

PROJECT

OMIS 6350 V – Advanced Spreadsheet Modelling



December 4, 2019

Name: Khaliapin, mukherjee, cao, gudimella

SECTION: OMIS 6350 V

Table of Contents

[SECTION 1: PROJECT DESCRIPTION 2](#_Toc26345726)

[SECTION 2: SPREADSHEETS AND WORKSHEETS 2](#_Toc26345727)

[SECTION 3: USER INTERFACE 2](#_Toc26345728)

[3.1 Functionalities 5](#_Toc26345729)

[3.1.1. Enter other data 5](#_Toc26345730)

[3.1.2. Show Graph 5](#_Toc26345731)

[3.1.3. Calculate n-step probabilities 5](#_Toc26345732)

[3.1.4 Calculate steady-state probabilities 6](#_Toc26345733)

[3.1.5. Is State j Reachable from State i? 6](#_Toc26345734)

[3.1.6. Absorbing States: 6](#_Toc26345735)

[SECTION 4: PROCEDURE OUTLINES 7](#_Toc26345736)

[4.1. Calculating n-Step Probabilities 7](#_Toc26345737)

[4.2. Steady State Probabilities 8](#_Toc26345738)

[4.3. State j Reachable from State i 8](#_Toc26345739)

[4.4. Absorbing States 8](#_Toc26345740)

[SECTION 5: CONCLUSION 9](#_Toc26345741)

[5.1. Difficulties & Challenges 9](#_Toc26345742)

[5.2. What would you have done if you had more time? 10](#_Toc26345743)

# SECTION 1: PROJECT DESCRIPTION

The goal of this project was to simulate the Markov Chain algorithm. Markov Chain is oven used to statistically model random processes or stochastic processes. Some common examples of stochastic processes are stock market and exchange rate fluctuations, signals such as speech; audio and video; medical data such as a patient's EKG, EEG, blood pressure or temperature. Markov chain has been and continues to be widely used in various other areas such as education, marketing, health services, finances, accounting etc.

Markov chain consists of states and probabilities and is based on the concept of “memorylessness”, that is, the next state of any process only depends on the previous state and is completely independent of the sequence of states that precede. This simple assumption allows the use of conditional probability in calculations of the various phases and steps in the Markov chain process.

Markov chain consists of a set of transitions, which are determined by probability distributions that satisfy the Markov property. This project is aimed at exploring the different concepts of the Markov chain algorithm. Ideally, the Markov chain can be implemented for n states, where n can be any number between two and infinity. However, in this project, to maintain lower complexity while ensuring a holistic view of the algorithm, the number of states used to explain the various concepts was restricted to 5 states. This can be at any time be scaled up to as many states as required.

# SECTION 2: SPREADSHEETS AND WORKSHEETS

As part of the interface users need to enter the initial transition matrix. The transition matrix is a square matrix used to describe the transitions of a Markov chain. Each of its entries is a nonnegative real number representing a probability.

The entire project is implemented through macros in MS Excel. There are two main worksheets that are visible to the end user named Welcome and Model.

The Welcome sheet is designed in a way that serves as the first interaction point for the users and allows the user to enter the initial matrix.

The macro enabled file that has the implementation for the Markov Chains has two worksheets that are visible to the end user – **Welcome** and **Model**.

The Welcome sheet is displayed on load of the file and provides an option for the user to enter the Markov Chain State Transition Matrix. The captured details are displayed in the Model worksheet, which also has several command buttons that can be used to answer various questions related to the Markov Chain.

All the matrix calculations are done in worksheets that are hidden to the end users.

