Applying Hidden Markov Models to My Capstone Project

As I delve deeper into my capstone project, "The Efficacy of Decision Driven Simulations in Developing Entrepreneurial Skills Among Liberian Secondary Students," I've been exploring various machine learning techniques that could enhance the intelligence and adaptability of the simulation platform. One such technique that holds significant promise is the Hidden Markov Model (HMM). I believe HMMs could be instrumental in understanding and modeling the progression of entrepreneurial skill development within the simulation, offering insights into student learning pathways and providing personalized feedback.

Observations: Measurable Data for the HMM

In the context of my decision-driven simulation, the observable data for an HMM would primarily consist of the discrete actions and performance metrics generated by students as they interact with the simulation. These observations would represent the outward manifestations of their underlying entrepreneurial skill development. Specifically, I envision the following measurable data points:

- **Decision Sequences:** The chronological series of choices a student makes within the simulation, such as pricing strategies, marketing campaign selections, investment decisions, and resource allocation. Each specific decision at a given step could be a unique observation.
- **Simulation Outcomes/Performance Metrics:** The immediate results or feedback provided by the simulation in response to a student's decisions. This could include changes in virtual company revenue, profit margins, customer satisfaction scores, market share, inventory levels, or even the success/failure of specific business ventures within the simulation. These numerical outcomes could be discretized into categories (e.g.,

Type of HMM Problem

Given that I am starting with a sequence of observations (student actions and performance metrics) but do not know the underlying entrepreneurial skill states, the primary HMM task I would be addressing is a **learning problem**. Specifically, I would be trying to learn the parameters of an HMM that best explains the observed data. This is often referred to as the "unsupervised learning" of an HMM, as I am not providing the model with pre-labeled hidden states. The goal is to uncover the hidden structure (the students' evolving skill levels) from the observable data.

Training Algorithm

To tackle this learning problem, I would employ the **Baum-Welch algorithm**, which is a specific instance of the Expectation-Maximization (EM) algorithm tailored for HMMs. The Baum-Welch algorithm is an iterative procedure that, given a set of observation sequences, estimates the HMM parameters that maximize the likelihood of observing that data.

a. What values are known at the start?

At the beginning of the training process, the primary known values would be the **sequences of observations** collected from each student's interaction with the simulation. For example, for each student, I would have a sequence like `O = (o1, o2, o3, ...)` where each `ot` is a specific decision or performance metric at a given time step. I would also need to pre-define the number of hidden states, which would correspond to the number of entrepreneurial skill levels I want to model (e.g., 'Novice', 'Developing', 'Proficient').

b. What values are unknown and need to be learned?

The core of the learning problem is to determine the unknown HMM parameters that best model the observed data. These parameters are:

- **Transition Probabilities (A):** The probability of transitioning from one hidden skill state to another (e.g., the probability of a student moving from a 'Developing' skill state to a 'Proficient' skill state after a certain number of successful decisions).

- **Emission Probabilities (B):** The probability of observing a particular action or performance metric given a specific hidden skill state (e.g., the probability of a student with a 'Proficient' skill level making a specific, optimal pricing decision).
- **Initial State Probabilities (π):** The probability that a student starts in a particular hidden skill state at the beginning of the simulation.

Parameter Updates

The Baum-Welch algorithm will iteratively update the HMM parameters (A, B, and π) to better fit the observed data. In each iteration, the algorithm performs two steps:

1. **Expectation (E-step):** Using the current HMM parameters, the algorithm calculates the expected number of times each state transition and emission occurs in the observed data.
2. **Maximization (M-step):** The algorithm re-estimates the HMM parameters based on the expectations calculated in the E-step. Specifically, it will update:
    - The **transition probabilities (A)** by dividing the expected number of transitions from one state to another by the total expected number of transitions out of the first state.
    - The **emission probabilities (B)** by dividing the expected number of times a particular observation is generated from a specific state by the total expected number of times that state occurs.
    - The **initial state probabilities (π)** by calculating the expected frequency of each state at the beginning of the observation sequences.

This iterative process continues until the HMM parameters converge, meaning they no longer change significantly between iterations. The resulting HMM will provide a model of how students transition between different entrepreneurial skill levels and how those skill levels manifest as observable actions and outcomes within the simulation.