Hidden Markov Models

Diljeet Jagpal

-

Contents

Preface

Preface

Chapter 1

Introduction

The goal of this project is to determine if HMMs are suitable as rain generators.

The first task will be to extend on the work done by Grando. In her testing, she has come to the conclusion that HMMs are suitable as rain generators however she has used the same data for testing as she used for training. This, quite likely, has lead to bias and thus we will extend her work by conducting out-of-sample tests.

If possible, I will build the software so it is user friendly and efficient. With this, I can test data for multiple locations. This will allow me to understand if the result is truly signficant, at least more so than just one location.

Chapter 2

Preliminaries

In this section, we will briefly visit foundations on which we will build throughout this paper. For most, this will be a simple refresher.

2.1 Mathematical Foundations

We start with a few key mathematical concepts.

2.1.1 Probability Theory

To discuss any probablisitic ideas we must first understand general probability theory. This can be done through the definition of a probability space.

Definition 2.1. Probability Space
A probability space is defined by $(\Omega, \mathcal{F}, \mathbb{P})$. Ω is the non-empty set of all possible outcomes, such that all events $\omega \in \Omega$. $\mathbb{P}$ is a probability measure, a function $\mathbb{P}$(A) that maps event A to a number within [0,1] based on the liklihood of the event. $\mathcal{F}$ is a σ-algebra on Ω if

1. $\Omega \in \mathcal{F}$
2. A $\in \mathcal{F}$ implies $A^c \in \mathcal{F}$
3. if A_1, A_2, A_3,... are in $\mathcal{F}$ then so is $A_1 \cup A_2 \cup A_3$...

2.1.2 Conditional Probability

Sometimes we require the probability of an event assuming another event has occured. In such situations we require conditional probability. Given two events A and B, the probability of event A occuring conditioned on the occurance of event B can be calculated as below.

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)},\ \forall A \in \mathcal{F} \tag{2.1}$$

From 2.1 and the fact that for dependednt events $\mathbb{P}$(A $\cap$ B) = $\mathbb{P}$(B $\cap$ A) we can see that:

$$\mathbb{P}(A \cap B) = \mathbb{P}(A|B)\mathbb{P}(B) = \mathbb{P}(B|A)\mathbb{P}(A), \ \forall A, B \in \mathcal{F} \tag{2.2}$$

Subsituting 2.2 into 2.1 we get the famous Bayes Theorem.

Theorem 2.2. *Bayes' Theorem*
For dependent events A and B with probability space $(\Omega, \mathcal{F}, \mathbb{P})$ *, where* $\mathbb{P}$ *(B)* $\neq$ *0,*

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(B|A)\mathbb{P}(A)}{\mathbb{P}(B)}, \ \forall A \in \mathcal{F} \tag{2.3}$$

2.1.3 Stochastic Process

To be able to define a Markov model, of any kind, we must first define a stochastic process.

Definition 2.3. Stochastic Process
Given an ordered set T and probability space $(\Omega, \mathcal{F}, \mathbb{P})$ a stochastic process is a collection of random variables X = $\{X_t;\ t \in T\}$. Based on t $\in$ T and $\omega \in \Omega$ we get a numerical realization of the process. For simplicity, this may be viewed as a function; $X_t(\omega)$.

2.2 Applied Foundations

Chapter 3

Standard Markov and Markov Property

3.1 History

Andrei Markov discovered the Markov model while analyzing the relationship between consecutive letters from text in the Russian novel "Eugene Onegin". With a two state model (states Vowel and Consonant) he proved that the probabililty of letters being in a particular state are not independent. Given the current state he could probabilistically predict the next. This chain of states, with various probabilities to and from each state, formed the foundation of the Markov Chain.

3.2 Markov Chain

A Markov chain is a network of connceted states. At any given time the model is said to be in a particular state. At a regular discrete interval the model has the ability to change states, which state will depend on the probaility and randomness. To define a Markov chain we must first address the Markov property. This simply states that the next state depends only on the current state. A more formal definition is given below.

Definition 3.1. Markov Property
Let $\{X_t$; t $\in \mathbb{N}_0\}$ denote a stochastic proccess 2.3, where t represents discrete time. The process has the Markov property if and only if,

$$\mathbb{P}\{X_{n+1} = i_{n+1} | X_n = i_n, X_{n-1} = i_{n-1}, ..., X_0 = i_0\} = \mathbb{P}\{X_{n+1} = i_{n+1} | X_n = i_n\} \tag{3.1}$$

A Markov chain is simply a model that obeys 3.1. Again a more formal definition is given below.

