

---

# Decoding Travel Purposes Through Sequential Probabilistic Models – Project Report

---

**Ajay Partha**  
apartha@ucsd.edu  
A17038162

**Albert Ding**  
axding@ucsd.edu  
A17007774

**Bhruhu Bharathi**  
bbharathi@ucsd.edu  
A16641798

**Yuandong Zhang**  
yuz371@ucsd.edu  
A69041602

## 1 Problem Description

Imagine you are a software designer looking to add quality of life improvements to the navigation systems found in vehicles and everyday devices. Technology feels like magic when it correctly identifies a user’s intent and does something the user wants without them having to be entirely explicit. Consider, then, the possibility of suggesting relevant destinations to a driver just based on the time they enter their car. Or retrieving an appropriate shopping list for a user who has just boarded a bus on their way to a store. Such applications motivated our project, which studies how a traveler’s choice of vehicle and the time of their trip can be used to predict what they are about to do, for example eat, shop, or work. Central to this model, is the assumption that the purpose of the traveler’s trip is never directly observed; instead, only the trip modes, like walking, driving, and biking, and their start time are measured. This choice is important: if we want our technology to anticipate what a user will do, we shouldn’t expect to already know this information at inference time. Thus, this problem takes a form amenable to Hidden Markov Models (HMMs), where trip purposes constitute an evolving latent sequence, and trip modes are their corresponding emissions.

Given a dataset which contains hundreds of thousands of paired trip mode and purpose trajectories measured within a single day, we set out to accomplish the following: (1) Train multiple Hidden Markov Models using Maximum Likelihood Estimation (MLE) on the observed training data. This split contains the true trip purpose labels, meaning we can directly estimate the initial, transition, and emission probabilities using normalized frequency counts. (2) Train non-HMM baselines, with less complicated modeling assumptions to serve as useful comparators. (3) Inference the trained Hidden Markov Models on a held-out test split of the data by applying the Viterbi algorithm, to identify the most likely sequence of latent purposes. (4) Evaluate the performance of the trained HMMs, by comparing the similarity between the inferred sequence of trip purposes and the true purposes. (5) Identify the best performing model along with the modeling assumptions that contributed to its stronger performance.

Our progress on this problem poses promising implications for personalized navigation and recommendation software, but could also be used in a broader context to inform transportation policies or infrastructure development.

The rest of our report will be structured as follows. In Section 2, we cover how our data was sourced and processed. In Section 3, we formalize our modeling assumptions. In Section 4, we outline all baseline methods, and in Section 5 we detail the different HMMs we developed. We discuss our results in Section 6 and we conclude in Sections 7, and 8. We include all auxiliary information in the Appendices.

## 2 Data Sourcing and Processing

Our data comes from the Microsoft Research Asia Geolife GPS Trajectory Dataset, which contains GPS traces from over 100 users in Beijing from April 2007 to August 2012.[Zheng et al., 2011]

The dataset shows GPS points (latitude and longitude) and pre-annotated trip metadata such as `trip_mode`, `start_purpose`, and `end_purpose`.

We undertook four main pre-processing steps: First, we grouped the data by `trip_id`, and sorted by timestamp to reconstruct full trip trajectories from GPS data points. This made obtaining ordered sequences for HMM training significantly easier. Secondly, we computed time differences between every pair of consecutive GPS points. Computing inter-point time deltas allowed us to validate trip continuity, detect gaps, and verify that a sequence of points corresponded to a continuous trip. Trips that had missing data or irregular sampling could distort our ordering which would then affect the state transitions in the HMM. Third, after reconstructing trips, we grouped by `user_id` and `trip_date` to obtain the users' daily trips, and we sorted the daily trips with respect to their timestamps to create properly ordered, labeled sequences. Finally, we filtered low-quality data by removing users with fewer than 10 total trips, daily sequences with fewer than 3 trips, and trips shorter than 5 minutes. This was important as sequences that are too short don't provide informative transitions for training an HMM and their inclusion would only contribute noise.

Our final dataset contains 1,158,825 entries. Each entry represents a single trip made by a user, containing a variable length list, where each element of the list contains the transportation mode, the trip purpose, and an index indicating its position within the user's daily sequence.