# SECTION 3: USER INTERFACE

This application contains several forms that allows the user to interact with the VBA application. The application opens with the following page displayed:

A screenshot of a social media post

Description automatically generated

*Fig 1: Welcome Page*

Upon clicking the Start button, the first form is displayed:

A screenshot of a cell phone

Description automatically generated

*Fig 2: Data Page*

This form is built to allow the user to input the initial transition matrix needed for the Markov Chain application. The user has two options:

1. A screenshot of a cell phone

   Description automatically generatedReading the data from the file. If this option is chosen, the user is asked to enter a file using the “Select a file to import” dialog box:

*Fig 3: File Explorer*

1. A picture containing screenshot

   Description automatically generatedEnter data manually. If this option is chosen, the user is asked to:
   1. Enter the number of states in the Markov process

*Fig 4: Manual Data Entry Form*

* 1. Depending on the number of states entered, the following form is displayed:

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

*Fig 5: Enter State Names Form*

In this form, the user is asked to enter the names of the states. This portion of the form is dynamic, in that the number of textboxes displayed depends on the number of states entered on the previous form.

* 1. A screenshot of a cell phone

     Description automatically generatedThe next step in the process is to prompt the user to enter the probabilities of moving from one state to another state in a Markov process. The values entered makeup the transition matrix. This form is also dynamic, in that based on the number of states entered in step a, the matrix shown in this form will vary from 2 x 2 matrix to a 5 x 5 matrix:

A screenshot of a cell phone

Description automatically generated

*Fig 6: Initial Transition Matrix Form*

With the probabilities in the transition matrix entered, either by reading a file directly, or by entering the values through the forms, the “Next🡪” button must be clicked. On clicking the “Next🡪” button the user is taken to the worksheet named “Model” that has several buttons corresponding to each functionality that this application is designed to serve:

A screenshot of a cell phone

Description automatically generated

*Fig 7: Model Sheet*

3.1 Functionalities:

A screenshot of a cell phone

Description automatically generated3.1.1. Enter other data:

This button will allow the user to enter a different transition matrix that pertains to a different Markov process, making this application re-usable. On clicking this button, the user is returned to the following form:

*Fig 8: Manual Data Entry Form*

A picture containing sky

Description automatically generated3.1.2. Show Graph:

On clicking this button, a graphical depiction of the Markov process is displayed to the user on the same worksheet, “Model”:

*Fig 9: Graph*

3.1.3. Calculate n-step probabilities: This button allows the user to see the probabilities of reaching one state from another after the Markov process has run for n-periods. On clicking this button, the following forms are displayed:

* 1. A screenshot of a cell phone

     Description automatically generated

*Fig 10: Calculate n-step transition matrix form*

* 1. A screenshot of a cell phone

     Description automatically generatedA screenshot of a cell phone

     Description automatically generated

A screenshot of a cell phone

Description automatically generated

* 1. A screenshot of a social media post

     Description automatically generatedA screenshot of a cell phone

     Description automatically generated

A screenshot of a cell phone

Description automatically generated

3.1.4 Calculate steady-state probabilities:

This functionality shows the probabilities of being in each state of the Markov process, when the process is stable.

A screenshot of a cell phone

Description automatically generated

*Fig 11: Steady State Probabilities*

3.1.5. Is State j Reachable from State i?

This functionality enables a user to see if a process can move from one state to another state, either directly or indirectly through other states. When this button is clicked, the user is prompted to enter the current state of the process. The user is also prompted to select the next state of the process.

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

*Fig 13: Message when direct route Fig 14: Message when indirect route*

*Fig 12: ReachableState Form*

* + 1. Absorbing States: This functionality allows a user to see the following:
  1. the states that are absorbing in nature, that is, its is possible for a process to enter these states, however, once entered, a process cannot leave these states
  2. number of times a process will enter each of the absorbing states given that the process starts in a transient (non-absorbing state) state
  3. number of periods or days or steps a process spends in a transient state before the process gets absorbed by one of the absorbing states
  4. probability of a process ending in an absorbing state, given that it started in a transient state

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedGiven a transition matrix that has absorbing states,when the button “Absorbing States” is clicked, all the absorbing states are displayed:

*Fig 15: Message when absorbing states found*

This is followed by a user form that allows the user to choose a property they wish to explore:

A screenshot of a cell phone

Description automatically generated A screenshot of a cell phone

Description automatically generated

*Fig 16: Absorbing State Option 1* *Fig 17: Absorbing State Option 1 Output*

A screenshot of a cell phone

Description automatically generated 

*Fig 18: Absorbing State Option 1* *Fig 19: Absorbing State Option 1 Output*

A screenshot of a cell phone

Description automatically generated 

*Fig 20: Absorbing State Option 1* *Fig 21: Absorbing State Option 1 Output*

If there are no absorbing states, an appropriate message is displayed.