Definition 3.2. Markov Chain
A stochastic process $\{X_t$; t $> 0\}$ is a Markov Chain if and only if it satisfies the Markov property 3.1.

To store the sequence of states a Markov chain has been through, we use the set $Q = \{q_t; t \in \mathbb{N}_0\}$, where q_t represents the state at time t. We will use this notation throughout the paper.

Example 3.3. Given a Markov Model with states S = $\{S_1, S_2, S_3\}$, if the model starts at S_2 and then goes to S_3 and then back to S_2 the state sequence Q will be $Q = \{q_1 = S_2, q_2 = S_3, q_3 = S_2\}$.

From the markov property we can see that the only thing that influences q_t is q_{t-1}. Thus we can make a prediction for q_{t+1} based on the outward transition probabililties from state q_t. If we calculate the probability of all possible states S_j and use a biased random variable we can make a prediction.
i.e.
Given a Markov chain with N states including i and j and discrete time $t \in \mathbb{N}_0$:

$$\mathbb{P}(q_t = S_j | q_{t-1} = S_i)_{1 \leq i,j \leq N} \tag{3.2}$$

These probabilities can vary with time but can become quite complex. Thus, we usually assume the probabilities are constant. These special Markov models are called time-homogenous.

Definition 3.4. Time homogenous
Let $\{X_t$; t $\in \mathbb{N}_0\}$ denote a stochastic proccess 2.3, where t represents discrete time, and $p(i,j)$ represent the transition probability from state i to state j. If

$$\mathbb{P}\{X_n = j | X_{n-1} = i\} = p(i,j), \forall n \in \mathbb{N}_0 \tag{3.3}$$

then the process is time-homogenous.

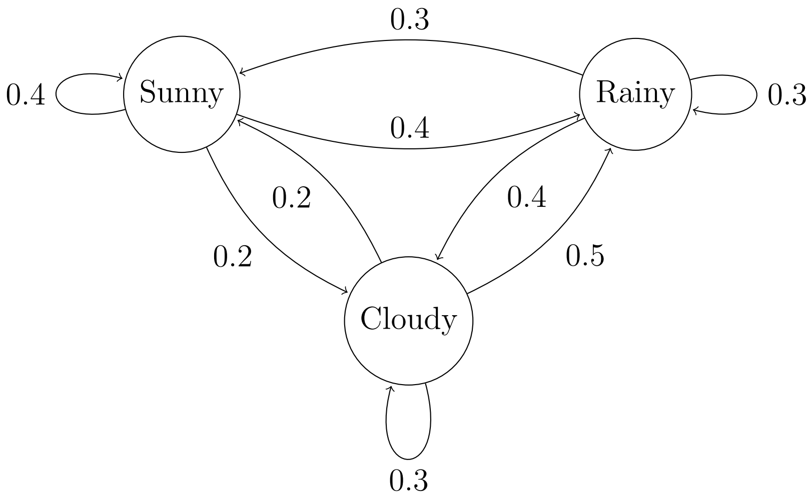
For a discrete Markov model with N states, there are N^2 possible transitions, where $p(i,j) = 0$ represents an impossible transition. We must store each of these probabilities. Given a time-homogenous Markov chain, we can create a 2-dimensional N x N matrix of transition probabilities p. Unique Markov chains have unique transition matrices. These matrices can be defined as below:

$$p = \{p(i,j) = \mathbb{P}\{X_n = j | X_{n-1} = i\}\}_{1 \leq i,j \leq N} \tag{3.4}$$

All p matrices have some special characteristics. The first, is that all values contained within p must be within $[0,1]$. This is quite natural as all values are probabililties and thus by definition must lie within $[0,1]$. The second is that all either the rows, columns or both form stochastic vectors. If it is the rows then the matrix is defined as a right-stochastic matrix, if it is the columns then it is called a left-stochastic matrix.

To demonstrate we will present an example where the weather is represented by the states.

Example 3.5. Let $\{X_t; t \in \mathbb{N}_0\}$ denote a Markov Chain, with state space S = {rainy, sunny, cloudy}, where t represents the number of days from start. Since any state can transition into any other state, we say this model is ergodic. This can also be seen through the figure below as each state is connected to all others.