To train our HMMs, we split our dataset into a training set and a testing set. After shuffling the data, we randomly assigned 80% of users to the training set and 20% to the testing set. We ensured that individuals' data was not split between the training and testing sets; either they appeared in the training set or the testing set. This was done to prevent behavioral patterns from potentially leaking between the two sets, preserving the integrity of our methodology.

Below, we have plotted the normalized distributions of trip modes and trip purposes for the entire cleaned dataset.

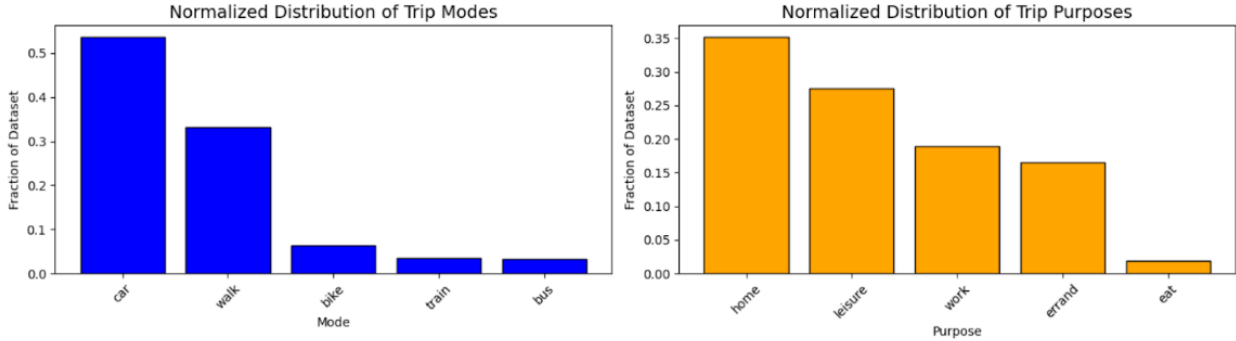


Figure 1: Normalized distributions of trip modes and trip purposes in the GeoLife dataset.

These histograms reveal a fundamental class imbalances for both the trip modes and purposes. Driving and walking dominate the trip mode distribution, ultimately making the input sequences less informative, and biasing the MLE procedure towards emitting `car` and `walk`. While the distribution of trip purposes is slightly more uniform, it is apparent that trips toward `home` appear almost as frequently as the bottom three purposes (`work`, `errand`, and `eat`) combined. As a result, MLE is susceptible to over-represent state transitions toward `home` and `leisure`.

Fortunately, the above doesn't tell the whole story because it omits *when* certain trips were taken. Thus, to inject more signal into the sequence of observed trip modes, we decided to granularize the mode bins according to when the trip was taken. We considered two schemes: mode plus the general time of day (e.g. `bus_afternoon`) and mode plus the hour of day (e.g. `bus_11am`).

Although the new distributions are still somewhat skewed, each observation is now more informative. Our results demonstrate that this change significantly improves the modeling performance of our HMMs, as it disambiguates the previously uniform transition and emission tendencies. Below, we have illustrated how this improved signal actually manifests. We use primary colors to denote individual modes and gradients to denote time of day. Here, decreased saturation implies a trip taken later in the day.

