Stochastic Processes

Lecture 21: Birth and Death Processes, Epidemic SEIR model

Hailun Zhang@SDS of CUHK-Shenzhen

April 15, 2021

(Discrete time) jump chain

- Consider a CTMC $X = \{X(t), t \geq 0\}$ with generator

$$G = \begin{pmatrix} -1 & 1 & 0 \\ 2 & -4 & 2 \\ 0 & 1 & -1 \end{pmatrix}$$

- Stationary distribution $\pi = \pi_X$
- Jump chain $Y = \{Y_n : n = 0, 1, \ldots, \}$

$$J = \begin{pmatrix} 0 & 1 & 0 \\ .5 & 0 & .5 \\ 0 & 1 & 0 \end{pmatrix}$$

π_Y

- Relationship between π_X and π_Y

Uniformization

- Sample path
- The holding rates at states 1 and 3 are smaller than that at state 2.
- How to "boost" the holding rates so that we get a "uniform" CTMC?
- The magic lies in the "jump matrix".
- Consider a CTMC $\tilde{X}$ with uniform holding time rates $\lambda = 4$ and "jump matrix"

$$J_{\text{Uniformization}} = \begin{pmatrix} 3/4 & 1/4 & 0 \\ 1/2 & 0 & 1/2 \\ 0 & 1/4 & 3/4 \end{pmatrix}.$$

- $N(t)$ is a Poisson process with rate λ.
- $\tilde{X}(t) = Z_{N(t)}$, where $Z = \{Z_n : n \geq 0\}$ is the "uniform jump chain".
- $X \stackrel{d}{=} \tilde{X}$. (Proof needed.)
- For each $t \geq 0$

$$\begin{aligned} P_{ij}(t) &= \mathbb{P}\{Z_{N(t)} = j | Z_0 = i\} \\ &= \sum_{n=0}^{\infty} \mathbb{P}\{Z_n = j | Z_0 = i\} \mathbb{P}\{N(t) = n\} \\ &= \sum_{n=0}^{\infty} (J_{\mathrm{U}})_{ij}^n \frac{(\lambda t)^n}{n!} e^{-\lambda t} \end{aligned}$$

$$P(t) = \sum_{n=0}^{\infty} (J_{\mathrm{U}})^n \frac{(\lambda t)^n}{n!} e^{-\lambda t}.$$

- Stationary distribution of Z π_Z

- A CTMC is uniformizable if

$$\sup_{i \in S} |G_{ii}| < \infty. \tag{1}$$

- Under condition (1), the CTMC is regular.
- How to uniformize the following CTMC?

$$G = \begin{pmatrix} -4 & 2 & 2 \\ 1 & -2 & 1 \\ 2 & 1 & -3 \end{pmatrix}$$

Birth and Death Process

- CTMC's that can only change state by increasing by one, or decreasing by one

$$J_{i,i-1} + J_{i,i+1} = 1.$$

- We say there is a *birth* whenever the state increases by one, and there is a *death* whenever it decreases by one.
- $S = \{0, 1, 2, ...\}$
- Birth/death rate: λ_i, μ_i.

Stationary distribution

Theorem

An irreducible Birth and Death process is positive recurrent if and only if

$$\sum_{j=1}^{\infty}\prod_{i=1}^{j}\frac{\lambda_{i-1}}{\mu_i}<\infty,$$

in which case,

$$\pi(0)=\frac{1}{1+\sum_{j=1}^{\infty}\prod_{i=1}^{j}\frac{\lambda_{i-1}}{\mu_i}}.$$

