Homework 10

Jonathan Hernandez

November 2, 2018

```
Smith is in jail and has 1 dollar; he can get out on bail if he has 8 dollars.
A guard agrees to make a series of bets with him. If Smith bets A dollars, he
wins A dollars with probability 0.4 and loses A dollars with probability 0.6.

Find the probability that he wins 8 dollars before losing all his money if

(a) he bets 1 dollar each time (timid strategy)
(b) he bets, each time, as much as possible but not more than necessary to
bring his fortune up to 8 dollars (bold strategy).
(c) Which strategy gives Smith the better chance of getting out of jail?
```

let $S = \{s_1, s_2, ..., s_8\}$ be the states where s_i = state of having i dollars.

let p = probability he wins A dollars and q = 1-p = probability he loses A dollars

for (a) with this given information we have the transition matrix

```r
# transition matrix
p <- 0.4
q <- 0.6
P <- matrix(c(1, 0, 0, 0, 0, 0, 0, 0, 0,
              q, 0, p, 0, 0, 0, 0, 0, 0,
              0, q, 0, p, 0, 0, 0, 0, 0,
              0, 0, q, 0, p, 0, 0, 0, 0,
              0, 0, 0, q, 0, p, 0, 0, 0,
              0, 0, 0, 0, q, 0, p, 0, 0,
              0, 0, 0, 0, 0, q, 0, p, 0,
              0, 0, 0, 0, 0, 0, q, 0, p,
              0, 0, 0, 0, 0, 0, 0, 0, 1), nrow = 9, ncol = 9, byrow = TRUE)
rownames(P) <- 0:8
colnames(P) <- 0:8
P
```

```
##     0   1   2   3   4   5   6   7   8
## 0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
## 1 0.6 0.0 0.4 0.0 0.0 0.0 0.0 0.0 0.0
## 2 0.0 0.6 0.0 0.4 0.0 0.0 0.0 0.0 0.0
## 3 0.0 0.0 0.6 0.0 0.4 0.0 0.0 0.0 0.0
## 4 0.0 0.0 0.0 0.6 0.0 0.4 0.0 0.0 0.0
## 5 0.0 0.0 0.0 0.0 0.6 0.0 0.4 0.0 0.0
## 6 0.0 0.0 0.0 0.0 0.0 0.6 0.0 0.4 0.0
## 7 0.0 0.0 0.0 0.0 0.0 0.0 0.6 0.0 0.4
## 8 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0
```

We can compute the probability of reaching the absorbing state that is at state s_8 by computing $B = NR$ where $N = (I - Q)^{-1}$ and I = 7x7 identity matrix and Q = square matrix of the transient states that is rows and columns 1:7 and

$R =$ square matrix of the column absorbing states and the row transient states

that is

```r
library(Matrix)
# matricies to build matrix B as the probability
Q <- P[c(2:8), c(2:8)]
I <- diag(nrow = nrow(Q))
Q
```

```
##     1   2   3   4   5   6   7
## 1 0.0 0.4 0.0 0.0 0.0 0.0 0.0
## 2 0.6 0.0 0.4 0.0 0.0 0.0 0.0
## 3 0.0 0.6 0.0 0.4 0.0 0.0 0.0
## 4 0.0 0.0 0.6 0.0 0.4 0.0 0.0
## 5 0.0 0.0 0.0 0.6 0.0 0.4 0.0
## 6 0.0 0.0 0.0 0.0 0.6 0.0 0.4
## 7 0.0 0.0 0.0 0.0 0.0 0.6 0.0
```

```r
N <- solve(I - Q)
N
```

```
##           1        2         3         4         5         6          7
## 1 1.6328311 1.054718 0.6693101 0.4123711 0.2410785 0.1268834 0.05075337
## 2 1.5820777 2.636796 1.6732752 1.0309278 0.6026963 0.3172086 0.12688343
## 3 1.5059477 2.509913 3.1792228 1.9587629 1.1451229 0.6026963 0.24107851
## 4 1.3917526 2.319588 2.9381443 3.3505155 1.9587629 1.0309278 0.41237113
## 5 1.2204600 2.034100 2.5765266 2.9381443 3.1792228 1.6732752 0.66931007
## 6 0.9635210 1.605868 2.0340999 2.3195876 2.5099128 2.6367962 1.05471848
## 7 0.5781126 0.963521 1.2204600 1.3917526 1.5059477 1.5820777 1.63283109
```

```r
R <- P[c(2:8), c(1, 9)]
R
```

```
##     0   8
## 1 0.6 0.0
## 2 0.0 0.0
## 3 0.0 0.0
## 4 0.0 0.0
## 5 0.0 0.0
## 6 0.0 0.0
## 7 0.0 0.4
```

```r
# compute B

B <- N %*% R
B
```

```
##           0          8
## 1 0.9796987 0.02030135
## 2 0.9492466 0.05075337
## 3 0.9035686 0.09643140
```

```
## 4 0.8350515 0.16494845
## 5 0.7322760 0.26772403
## 6 0.5781126 0.42188739
## 7 0.3468676 0.65313243
```

```r
round(B[1,2], 4)
```

```
## [1] 0.0203
```