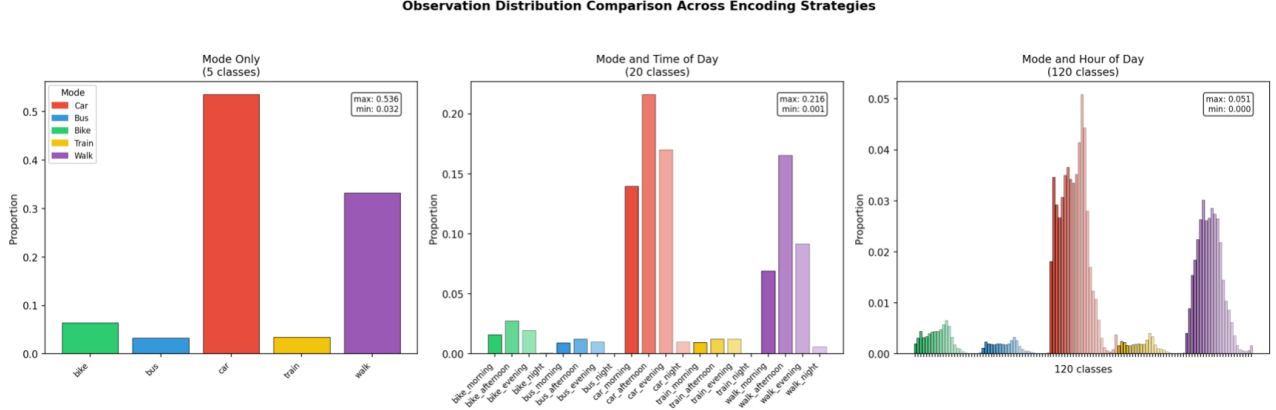


Figure 2: Different observation encoding schemes and their normalized class frequencies. On the left, we plot “Mode Only”, which doesn’t incorporate any temporal information. In the middle, we discretize each original trip mode into four more categories, morning, afternoon, evening, and night, and plot the results. This regime is “Mode and Time of Day”. On the right, we increase the total number of classes to 120 to capture trips taken at every possible hour of the day (24 hours \* 5 modes = 120 classes). This regime is “Mode and Hour of Day”.

Importantly, not all models utilize information from low-sample modes in the same way. As we will see, certain baselines completely ignore them.

### 3 First Order Hidden Markov Modeling and Inference

We leverage the standard formalization of first-order Hidden Markov Models, and include MLE formulas for the initial, transition, and emission probabilities in Appendix B. Importantly, we apply the Markov assumption: that  $P(Z_{t+1} | Z_{1:t}) = P(Z_{t+1} | Z_t)$ , where  $Z_t \in \{1, \dots, K\}$  represents the set of hidden trip purposes. Likewise, emissions only depend on the current trip purpose:  $P(O_t | Z_{1:t}, O_{1:t-1}) = P(O_t | Z_t)$ , where  $O_t \in \{1, \dots, M\}$  represents the set of observed trip modes.

During training, all of the hidden trip purposes are actually observable, so we can directly estimate the HMM parameters using MLE. During testing, we hide the true purposes, and apply the Viterbi decoding algorithm to return the most likely sequence of trip purposes.

### 4 Baselines

#### Home Only

For our first baseline, we use the imbalance of the data to our advantage and simply only predict the majority purpose: home.

#### Time-of-day

Our second baseline leverages the time of day to make a simple prediction about the trip purpose, with the rationale being that certain trip purposes follow predictable daily patterns. We use the following rules: Work (7–10 AM, 1–5 PM), Home (10 AM–12 PM), Eat (12–1 PM, 5–7 PM), and Leisure (7–10 PM). Trips assigned to the Other category are mapped to Home for the entire day.

This model does not depend on properties of the data distribution and serves as a useful lower bound.

#### Frequency-Based

The probability-based model counts the number of each purpose given an observation in the dataset to create a probability distribution of purposes for each observed state. So from the training set we compute:

$$P(\text{purpose} = p \mid \text{mode} = m)$$

Then for each observed mode, we predict the purpose with the highest conditional probability. This ignores temporal structure, treating all trips independently.

### Random Forest Baseline

This is a non-sequential machine learning baseline using a Random Forest classifier. Each trip is treated independently where the inputs include observed attributes such as mode of transportation, and the output label is the trip purpose. For each observed trip, we construct a input feature (mode, time) and use the purpose as the ground truth. The Random Forest learns a set of decision trees that separate feature space into sections corresponding to different purposes. At test time, for each observed mode, the Random Forest predicts a purpose by calculating votes across trees. This baseline is more flexible than the rule-based and pure frequency-based baselines. Comparing our HMM against this Random Forest baseline can help us understand the additional value coming from modeling temporal dependencies.

## 5 Other Hidden Markov Models

### 3-gram Hidden Markov Model

We extend our first-order HMM to a 3-gram (third-order) Markov structure over latent purposes. Here, each latent purpose now depends on the two preceding purposes, while emissions remain the same as in our first-order HMM:

$$P(Z_t | Z_{1:t-1}) = P(Z_t | Z_{t-1}, Z_{t-2}), \forall t \geq 3$$

The full definition and MLE derivation for the 3-gram HMM are provided in Appendix A.

