t) + 2k_{02}(t) + p(k_{10}(t) + 2k_{11}(t))).
\end{aligned}$$

Since our initial condition only has one term, it is clear that $k_{10}(0) = a$ and $k_{ij}(0) = 0$ for all other i, j.

1.6 Part f

With Mathematica, we find that the solution to this system is given by

$$\begin{aligned}
k_{10}(t) &= ae^{(1-p)\lambda t}\\
k_{01}(t) &= a(e^{\lambda t} - e^{(1-p)\lambda t})\\
k_{11}(t) &= \frac{1}{2}a(1-p)p(e^{(1-p)\lambda t} - e^{(2-p)\lambda t} + e^{(1-p)\lambda t}\lambda t)\\
k_{20}(t) &= \frac{1}{2}ae^{(1-p)\lambda t}(1-p)\lambda t\\
k_{02}(t) &= -\frac{a}{2p}(e^{\lambda t}(1-p\lambda t) - e^{(1-p)\lambda t}(-1+p^3-p^2\lambda t+p^3\lambda t) - e^{(2-p)\lambda t}p^3).
\end{aligned}$$

Of course, this means that

$$\begin{aligned}
E[X(t)] &= ae^{(1-p)\lambda t}\\
E[Y(t)] &= a(e^{\lambda t} - e^{(1-p)\lambda t})\\
\mathrm{Var}[X(t)] &= \frac{1}{2}ae^{(1-p)\lambda t}(1-p)\lambda t\\
\mathrm{Var}[Y(t)] &= -\frac{a}{2p}(e^{\lambda t}(1-p\lambda t) - e^{(1-p)\lambda t}(-1+p^3-p^2\lambda t+p^3\lambda t) - e^{(2-p)\lambda t}p^3).
\end{aligned}$$

[1] See Appendix A for code

2 Problem 2

2.1 Part a

Consider a biased random walk with probability $p = b/(b+d)$ of moving $+1$ and $1-p$ of moving -1. Assuming that we start at position x, let X_n denote the position of the walker at time n. Then,

$$\text{Prob}(X_{2n} = x) = \binom{2n}{n} p^n (1-p)^n$$

for $n \geq 1$ as we must make each move an equal amount of times and there are $\binom{2n}{n}$ ways to do this. Now, let N_x denote the number of returns to x. Then, using Mathematica to compute the sum,

$$E[N_x] = \sum_{n=1}^{\infty} \text{Prob}(X_{2n} = x) = \frac{1}{\sqrt{(2p-1)^2}} - 1.$$

Of course, this implicitly assumes that $p \neq \frac{1}{2}$. In that case,

$$E[N_x] = \sum_{n=1}^{\infty} \binom{2n}{n} \frac{1}{2^{2n}} = \infty.$$

2.2 Part b

To compute the expected number of times a surviving birth-death process will visit a (large) positive integer x, we note that each x is visited at least once and use that first visit to initialize our random walk. This means that the expected total visits are given by

$$E[N_x] + 1 = \begin{cases} \frac{1}{\sqrt{(2p-1)^2}}, & p \neq \frac{1}{2} \\ \infty, & p = \frac{1}{2}. \end{cases}$$

2.3 Part c

If we assume that each birth of a birth-death process can result in one normal cell and one mutant with some small probability u, there are currently x normal cells, and at most one mutant can be produced before leaving state x, the probability that a mutant will be produced before the number of normal cells changes to $x-1$ or $x+1$ is simply the probability that the next event is a birth which results in a mutant. This is given by

$$pu = \frac{bp}{b+d}.$$

2.4 Part d

Now, the expected number of mutants produced when there are exactly x normal cells is simply given by the expected number of visits to x multiplied by the probability of producing a mutant at state x. Thus, it is given by

$$\begin{cases} \frac{bp}{(b+d)\sqrt{(2p-1)^2}}, & p \neq \frac{1}{2} \\ \infty, & p = \frac{1}{2}. \end{cases}$$

Of course, if our random walk is truly biased, we need not consider the $p = \frac{1}{2}$ case for any part of these problems meaning that all values are finite.

3 Appendix A

The following Mathematica code was used to solve problem 1.

```
K = k10[t]*\[Theta] + k01[t] \[Phi] + k11[t]*\[Theta]*\[Phi] +
k20[t]*\[Theta]^2/2 + k02[t]*\[Phi]^2/2
spaceder[\[Theta]_, \[Phi]_] = \[Lambda]*((1 - p)*(E^\[Theta] - 1) +
p*(E^\[Phi] - 1))*D[K, \[Theta]] + \[Lambda]*(E^\[Phi] - 1)*
D[K, \[Phi]]
timeder[\[Theta]_, \[Phi]_] = D[K, t]
spacederseries =
Normal[Series[spaceder[\[Theta]*g, \[Phi]*g], {g, 0, 2}]] /. g -> 1
timederseries =
Normal[Series[timeder[\[Theta]*g, \[Phi]*g], {g, 0, 2}]] /. g -> 1
```

After noticing the coefficients on the LHS series, we can use

```
c10 = FullSimplify[
Coefficient[spacederseries, \[Theta]] /. \[Phi] -> 0]
c01 = FullSimplify[
Coefficient[spacederseries, \[Phi]] /. \[Theta] -> 0]
c11 = FullSimplify[Coefficient[spacederseries, \[Phi]*\[Theta]]]
c20 = 2*FullSimplify[Coefficient[spacederseries, \[Theta], 2]]
c02 = 2*FullSimplify[Coefficient[spacederseries, \[Phi], 2]]
diffeqn = { k10'[t] == d10,
        k01'[t] == d01,
        k11'[t] == d11,
        k20'[t] == d20,
        k02'[t] == d02,
        k10[0] == a,
        k01[0] == 0,
        k11[0] == 0,
        k20[0] == 0,
        k02[0] == 0
}
ans = FullSimplify[DSolve[ODEsystem, {k10, k01, k11, k20, k02}, t]]
```

The following Mathematica code was used to compute infinite sums in problem 2.

```
FullSimplify[Sum[Binomial[2 n, n] p^n (1 - p)^n, {n, 1, Infinity}]]
FullSimplify[Sum[Binomial[2 n, n] (1/2)^2 n, {n, 1, Infinity}]]
```