A Case Study with implementation of Hidden Markov Model (HMM)

Nithin Sai Jalukuru
School of Engineering and Applied Sciences
University at Buffalo, Buffalo, NY, USA
njalukur@buffalo.edu

Abstract : In this present work, A case study of burglar alarm has taken, in which the task is to find the most probable location of the burglar. This is a well-known instance of an HMM implementation. In many branches of science, HMM is a widely used technique. It is a temporal probabilistic model in which a single discrete random variable describes the state of the process. In the latter half of the 1960s, the HMM theory was developed. It is now particularly well-known for its use in bioinformatics, speech, handwriting, and temporal pattern recognition.

Keywords : Hidden Markov model, Case study, Markov chains,

1.INTRODUCTION

Hidden Markov Model (HMM) is a statistical model named after Russian mathematician Andrey Markov. It is a large and useful class of stochastic processes. It is characterized by *Markov Property* which means that *future state of the process depends only upon the present state*, not on the sequence of events that preceded it. Markov Models are very rich in mathematical structure and when applied properly, work very well in practice for several applications.

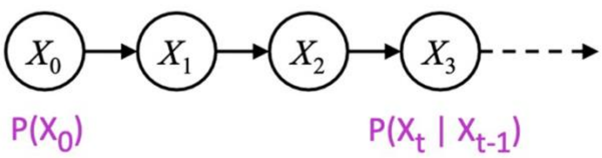

Fig 1: Markov Model Structure

Observations generated by real-time processes can be discrete, continuous, stationary, time-variant, or noisy. The main difficulty is defining the observations as a parametric random process, whose parameters should be estimated using a well-defined methodology. As a result, we are able to create a theoretical model of the underlying process that lets us forecast the process output and distinguish between the statistical characteristics of the observation itself. The *Hidden Markov Model* is one such statistical model.

1.1 HIDDEN MARKOV MODEL

Markov model *may be too restrictive to be of practical use* in realistic problems in which states cannot directly correspond to a physical event. You expand the model to one in which the observed output is a probabilistic function of a state to increase its flexibility. According to a distinct probability distribution, each state has the potential to produce a number of distinct outputs, and each output can potentially be produced at any state. The doubly embedded stochastic model known as the HMM is the final model.

The (non-observable) process is deciphered by HMM by examining the pattern of a series of observed symbols. In an HMM, the underlying (or hidden) stochastic process can be inferred indirectly by looking at the sequence of observed symbols from another set of stochastic processes.

HMM comprises (hidden) states that represent an unobservable, or latent, attribute of the process being modeled.

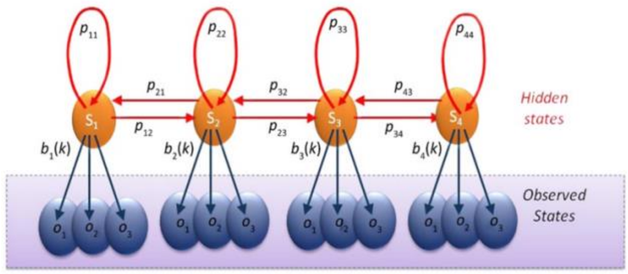

Fig 2 : Hidden Markov Model Representation with 4 hidden states and 3 observed states

The Fig 2 represents the underlying Markov chain.
$P_{ij} = [Transistion\ Model]$, transition probability from Hidden state i to Hidden state j.
$b_j(k) = [Observation\ Model]$, Observation probability distribution for State j and $k = \{O_1, O_2, \ldots . O_n\}$

1.2 COMPONENTS HIDDEN MARKOV MODEL

I. *Number of Hidden states* : (N) in the model. The individual states are represented as $S = \{S_1, S_2, \ldots \ldots . . S_N\}$, and the state at the time t is represented as q_t

II. *State transition probability distribution* :P = $\{P_{ij}\}$, to represent the state transition from state i to state j

III. *Observation symbol probability distribution:* for state j

$$B = b_j(k) = P(x_t = o_k \mid q_t = S_j)$$

IV. *Initial State Distribution :* $(\pi = \{\pi_i\})$, Where $\pi_i = P(q_i = S_j)$, and $1 \le i \le N$

Formally, an HMM can be defined by specifying model parameters N and M, observation symbols O, and three probability matrices P, B, and π. For simplicity, you can use the compact form, $\boldsymbol{\lambda}$ **= (** $\boldsymbol{\pi}$**, P, B),** to indicate the complete parameter set of the model.

