Topic 5: Markovian (and Deterministic) Dynamic System

Group Members: Xiaomeng Cai, Shibo Ma, Gaoyu Lu, Ruiling Chen, Yang Len

The link of video: https://youtu.be/9WpuBwX-Q8Q (https://youtu.be/9WpuBwX-Q8Q)

Introduction

Definition

Markov chain is the simplest Markov model. It is a time series analysis method, it predicts the state of an event and development trend by using state-transition matrices.

Let us take the traffic light system as an example. Considered such a traffic light in practice, there are red, green and yellow. Three different colours run in a designed subsequence, red firstly, then green and yellow lastly. Then, this sequence can be seen as a state machine, each time of a colour appearing only depends on the last colour. Described in mathematics notation, x_1 = red, x_2 = green, x_3 = yellow, x_4 = red again, the state x_i is a deterministic pattern, with each state only determined by the last state.

Thus, we can deduce a linear dynamical system that is a simple model for a sequence $x_1, x_2, \ldots$ in which each x_{t+1} is a linear function of x_t:

$$x_{t+1} = A_t x_t, \quad t = 1, 2, \ldots$$

The $n \times n$ matrices A_t are called the dynamics matrices. The equation above is called the dynamics or update equation.

This linear dynamical system is sometimes called a Markov model (after the mathematician Andrey Markov). Markov studied systems assumed that future states depend only on the current state, i.e. x_{t+1} depends only on x_t, and not additionally on $x_{t-1}, x_{t-2}, \ldots$

Transition matrix

In the example above there are three states for the system. Define X_i to be the probability of the system to be in state t after it was in state $t-1$ (at any observation). The matrix $A = X$ is called the Transition matrix of the Markov Chain.

Let us consider the transition matrix of traffic lights. Presume the sequence of the traffic lights starting from green, then yellow, and red. Here we can have a simple transition state table to show three different situations as below, in which each row represents the initial state of traffic lights and each column is the state after transiting.

	Green	Yellow	Red
Green	0	0	1
Yellow	1	0	0
Red	0	1	0

It can also be shown as a 3×3 marix $p = \left(\begin{smallmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{smallmatrix}\right)$. Take the 1 in the first row as an example. It means the green light will be on after the red is off.

Here we assume y as the state of each traffic light. If the red light is on, then we have
$$y_t = \left[\begin{array}{c|c} Green & 0 \\ Yellow & 0 \\ Red & 1 \end{array}\right].$$

Therefore, the sequence for which light will be on can be expressed as below,
$$y_{t+1} = p \cdot y_t = \left[\begin{array}{c|c} Green & 1 \\ Yellow & 0 \\ Red & 0 \end{array}\right].$$

All the states thus can have the expression $y_t = p^t \cdot y_0$.

Property

Indenpency

A state S_t is Markov if and only if
$$P[S_{t+1}|S_t] = P[S_{t+1}|S_1, \cdots, S_t],$$
which means the memoryless attributes of the stochastic process. This process will own the attribute of Markov, if the conditional probability distrbution of this process's future state only depends on the current state, rather than the former sequence. Thus, these two events are independent from each other.

State Transition Matrix

For a Markov state s and successor state s', the state transition probability is defined by
$$P_{ss'} = P[S_{t+1} = s'|S_t = s].$$
State transition matrxi P defines transition probabilities from all states s to all successor states s',
$$P = \begin{bmatrix} P_{11} & \cdots & P_{1n} \\ \vdots & \ddots & \vdots \\ P_{n1} & \cdots & P_{nn} \end{bmatrix},$$
where each row of the matrix sums to 1.

Application

Take the weather obeservation data of Shanghai as an example, during which the Markov model will be deployed to forecast local weather.

Firstly, we will introduce the data that has been preprocessed throughout Python, in which the state 1 represents the sunny day, 2 is the cloudy day and 3 means rainy days. In the meantime, we choose the last day as the test set to verify the model.