In this Markov Chain diagram, as per usual, the arrows indicate the transition between states and the values next to these correspond to the probability of this transition.

Using 3.5 we can create a matrix p. To make this clear, we first create a table with our states labeled for rows and columns, where the $p(i, j)$ is the cell corresponding to row i and column j.

x	Sunny	Rainy	Cloudy
Sunny	0.4	0.4	0.2
Rainy	0.3	0.3	0.4
Cloudy	0.2	0.5	0.3

This content of this table forms the matrix p.

$$p = \begin{bmatrix} 0.4 & 0.4 & 0.2 \\ 0.3 & 0.3 & 0.4 \\ 0.2 & 0.5 & 0.3 \end{bmatrix} \tag{3.5}$$

Now that we have a Markov Model with its p we must reflect on its potential uses. Some natural questinos one may ask are:

1. Given at time t the state was S_i, what is the most likely state time $t + 1$?

2. What is the probability of getting a particular state sequence O?

3. What is the probability of staying within a state for d time steps?

This first problem can be addressed simply using the matrix p. Our motivation behind it was to build a matrix where p_{ij} contains the probability of moving from i to j. Thus, we can simply look at the current row and find the largest probability and its corresponding j.

Example 3.6. Let us refer back to 3.5 and assume the current state is Cloudy. We can see that $0.2 < 0.3 < 0.5$ and that 0.5 corresponds to Rainy. Thus our the most likely next state would be Rainy.
If the current state was sunny, we would have two maximums of 0.4. In such case, the process is equally likely to go to either.

The second problem provides a state sequence $Q = \{q_t, q_{t+1}, q_{t+2}...\}$ and asks what the probability of this occuring is, i.e. $\mathbb{P}(Q|model)$. This can be simplified quite easily as shown below.

$$\begin{aligned}
\mathbb{P}(Q|model) &= \mathbb{P}(\{q_t, q_{t+1}, q_{t+2}...\}|model) && (3.6)\\
&= \mathbb{P}(q_t)\mathbb{P}(q_{t+1}|q_t)\mathbb{P}(q_{t+2}|q_{t+1})... && (3.7)\\
&= \mathbb{P}(q_t)p(q_t, q_{t+1})p(q_{t+1}q_{t+2})... && (3.8)
\end{aligned}$$

Example 3.7. We can now use 3.6 to demonstrate the probability of a state squence from 3.5. Let $Q = \{\text{Sunny, Sunny, Cloudy, Rainy}\}$. Given that we start from Sunny:

$$\begin{aligned}
\mathbb{P}(Q|model) &= \mathbb{P}(\{Sunny, Sunny, Cloudy, Rainy\}|model) && (3.9)\\
&= \mathbb{P}(Sunny)\mathbb{P}(Sunny|Sunny)\mathbb{P}(Cloudy|Sunny)\mathbb{P}(Rainy|Cloudy) && (3.10)\\
&= 1 * 0.4 * 0.2 * 0.5 && (3.11)\\
&= 0.04 && (3.12)
\end{aligned}$$

The third problem asks how long is the model likely to stay in any given state. Assume that the model stays in state S_i for d days. We can create a state sequence for this, $Q = \{q_t = S_i, q_{t+1} = S_i, ..., q_{t+d-1} = S_i, q_{t+d} \neq S_i\}$. Using the state sequence probabililty calculation from before we can compute the following equation:

$$\mathbb{P}(Q|model) = p(i,i)^{d-1}(1 - p(i,i)) = p_i(d) \qquad (3.13)$$

We label this $p_i(d)$ to represent the discrete probability density function of the duration d in state i. From this we can calculate the expected stay in any particular state. This is done using the following formula:

$$\begin{aligned}
\bar{d_i} &= \sum_{d=1}^{\infty} d p_i(d) && (3.14)\\
&= \frac{1}{1 - p(i,i)} && (3.15)
\end{aligned}$$

Example 3.8. Reffering back to 3.5, suppose we would like to see how many days in a row we expect it to be sunny. We see that $p(Sunny, Sunny) = 0.4$. Now we can use 3.14 to find,

$$\begin{aligned}
\bar{d}_{Sunny} &= \sum_{d=1}^{\infty} d p_{Sunny}(d) && (3.16)\\
&= \frac{1}{1 - p(Sunny, Sunny)} && (3.17)\\
&= \frac{1}{1 - 0.4} && (3.18)\\
&= 1.67 && (3.19)
\end{aligned}$$

Thus we expect it to remain sunny for 1.67 days. Since we are dealing with discrete data, it is more approprate to round up to 2 days.