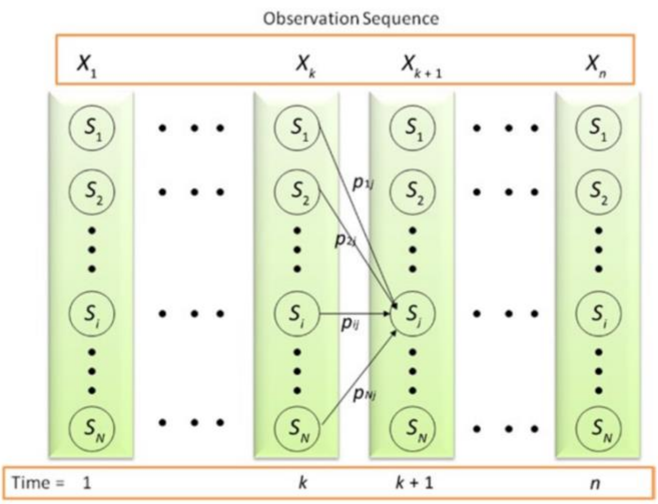

Fig 3 : Hidden Markov Model : Trellis Representation

HMM described here makes *two Assumptions*

I. *Markov Assumption :* "The current state is dependent only on the previous state; this represents the memory of the model".

II. *Independence Assumption :* "Output observation O_t at the time t is dependent only on the current state ; it is independent of previous observation and states".

2.EXPERIMENTATION

Problem Statement : Consider a house with *4 rooms* (0,1,2,3) and sure burglar entered through room 0. Every room has a motion detector, and those in rooms 0, 0, 2, and 3 have activated in the previous five hours. The probability matrix P for visible features given the hidden ones and transition matrix T. To find most probable location of the burglar now ?.

In this Probability matrix P , the visible feature is the alarm going off (column) and hidden variable is burglar location (Rows)

Hence ,

$$P = \begin{pmatrix} 0.7 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.7 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.7 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.7 \end{pmatrix}$$

And in Transition matrix T, column is next state of hidden variable (burglar location) and row is current state of hidden variable.

$$T = \begin{pmatrix} 0.15 & 0.3 & 0.3 & 0.25 \\ 0.35 & 0.05 & 0.3 & 0.3 \\ 0.45 & 0.35 & 0.05 & 0.15 \\ 0.45 & 0.25 & 0.05 & 0.25 \end{pmatrix}$$

As alarms have gone off in the last *5 hours* in the rooms 0,0,2,2,3, hence

$$observations = [0,0,2,2,3]$$

Let's start at hour 0, At 0th hour we know that the burglar has entered room 0. So, hidden probability that the burglar at 1st hour is [1,0,0,0]

Coming to 1st hour, probability that the burglar is in room 0 is calculated by probability that the burglar is in room 0, given the observations at hour 1 plus the next states of the hidden variables for room 0.Similary calculate the hidden probabilities for the rooms at hour 1

```
# HOUR 1
a_hidden_prob_1 = np.zeros(4)

a_hidden_prob_1[0] = P[0,Obs[1]] * (T[0,0]*hidden_prob_0[0] + T[1,0]*hidden_prob_0[1] + T[2,0]*hidden_prob_0[2] + T[3,0]*hidden_prob_0[3])
a_hidden_prob_1[1] = P[1,Obs[1]] * (T[0,1]*hidden_prob_0[0] + T[1,1]*hidden_prob_0[1] + T[2,1]*hidden_prob_0[2] + T[3,1]*hidden_prob_0[3])
a_hidden_prob_1[2] = P[2,Obs[1]] * (T[0,2]*hidden_prob_0[0] + T[1,2]*hidden_prob_0[1] + T[2,2]*hidden_prob_0[2] + T[3,2]*hidden_prob_0[3])
a_hidden_prob_1[3] = P[3,Obs[1]] * (T[0,3]*hidden_prob_0[0] + T[1,3]*hidden_prob_0[1] + T[2,3]*hidden_prob_0[2] + T[3,3]*hidden_prob_0[3])
print(a_hidden_prob_1)

hidden_prob_1 = a_hidden_prob_1/np.sum(a_hidden_prob_1)
print(hidden_prob_1)

[0.105 0.03  0.03  0.025]
[0.55263158 0.15789474 0.15789474 0.13157895]
```

Fig 4 : For hour 1, Hidden probabilities

From the results obtained after calculating all the probabilities for each room, It can be seen that there is 55% chance that the burglar is in room 0 at hour 1.

