Markov Models and Hidden Markov Models

Akshan Sameullah

April 2019

1 Introduction

Imagine that you are locked in a room with no way to observe anything outside the room. Person A comes into the room every few hours and you can see what they are wearing. You are then asked by someone (that presumably didn't know about your lonely whereabouts) what the weather will be. Although you haven't seen anything outside, you can still predict the weather outside your room based on the clothing Person A was wearing when they entered your room. This is similar to how a **Hidden Markov Model (HMM)** works.

A similar scenario involves a person asking you about the future weather but this time you are not locked in a room and have observed the weather in your environment for large amount of time. Now you can use your observations to find patterns and predict what the weather will be in the future using probabilities in sequential data. This is similar to how a **Markov Model** works.

2 Concept

A **Markov Model** models the dependencies of current information with previously gained information. They are composed of states, transitions between states and discrete or continuous output. These models can be used to learn statistics of sequential data, recognize patterns, and make predictions and estimations.

A **Hidden Markov Model (HMM)** is a system in which a variable switches states and generates one of many possible outputs. The outputs are observable but the internals are "hidden". An example could be to predict the weather based on clothes people wear. The weather is hidden and the observable outputs are the clothes people are wearing.

Possible uses of HMMs include to:

Infer Given a sequence of outputs, compute the probabilities of switch sequences. Match Patterns Find the state switch sequence most likely to generate an output Train Compute the probabilities (sequence) based on the data

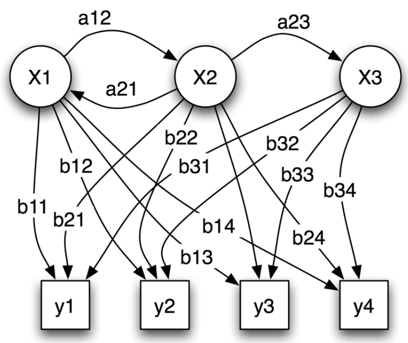

Figure 1: Hidden Markov Model (HMM)

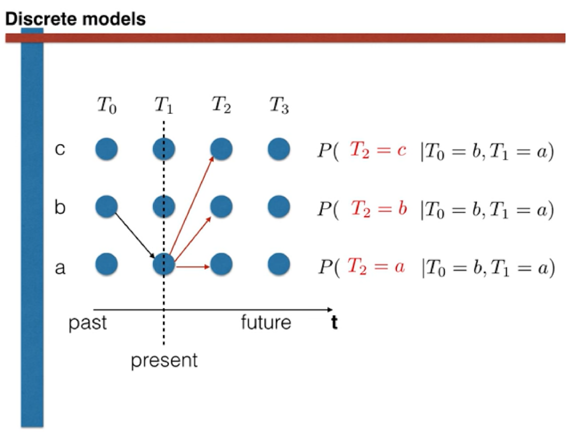

Figure 2: Visual Example (Markov Model)

3 Visual Example (Markov Model)

Let's say we have three states, a, b, and c. State a involves having a low chance of a certain disease, while state b is a moderate risk, and state c is a dangerously high risk of the disease.

4 Visual Example (Hidden Markov Model)

For this example we have data of the wrist motion of an individual eating a meal. We are trying to find out what the behavior of the person is during the interval of the wrist motion marked by the question mark.

Now, given an HMM we can simply find the best sequence that best explains

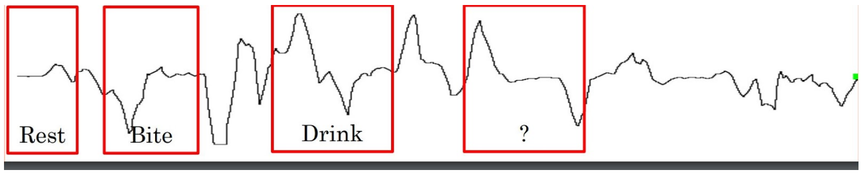

Figure 3: Visual Example (Hidden Markov Model)

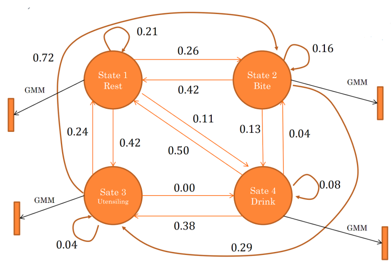

Figure 4: first figure

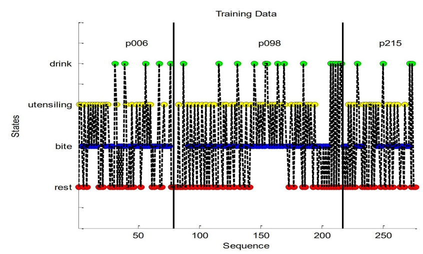

Figure 5: second figure

the observation. This way of proceeding is known as the **Viterbi Algorithm**. The Viterbi algorithm is very efficient and is useful in signal processing. Another way to proceed involves evaluating a series of observations using multiple HMMs and determines which one is the most highly recognized sequence. This is known as the **Forward-Backward Algorithm**.

5 Program Example (Markov Model)

This program uses an HMM to predict the weather based on the mood of a person. There are two possible hidden states, r for rainy weather, and s for sunny weather. The observed variable is the mood of the person which is either h for Happy, or g for Grumpy. This model returns the predicted weather as output after taking in the moods of the person as input.

```
import numpy as np
# Transition Probabilities
p_ss = 0.8
p_sr = 0.2
p_rs = 0.4
p_rr = 0.6

# Initial Probabilities
p_s = 2/3
p_r = 1/3

# Emission Probabilities
```

```
p_sh = 0.8
p_sg = 0.2
p_rh = 0.4
p_rg = 0.6

moods = ['H', 'H', 'G', 'G', 'G', 'H']
probabilities = []
weather = []

if moods[0] == 'H':
    probabilities.append((p_s*p_sh, p_r*p_rh))
else:
    probabilities.append((p_s*p_sg, p_r*p_rg))

for i in range(1,len(moods)):
    yesterday_sunny, yesterday_rainy = probabilities[-1]
    if moods[i] == 'H':
        today_sunny = max(yesterday_sunny*p_ss*p_sh,
             yesterday_rainy*p_rs*p_sh)
        today_rainy = max(yesterday_sunny*p_sr*p_rh,
             yesterday_rainy*p_rr*p_rh)
        probabilities.append((today_sunny, today_rainy))
    else:
        today_sunny = max(yesterday_sunny*p_ss*p_sg,
             yesterday_rainy*p_rs*p_sg)
        today_rainy = max(yesterday_sunny*p_sr*p_rg,
             yesterday_rainy*p_rr*p_rg)
        probabilities.append((today_sunny, today_rainy))

for p in probabilities:
    if p[0] > p[1]:
        weather.append('S')
    else:
        weather.append('R')
```

6 Applications

HMMs are used in Natural Language Processing for Part of Speech tagging. This returns the Part of Speech tag each word in a sentence is. The observed variables are the words and the hidden states are the tags. HMMs can be used in Speech Recognition in which the observed variable is the spoken audio received and the hidden states are the words being spoken. HMMs are also used in robotics, such as when a sensor takes in raw input which would be the observed variable and the current state of the robot would be one of the possible hidden states. HMMs are commonly used in genetics especially for analyzing sequences of DNA.

7 Acknowledgements

The figures and the source code used in this lecture were not created by me. Credit goes to:

Clemson University Markov Models Paper

NCBI Natural Language Processing Introduction by
Prakash M Nadkarni, Lucila Ohno-Machado, and Wendy W Chapman

Luis Serranos A friendly introduction to Bayes Theorem and Hidden Markov Models