### Edge-Emitting Hidden Markov Model

The standard state-emitting HMM assumes that each observation (mode) is emitted independently from the current hidden state (purpose), as illustrated in Fig. 3a.

However, this assumption may be overly restrictive, as transitions between purposes may have an influence comparable to or stronger than that of individual purposes alone.

To capture such transition-level behavioral patterns, we adopt an edge-emitting HMM [Rabiner, 1989], shown in Fig. 3b. Unlike the standard HMM, observations are emitted by transitions between consecutive latent states:

$$P(O_t | Z_{t-1}, Z_t).$$

To maintain a consistent formulation across time steps, we introduce a special start state  $Z_0 = \text{START}$ . The joint distribution then factorizes as

$$P(Z_{1:T}, O_{1:T}) = P(Z_1 | \text{START}) \prod_{t=2}^T P(Z_t | Z_{t-1}) \prod_{t=1}^T P(O_t | Z_{t-1}, Z_t).$$

The model parameters are

$A_{ij} = P(Z_t = j | Z_{t-1} = i)$  and  $B_{ijo} = P(O_t = o | Z_{t-1} = i, Z_t = j)$  and  $\pi_j = P(Z_1 = j | Z_0 = \text{START})$ .

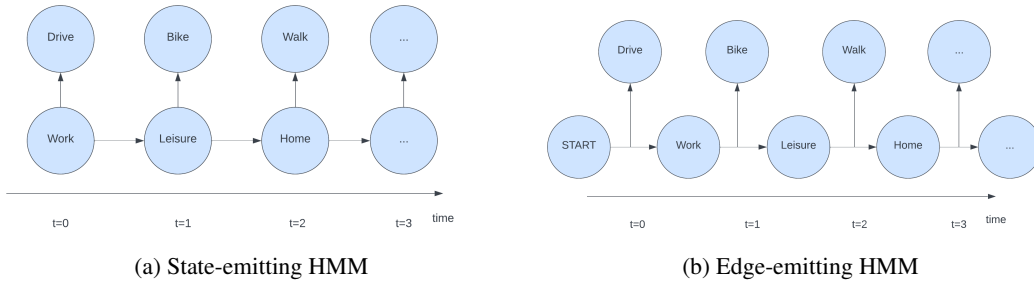


Figure 3: Comparison between state-emitting and edge-emitting Hidden Markov Models. In the state-emitting HMM, observations are generated by individual latent states, whereas in the edge-emitting HMM, observations are generated by transitions between consecutive latent states.

The training and prediction procedures follow Maximum Likelihood Estimation and Viterbi decoding with minor modifications, which are detailed in Appendix B.

## 6 Results and Discussion

We use a binary accuracy function to measure the model’s performance, where the predicted purpose is compared to the ground truth purpose with an indicator function. The total accuracy of a predicted sequence is computed element-wise and ranges from  $[0.0, 1.0]$ . The model’s total accuracy is averaged over the per-sequence accuracy on the held-out test set.

Category	Method	Accuracy
<b>Non-sequential baselines</b>	Home baseline	0.3522
	Time-of-day baseline	0.2092
	Frequency baseline	0.3449
	Random Forest	<b>0.4617</b>
<b>Sequential models (Mode Only)</b>	First-order HMM	0.3561
	3-gram HMM	0.3240
	Edge-Emitting HMM	<b>0.3950</b>
<b>Sequential models (Mode plus Hour of Day)</b>	First-order HMM	0.4706
	3-gram HMM	0.4510
	Edge-Emitting HMM	<b>0.4849</b>

Table 1: Accuracy comparison between non-sequential baselines and sequential models (before vs. after granular observation bins).