By looking at the first row second column of B which represents the probability that at the start of this series of bets, he has only about 0.0203 or about 2.03% percent chance of getting out of jail.
for (b) the transition matrix will be different and smith will go for a all-or-nothing approach.
the transition matrix will look something like this and re-calculation of the matricies:

```r
P <- matrix(c(1, 0, 0, 0, 0, 0, 0, 0, 0,
              q, 0, p, 0, 0, 0, 0, 0, 0,
              q, 0, 0, 0, p, 0, 0, 0, 0,
              0, 0, 0, 0, 0, 0, 0, 0, 0,
              q, 0, 0, 0, 0, 0, 0, 0, p,
              0, 0, 0, 0, 0, 0, 0, 0, 0,
              0, 0, 0, 0, 0, 0, 0, 0, 0,
              0, 0, 0, 0, 0, 0, 0, 0, 0,
              0, 0, 0, 0, 0, 0, 0, 0, 1), nrow = 9, ncol = 9, byrow = TRUE)
rownames(P) <- 0:8
colnames(P) <- 0:8
P
```

```
##     0 1   2 3   4 5 6 7   8
## 0 1.0 0 0.0 0 0.0 0 0 0 0.0
## 1 0.6 0 0.4 0 0.0 0 0 0 0.0
## 2 0.6 0 0.0 0 0.4 0 0 0 0.0
## 3 0.0 0 0.0 0 0.0 0 0 0 0.0
## 4 0.6 0 0.0 0 0.0 0 0 0 0.4
## 5 0.0 0 0.0 0 0.0 0 0 0 0.0
## 6 0.0 0 0.0 0 0.0 0 0 0 0.0
## 7 0.0 0 0.0 0 0.0 0 0 0 0.0
## 8 0.0 0 0.0 0 0.0 0 0 0 1.0
```

```r
Q <- P[c(2:8), c(2:8)]
I <- diag(nrow = nrow(Q))
Q
```

```
##   1   2 3   4 5 6 7
## 1 0 0.4 0 0.0 0 0 0
## 2 0 0.0 0 0.4 0 0 0
## 3 0 0.0 0 0.0 0 0 0
## 4 0 0.0 0 0.0 0 0 0
## 5 0 0.0 0 0.0 0 0 0
## 6 0 0.0 0 0.0 0 0 0
## 7 0 0.0 0 0.0 0 0 0
```

```r
N <- solve(I - Q)
N
```

```
##   1   2 3    4 5 6 7
## 1 1 0.4 0 0.16 0 0 0
## 2 0 1.0 0 0.40 0 0 0
## 3 0 0.0 1 0.00 0 0 0
## 4 0 0.0 0 1.00 0 0 0
## 5 0 0.0 0 0.00 1 0 0
## 6 0 0.0 0 0.00 0 1 0
## 7 0 0.0 0 0.00 0 0 1
```

```r
R <- P[c(2:8), c(1, 9)]
R
```

```
##     0   8
## 1 0.6 0.0
## 2 0.6 0.0
## 3 0.0 0.0
## 4 0.6 0.4
## 5 0.0 0.0
## 6 0.0 0.0
## 7 0.0 0.0
```

```r
# compute B

B <- N %*% R
B
```

```
##       0     8
## 1 0.936 0.064
## 2 0.840 0.160
## 3 0.000 0.000
## 4 0.600 0.400
## 5 0.000 0.000
## 6 0.000 0.000
## 7 0.000 0.000
```

```r
round(B[1,2], 4)
```

```
## [1] 0.064
```

We see that using this all-or-nothing approach will give Smith about 6.4% of getting out of jail.
for (c), we see that going the bold route/all-or-nothing approach is more feasible than the timid approach and more likley for Smith to get out of jail.