Homework 5

DATA604 Simulation and Modeling

Daniel Dittenhafer

April 10, 2016

1

At the start of each week, the condition of a machine is determined by measuring the amount of electrical current it uses. According to its amperage reading, the machine is categorized as being in one of the following four states: low, medium, high and failed. A machine in the low state has a probability of 0.05, 0.03, and 0.02 of being in the medium, high, or failed state, respectively, at the start of the next week. A machine in the medium state has a probability of 0.09 and 0.06 of being in the high or failed state, respectively, at the start of the next week (it cannot, by itself, go to the low state). And, a machine in the high state has a probability of 0.1 of being in the failed state at the start of the next week (it cannot, by itself, go to the low or medium state). If a machine is in the failed state at the start of a week, repair is immediately begun on the machine so that it will (with probability 1) be in the low state at the start of the following week. Let X be a Markov chain where Xn is the state of the machine at the start of week n.

a) Given the Markov transition matrix for X.

The following code creates and prints the Markov transition matrix for X.

```
tmX <- matrix(c(0.90, 0.05, 0.03, 0.02,
                0, 0.85, 0.09, 0.06,
                0, 0.00, 0.90, 0.10,
                1, 0.00, 0.00, 0.00), nrow=4, byrow=TRUE)
tmX
```

```
##      [,1] [,2] [,3] [,4]
## [1,]  0.9 0.05 0.03 0.02
## [2,]  0.0 0.85 0.09 0.06
## [3,]  0.0 0.00 0.90 0.10
## [4,]  1.0 0.00 0.00 0.00
```

b) A new machine always starts in the low state. What is the probability that the machine is in the failed state three weeks after it is new?

First we define a helper function:

```
markovChainStep <- function(initialState, tranMatrix, steps)
{
  n <- steps
  step <- initialState
  for(i in 1:n)
  {
    step <- step %*% tranMatrix
  }
  return (step)
}
```

Next we call the helper function to compute the requested probability. We step twice in the function as it is 2 steps beyond the initial probability state.

```
bStep <- markovChainStep(tmX, tmX, 2)
kable(bStep)
```

0.771	0.115875	0.085425	0.0277
0.114	0.617125	0.208575	0.0603
0.180	0.005000	0.732000	0.0830
0.830	0.087500	0.058500	0.0240

The probability that the machine is in the failed state 3 weeks after it is new is 0.0277.

c) What is the probabililty that a machine has at least one failure three weeks after is is new?

Adding up the failure probabilities for each of the 3 weeks would yiield the probabilty of at least one failure during the 3 three week time frame. Here we use the low -> failed probability from week 1, plus the full array of failure probabilites from weeks 2 and 3.

```
firstWeek <- tmX
secondWeek <- markovChainStep(tmX, tmX, 1)
thirdWeek <- bStep

# Add up the probabilities for each week
noFailures <- tmX[1:3,1:3]
noFailsIn3Wks <- markovChainStep(noFailures, noFailures, 2)
noFailsIn3Wks
```

```
##       [,1]     [,2]     [,3]
## [1,] 0.729 0.114875 0.084825
## [2,] 0.000 0.614125 0.206775
## [3,] 0.000 0.000000 0.729000
```

```
#atLeastOneFailaure3Wks <- firstWeek[1,4] + sum(secondWeek[,4]) + sum(thirdWeek[,4])
#atLeastOneFailaure3Wks
```

d) What is the expected number of weeks after a new machine is installed until the first failure occurs?

```
longRun <- markovChainStep(tmX, tmX, 25)
longRun
```

```
##           [,1]      [,2]      [,3]       [,4]
## [1,] 0.4923490 0.1679188 0.2907824 0.04894977
## [2,] 0.4852857 0.1626683 0.3022538 0.04979218
## [3,] 0.4943593 0.1572251 0.2991421 0.04927356
## [4,] 0.4927356 0.1685671 0.2898104 0.04888698
```

```
1 / longRun[1,4]
```

```
## [1] 20.4291
```

e) On average, how many weeks per year is the machine working?

```
# Use Metropolis-Hastings to estimate?
```