Analytics for Competitive Advantage: Lab Exercise 4

Kristin Meier

November 22, 2016

```
##
## Attaching package: 'expm'

## The following object is masked from 'package:Matrix':
##
##     expm
```

Problem 1

In Markov100.txt, the one step transition probability matrix for a Markov chain with 100 states (State 1 to State 100) is given. Note that the data has no heading. Name of the data set Markov100. Number of rows 100. Number of columns 100

```
Markov100 <- read.table(file.path(filepath,"ex4_markov100.txt"),stringsAsFactors = F,header=F)
```

Problem 1(a)

Suppose we are at State 1 now. Find and display the probability of being in State 5 after 10 transitions.

```
# at state 1 now = initial vector
a = c(1,rep(0,99))
P = as.matrix(Markov100)
# Calculate prob distribution after 10 steps given initial vector
probDist10 <- a %*% (P %^%10)
# state 5
state5 <- probDist10[5]
state5
```

```
## [1] 0.045091
```

Problem 1(b)

Suppose we are at one of States 1,2, and 3 with equal probabilities. Find and display the probability of being in State 10 after 10 transitions.

```
# equal probabilities for 1,2,3
three <- 1/3
a2 = c(rep(three,3),rep(0,97))
# Calculate prob distribution after 10 steps given initial vector
probDist10.2 <- a2 %*% (P %^%10)
# state 10
state10 <- probDist10.2[10]
state10
```

```
## [1] 0.08268901
```

Problem 1(c)

Find the steady state probability of being in State 1.

```
# diag(n) # creates identity matrix with size n
# solve(M) # returns inverse of matrix M
n = 100
Q = t(P) - diag(n)
Q[n,] = c(rep(1,n))
rhs = c(rep(0,(n-1)),1)
# steady state probabilities
Pi = solve(Q) %*% rhs
# state 1
Pi[1]
```

```
## [1] 0.01256589
```

Problem 1(d)

Find the mean first passage time from State 1 to State 100.

```
# What is the expected number of steps to get to state 100?
B = P[1:(n-1),1:(n-1)]
Q = diag(n-1) - B
e = c(rep(1,n-1))
m = solve(Q) %*% e
# mean steps from 1 to 100
m[1]
```

```
## [1] 254.9395
```

Problem 2

You are asked to analyze the data from an website with 8 pages. Let us assume that there is a virtual page 9 that a visitor must automatically visit when the visitor leaves the website. The visitors always start their visit from Page 1. Let us formulate a Markov chain for this website.

```
webtraffic <- read.table(file.path(filepath,"ex4_webtraffic.txt"),stringsAsFactors = F,header=T)
```

Problem 2(a)

Construct 9 by 9 matrix Traffic that counts total traffic between State i to State j for all i and j. Display Traffic. Hint colSums() adds all rows for each column.

```
# Total traffic b/w state i to state j
# 1000 rows are 1000 obs. so need to sum them up
ij <- colSums(webtraffic)
# now put them in order
Traffic <- t(ij[1:9])
```

```r
for(i in 2:9){
  Traffic <- rbind(Traffic,t(ij[((9*(i-1))+1):(9*i)]))
}
Traffic <- as.matrix(Traffic)
listname <- seq(1,9)
colnames(Traffic) <- paste("to",listname,sep="")
rownames(Traffic) <- paste("from",listname,sep="")
Traffic
```

```
##       to1 to2 to3 to4 to5 to6 to7 to8 to9
## from1   0 447 553   0   0   0   0   0   0
## from2   0  23 230 321   0   0   0   0  63
## from3   0 167  43 520   0   0   0   0  96
## from4   0   0   0  44 158 312 247   0 124
## from5   0   0   0   0  22  52  90 127 218
## from6   0   0   0   0  67  21   0 294  97
## from7   0   0   0   0   0  94   7 185  58
## from8   0   0   0   0 262   0   0  30 344
## from9   0   0   0   0   0   0   0   0   0
```

Problem 2(b)