In [8]:

```
import numpy as np
import pandas as pd
sh_weather = pd.read_csv("weather.csv")
print(sh_weather)
```

```
           ymd weather  degree  Unnamed: 3  Unnamed: 4 Unnamed: 5
0   2017/10/16    rain       3         NaN         NaN    sunny 1
1   2017/10/17  cloudy       2         NaN         NaN   cloudy 2
2   2017/10/18  cloudy       2         NaN         NaN     rain 3
3   2017/10/19    rain       3         NaN         NaN        NaN
4   2017/10/20  cloudy       2         NaN         NaN        NaN
5   2017/10/21   sunny       1         NaN         NaN        NaN
6   2017/10/22   sunny       1         NaN         NaN        NaN
7   2017/10/23  cloudy       2         NaN         NaN        NaN
8   2017/10/24   sunny       1         NaN         NaN        NaN
9   2017/10/25   sunny       1         NaN         NaN        NaN
10  2017/10/26  cloudy       2         NaN         NaN        NaN
11  2017/10/27   sunny       1         NaN         NaN        NaN
12  2017/10/28  cloudy       2         NaN         NaN        NaN
13  2017/10/29   sunny       1         NaN         NaN        NaN
14  2017/10/30  cloudy       2         NaN         NaN        NaN
15  2017/10/31  cloudy       2         NaN         NaN        NaN
16   2017/11/1   sunny       1         NaN         NaN        NaN
17   2017/11/2  cloudy       2         NaN         NaN        NaN
18   2017/11/3  cloudy       2         NaN         NaN        NaN
19   2017/11/4  cloudy       2         NaN         NaN        NaN
20   2017/11/5  cloudy       2         NaN         NaN        NaN
21   2017/11/6  cloudy       2         NaN         NaN        NaN
22   2017/11/7  cloudy       2         NaN         NaN        NaN
23   2017/11/8   sunny       1         NaN         NaN        NaN
24   2017/11/9  cloudy       2         NaN         NaN        NaN
25  2017/11/10  cloudy       2         NaN         NaN        NaN
26  2017/11/11  cloudy       2         NaN         NaN        NaN
27  2017/11/12  cloudy       2         NaN         NaN        NaN
28  2017/11/13    rain       3         NaN         NaN        NaN
29  2017/11/14    rain       3         NaN         NaN        NaN
..         ...     ...     ...         ...         ...        ...
47   2017/12/2  cloudy       2         NaN         NaN        NaN
48   2017/12/3  cloudy       2         NaN         NaN        NaN
49   2017/12/4  cloudy       2         NaN         NaN        NaN
50   2017/12/5   sunny       1         NaN         NaN        NaN
51   2017/12/6   sunny       1         NaN         NaN        NaN
52   2017/12/7  cloudy       2         NaN         NaN        NaN
53   2017/12/8  cloudy       2         NaN         NaN        NaN
54   2017/12/9  cloudy       2         NaN         NaN        NaN
55  2017/12/10   sunny       1         NaN         NaN        NaN
56  2017/12/11   sunny       1         NaN         NaN        NaN
57  2017/12/12  cloudy       2         NaN         NaN        NaN
58  2017/12/13  cloudy       2         NaN         NaN        NaN
59  2017/12/14  cloudy       2         NaN         NaN        NaN
60  2017/12/15    rain       3         NaN         NaN        NaN
61  2017/12/16   sunny       1         NaN         NaN        NaN
62  2017/12/17   sunny       1         NaN         NaN        NaN
63  2017/12/18   sunny       1         NaN         NaN        NaN
64  2017/12/19   sunny       1         NaN         NaN        NaN
65  2017/12/20   sunny       1         NaN         NaN        NaN
66  2017/12/21   sunny       1         NaN         NaN        NaN
67  2017/12/22   sunny       1         NaN         NaN        NaN
68  2017/12/23  cloudy       2         NaN         NaN        NaN
69  2017/12/24   sunny       1         NaN         NaN        NaN
70  2017/12/25  cloudy       2         NaN         NaN        NaN
71  2017/12/26   sunny       1         NaN         NaN        NaN
72  2017/12/27  cloudy       2         NaN         NaN        NaN
73  2017/12/28    rain       3         NaN         NaN        NaN
74  2017/12/29  cloudy       2         NaN         NaN        NaN
75  2017/12/30  cloudy       2         NaN         NaN        NaN
```

```
76  2017/12/31   sunny        1          NaN          NaN         NaN

[77 rows x 6 columns]
```

