Final Exam

Austin Pesina

7/7/2021

1. Broken Bottles

A bottling plant has one repairman and three machines. Each machine randomly breaks down at an exponential rate of two times every hour, and the repairman fixes them at an exponential rate of three machines every hour. Whenever all three machines are out of order, however, the plant manager assists the repairman, and the service rate becomes four machines per hour.

(a) Let the state of the system be the number of machines that are broken. Give the transition diagram for the system, and calculate the steady-state probabilities for the system.

$$\rho_n = \frac{\lambda_0 \lambda_1 ... \lambda_{n-1}}{\mu_1 \mu_2 ... \mu_n} \quad n = 1, 2, 3, .., n$$

$$\begin{aligned} \lambda_n &= 2(3-n), && \text{from } n \geq 3 \text{ and} \\ \mu_n &= 3, && \text{for } n < 3 \\ \mu_n &= 4, && \text{for } n = 3 \end{aligned}$$

$$\begin{aligned} \rho_1 &= \frac{\lambda_0}{\mu_1} \\ &= \frac{6}{3} \\ &= 2 \end{aligned}$$

$$\begin{aligned} \rho_2 &= \frac{\lambda_0 \lambda_1}{\mu_1 \mu_2} \\ &= \frac{6 * 4}{3 * 3} \\ &= \frac{24}{9} = \frac{8}{3} \end{aligned}$$

$$\begin{aligned} \rho_3 &= \frac{\lambda_0 \lambda_1 \lambda_2}{\mu_1 \mu_2} \mu_3 \\ &= \frac{6 * 4 * 2}{3 * 3 * 4} \\ &= \frac{48}{36} = \frac{4}{3} \end{aligned}$$

$$
\begin{aligned}
\rho_0 &= \left(1 + \sum_{n=1}^{3} \rho_n\right)^{-1} \\
&= \left(1 + 2 + \frac{8}{3} + \frac{4}{3}\right)^{-1}
\end{aligned}
$$

```
rho_0 <- (1+2+(8/3)+(4/3))^(-1)
rho_0
```

```
## [1] 0.1428571
```

```
rho_1 <- rho_0*2
rho_1
```

```
## [1] 0.2857143
```

```
rho_2 <- rho_0*3
rho_2
```

```
## [1] 0.4285714
```

```
rho_3 <- rho_0*(4/3)
rho_3
```

```
## [1] 0.1904762
```

(b) Calculate the average number of machines that are broken and the average rate at which machine breakdowns occur under steady-state conditions.

```
broken <- rho_0+rho_1+rho_2+rho_3
broken
```

```
## [1] 1.047619
```

The average number of broken machines is 1.047619.

```
time <- 60/broken
time
```

```
## [1] 57.27273
```

The average rate at which a machine breaks down is 57.2727273 minutes.

2. A Weibull Series-Parallel System

Suppose the system in figure 9.5 has components whose lifetimes are IID Weibull distributed with parameters α and λ.

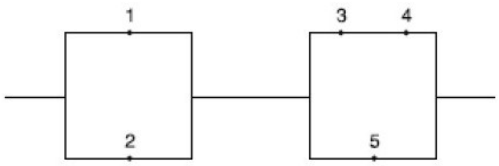

Figure 9.5

What is the distribution for the system lifetime? Give the system reliability function, $R(t) = 1 - F(t)$, and the density function, $f(t)$. For what values of the α parameter is the system IFR or DFR?

Two systems are connected in series: S_1, S_2. In S_1, components 1 and 2 are connected in parallel. In S_2, components 3 and 4 are connected in series and then connected with component 5 in parallel.

The lifetime of each component is given by:

$$P(X \leq t) = 1 - e^{-(\frac{t}{\alpha})^\lambda}$$
$$f(t) = e^{-(\frac{t}{a})^\lambda}$$

Because S_1 has components 1 and 2 in parallel:

$$f(t) = 1 - (1 - e^{-(\frac{t}{a})^\lambda})^2$$

For S_2 components 3 and 4:

$$f(t) = e^{-(\frac{t}{\alpha})^{2\lambda}}$$

and for components 3, 4, and 5:

$$f(t) = 1 - (1 - e^{-(\frac{t}{\alpha})^{2\lambda}})(1 - e^{-(\frac{t}{\alpha})^\lambda})$$

For the whole system, we get:

$$f(t) = \left[1 - (1 - e^{-(\frac{t}{\alpha})^{2\lambda}})(1 - e^{-(\frac{t}{\alpha})^\lambda})\right] \left[1 - (1 - e^{-(\frac{t}{a})^\lambda})^2\right]$$

3. Heavy Training

A coach can give two types of training, “light” or “heavy” to his sports team before a game. If the team wins the prior game, the next training is equally likely to be light or heavy. But, if the team loses the prior game, the next training is always heavy. The probability that the team will win the game is 0.4 after light training and 0.8 after heavy training.