# SECTION 4: PROCEDURE OUTLINES

Following procedures have been written to implement the mathematical properties of the Markov Chain. These procedures are linked to each of the functionalities presented in this application.

## 4.1. Calculating n-Step Probabilities

The procedure is designed to calculate the probabilities of a process transitioning from one state(state i) to another state(state j) in n-periods. This involves matrix multiplication based on the State Transition Matrix and the initial Probabilities vector. The worksheet function MMULT has been used for multiplying the matrices. This operation is repeated in a loop based on the value for the “Number of States” entered by the user.

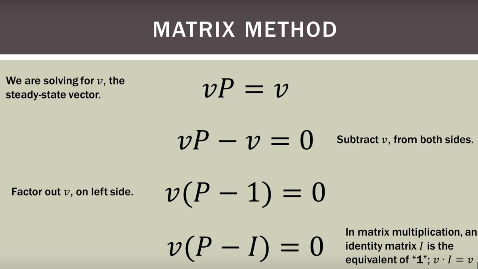
## 4.2. Steady State Probabilities[[1]](#footnote-1)

For Markov processes that are finite, irreducible, and aperiodic, there is a long-run equilibrium that is reached despite the starting state.

This equilibrium is called the steady-state distribution of the Markov.

Instead of raising PP to sufficiently high powers, this steady-state distribution can be found more easily in a few ways, all of which come down to solving μP=μ, such that ∑iμi=1.

The steady state probabilities are determined based on the solutions to matrix equations above and these are identified using the Solver feature of the VBA. Based on the user’s input, the algorithm checks the following:

1. First, transfer the transition matrix to a new worksheet called “SteadyState” worksheets. The “SteadyState” worksheet is marked as “VeryHidden”.
2. According to the matrix method listed below, compute (P-I) Matrix, which stands for the transition matrix minus the identity matrix.
3. Run solver. Within the solver setting, set the sum of all steady state probabilities equal to one as the objective. If the transition matrix is n x n, there will be n constraints in the solver, which represents the steady state matrix \* (P-I) Matrix equals zero. *Fig 18: Formula for steady state*
4. Run solver in simplex LP since this is a linear problem. And transfer the steady state probabilities to the model worksheet.

## 4.3. State j Reachable from State i

This functionality has been developed to allow a user to determine if a state can be reached from any other state. The user will be prompted to enter any two states (state i and state j) that is available on the transition matrix. Based on the user’s input, the algorithm checks the following:

1. If a direct route from State i to State j exists. If this exists, an appropriate message is displayed to user.
2. If a direct route from State i to State j does not exist, the algorithm will then check if an indirect route from state i to state j exists. To determine the indirect route, the concept of Markov Sequence is used. The concept of Markov Sequence relies on the conditional probability that some sequence leads from state i to state j.

To check if a direct route exists between states i and j, the algorithm checks if an element sij of the transition matrix is greater than 0. If it is greater than 0, a direct route exists, if not there are two possible options:

1. Indirect route exists
2. No routes at all

To check if an indirect route exists between states i and j, the algorithm performs a matrix multiplication of the elements in row i and of the elements in column j. The multiplication of 1 x n matrix and n x 1 matrix results in a 1 x 1 matrix. If the value of this 1 x matrix is greater than 0, then a process can reach state j from state i indirectly.

If the value of the resulting matrix multiplication is 0, it is indicative of the fact that no direct route exists.