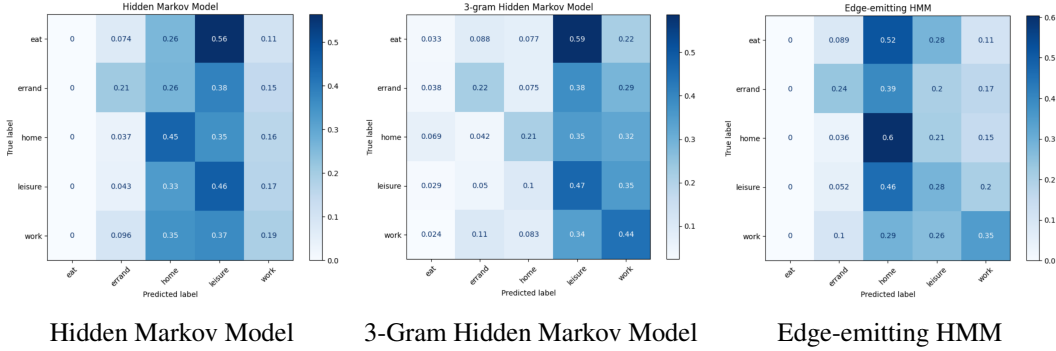
Among the non-sequential baselines, the Time-of-day baseline performs the worst, as it relies solely on arbitrary hard-coded rules with no trip-specific information. The Frequency baseline achieves an accuracy of 0.3449, which is comparable to the home-only baseline (0.3522). This similarity is largely due to the imbalance in the trip purpose classes: home is the dominant purpose among all trips and its count exceeds other purposes by a large margin. This is confirmed in the plot of the confusion matrices for both baselines as seen in Appendix D where both matrices distribute their probability masses similarly.

The Random Forest baseline achieves the highest accuracy among all non-sequential models (0.4617), which is not surprising given the strong empirical evidence that tree-based models are highly effective learners for tabular data [Grinsztajn et al., 2022]. By jointly exploiting travel mode and temporal features, Random Forests can capture rich point-wise associations between inputs and trip purposes. However, Random Forests operate on individual trips independently and do not model dependencies across consecutive trips. Thus, there is a natural limit to what they can effectively model and they eventually get outclassed by the Sequential models.

When using only the raw travel modes as observations, our HMMs barely exceed the performance of the baselines, and are outperformed by the Random Forest baseline. This indicates that the combination of class imbalance and weak input signal we observed in Section 2 thwarts the use of HMMs unless sufficient countermeasures are taken. We see evidence for this in the corresponding confusion matrices at the top of Figure 4, where the first-order HMM and 3-Gram HMM over-predict home and leisure, and the Edge-emitting HMM concentrates most of its predictions on home as well.

Fortunately, our strategy to inject more signal in the observed sequences appears to have the intended effect, as performance on the sequential models when time of day factors into the observation improves by around 10 percent and outscores all the baselines (from Table 1). The corresponding confusion matrices at the bottom of Figure 4 show a strong visible improvement: probability mass is shifted towards the diagonals, as fewer “blanket” predictions are being made. Still, there is room for improvement. The existing class imbalance prevents the first-order HMM from ever predicting eat, an infrequent trip purpose (something similar happens for the Edge-Emitting HMM in the Mode only regime). Additionally, the 3-Gram HMM and Edge-Emitting HMM rarely predict eat and leisure in the Mode plus Hour of day setting. We also report the performance of the Mode plus Time of Day setting in Appendix C, which is an improvement over the Mode only setting, but is outperformed by further discretizations which uses all 24-hour labels. This result also validates our hypothesis, that observations need to be sufficiently informative for our modeling assumptions to hold, otherwise we do no better than predicting the same label every time.

(a) Trip Mode Only

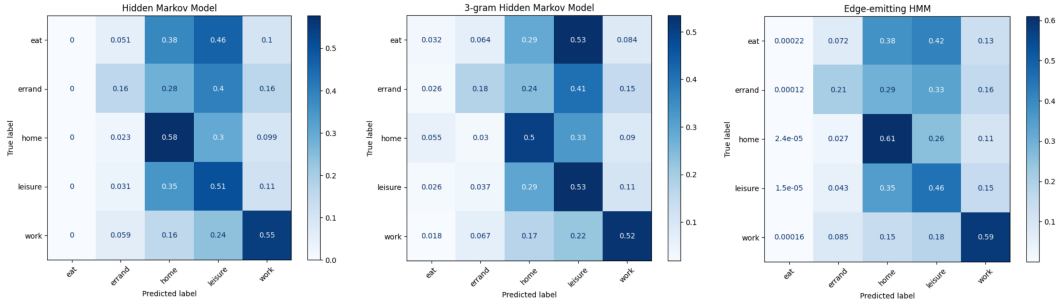


Hidden Markov Model

3-gram Hidden Markov Model

Edge-emitting HMM

(b) Trip Mode Plus Hour of Day



Hidden Markov Model

3-gram Hidden Markov Model

Edge-emitting HMM

Figure 4: Confusion matrices for the first-order HMM, 3-gram HMM, and edge-emitting HMM under two labeling granularities: (a) raw modes and (b) hourly aggregation. Each entry is normalized by row, indicating the proportion of predicted labels given the true label.