Calculate the long run proportion of time that the coach will give heavy training to the team.

```r
library(markovchain)
```

```
## Package:   markovchain
## Version:   0.8.6
## Date:      2021-05-17
## BugReport: https://github.com/spedygiorgio/markovchain/issues
```

```r
# Setting up transitional matrix

state <- c("win", "lose")
transmat <- matrix(c(0.4, 0.6, 0.8, 0.2), nrow = 2, byrow = T)
row.names(transmat) <- c("light", "heavy")
colnames(transmat) <- state
transmat
```

```
##       win lose
## light 0.4  0.6
## heavy 0.8  0.2
```

```r
train_mc <- new("markovchain", transitionMatrix=transmat,
                states=row.names(transmat), name="Training Markov Chain")
train_mc
```

```
## Training Markov Chain
##  A  2 - dimensional discrete Markov Chain defined by the following states:
##  light, heavy
##  The transition matrix  (by rows)  is defined as follows:
##       light heavy
## light   0.4   0.6
## heavy   0.8   0.2
```

```r
P <- transmat
P2 <- P%*%P; P2
```

```
##        win lose
## light 0.64 0.36
## heavy 0.48 0.52
```

```r
P3 <- P2%*%P2; P3
```

```
##          win   lose
## light 0.5824 0.4176
## heavy 0.5568 0.4432
```

```r
P4 <- P3%*%P3; P4
```

```
##             win      lose
## light 0.5717094 0.4282906
## heavy 0.5710541 0.4289459
```

```r
P5 <- P4%*%P4; P5
```

```
##             win      lose
## light 0.5714288 0.4285712
## heavy 0.5714283 0.4285717
```

As we can see,the probability of a win converges to 0.5714288 or $\frac{4}{7}$, meaning the probability of a loss is 0.4285717 or $\frac{3}{7}$. With heavy training guaranteed after a loss, the probability of heavy training is 42.9%.

4. Pull Them Strings!

At Metropolis City Hall, two workers "pull strings" every day. Strings arrive to be pulled on average of every 10 minutes throughout the day. It takes an average of 15 minutes to pull a string. Both times between arrivals and service times are exponentially distributed.

```
library(queueing)

strings <- NewInput.MMC(lambda = 1/10, mu = 1/15, c = 2, n=0)
out <- QueueingModel(strings)
Report(out)
```

```
## The inputs of the model M/M/c are:
## lambda: 0.1, mu: 0.0666666666666667, c: 2, n: 0, method: Exact
##
## The outputs of the model M/M/c are:
##
## The probability (p0, p1, ..., pn) of the n = 0 clients in the system are:
## 0.1428571
## The traffic intensity is: 1.5
## The server use is: 0.75
## The mean number of clients in the system is: 3.42857142857143
## The mean number of clients in the queue is: 1.92857142857143
## The mean number of clients in the server is: 1.5
## The mean time spend in the system is: 34.2857142857143
## The mean time spend in the queue is: 19.2857142857143
## The mean time spend in the server is: 15
## The mean time spend in the queue when there is queue is: 30
## The throughput is: 0.1
```

(a) What is the probability that there are no strings to be pulled in the system at a random point in time?

As shown in our queueing model, the probability that there are no strings to be pulled at a random time is $Qn = 0.14$

(b) What is the expected number of strings waiting to be pulled?

As shown in our queueing model, the expected number of strings waiting to be pulled is $Lq = 1.928571$

(c) What is the probability that both string pullers are busy?

```
both <- B_erlang(c=2, u=((1/10)/(1/15)))
both
```

```
## [1] 0.3103448
```

The probability that both string pullers are busy is 0.3103448.

