Part 2: Applying Hidden Markov Models (HMM) in My Capstone Project

---

Capstone Project Summary

My project aims to address inefficiencies in public transport systems in urban areas by developing a smart web/mobile platform. This platform will:

- Track bus locations using real-time and historical GPS data.
- Identify available parking slots.
- Recommend the fastest travel routes.

Observations (Measurable Data for HMM)

The observable inputs (emissions) for the HMM model in this system include:

- **Real-time GPS coordinates** of buses.
- **Timestamps** for location updates.
- **Estimated arrival times** at stops.
- **Traffic density data** from sensors or APIs.

These are all measurable and time-sequenced, making them ideal for modeling using HMMs.

Type of HMM Problem

This is a **Hidden State Inference Problem**, where:

- The observable data (locations, times, speeds) are known.
- The **hidden states** (e.g., "Traffic Flow State": Free, Moderate, Congested) are not directly known.

Thus, it is an **unsupervised learning problem**, where HMM helps uncover the underlying state sequences that explain the observed transit behavior.

Training the HMM

a. Known Values at Start:

- GPS position logs with timestamps.
- Initial transit data and rough labels (optional, if semi-supervised).

b. Unknown Values to Learn:

- Transition probabilities between hidden traffic states.
- Emission probabilities (likelihood of GPS data given a traffic state).
- Initial state distribution (π).

These are learned using the **Baum-Welch algorithm**, an Expectation-Maximization method.

Parameter Updates in HMM

The HMM training algorithm will iteratively update:

- **A (Transition Matrix):** Probability of transitioning from one traffic condition to another (e.g., from "Free" to "Congested").
- **B (Emission Matrix):** Probability of observing a certain GPS + timestamp pattern given a traffic state.
- **π (Initial Probabilities):** Likelihood of starting in each hidden state.