## 7 Conclusion

For this project, we tasked ourselves with decoding travelers' latent purposes utilizing sequential probabilistic models trained on the Geolife mobility dataset. The results show that although a sequential structure can offer modeling advantages, performance is ultimately led by the descriptive and balance of the observation space.

From all baselines, the Random Forest model performs the best with accuracy of 0.4617, showing its flexibility even though it doesn't pay attention to sequential dependencies. The initial HMM model that we created with raw trip modes did not perform as well as this baseline, with accuracies between 0.3240 and 0.3950. This shows that this model is insufficiently powerful, when observations are insufficiently informative. The decision to enrich observations by splitting each trip mode by the hour in the day helped our models thoroughly. The HMM variants all showed substantial improvement: the 3-gram model rose to 0.4510, first-order to 0.4706, and edge-emitting HMM to 0.4849. This adds validation to our theory that additional context is needed for uncovering latent behavioral patterns, when the observation data is highly skewed.

There are still limitations to our model. The Geolife dataset is very imbalanced, as seen in Figure 1. Due to this, our baseline and sequential models over-predict purposes such as home. On top of that, travel mode is not the strongest proxy of intent and the models assume clean trip segmentation, increasing potential noise.

Future facing work might leverage a more abundant feature space, including time-of-week labels, or geographic location, which would increase the signal in the observations. Higher-order HMMs that operate on state vectors with continuous elements could potentially capture high-dimensional, multi-step behavioral routines better. Another option could be applying neural sequence models which could allow for pattern discovery which goes beyond the constraints imposed by the Markov assumption.

In conclusion, our project shows that sequential probabilistic models with temporal encoding can find out patterns in mobility behavior. This project allowed our group to deeply understand Markov modeling, and provided a foundation for more mobility modeling in the future.

All code for data processing, model implementation, and experiments is available in the main branch of <https://github.com/Ydz0616/Trip-Purpose-Prediction>.

## 8 Reflections & Contributions

### Advice for Future Students

Suggestions we have for future teams is to start data preprocessing early, due to the fact the Geolife dataset is very large and noisy and needs considerable cleaning before it can be modeled. Histograms and/or confusion matrices for exploratory analysis is also valuable for understanding data imbalance and figuring out how to model (MLE vs. EM). Finally, testing the models on smaller subsets before scaling can help with efficient debugging.

### Team Contributions

**Ajay Partha:** Worked with conceptual and technical design of the modeling, including helping with outlining the code structure, modeling write up, and translating probabilistic modeling ideas into a clear implementation scheme.

**Albert Ding:** Contributed to prompt/idea formulation, code implementation report writing, and visualizations. Implemented Viterbi's algorithm, baselines, and 3-gram HMM.

**Bhruhu Bharathi:** I organized and scaffolded the code repository, wrote sections of the final report, and implemented the discretization scheme to include Time of Day and Hour of Day to improve the signal in the observations.

**Yuandong Zhang:** Contributed to idea formulation, code implementation (implemented data processing, baselines, edge-emitting HMM) and report writing (models, experiments).

### Individual Reflections

**Ajay:** The project really helped complete my understanding of sequential probabilistic models by connecting the complicated math of HMMs to real world sequences. Working with the Geolife dataset also highlighted the obstacles that come with real-world datasets. These obstacles being imbalances in the data, and some clear data outliers.

**Albert:** I learned a lot from this project and would say that my key takeaway has been the process of analyzing and manipulating data and formulating the models that would work best given that data. It gave me a deeper awareness of data quality, data imbalance, and the importance of preprocessing choices in shaping a model's performance. It also helped me better understand sequential approaches as well as their tradeoffs.

**Bhruhu:** This project reinforced that one cannot make modeling assumptions a priori of understanding the data. It also helped me learn how to combat class imbalance, by ensuring that every observation is sufficiently informative. I also gained that the Markov assumption might be a limiting factor in realistic modeling, and I feel motivated to explore better modeling assumptions.