The Markov model we have been discussing so far is called an observable Markov Model as we can observe its events. This is not always the case.

3.3 Motivating the Hidden Markov Model

Sometimes you do not get an observation from your states but only see the effect of the state change. To help develop this idea we borrow an example from Rabiner and Juang, 1986.

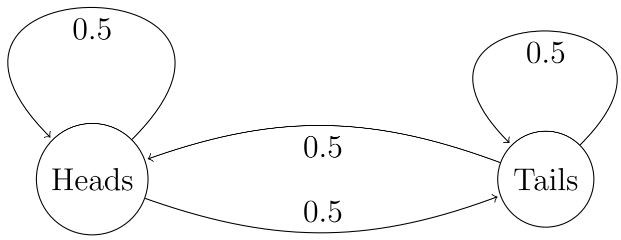
Example 3.9. Suppose there is someone hidden behind a barrier or curtain, where you cannot see what they are doing. This person is doing some kind of experiment with flipping coins and shouting heads or tails at regular intervals. You do not know:

i How many coins there are.

ii If the coin/s is/are fair.

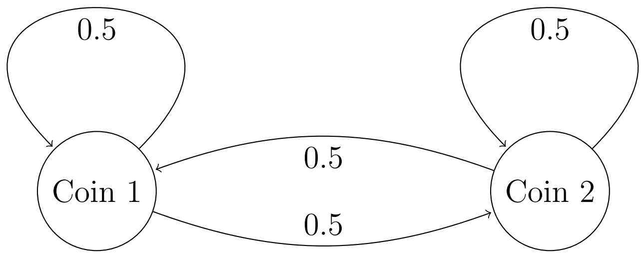
Since the problem is quite vague, we must experiment with various models and see which fits the data the best. Since the only thing we can observe is the outcome, heads or tails, we will refer to this as our observation. Lets start with the simplest.

1. 1-Fair Coin
   In this model we have two states, heads and tails. There is a 0.5 probability for the model to change state and equally for staying in the same. The simplicity of this model comes at the cost of many assumptions that may not nessesarily hold true.



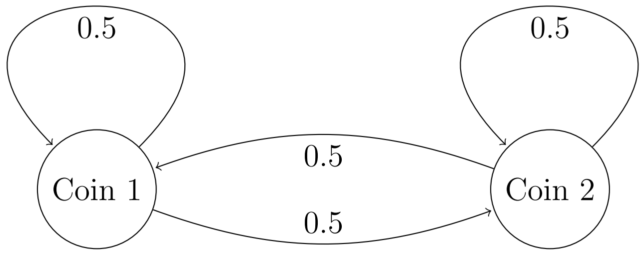
$\mathbb{P}(Heads)$	0.5
$\mathbb{P}(Tails)$	0.5

2. 2-Fair Coin
   This model also has two states but this time they reprsent two different coins. Both can produce heads and tails, thus we do not know which produced the observation. The fact that they are both Fair coins means that as an external observer, we will not be able to see a difference between this model and the 1-Fair Coin. However, it is clear that the observations are now independent of the hidden states transitions, as they are equally likely regardless of the state. In the 1-Fair Coin model we can determine perfectly which state the model is in from the observation, but this is no longer possible with this model.



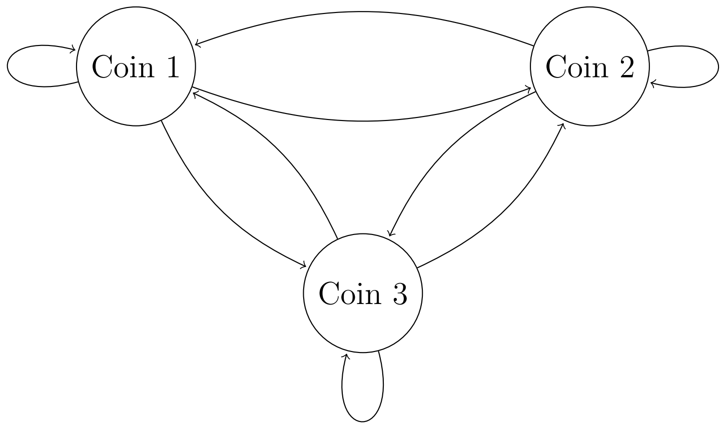
	$\mathbb{P}(Coin1)$	$\mathbb{P}(Coin2)$
$\mathbb{P}(Heads)$	0.5	0.5
$\mathbb{P}(Tails)$	0.5	0.5