(d) What is the effect on performance if a third string puller, working at the same speed as the first two, is added to the system?

```
strings3 <- NewInput.MMC(lambda = 1/10, mu = 1/15, c = 3, n=0)
out3 <- QueueingModel(strings3)
Report(out3)
```

```
## The inputs of the model M/M/c are:
## lambda: 0.1, mu: 0.0666666666666667, c: 3, n: 0, method: Exact
##
## The outputs of the model M/M/c are:
##
## The probability (p0, p1, ..., pn) of the n = 0 clients in the system are:
## 0.2105263
## The traffic intensity is: 1.5
## The server use is: 0.5
## The mean number of clients in the system is: 1.73684210526316
## The mean number of clients in the queue is: 0.236842105263158
## The mean number of clients in the server is: 1.5
## The mean time spend in the system is: 17.3684210526316
## The mean time spend in the queue is: 2.36842105263158
## The mean time spend in the server is: 15
## The mean time spend in the queue when there is queue is: 10
## The throughput is: 0.1
```

The probability that there are no strings waiting to be pulled is $Qn = 0.2105263$. The expected number of strings waiting to be pulled is $Lq = 0.2368421$.

5. The New Copier

A new copy machine is being installed in a library. The librarian estimates that each user will spend 3 minutes on aver with the machine, and wants the average number of users, L, at the facility at any moment to be at most three.

$$\lambda = \text{average number of users coming into the library to use the new copier per hour}$$

$$\mu = \text{average number of users using the new copier per hour} = \frac{60}{3} \therefore \mu = 20$$

(a) Under these conditions, what is the maximum average number of users per hour that the machine can serve? Assume Poisson arrivals, exponential service times, and steady-state conditions.

Since no more than 3 people are allowed, $\lambda \leq 3$ is the maximum number of people allowed at any given moment. With 60 minutes in an hour and less than 4 people allowed at a time, $\frac{60}{4} = 15$.

15 is the maximum number of users the machine can serve per hour.

(b) With the maximum allowed arrival rate from (a), what is the average queue time for a user of the machine?

```
copier3 <- NewInput.MM1(15,20,0)
out_copier3 <- QueueingModel(copier3)
Report(out_copier3)
```

```
## The inputs of the M/M/1 model are:
## lambda: 15, mu: 20, n: 0
##
## The outputs of the M/M/1 model are:
##
## The probability (p0, p1, ..., pn) of the n = 0 clients in the system are:
## 0.25
## The traffic intensity is: 0.75
## The server use is: 0.75
## The mean number of clients in the system is: 3
## The mean number of clients in the queue is: 2.25
## The mean number of clients in the server is: 0.75
## The mean time spend in the system is: 0.2
## The mean time spend in the queue is: 0.15
## The mean time spend in the server is: 0.05
## The mean time spend in the queue when there is queue is: 0.2
## The throughput is: 15
```

According to our model, the average time in the queue is 0.15 minutes. Multiplied by 60 minutes, the average queue time is 9 minutes.

(c) Suppose more space were provided for the machine, so that the facility could accommodate twice as many users ($L = 6$), on average. What would the results of parts (a) and (b) be in this case?

$$\text{Max}\lambda : \frac{\lambda}{20 - \lambda} \leq 6 \text{ or } \lambda \leq 17$$

```r
copier6 <- NewInput.MM1(17,20,0)
out_copier6 <- QueueingModel(copier6)
Report(out_copier6)
```

```
## The inputs of the M/M/1 model are:
## lambda: 17, mu: 20, n: 0
##
## The outputs of the M/M/1 model are:
##
## The probability (p0, p1, ..., pn) of the n = 0 clients in the system are:
## 0.15
## The traffic intensity is: 0.850000000000001
## The server use is: 0.85
## The mean number of clients in the system is: 5.66666666666667
## The mean number of clients in the queue is: 4.81666666666667
## The mean number of clients in the server is: 0.850000000000001
## The mean time spend in the system is: 0.333333333333333
## The mean time spend in the queue is: 0.283333333333333
## The mean time spend in the server is: 0.05
## The mean time spend in the queue when there is queue is: 0.333333333333333
## The throughput is: 17
```

According to our model, the average time in the queue is 0.283 minutes. Multiplied by 60 minutes, the average queue time is 17 minutes.