Observe that Traffic has 0's in row 9 and 0's in column 1. Set Traffic[9,1]=1000. Construct the one step transition probability matrix P and display it.

```r
Traffic[9,1] = 1000

P = Traffic
# prob matrix, x/n, n = row total
rowtot <- rowSums(P)
for(i in 1:9){
  P[i,] <- P[i,]/rowtot[i]
}
P = round(P,4)
P
```

```
##       to1    to2    to3    to4    to5    to6    to7    to8    to9
## from1   0 0.4470 0.5530 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
## from2   0 0.0361 0.3611 0.5039 0.0000 0.0000 0.0000 0.0000 0.0989
## from3   0 0.2022 0.0521 0.6295 0.0000 0.0000 0.0000 0.0000 0.1162
## from4   0 0.0000 0.0000 0.0497 0.1785 0.3525 0.2791 0.0000 0.1401
## from5   0 0.0000 0.0000 0.0000 0.0432 0.1022 0.1768 0.2495 0.4283
## from6   0 0.0000 0.0000 0.0000 0.1399 0.0438 0.0000 0.6138 0.2025
## from7   0 0.0000 0.0000 0.0000 0.0000 0.2733 0.0203 0.5378 0.1686
## from8   0 0.0000 0.0000 0.0000 0.4119 0.0000 0.0000 0.0472 0.5409
## from9   1 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
```

Problem 2(c)

Calculate and display the steady state probability vector Pi

```
n = 9
Q = t(P) - diag(n)
Q[n,] = c(rep(1,n))
rhs = c(rep(0,(n-1)),1)
# steady state probabilities
Pi = solve(Q) %*% rhs
rownames(Pi) = listname
Pi
```

```
##         [,1]
## 1 0.15833144
## 2 0.10086153
## 3 0.13079268
## 4 0.14012219
## 5 0.08057655
## 6 0.07583329
## 7 0.05445957
## 8 0.10069131
## 9 0.15833144
```

Problem 2(d)

The following table presents the average time that the visitors spend on each page. Page 1 2 3 4 5 6 7 8. Avg(minute) 0.1 2 3 5 5 3 3 2. Calculate and display the average time a visitor spend on the website (until she leaves).

```
# calcualte first mean passage time
# multiply that by the time
avgpg <- c(0.1, 2, 3, 5, 5, 3, 3, 2)

B = P[1:(n-1),1:(n-1)]
Q = diag(n-1) - B
# e is the average time spent on each page
e = avgpg
m = solve(Q) %*% e
# mean time spent starting from pg 1
passagevec <- t(m)
timespent <- passagevec[1]
timespent
```

```
## [1] 14.56242
```

Problem 2(e)

In the output of Problem 2(c), observe that Pages 3 and 4 are one of the most crowded pages except Pages 1 and 9. To balance the traffic, the owner of the website decided to create links from Page 2 to Pages 6,7 (hence, from State 2 to States 6,7). By adding the links, the owner anticipates that, from Page 2, 30% of the current outgoing traffic to State 3 would move to State 6, and 20% of the current outgoing traffic to State 4 would move to State 7. Calculate new steady state probability vector Pi2 to check the effect of the new links. Decide if the link helped balancing the traffic by comparing the variance of Pi and Pi2. Hint Start with matrix Traffic from Problem 2(a).

```r
# .3 2,3 -> 2,6
#  .2 2,4 -> 2,7
## first add the additional traffic to pg 6 and 7
Traffic[2,6] <- Traffic[2,6] + .3*Traffic[2,3]
Traffic[2,7] <- Traffic[2,7] + .2*Traffic[2,4]
## next reduce the current traffic
Traffic[2,3] <- .7*Traffic[2,3]
Traffic[2,4] <- .8*Traffic[2,4]

## recalc prob. matrix
P2 = Traffic
# prob matrix, x/n, n = row total
rowtot <- rowSums(P2)
for(i in 1:9){
  P2[i,] <- P2[i,]/rowtot[i]
}
P2 <- round(P2,4)

## recalculate steady state prob
n = 9
Q = t(P2) - diag(n)
Q[n,] = c(rep(1,n))
rhs = c(rep(0,(n-1)),1)
# steady state probabilities
Pi2 = solve(Q) %*% rhs
rownames(Pi2) <- listname
Pi2
```

```
##         [,1]
## 1 0.16163806
## 2 0.10035150
## 3 0.12105145
## 4 0.12275448
## 5 0.08163241
## 6 0.08250087
## 7 0.06002737
## 8 0.10840580
## 9 0.16163806
```

```r
varPi = var(Pi)
varPi2 = var(Pi2)
varPi2 < varPi
```

```
##      [,1]
## [1,] TRUE
```

The variance decreased slightly, from 0.0014 to 0.0012.