**Yuandong:** The key takeaway for me is the importance of choosing a model that best captures the structure of the problem. In this project, I learned that human mobility is inherently sequential, making Markov-based models a natural fit. Their superior performance over rule-based and frequency-based approaches validated this choice. Moreover, our comparison between edge-emitting and state-emitting HMMs showed that modeling transitions between hidden states plays a more critical role in explaining the observations.

### Use of Generative AI

Our group utilized Gen AI to help structure our files and helping with a skeleton for our code. Before utilizing Gen AI, we knew how we wanted to implement our models but the use of Gen AI helped

us to get a skeleton of the files and function headers (after a few back and forth prompts, detailing what we wanted). The original prompt is in our GitHub under `prompt_plan.md`, and we formatted what we were given in `scaffolding_plan.md`. The Gen AI was just used for planning purposes and not implementing the functionality of the models.

## References

- Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on tabular data? *arXiv preprint arXiv:2207.08815*, 2022. doi: 10.48550/arXiv.2207.08815.
- Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989. doi: 10.1109/5.18626.
- Yu Zheng, Hao Fu, Xing Xie, Wei-Ying Ma, and Quannan Li. *GeoLife GPS Trajectory Dataset: User Guide*. Microsoft Research Asia, July 2011. URL <https://www.microsoft.com/en-us/research/publication/geolife-gps-trajectory-dataset-user-guide/>.

## A 3-Gram Hidden Markov Model Details

The initial distribution must assign probability to the first *pair* of latent purposes:

$$\pi_{ij} = P(Z_1 = i, Z_2 = j).$$

**Model parameters.** We define the parameters of the 3-gram HMM as:

$$A_{ijk} = P(Z_t = k \mid Z_{t-2} = i, Z_{t-1} = j), \quad B_{ko} = P(O_t = o \mid Z_t = k), \quad \pi_{ij} = P(Z_1 = i, Z_2 = j).$$

**Normalization constraints.**

$$\sum_{i,j} \pi_{ij} = 1, \quad \sum_{k=1}^K A_{ijk} = 1 \quad \forall (i, j), \quad \sum_{o=1}^M B_{ko} = 1 \quad \forall k.$$

**MLE estimation.** Training data contains fully observed purposes, so we may estimate all parameters via normalized frequency counts (no EM needed). Let

$$C_{ijk}^{(\text{tr})} = \#\{t \geq 3 : (Z_{t-2}, Z_{t-1}, Z_t) = (i, j, k)\},$$

$$C_{ko}^{(\text{em})} = \#\{t : Z_t = k, O_t = o\},$$

$$C_{ij}^{(\text{init})} = \#\{(Z_1, Z_2) = (i, j)\}.$$

Then the MLE updates are:

$$A_{ijk} \leftarrow \frac{C_{ijk}^{(\text{tr})}}{\sum_{k'} C_{ijk'}^{(\text{tr})}} \quad B_{ko} \leftarrow \frac{C_{ko}^{(\text{em})}}{\sum_{o'} C_{ko'}^{(\text{em})}} \quad \pi_{ij} \leftarrow \frac{C_{ij}^{(\text{init})}}{\sum_{i',j'} C_{i'j'}^{(\text{init})}}$$

In words:

- $A$  is estimated by counting how often each state-pair  $(i, j)$  transitions into a next state  $k$ , then normalizing over  $k$ .
- $B$  follows the same MLE rule as the first-order HMM, normalizing counts over observation symbols.
- $\pi$  is the empirical distribution of the initial latent-purpose pair  $(Z_1, Z_2)$ .

## B MLE Parameter Estimation for the HMM

Training data contains fully observed  $(Z_t, O_t)$  pairs, so we estimate parameters via normalized frequency counts.

### Hidden States and Observations

Let  $Z_t \in \{1, \dots, K\}$  denote the latent purpose at trip  $t$ , and  $O_t \in \{1, \dots, M\}$  denote the observed travel mode.