In the similar way, we calculate the probabilities for each hour for 4 rooms (0,1,2,3).

For hour 2,

```
# HOUR 2
a_hidden_prob_2 = np.zeros(4)

a_hidden_prob_2[0] = P[0,Obs[2]] * (T[0,0]*hidden_prob_1[0] + T[1,0]*hidden_prob_1[1] + T[2,0]*hidden_prob_1[2] + T[3,0]*hidden_prob_1[3])
a_hidden_prob_2[1] = P[1,Obs[2]] * (T[0,1]*hidden_prob_1[0] + T[1,1]*hidden_prob_1[1] + T[2,1]*hidden_prob_1[2] + T[3,1]*hidden_prob_1[3])
a_hidden_prob_2[2] = P[2,Obs[2]] * (T[0,2]*hidden_prob_1[0] + T[1,2]*hidden_prob_1[1] + T[2,2]*hidden_prob_1[2] + T[3,2]*hidden_prob_1[3])
a_hidden_prob_2[3] = P[3,Obs[2]] * (T[0,3]*hidden_prob_1[0] + T[1,3]*hidden_prob_1[1] + T[2,3]*hidden_prob_1[2] + T[3,3]*hidden_prob_1[3])
print(a_hidden_prob_2)

hidden_prob_2 = a_hidden_prob_2/np.sum(a_hidden_prob_2)
print(hidden_prob_2)

[0.02684211 0.02618421 0.15934211 0.02421053]
[0.1134594  0.11067853 0.67352614 0.10233593]
```

Fig 5 : For hour 2 Hidden probabilities

There is 67% chance that the burglar is in room 2 , for the 2nd hour.

For hour 3,

```
# HOUR 3
a_hidden_prob_3 = np.zeros(4)

a_hidden_prob_3[0] = P[0,Obs[3]] * (T[0,0]*hidden_prob_2[0] + T[1,0]*hidden_prob_2[1] + T[2,0]*hidden_prob_2[2] + T[3,0]*hidden_prob_2[3])
a_hidden_prob_3[1] = P[1,Obs[3]] * (T[0,1]*hidden_prob_2[0] + T[1,1]*hidden_prob_2[1] + T[2,1]*hidden_prob_2[2] + T[3,1]*hidden_prob_2[3])
a_hidden_prob_3[2] = P[2,Obs[3]] * (T[0,2]*hidden_prob_2[0] + T[1,2]*hidden_prob_2[1] + T[2,2]*hidden_prob_2[2] + T[3,2]*hidden_prob_2[3])
a_hidden_prob_3[3] = P[3,Obs[3]] * (T[0,3]*hidden_prob_2[0] + T[1,3]*hidden_prob_2[1] + T[2,3]*hidden_prob_2[2] + T[3,3]*hidden_prob_2[3])
print(a_hidden_prob_3)

hidden_prob_3 = a_hidden_prob_3/np.sum(a_hidden_prob_3)
print(hidden_prob_3)

[0.04048943 0.03008899 0.07422414 0.01881813]
[0.24745913 0.18389476 0.45363541 0.11501071]
```

Fig 6 : For hour 3 Hidden probabilities

There is 45% chance that the burglar is in room 2 , for the 3rd hour.

For hour 4,

```
# HOUR 4
a_hidden_prob_4 = np.zeros(4)

a_hidden_prob_4[0] = P[0,Obs[4]] * (T[0,0]*hidden_prob_3[0] + T[1,0]*hidden_prob_3[1] + T[2,0]*hidden_prob_3[2] + T[3,0]*hidden_prob_3[3])
a_hidden_prob_4[1] = P[1,Obs[4]] * (T[0,1]*hidden_prob_3[0] + T[1,1]*hidden_prob_3[1] + T[2,1]*hidden_prob_3[2] + T[3,1]*hidden_prob_3[3])
a_hidden_prob_4[2] = P[2,Obs[4]] * (T[0,2]*hidden_prob_3[0] + T[1,2]*hidden_prob_3[1] + T[2,2]*hidden_prob_3[2] + T[3,2]*hidden_prob_3[3])
a_hidden_prob_4[3] = P[3,Obs[4]] * (T[0,3]*hidden_prob_3[0] + T[1,3]*hidden_prob_3[1] + T[2,3]*hidden_prob_3[2] + T[3,3]*hidden_prob_3[3])
print(a_hidden_prob_4)

hidden_prob_4 = a_hidden_prob_4/np.sum(a_hidden_prob_4)
print(hidden_prob_4)

[0.03573728 0.02709575 0.01578385 0.14968184]
[0.15653736 0.11868553 0.06913682 0.65564029]
```

Fig 7 : For hour 4 Hidden probabilities

There is 65% chance that the burglar is in room 3 , for the 4th hour.

