# RO54 - TD3 - Markov Models

> This report is made by **RASSIÉ Nathan** and **DERAISIN Nicolas**.
>
> It has been written in **Markdown**.
>
> Class Diagram has been created with **Mermaid**.
>
> Implementation has been done in **HTML**, **PHP**, **CSS** and **JavaScript**.

# Report on the representation of the Markov model

## Introduction

The Markov model is a statistical tool used to predict the probability of a future event based on the present state and the transition probabilities between states. It is used in various fields such as finance, biology, and computer science. In this report, we will explain how we represented the Markov model to predict the user behavior on a website.

## Explanation of the Markov model

The Markov model is based on the assumption that the probability of a future event depends only on the current state and not on the previous states. Thus, to predict the next action of a user on a website, we must consider the current state, which is the page the user is on, and the transition probabilities between this page and other possible pages.

In our case, we represented these transition probabilities in the form of a matrix, where each row represents the current state and each column represents the possible next state. The values of the matrix indicate the transition probabilities between each state.

## Explanation of the functioning via a matrix

The transition matrix is updated every time the user changes the page. When the user arrives on a new page, we increment the corresponding counter in the transition matrix. Thus, the more the user visits pages, the more accurate our model becomes.

## Global functioning of the code

Our website is composed of two pages: a main page displaying the possible pages and the transition probabilities, and a second page containing a PHP script for the creation and incrementation of the transition matrix.

The main page displays the transition matrix in the form of a table and a graph estimating the probability of the user's origin and destination. Each page is represented by a button redirecting to the corresponding page and allowing to update the transition matrix.

The PHP script allows creating the transition matrix if it does not exist yet and incrementing it at each page change. We use variables such as $prevPage$,currentPage, and nextPage to determine the current state and the transition probabilities to the next states. All this data are return to the index page to be display on the main page.

```
+-----------------+     +-----------------+     +------------------------+
|     index.php    |     |    script.php   |     |       Algorythem       |  |
+-----------------+     +-----------------+     +------------------------+
         |                        |                           |
         |     user navigates     |                           |
         |     to new page        |                           |
         |----------------------> |                           |
         |    (click on button)   |  reads current state      |
         |      (send a form)     |------------------------>  |
         |                        |  reads transition matrix  |
         |                        |------------------------>  |
         |                        |   updates matrix          |
         |                        |<------------------------  |
         |                        |                           |
         |                        |  calculates next state    |
         |                        |------------------------>  |
         |                        |  returns next state       |
         |                        |<------------------------  |
         | updates transition     |                           |
         | send the matrix        |                           |
         |<---------------------- |                           |
         |                        |                           |
         | updates graph          |                           |
         | and displays it        |                           |
         |<---------------------- |                           |
```

## Example

You can test the code on the following URL :

- nicolasderaisin.yj.fr/pages/HMM/index.php

### Markov Matrice reading :

The color represents the page you are on, as does the annotation at the top right of the site.

Capture d’écran 2023-06-01 162040.png

Capture d’écran 2023-06-01 141105.png

In this array, there is 3 parts per case :

- First value is the numbre of visite from the line title(Page x on the left) to the column title (Page x on the top).
- Seconde value is the probability that the user go from the line title to the column title. This value is calculated according to the sum of the values of the second values in the line. You can see the reading direction in the second screenshot below.
- Third value is the probability that the user go from the column title to the line title. This value is calculated according to the sum of the values of the third values in the columne. You can see the reading direction in the third screenshot below.

markov_prediction_cases.png

markov_prediction_future.png

markov_prediction_passe.png

markov_prediction_stat.png

The graph with the bars represents the statistics in percentage of the prediction of the future and previous pages on which we are located.

bar_graph.png

bar_graph_markov_matrix_2.png

---

## Conclusion

In conclusion, we have presented how we represented the Markov model to predict the user behavior on a website. We have explained how we used a transition matrix to represent the transition probabilities between the different possible pages and how we updated this matrix at each page change.