### Initial State Distribution

The initial state distribution  $\pi \in \mathbb{R}^K$  defines the probability that a day begins in purpose state  $i$ :

$$\pi_i = P(Z_1 = i).$$

Because the training data contains true purpose labels, we estimate  $\pi$  using normalized frequency counts:

$$\pi_i = \frac{\#\{\text{days whose first trip has purpose } i\}}{\#\{\text{total days}\}}.$$

### Transition Matrix

The transition matrix  $A \in \mathbb{R}^{K \times K}$  models purpose-to-purpose transitions:

$$A_{ij} = P(Z_{t+1} = j \mid Z_t = i).$$

The Maximum Likelihood Estimation (MLE) update is given by:

$$A_{ij} = \frac{\#\{Z_t = i, Z_{t+1} = j\}}{\#\{Z_t = i\}}.$$

### Emission Matrix

The emission matrix  $B \in \mathbb{R}^{K \times M}$  defines the probability of observing mode  $k$  when in purpose state  $i$ :

$$B_{ik} = P(O_t = k \mid Z_t = i).$$

The MLE estimate is:

$$B_{ik} = \frac{\#\{Z_t = i, O_t = k\}}{\#\{Z_t = i\}}.$$

## C Confusion Matrices: Time of Day

Model	Accuracy
Hidden Markov Model (HMM)	0.4398
3-Gram HMM	0.4168
Edge-Emitting HMM	0.4475

Table 2: Model accuracies under the time-of-day encoding.

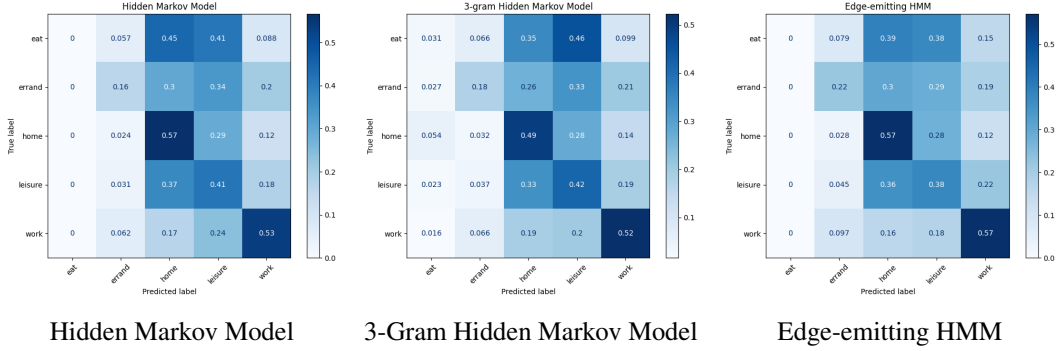


Figure 5: Confusion matrices for the first-order HMM, 3-gram HMM, and edge-emitting HMM (Trip Mode Plus Time of Day). Each entry is normalized by row, indicating the proportion of predicted labels given the true label.

## D Confusion Matrices: Baselines

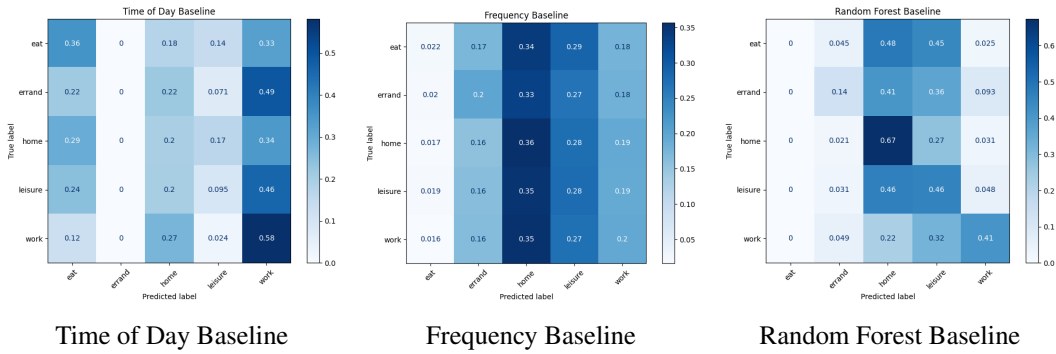


Figure 6: Confusion matrices for the time of day, frequency based, and random forest baselines. Each entry is normalized by row, indicating the proportion of predicted labels given the true label.