3. 2-Biased Coins
   Although this model is similar to 2-Fair Coins, the change in $\mathbb{P}(Heads)$ and $\mathbb{P}(Tails)$ for each state has a large impact on the observation likelihood. It can be seen that if the model is in state 1 heads is more likely and if the state is in state 2 tails is more likely. As an observer we can now say if we see a tails, it was most likely from state 2 and if we see a heads, it was most likely from state 1. The change between these two states must be an unrealted proabbility, such as a third coin or another source of randomness.



	$\mathbb{P}(Coin1)$	$\mathbb{P}(Coin2)$
$\mathbb{P}(Heads)$	0.75	0.25
$\mathbb{P}(Tails)$	0.25	0.75

4. 3-Biased Coins
   Similar to the 2-Biased Coins model, the probabilities for heads and tails vary with the three states.



	$\mathbb{P}(Coin1)$	$\mathbb{P}(Coin2)$	$\mathbb{P}(Coin3)$
$\mathbb{P}(Heads)$	0.6	0.25	0.45
$\mathbb{P}(Tails)$	0.4	0.75	0.55

An important conclusion one may make, after contemplating which model is best, is that it is quite difficult to decide on how many states are needed without priori information. In such case, we must ensure that there are enough states, such that the model does not over generalize but also not so many that it requires to much data to train. Naturally, one would assume choosing the larger number of states is more suitable as they can take the shape of a model with fewer states but the opposite is not true. However, larger models require much more data to be statisticaly reliable as there are alot more uknown variables that must be found. Below we show this for our models, along with where the uknowns come from (-s being from the state transition proabilities and -O being from the observation probabililties).

Model	Number of Unknowns	From
1-Fair Coin	0	-
2-Fair Coins	1	1-S
2-Biased Coins	4	2-S 2-O
3-Biased Coins	9	6-S 3-O

Hence, the best approach is usually to base the model state size on the amount of available data. This is not always garenteed to give reliable results but can sometimes be the only option, for example when limited by data.

The first model having 0 unknowns is also interesting. This is because here each state is linked with only one observation. Thus, as an observer, given an observation we can assert what state the model is in. This means the states are no longer hidden and the model is simply a standard markov model.

Another important detail the avid reader may have noticed is in the case of 2-Biased Coins and 3-Biased Coins it is possible to make a reasonable guess of what state the model is in, i.e. which coin was flipped, based on the observation. Furthermore, the statistical properties of predictions generated using 1-Fair Coin and 2-Fair Coins should be identical but for 2-Biased Coins and 3-Biased Coins be somewhat unqiue. Due to this, unless the underlying system is completly fair, like in the first two, it should be possible to determine which model best fits the data, and thus determine the model.

This brings us to the core idea behind hidden markov models. Although we cannot directly observe the model or its properties, through sufficient observation data, we can attempt to create one that fits this data the best.

Chapter 4

Hidden Markov Model

4.1 Definition

A hidden markov model is a doubly stochastic markov process. This comes from the fact that there are two stochastic processes, one determining the transition between states and one determining the output observation.

To define a hidden markov model we need 5 things.

1. N

   i N is the number of hidden states. This is usually based of something in the real world but sometimes can be uknown, as in 3.9.

   ii The states are usually ergodic, i.e. from any given state you can reach another eventually.

   iii These states from the state space $S = \{s_1, s_2, ..., s_N\}$.

2. M

   i M is the number of observable outputs.

   ii These combine to make a discrete alphabet of observations called $V = \{v_1, v_2, ..., v_M\}$.

