Predicting Movement of Taxi

Shuvro Ahmed
dept. of CSE
BRAC University
Dhaka, Bangladesh
ID: 18201119
shuvro.ahmed@g.bracu.ac.bd

I. Introduction

The goal of our work is to be able to predict taxi routes in a certain area based upon probabilities of taxies remaining or moving to a certain block.

A. *Defining the Problem*

Data is available for movement of taxies in ”X city divided into ”East” ,”South”, ”West”. Movement of one taxi to another will depend upon it’s current position. For example, of all taxis in the east, 30% will remain there, 30% move south while 40% go to west. Similiarly, in south, 40% will remain there, 40% will move to East and 20% will go to west. And lastly, for west, 50% will remain in the west while 30% going to east and rest to the west.

II. Methodology

Here, for driver movement a 3 dimensional discrete markov chain model is selected with the states: east, south and west. From there, we get the following transition matrix:

```
DriverMovement
 A  3 - dimensional discrete Markov Chain defined by the following states:
 East, South, West
 The transition matrix  (by rows)  is defined as follows:
      East South West
East   0.3   0.3  0.4
South  0.4   0.4  0.2
West   0.5   0.3  0.2
```

In the code, MCZone is the class for markov chain class as seen below:

```
class(MCZone)

[1] "markovchain"
attr(,"package")
[1] "markovchain"
```

III. Analysis

First of all, we imported library(markovchain) to get the markov chain library. Next, we got our states by using DriverZone as the following:

```
library(markovchain)
DriverZone <- c("East","South","West")
DriverZone

[1] "East"  "South" "West"
```

Then, we defined define a transition matrix named ”ZoneTransition” with 3 rows and byrows= true. and using dimname = DriverZone for both rows and columns as the following:

```
ZoneTransition <- matrix(c(0.1,0.4,0.9,0.3,0.3,0.4,0.6,0.7,0.3),nrow = 3, byrow = TRUE, dimname = list(DriverZone,DriverZone))
ZoneTransition

      East South West
East   0.3   0.3  0.4
South  0.4   0.4  0.2
West   0.5   0.3  0.2
```

A. *Forecast the states*

Now, after creating markov chain class as already shown, we start forcasting future routes for taxis. First, we found what will happen after 2 trips which is given below:

```
#After 2 trips
MCZone^2

DriverMovement^2
 A  3 - dimensional discrete Markov Chain defined by the following states:
 East, South, West
 The transition matrix  (by rows)  is defined as follows:
      East South West
East  0.41  0.33 0.26
South 0.38  0.34 0.28
West  0.37  0.33 0.30
```

Next, we found out what will happen after 4 and 9 trips respectively as seen below:

```
 East, South, West
 The transition matrix  (by rows)  is defined as follows:
        East  South   West
East  0.3897 0.3333 0.2770
South 0.3886 0.3334 0.2780
West  0.3881 0.3333 0.2786
```

```
 East, South, West
 The transition matrix  (by rows)  is defined as follows:
        East  South   West
East  0.3897 0.3333 0.2770
South 0.3886 0.3334 0.2780
West  0.3881 0.3333 0.2786
```

Interestingly, we can see that after 9 trips, the values are close to steady states distribution.
Now, finally let's find out the steady states by using the function shown below:

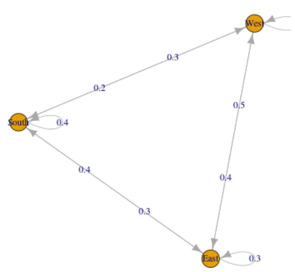
```
MCZone^9

DriverMovement^9
 A  3 - dimensional discrete Markov Chain defined by the following states:
 East, South, West
 The transition matrix  (by rows)  is defined as follows:
           East     South      West
East  0.3888886 0.3333333 0.2777780
South 0.3888890 0.3333333 0.2777777
West  0.3888891 0.3333333 0.2777775
```

As stated above, the steady states are very close to trip 9, so the steady state should be around that.

IV. Data Visualization

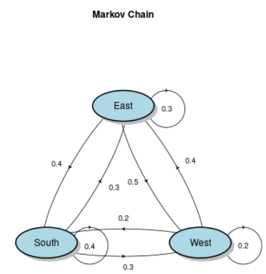
Since we have an absorbing Markov chain, we calculate the expected time until absorption. Below is the diagram where every entry will output the expected number if we start from any of the states.



Markov Chain
As stated above, we get an absorbing markov chain. For example, we can see that in south probability of taxi being in south is 0.4

‘

The Markov Chain depicted in the state diagram has 3 possible states: east, south, west. So, the transition matrix will be 3 x 3 matrix. Notice, the arrows exiting a state always sums up to exactly 1, similarly the entries in each row in the transition matrix must add up to exactly 1 - representing probability distribution. In the transition matrix, the cells do the same job that the arrows do in the state diagram.



State Diagram
Here, the state diagram is shown above.

V. Output Explanation

Here, we can see that after 9 trips the output almost matches with the steady state. Thus, we can say that the steady state will be around 9 trips.

```
MCZone^9

DriverMovement^9
 A  3 - dimensional discrete Markov Chain defined by the following states:
 East, South, West
 The transition matrix  (by rows)  is defined as follows:
           East     South      West
East  0.3888886 0.3333333 0.2777780
South 0.3888890 0.3333333 0.2777777
West  0.3888891 0.3333333 0.2777775

steadyStates(MCZone)

          East     South      West
[1,] 0.3888889 0.3333333 0.2777778
```