In [9]:

```
# There are totally 77 rows of data.
# Here, 76 of 77 rows are picked to train the model, with the last one remaining
as the test set.
state = sh_weather.loc[0:75,'degree']
state
```

```
Out[9]:

0     3
1     2
2     2
3     3
4     2
5     1
6     1
7     2
8     1
9     1
10    2
11    1
12    2
13    1
14    2
15    2
16    1
17    2
18    2
19    2
20    2
21    2
22    2
23    1
24    2
25    2
26    2
27    2
28    3
29    3
     ..
46    2
47    2
48    2
49    2
50    1
51    1
52    2
53    2
54    2
55    1
56    1
57    2
58    2
59    2
60    3
61    1
62    1
63    1
64    1
65    1
66    1
67    1
68    2
69    1
70    2
71    1
72    2
73    3
```

```
74    2
75    2
Name: degree, Length: 76, dtype: int64
```

In [10]:

```
# Train and extract each data.
State=[]
for i in range(76):
    State.append(state[i])
State
```

Out[10]:

```
[3,
 2,
 2,
 3,
 2,
 1,
 1,
 2,
 1,
 1,
 2,
 1,
 2,
 1,
 2,
 2,
 1,
 2,
 2,
 2,
 2,
 2,
 2,
 1,
 2,
 2,
 2,
 2,
 3,
 3,
 2,
 3,
 2,
 2,
 3,
 2,
 3,
 2,
 2,
 2,
 2,
 1,
 1,
 3,
 3,
 2,
 2,
 2,
 2,
 2,
 1,
 1,
 2,
 2,
 2,
 1,
 1,
 2,
 2,
 2,
 3,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 2,
 1,
 2,
 1,
 2,
 3,
 2,
 2]
```

In [12]:

```
R=[[0,0,0],[0,0,0],[0,0,0]]
```

In [8]:

```
# Here we define two functions to calculate state transition matrix.
def Pijm(i,j,m):
    a=0
    for n in range(len(State)):
         if State[n]==i:
            try:
                if State[n+m]==j:
                    a=a+1
            except:
                pass
    return  a/(State[0:len(State)-m].count(i))

def Rm(m):
    for i in range(1,4):
        for j in range(1,4):
            R[i-1][j-1]=Pijm(i,j,m)
    return R
```

Hypothesis test of independency

The Markov chain refers that the probability distribution of the next state can only be determined by the current state, and the events before it in the time series are irrelevant. This means changes of each state are independent from each other. Thereofre, hypothesis test of independence is necessary. Then, we have

H_0: The conditions of daily weather are independent.

The frequency of daily weather conditions

$$\begin{matrix} 1 & 11 & 1 \\ 11 & 23 & 7 \\ 1 & 7 & 2 \end{matrix}$$

After the calculation, we have $\chi^2_4 = 6.495, p - value = 0.165.$ Hence, this is not evidence against H_0. We can get state transition matrices as below.

These Markov matrices, with $m = 1$ and $m = 2$, are the first and second order Markov tranision probability matrices, and they respectively predict tomorrow and the day after tomorrow's weather conditions, in which each rows from the top to the bottom mean sunny days, cloudy days and rainy days, with each columns representing the probability of next states of sunny days, cloudy days and rainy days.

In [10]:

```
m=1
R1=np.matrix(Rm(m))
R1
```

Out[10]:

```
matrix([[0.47826087, 0.47826087, 0.04347826],
        [0.26829268, 0.56097561, 0.17073171],
        [0.09090909, 0.72727273, 0.18181818]])
```

In [11]:

```
m=2
R2=np.matrix(Rm(m))
R2
```

Out[11]:

```
matrix([[0.43478261, 0.43478261, 0.13043478],
        [0.275     , 0.6       , 0.125     ],
        [0.18181818, 0.63636364, 0.18181818]])
```