$M/M/1$ queue

- $\lambda < \mu$
- $\lambda > \mu$
- $\lambda = \mu$

$M/M/1$ loss system

$M/M/c$ queue

$M/M/\infty$ queue

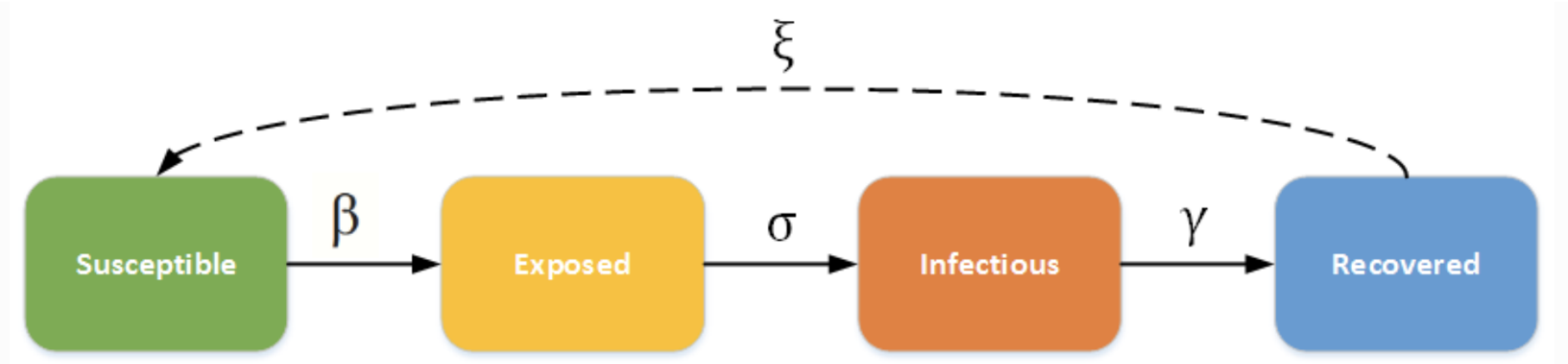
$M/M/c$ loss system

Theorem (Erlang's loss formula)

Loss probability $\pi(c) = \frac{\rho^c}{c!}\left(1 + \sum_{n=1}^{c} \frac{\rho^n}{n!}\right)^{-1}$.

It even holds when the service times are *not* exponential!

Epidemic model: SEIR



- Four types of population: susceptible, exposed, infectious, recovered
- ξ: the rate at which recovered people become susceptible again
- β: the rate of transmitting disease between a susceptible and an infectious individual
- σ: the rate of exposed individuals becoming infectious
- γ: recovery rate determined by average duration of infection.

Four dimensional CTMC

- Consider a population of N.
- $X_t = (S_t, E_t, I_t, R_t)$ with $S_t + E_t + I_t + R_t = N$.
- Transition rate diagram

$$\lambda_{(s,e,i,r),(s-1,e+1,i,r)} = \frac{\beta}{N} si, \ \lambda_{(s,e,i,r),(s,e-1,i+1,r)} = \sigma e, \ldots$$

- Hard to describe evoulution dynamics when N is large

Approximation by ODE

Consider the proportions $X_t^N = (S_t^N, E_t^N, I_t^N, R_t^N) = (S_t/N, E_t/N, I_t/N, R_t/N)$. It can be shown that when $N \to \infty$,

$$X_t^N \to (s_t, e_t, i_t, r_t)$$

where (s_t, e_t, i_t, r_t) is solution to the ODEs

$$\begin{aligned} ds(t)/dt &= -\beta s(t) i(t), \\ de(t)/dt &= \beta s(t) i(t) - \sigma e(t), \\ di(t)/dt &= \sigma e(t) - \gamma i(t), \\ dr(t)/dt &= \gamma i(t). \end{aligned}$$

Simulation code of SEIR process

```
import math
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
def calc(T):
    for i in range(0, len(T) - 1):
        S.append(S[i] - r * b * S[i] * I[i] / N )
        E.append(E[i] + r * b * S[i] * I[i] / N - sigma * E[i]
        I.append(I[i] + sigma * E[i] - gamma * I[i])
        R.append(R[i] + gamma * I[i])
def plot(T,S,E,I,R):
    plt.figure()
    plt.title("SEIR-virus spread curve")
    plt.plot(T,S,color='r',label='Susceptible')
    plt.plot(T, E, color='k', label='Exposed')
    plt.plot(T, I, color='b', label='Infected')
    plt.plot(T, R, color='g', label='Recovered')
```

Simulation code of SEIR process (continued)

```
    plt.grid(False); plt.legend();
    plt.xlabel("time(day)"); plt.ylabel("number of people")
    plt.show()
if __name__ == '__main__':
    N = 10000 # total number of population
    E = []; E.append(0);    # Exposed
    I = []; I.append(1);    # Infectious
    S = []; S.append(N - I[0]) # Susceptible
    R = []; R.append(0) # Recovered

    r = 20 # Average contact number
    b = 0.03 # Infectious rate
    sigma = 0.2 # Incubation rate
    gamma = 0.1 # Recovery rate
    T = [i for i in range(0, 160)] # time horizon

    calc(T); plot(T,S,E,I,R)
```

Evolution dynamics of SEIR process