Let's see the path of most probable states at each hour.

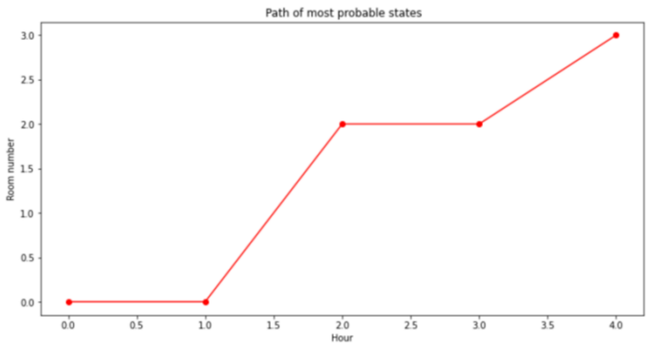

Fig 8 : Path of most probable states

From figure 8, it is evident that 0 hour → burglar is in room 0, Hour 1 → room 0, Hour 2 → room 2, Hour 3 → room 2 ,and hour 4 → room 3.

This is only one path we got [0,0,2,2,3], but there are many paths . Let's generate those all paths and compute probability for each path.

Path	Probability
[0, 0, 2, 0, 3]	0.10341600757557712
[0, 0, 2, 1, 3]	0.09652160707053864
[0, 0, 2, 2, 3]	0.04826080353526931
[0, 0, 1, 2, 3]	0.04136640303023084
[0, 0, 2, 3, 3]	0.03447200252519237
[0, 1, 2, 0, 3]	0.02954743073587917
[0, 1, 2, 1, 3]	0.02757760202015389
[0, 0, 0, 2, 3]	0.02068320151511542
[0, 0, 2, 2, 0]	0.02068320151511542
[0, 0, 2, 0, 1]	0.017728458441527508
[0, 0, 2, 0, 2]	0.017728458441527508
[0, 0, 1, 2, 0]	0.017728458441527505
[0, 2, 0, 2, 3]	0.017728458441527505
[0, 0, 2, 2, 1]	0.01608693451175644
[0, 0, 2, 1, 0]	0.01608693451175644
[0, 3, 0, 2, 3]	0.01477371536793959
[0, 0, 2, 1, 2]	0.013788801010076951

Fig 9 : Probabilities of each path

From the Fig 9, the path [0,0,2,0,3] has the highest chance ~ 10% the burglar will be in those rooms for 5 hours.

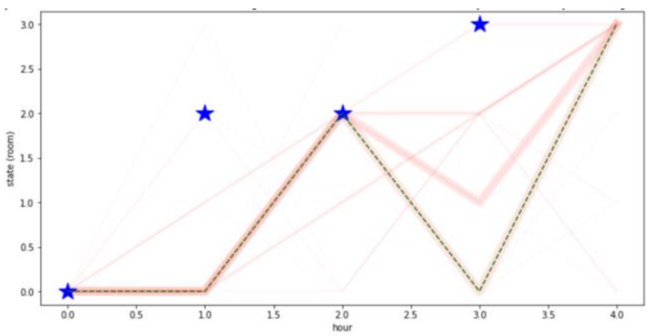

Fig 10: All possible paths

Paths shown as blue lines with probability indicated by transparency and thickness.

3.CONCLUSION

This is one the problem that utilizes the hidden Markov model property and the independent assumption. There many case real time case studies where Markov models are really helpful.

4.REFERENCES

1) CSE 474/574 Lecture slides, Prof Chen
2) Mariette Awad at. El " Hidden Markov Model"
3) Marcin PIETRZYKOWSKI and Wojciech SAŁABUN , "Applications of Hidden Markov Model: state-of-the-art"
4) Hidden Markov model, Kaggle.