3. A

   i A is the state transition matrix.

   ii This is the same as the p matrix 3.4.

   iii $\mathrm{A} = \{a_{ij}\}$

   iv $a_{ij} = \{p(i,j) = \mathbb{P}\{X_n = j | X_{n-1}\}$

4. B

   i B is the observation probability matrix.

   ii $\mathrm{B} = \{b_j(k)\}$

   iii $b_j(k) = \mathbb{P}(V_k \ at \ t | q_t = S_j)_{1 \leq j \leq N, 1 \leq k \leq M}$

5. π

i π is a vector containing all inital state probabilities.

ii $\pi = \{\pi_i\}$

iii $\pi_i = \mathbb{P}(q_1 = S_i) for 1 \leq i \leq N$

We can now combine the above to provide a formal definition of hidden markov models.

Definition 4.1. Hidden Markov Model
A Hidden Markov Model is a 5-tuple $\{N, M, A, B, \pi\}$ that is used to represent a doubly stochastic process where the hidden process is markovian.

Before we continue we will provide notation that will be used for the rest of the paper.

i We will continue to use Q 3.3 to represent the state sequence for markov model, i.e. the hidden process.

ii We will use $O = \{o_1, o_2, ..., o_T\},\ o_i \in V$ to represent the observation sequence.

iii we will occasionally represent the hidden markov model as $\{N, M, \lambda\}$ where $\lambda = \{A, B, \pi\}$

4.2 Using HMM

As with any mathematical model, we can use HMMs prediction. Through the output liklihood we can find the liklihood of paticular sequences and through the given probablities and random variables we can create predictions.

4.2.1 Predicitive Model

We can use HMM to generate a sequence of potential observations. Given a a hidden markov model, lets call it H $= \{N, M, A, B, \pi\}$, to generate a sequence of T observations O we do the following steps:

1. Using π as a probability distribution, set $t = 1$ and randomly select a state as the first $q_1 = s_i$.

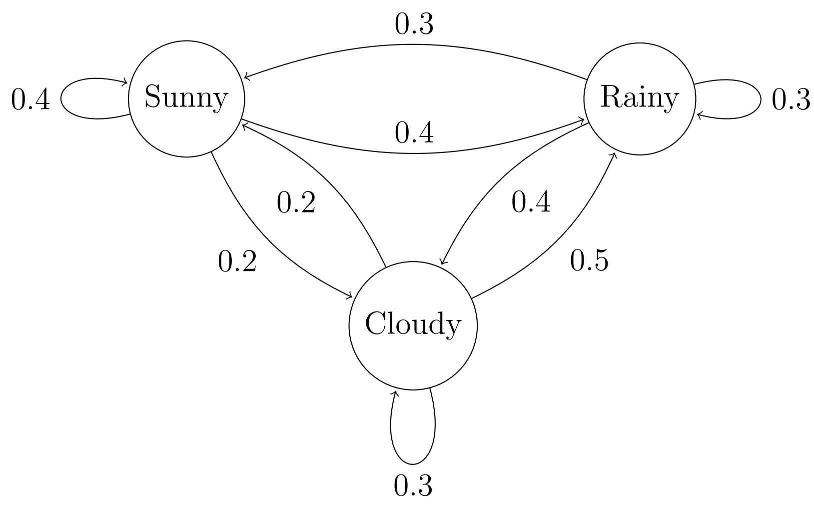
2. Using $b_i(k)$ as a probability distribution, randomly select the observation $O_t = v_k$.

3. Using a_{ij} as a probability distribution, set $t = t + 1$ and randomly select a the next state $q_{t+1} = s_j$.

4. Repeat steps 2 and 3 until $t = T$

To demonstrate this method, we will use an adapted version of 3.5. This version will have the markov chain from before as the underlying hidden process and another process with state space {Happy, Sad}, which we can observe.

Example 4.2. Suppose Alice is hidden away from the world and has no access to information regarding the weather. She meets Bob everyday and knows how weather affects his mood. For simplicity, assume Bob only has two moods, happy and sad. Given matrix A, B and vector π can she predict his mood for 3 consecutive days?
From context we can deduce the following:



i N = 3, number of hidden states

ii M = 2, number of possible observations

iii

$$A = \begin{bmatrix} 0.4 & 0.4 & 0.2 \\ 0.3 & 0.3 & 0.4 \\ 0.2 & 0.5 & 0.3 \end{bmatrix} \tag{4.1}$$

iv

$$B = \begin{bmatrix} 0.8 & 0.2 \\ 0.7 & 0.3 \\ 0.6 & 0.4 \end{bmatrix} \tag{4.2}$$

v

$$\pi = \begin{bmatrix} 0.5 \\ 0.3 \\ 0.2 \end{bmatrix} \tag{4.3}$$

We can now use 4.2.1 to create an observation sequence $O = \{o_1, o_2, o_3\}$. At multiple occasions we will require a random variables. I will generate uniform random variables through python using my "rand.py" file. I will label these r.v.. I select the state that corresponds to the region on the probability distribution that the random variable lies on.

