Hidden Markov Model

Lab 4.2

What is HMM

A Hidden Markov Model (HMM) is a statistical model used to describe the probabilistic relationship between a sequence of observations and a sequence of hidden states.

HMMs are particularly useful for modeling sequential data and time series.

An HMM consists of two types of variables: hidden states and observations.

- The **hidden states** are the underlying variables that generate the observed data, but they are not directly observable.
- The **observations** are the variables that are measured and observed.

The relationship between the hidden states and the observations is modeled using a probability distribution.

The Hidden Markov Model (HMM) is the relationship between the hidden states and the observations using two sets of probabilities: the transition probabilities and the emission probabilities.

- The **transition probabilities** describe the probability of transitioning from one hidden state to another.
- The **emission probabilities** describe the probability of observing an output given a hidden state.

Break down components and HMMs:

1. **Sequence of Observations:** In many real-world applications, you have a sequence of observations or data points. These observations can be anything, such as words in a sentence (in natural language processing), stock prices over time (in finance), or DNA base pairs in a genome (in bioinformatics).
2. **Sequence of Hidden States:** In addition to the observations, there is a corresponding sequence of hidden states. These states are not directly observable but influence the observations. Each hidden state represents a certain "underlying" or "hidden" condition or situation that generates the observed data at a given time step.
3. **State Transition Probabilities:** HMMs use probabilities to model the transitions between hidden states. These transition probabilities represent the likelihood of moving from one hidden state to another at each time step. The transitions are often assumed to follow a Markov process, meaning that the future state depends only on the current state (not on the past states).
4. **Emission Probabilities:** Emission probabilities represent the likelihood of observing a particular data point given the current hidden state. These probabilities are specific to each state and are used to generate the observed data.

Here's how the HMM process works

- You start with an initial hidden state, often defined by a probability distribution.
- At each time step, you use the state transition probabilities to determine the next hidden state.
- Given the current hidden state, you use the emission probabilities to generate an observation.
- You continue this process for each time step to generate a sequence of observations and hidden states.

Hidden Markov Models (HMMs) with example

Sequence of Observations:

- Example: In natural language processing, consider a sequence of words in a sentence. Each word is an observation.
- Sentence: "The cat is on the mat."
- Observations: ["The", "cat", "is", "on", "the", "mat"]

Sequence of Hidden States:

- Example: In part-of-speech tagging, we have a sequence of words (observations) and a corresponding sequence of part-of-speech tags (hidden states).
- Sentence: "The cat is on the mat."
- Hidden States: ["Det", "Noun", "Verb", "Prep", "Det", "Noun"]
- Here, "Det" represents a determiner, "Noun" represents a noun, "Verb" represents a verb, and "Prep" represents a preposition.

State Transition Probabilities:

- Example: In a simple weather model, we have two hidden states: "Sunny" and "Rainy." We can define transition probabilities to represent how likely the weather is to change.
- Transition Probabilities:
  - P(Sunny to Sunny) = 0.7
  - P(Sunny to Rainy) = 0.3
  - P(Rainy to Sunny) = 0.4
  - P(Rainy to Rainy) = 0.6
- These probabilities determine how often the weather transitions from sunny to rainy or vice versa.

Emission Probabilities:

- Example: In a simple ice cream sales model, we have hidden states "Hot" and "Cold." We want to model the number of ice creams sold as observations.
- Emission Probabilities:
  - P(Ice Creams Sold | Hot) = 0.8
  - P(Ice Creams Sold | Cold) = 0.2

In this example, we'll use a simple two-state HMM to model weather conditions as "Sunny" and "Rainy." The observations represent whether a person carries an umbrella, with "Yes" and "No" as possible observations.

Sequence of Observations:

- Observations represent whether a person carries an umbrella on a given day.
- Observations: ["Yes", "No", "No", "Yes", "Yes"]

Sequence of Hidden States:

- Hidden states represent the underlying weather conditions, which can be "Sunny" or "Rainy."
- Hidden States: ["Sunny", "Rainy", "Sunny", "Rainy", "Sunny"]
- These hidden states are not directly observable but influence whether people carry umbrellas.

State Transition Probabilities:

- Transition probabilities govern how the weather conditions change from one day to the next.
- For simplicity, let's assume:
  - P(Sunny to Sunny) = 0.7
  - P(Sunny to Rainy) = 0.3
  - P(Rainy to Sunny) = 0.4
  - P(Rainy to Rainy) = 0.6
- These probabilities represent how likely it is for the weather to transition between "Sunny" and "Rainy" states.

Emission Probabilities:

- Emission probabilities describe how likely a person is to carry an umbrella based on the current weather condition.
- For simplicity, let's assume:
  - P(Carry Umbrella = Yes | Sunny) = 0.2
  - P(Carry Umbrella = No | Sunny) = 0.8
  - P(Carry Umbrella = Yes | Rainy) = 0.9
  - P(Carry Umbrella = No | Rainy) = 0.1
- These probabilities represent how likely people are to carry umbrellas based on whether it's "Sunny" or "Rainy."

Hidden Markov Model Algorithm

The Hidden Markov Model (HMM) algorithm can be implemented using the following steps:

Step 1: Define the state space and observation space

The state space is the set of all possible hidden states, and the observation space is the set of all possible observations.

Step 2:Define the initial state distribution

This is the probability distribution over the initial state.

Step 3:Define the state transition probabilities

These are the probabilities of transitioning from one state to another. This forms the transition matrix, which describes the probability of moving from one state to another.

Step 4:Define the observation likelihoods:

These are the probabilities of generating each observation from each state. This forms the emission matrix, which describes the probability of generating each observation from each state.

Step 5:Train the model

The parameters of the state transition probabilities and the observation likelihoods are estimated using the Baum-Welch algorithm, or the forward-backward algorithm. This is done by iteratively updating the parameters until convergence.

Step 6:Decode the most likely sequence of hidden states

Given the observed data, the Viterbi algorithm is used to compute the most likely sequence of hidden states. This can be used to predict future observations, classify sequences, or detect patterns in sequential data.

Step 7:Evaluate the model

The performance of the HMM can be evaluated using various metrics, such as accuracy, precision, recall, or F1 score.

Simple HMM Model

[Weather Model]

LIBRARY NEEDED

1.Numpy (import numpy as np): numpy is a fundamental library for numerical operations in Python. It is widely used for handling arrays and mathematical computations, which can be useful when working with HMMs.

2. hmmlearn (from hmmlearn import hmm): This library is essential for creating and training Hidden Markov Models in Python

3. matplotlib (from matplotlib import cm, pyplot as plt): Matplotlib is used for data visualization, and you've imported components like colormaps (cm) and plotting functions (pyplot) to create plots and charts to visualize your HMM and its results.

4. %matplotlib inline: magic command that allows you to display matplotlib plots directly within the notebook interface.

5. pandas (import pandas as pd): Pandas is a popular data manipulation and analysis library. You can use it to handle, analyze, and manipulate data, which can be valuable when working with datasets that you want to feed into your HMM.

6. DataFrame, Series (from pandas): These are specific data structures provided by Pandas. DataFrames are used for tabular data, and Series are used for one-dimensional data structures.

Complex HMM Model

Weather HMM and Dataset