LAB 6: Hidden Markov Model

Credit Health Prediction

Hidden Markov Models (HMMs) are useful for modeling systems where the underlying states that influence observed behaviors are hidden. In this lab, you will implement an HMM to analyze credit health trends and observable financial behaviors. The hidden states represent an individual's credit health conditions, while the observable data (e.g., payment behaviors) are emissions from these states. This can be used to predict future credit health trends based on payment history and other observable actions.

Key Components:

- **Hidden States:** These represent the underlying financial conditions we want to model but cannot observe directly.
- **Observations:** These represent the observable behaviors in an individual's financial activity, influenced by the hidden states.
- **State Transition Probabilities:** These define the likelihood of transitioning from one hidden state to another.
- **Emission Probabilities:** These represent the likelihood of observing specific behaviors given a particular hidden state.
- **Initial State Distribution:** This defines the probability of each hidden state being the starting point in the model.

In the given context of credit health:

<u>Hidden States</u> (Credit Health Conditions):

1. **Good Standing:** Financially stable, making on-time payments.
2. **Mild Financial Strain:** Experiencing some financial pressure but still managing.
3. **High Financial Risk:** At significant risk of financial trouble.
4. **Default:** Inability to make payments, resulting in default.
5. **Recovering Credit:** Working towards improving credit after previous issues.

<u>Emissions</u> (Observable Financial Behaviors):

1. **On-Time Payments:** Consistently making payments on time.
2. **Minimum Payment Only:** Only making the minimum required payment.
3. **Late Payments:** Delayed payments, signaling financial difficulties.
4. **Missed Payments:** Completely missing payments, indicating financial trouble.
5. **Debt Consolidation:** Using debt consolidation strategies to manage debts.

<u>Variables Legend:</u>

- **avocado:** Represents the state transition matrix, which defines the probabilities of moving from one hidden state to another.
- **bubblegum:** The initial state distribution (priors), representing the probability of each hidden state being the initial state.
- **mushroom:** A list of all possible hidden states
- **cheese:** The number of hidden states in the model.
- **kangaroo:** Emission probabilities, defining how likely each observable state is given a hidden state.
- **spaceship:** A list of observations
- **jellybean:** The number of observations in the dataset.
- **pancake:** The likelihood value calculated for the first observation sequence.
- **lemon:** A list of observations for the second likelihood calculation.
- **jam:** The likelihood value for the second observation sequence.
- **rat:** The output of the Viterbi algorithm, which is the most likely sequence of hidden states given the observations.
- **mouse:** The expected output of the Viterbi algorithm, which the model should predict.

Implementation

You are provided with two files:

1. **HMM.py** - This file contains the structure of the Hidden Markov Model and includes the following functions that need to be filled in:

Function Name	Input	Output
viterbi_algorithm()	List of observation sequences	The most likely sequence of hidden states
calculate_likelihood()	List of observation sequences	Likelihood of the given observation sequence

You will need to complete the provided function skeletons to implement the Viterbi algorithm and calculate the likelihood of a sequence. **Ensure that you use Dynamic Programming ONLY.**

2. **test.py** - This file is used to test your HMM implementation. It contains predefined test cases based on the Credit Health example described above. The test cases will validate both the Viterbi algorithm and the likelihood calculations.

Important Points:

1. **File Naming:** Rename your solution file to CAMPUS_SECTION_SRN_Lab6.py and run the command python3 test.py --ID CAMPUS_SECTION_SRN_Lab6
2. **Do not change function definitions** as provided in HMM.py. Write your implementation inside the given structure.
3. **Do not hardcode values**. Your functions should be generalizable to any dataset.
4. Ensure that your code is independent of the dataset schema and can work with different types of data.
5. You may write additional helper functions if needed.
6. **PyTorch** is required for this implementation. Please ensure your environment is set up with PyTorch before running the code.
7. After submission, your code will **also** be tested against hidden test cases.

Submission Guidelines

You need to submit the following files:

1. **Python solution:** CAMPUS_SECTION_SRN_Lab6.py
2. **Test case screenshot:** A screenshot showing the passing test cases in your terminal should be submitted as CAMPUS_SECTION_SRN_Lab6.png

Please submit these files via the provided Google form. **Ensure that you double-check the files** before submission, as resubmission is not allowed.

Important: If the naming convention is not followed, no marks will be allotted.