1. We generate a r.v. = 0.0058. Using Pi as the probaility distribution, we select Sunny. We can now set $q_1 = s_1$.

2. We generate a r.v. = 0.1947. Using $b_1(k)$ as the probability distribution we select Happy as the observation. We can now set $o_1 = v_1$.

3. we generate a r.v. = 0.7168. Using a_{1j} as the probability distribution we select Rainy as the next state. We now set $q_2 = s_2$.

4. We generate a r.v. = 0.1060. Using $b_2(k)$ as the probability distribution we select Happy as the observation. We can now set $o_2 = v_1$.

5. we generate a r.v. $= 0.8977$. Using a_{2j} as the probability distribution we select Cloudy as the next state. We now set $q_3 = s_3$.

6. We generate a r.v. $= 0.1369$. Using $b_3(k)$ as the probability distribution we select Happy as the observation. We can now set $o_2 = v_1$.

Finally we can look back on our prediction O and see that it is equal to $\{v_1, v_1, v_1\}$, i.e. we predict three consecutive Happy days.

4.2.2 Three Key Problems

There are many interesting questions one may have regarding the HMM but there are three famous ones which we will focus on.

1. Evaluation
   Given model H $= \{N, M, A, B, \pi\}$ what is the probability that it generated the sequence of observations $O = \{o_1, o_2, ..., o_T\}$? i.e. $\mathbb{P}(O \mid H)$

2. Decoding
   What sequence of states $Q = \{q_1, q_2, ..., q_3\}$ best explains a sequence of observations $O = \{o_1, o_2, ..., o_T\}$?

3. Learning
   Given a set of observation $O = \{o_1, o_2, ..., o_T\}$, how can we learn the model H $= \{N, M, A, B, \pi\}$ that would generate them?

In the coming sections for each problem we will be motivating its uses and exploring its solutions.

4.3 Problem 1: Evaluation

Lets start by addressing question 1. Informally, we are looking for the proability that a given model generated a sequence of observations, i.e. $\mathbb{P}(O|\lambda)$.

This probability has many useful applications. For example, you may have multiple potential models λ_i, for a given senario, but are unable to decide which one is the most suitable. In such case, you can now calculate this probability for each λ_i and select the one that gives the highest proability as you can now concretly say this model has the highest liklihood to generate the given observations.

To find this probability we must consider the internal hidden states of the model. Since our probability of observations $\{b_j(k)\}$ is conditioned on the hidden state, we can start by calculating this probaility conditioned on these states. Lets assume we know what the state sequence $Q = \{q_1, q_2, ..., q_t\}$ is.

$$\mathbb{P}(O|Q, \lambda) = \prod_{t=1}^{T} \mathbb{P}(O_t|q_t, \lambda) \tag{4.4}$$

All that is happening in the above equation is the probability that an observation was generated given a particular model and a particular hidden state at time t, is multiplied by all by the same for all these probabilities up to time T. In essence this breaks the lhs into T parts.

An observation one may make is that these probabilities are simply taken from the matrix B.

$$\mathbb{P}(O_t|q_t, \lambda) = b_{q_t}(O_t), \quad \forall t \in [0, T]$$

Thus we can rewrite 4.4 as:

$$\mathbb{P}(O|Q, \lambda) = b_{q_1}(O_1)b_{q_2}(O_2)...b_{q_T}(O_T) \tag{4.5}$$

Our next objective is to remove Q from the conditioned part of the probability. To do this we must first calculate $\mathbb{P}(Q|\lambda)$. This is simply the probability of transitioning from q_1 to q_2, q_2 to q_3 etc. More formally we can use matrix A to find the probability of each of these transitions and since we are finding the total for the entire sequence, we just multiply them all togther. We start with π_{q_1} as we also need the probability of starting at q_1.