## 4.4. Absorbing States

This functionality is provided to the user to determine if a state is an absorbing state. A state is called absorbing if it is possible to reach that state from any other state, but impossible to leave it. This translates to the following mathematically, the probability of a process entering and remaining in its current state is 1, and the probability of exiting that state to reach any other state is 0. With this defined, many questions arise, such as:

1. If the chain begins in a given transient state, and before we reach an absorbing state, what is the expected number of times that each state will be entered?
2. How many periods do we expect to spend in a given transient state before absorbing takes place?
3. If a chain begins in a given transient state, what is the probability that we end up in each absorbing state?

To model these, the canonical transformation of the transition matrix was used. In this technique, matrices are converted to standard forms or mathematical expressions. The below steps were followed:

Step 1: The states in the transition matrix were re-ordered in a way such that the transient states appeared first.

Step 2: The canonical form of the transition matrix was created such that if there are r absorbing states, and t transient states, the canonical form of the transition matrix will have the following form:

A close up of a clock

Description automatically generatedWhere Q : t x t matrix

R : non-zero t x r matrix

I : r x r matrix

0 : zero r x t matrix

Each element pij  of Matrix Pn gives the probability of a process reaching in state j after n steps, starting at state i. A matrix representation of the same is as shown on the below:

A picture containing object, sky

Description automatically generated

With the canonical forms of the transition matrix created, the following checks were done to determine answers to the above questions:

1. *QUESTION 1:* From the Pn matrix the following were computed
   1. matrix Q
   2. matrix (I – Q)
   3. matrix N by computing the inverse of matrix (I-Q)
   4. Each element of matrix N, nij gives the expected number of times a process will enter state j, given it started in state i
2. *QUESTION 2:* From the canonical form the following are computed:
   1. As part of the first question, matrix N was calculated, which is the inverse of matrix (I-Q)
   2. A column vector, c, is created, with all cij = 1
   3. Matrix t is calculated by multiplying matrices N and c
   4. Each element of t, ti , states the number of steps a process takes, starting from state i, till it gets absorbed
3. *QUESTION 3:* From the canonical matrix P the following were computed:
   1. Matrix R (right corner)
   2. As part of the first question, matrix N was calculated, which is the inverse of matrix (I-Q)
   3. Matrix B is calculated by multiplying matrices N and R
   4. Each element of matrix B, bij gives the probability that starting in a transient state i, the process will end in an absorbing state j.

# SECTION 5: CONCLUSION

In conclusion, this application is suitable for experimenting with the different concepts of Markov Chain. As mentioned at the beginning of this report, Markov Chain is a widely applicable concept, being applied to a plethora of industries. This application allows a user to interact and see the various concepts through real life examples easily, without needing to use complicated mathematical formulae to work them out manually. Below, some of the challenges and ways to improves this application are mentioned.

## 5.1. Difficulties & Challenges

* Defining the scope to ensure all concepts are covered in the most efficient and succinct manner
* The underlying mathematical concepts seem easy in theory, but reasonably difficult to implement in VBA, especially the functionalities related to Steady State Probabilities, State Reachability and Absorbing States.
* Understanding the implementation of Matrix operations in VBA.
* Integrating all the functionalities of Markov Chain in one application.

## 5.2. What would you have done if you had more time?

* Generalized the solution to consider any number of states in the Markov Chain. This application is restricted to the use of a maximum of 5 states.
* Implemented a more dynamic user interface
* Integrated the solution with MS Access database and store all the Transition Matrix values entered by the user to the database. An option can then be provided to the user to select one of the previously entered matrices and run the simulation.
* Compare the results based on multiple transition matrices when the same states are involved and identify the best choice depending on the business need.
* Provided a feature to email the reports and results to the users.

1. <https://www.youtube.com/watch?v=cP3c2PJ4UHg>

   <https://www.stat.auckland.ac.nz/~fewster/325/notes/ch9.pdf>

   <https://nicolewhite.github.io/2014/06/10/steady-state-transition-matrix.html> [↑](#footnote-ref-1)