Hidden Markov Model (HMM) Application in My Capstone Project

Capstone Title:
"Machine Learning-Powered Mobile Nutrition Assistant for Chronic Disease Management in Nairobi's Low-Income Communities"

1. Describe the Observations:

We would like to assist users in better monitoring their nutrition and chronic diseases in this project by utilizing a mobile assistant. In this case, a Hidden Markov Model (HMM) can be beneficial in detecting unseen health states-such as when a user is nutritionally stable, at risk, or in a critical stage-based on the information they input over time.
The measurable data include users' macronutrients, composition of their meals, biometric readings (e.g., weight, blood glucose), exercise diaries, and symptom reporting (e.g., weakness, dizziness). These are captured through app recording or wearables.

2. HMM Problem Type:

This is an unsupervised HMM problem as the underlying health states (e.g., "balanced," "at risk," "critical") are not given. The model must derive these latent states from patterns in observable sequences.
The app will monitor observable data like daily food, symptoms (e.g., weakness or dizziness), blood glucose levels, and exercise. These are observed by the model to look for patterns. We do not know the user's current health state at any time, so this is an unsupervised HMM problem.

3. Training Algorithm:

a. Known values:

- Time series of observable user logs
- Fixed pre-defined number of hidden states (identified using domain knowledge)

b. Values to be found:

- The actual hidden states

- Transition probabilities (the manner in which users switch from one health state to another)
- Emission probabilities (probability of observed symptoms given a state)

4. Updating parameters:

The model estimates the following utilizing the Baum-Welch algorithm:
- Initial state probabilities

- Transition matrix (A): probability of a move from one state to another

- Emission matrix (B): probability of observing a symptom/behavior under a state

The model starts off with the observations themselves. By using the Baum-Welch algorithm (HMM training procedure), it learns how likely a person is to move from one state of health to another (transition probabilities) and how likely it is to see a group of symptoms in a hidden state (emission probabilities).
Following training, the model updates three things: transition probabilities (e.g., from "at risk" to "critical"), emission probabilities (e.g., the likelihood of low energy given a poor nutrition day), and the probability of starting in each state of health.

By learning from each patient's data over time with HMMs, the system can identify patterns and warn users if their health seems to be deteriorating. It enables them to take precautions early-before they get worse. It's a feasible way of combining real-time data and predictive modeling to make management of chronic disease smarter and more personalized.