$$\mathbb{P}(Q|\lambda)\pi_{q_1}a_{q_1q_2}a_{q_2q_3}...a_{q_{T-1}q_T} \tag{4.6}$$

We can now successfully remove Q from the condition using 4.5 and 4.6:

$$\begin{aligned}
\mathbb{P}(O, Q|\lambda) &= \mathbb{P}(O|Q, \lambda)\mathbb{P}(Q|\lambda) && (4.7)\\
&= \pi_{q_1}b_{q_1}(O_1)a_{q_1q_2}b_{q_2}(O_2)a_{q_2q_3}...a_{q_{T-1}q_T}b_{q_T}(O_T) && (4.8)
\end{aligned}$$

This gives us the joint probability of observations and the internal states. In other words, it provides the probabilty that given observations O and internal state sequence Q was generated by model λ. To achieve our desired probablity all we need to do is get rid of the Q. Since it is another input, all we must do is sum each value of 4.7. As we have accounted for every possible Q, we no longer need to worry about its particular value. This leaves us with:

$$\begin{aligned}
\mathbb{P}(O|\lambda) &= \sum_{allQ} \mathbb{P}(O|Q, \lambda)P(Q|\lambda) && (4.9)\\
&= \sum_{q_1,q_2,...,q_T} \pi_{q_1}b_{q_1}(O_1)a_{q_1q_2}b_{q_2}(O_2)a_{q_2q_3}...a_{q_{T-1}q_T}b_{q_T}(O_T) && (4.10)
\end{aligned}$$

Although this solution is correct, calculating this is infeasible. This is because it requires to many computations. For T timesteps and N states, to find every possible Q we must sum over N^T state sequences. For each timestep we require a multiplication to $a_{q_{i-1}q_i}$ and $b_{q_i}(O_i)$, except the last where there are no transitions. This leads to $2T - 1$ multiplications for each state sequence. Lastly we require N^T addition operations to sum the result for each state sequence. This gives us a final total number of operations of $(2T - 1)N^T + (N^T - 1)$. This is a problem as even with a smaller model we require an infeasible amount of calculations. To overcome this problem we look for a more efficient method, the Forward-Backward algorithm.

4.3.1 Forward-Backward Algorithm

The Forward-Backward algorithm (F-B) is composed of two helper functions α and β. We will start by discussing the former.

$$\alpha_t(i) = \mathbb{P}(O_1, O_2, O_3, ..., O_t, q_t = S_i|\lambda) \quad (4.11)$$

α is an extremly powerful tool in reducing the number of calculations. As given by 4.3.1, it providse the probability that at time t we have seen a sequence of observations and are currently at state $q_t = S_i$. This is not quite $\mathbb{P}(O|\lambda)$ but it represents a part of it. Instead of the probability of the whole sequence, it breaks it into a chunk of size t, commits to end at a particular state and then calculates the same proabbility for this.

We can combine with induction to produce an iterative process that can calculate 1.

i Base case
For the base case we require the probability of the q_1 being equal to S_1 and giving us observation O_1. The former is addressed by π_i and the later by $b_i(O_1)$. This gives us:

$$\alpha_1(i) = \pi_i b_i(O_1), \quad i \in [1, N] \quad (4.12)$$

ii Inductive step:
For the inductive step we must consider how to approach the next timestep. We will again be calculating for all $j \in [1, N]$ and as such must take into consideration, for each j, every possible i. This is again the same set of $[1, N]$. Therefore, to account for all possible previous states and their transition to the current state, we must sum over 1 to N the product of $\alpha_t(i)$ and a_{ij}. For the given observation, as before, we compute $b_j(O_{t+1})$. Additionally, we must stop before reaching the final step as there is no outward transition and thus this would not be applicable. This gives us:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}), \quad j \in [1, N], t \in [1, T-1] \quad (4.13)$$

iii Termination Step:

4.4 Problem 2: Decoding

4.4.1 Viterbi Algorithm

4.5 Problem 3: Learning

4.5.1 Expectation Maximization

4.5.2 Baum-Welch Algorithm

4.6 Modified HMM

4.6.1 GMM

Bibliography

Rabiner, L. and B. Juang (1986). “An introduction to hidden Markov models”. In: *IEEE ASSP Magazine* 3.1, pp. 4–16. DOI: 10.1109/MASSP.1986.1165342.