Human Mobility Pattern Prediction Algorithm using Mobile Device Location and Time Data

Myungjun Choi, Yonghyun Ro, Han Lee

0. Notations

N = number of states in the model
T = length of observation sequence (number of state transitions per day)
x_t = the state in which the user is at time t
y_t = the observation symbol observed at instant t
$\pi_i = P(x_o = i)$, the probability of being in state i at the beginning of the each sequence
$Q = \{ q_{ij} \}$, where $q_{ij} = P(x_{t+1} = j \mid x_t = i)$, the transition matrix, the probability of being in state j at time $t+1$ given that user was in state i at time t
$P(y_t \mid x_t = j)$, emission probability of observation given that user is in state j
μ_i = the mean vector of each state
$\sum$ = the covariance matrix of each state
$\lambda = (Q, \mu_i, \sum, \pi_i)$, our Hidden Markov Model

1. Motivation and Goal

Advent of smartphones enabled us to use GPS information for personal use. In addition, law enforcement are utilizing next-generation technology, and applications monitor known targets and identify suspects to predict their behavior and prevent attacks before they take place. In order to implement these applications, it is necessary to track the movements of mobile user in real-time or over a defined time period among mass location. Mobile user's mobility pattern is a Hidden Markov Model(HMM), because we do not get to observe the actual sequence of states(the location state), we can only observe the measurements(location coordinate data) generated by each state. Hence, we designed the algorithm to predict mobile user mobility pattern using machine learning algorithm and HMM. We relaxed data and our goal was to predict the best estimate of state sequence.

2. Model Description

2.1 Introduction of Model

Mobile user's mobility pattern is a Hidden Markov Model, because we do not get to observe the actual sequence of states (the location state), we can only observe the measurements (location coordinate data) generated by each state. Specifically, if we projected an uniform grid onto the region of interest like Figure 1A and each grid stands for each state, every single different colored points tell us the coordinate of position of mobile device through GPS or any other navigation aids, but in this region the values of the states are unobserved when we observed location measurement. That is, we do not know which grid contains the specific measurement before we label it.

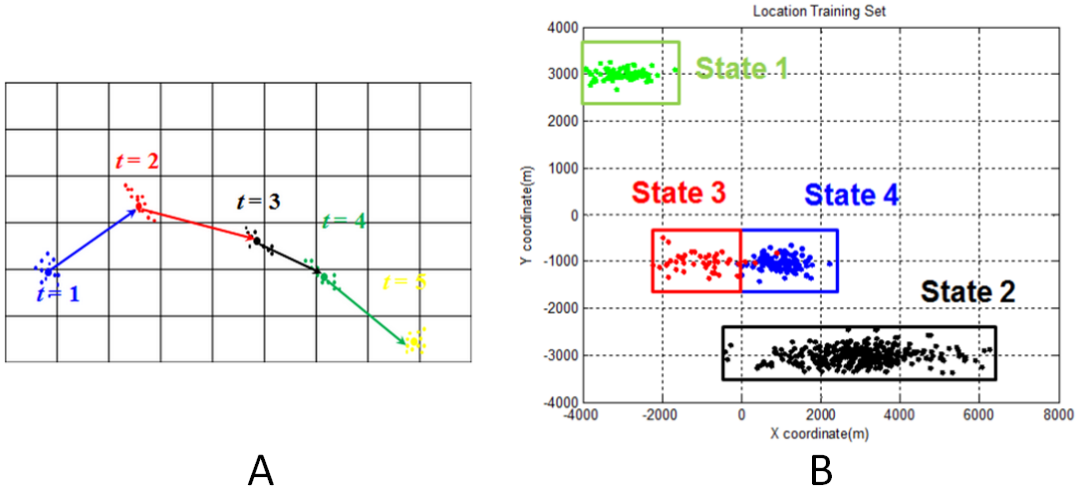


Figure 1. A. An example of human mobility pattern. Plot shows a sequence of location measured at different times, *t*, from 1 to 5. Measurements at each time follow Gaussian distribution. **B.** An example of 4 states, measured at 4 different time points represented by different colors. X, Y coordinates are shown on x-, y-axis. Each state could represent locations of home, work, library, shopping, etc.

Hence, if we measure 2 dimensional position data at every hour of a day for considerable days. we can have a series of observed measurements which follows Markov property without state assignment.

2.2 Model parameters for Hidden Markov Model

If we are given the data set, we have to specify which parameters we have to find based on Hidden Markov Model. As we mentioned earlier, human mobility pattern follows Markov property which the next state depends only on the current state and the equation is given below :

$$P(x_{t+1} | x_0, x_1, x_2, \ldots x_t) = P(x_{t+1} | x_t) \text{ where } x_t \text{ is the state at time } t$$

Since our purpose is to build the maximum likelihood mobility pattern parameter model in order to find the optimal state sequence and the probability of an observed sequence, we have to learn values for the Hidden Markov Model's parameters and apply that to the probability of an observed sequence. From figure 2, we can represent Hidden Markov Model parameters given as :

$$P(x_0, x_1, \ldots, x_T) = P(x_0) \prod_{t=1}^{T} P(x_t | x_{t-1}) P(y_t | x_t)$$

where y_t is the measurement at time t

$P(x_0, x_1, \ldots, x_T)$ is probability of series of state

$P(x_0)$ is initial state probability

$P(x_t | x_{t-1})$ is state transition probability

$P(y_t | x_t)$ is emission probability (measurement probability distribution at given state)

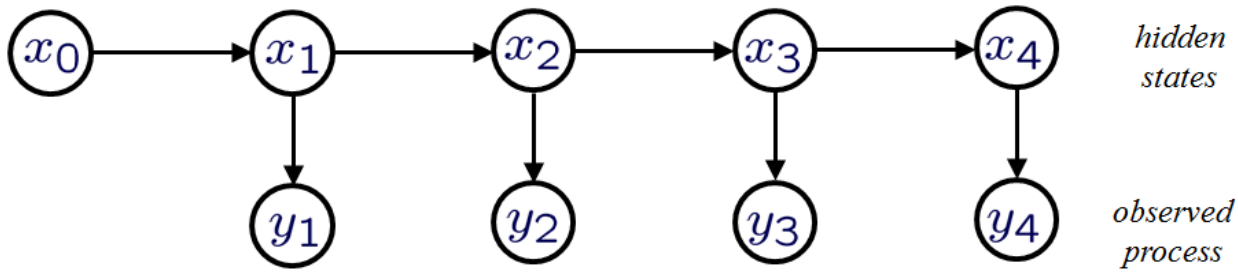


Figure 2

Especially, state transition probability can be extended to the state transition probability matrix given as figure 3 :

$$Q = \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{bmatrix}$$

$$q_{ij} \triangleq p(x_{t+1} = i \mid x_t = j)$$

Figure 3

Hence, we will learn the above parameters which are state transition probability matrix, initial probability and emission probability from the given data set and construct a human mobility pattern model.

2.3 Model Assumption

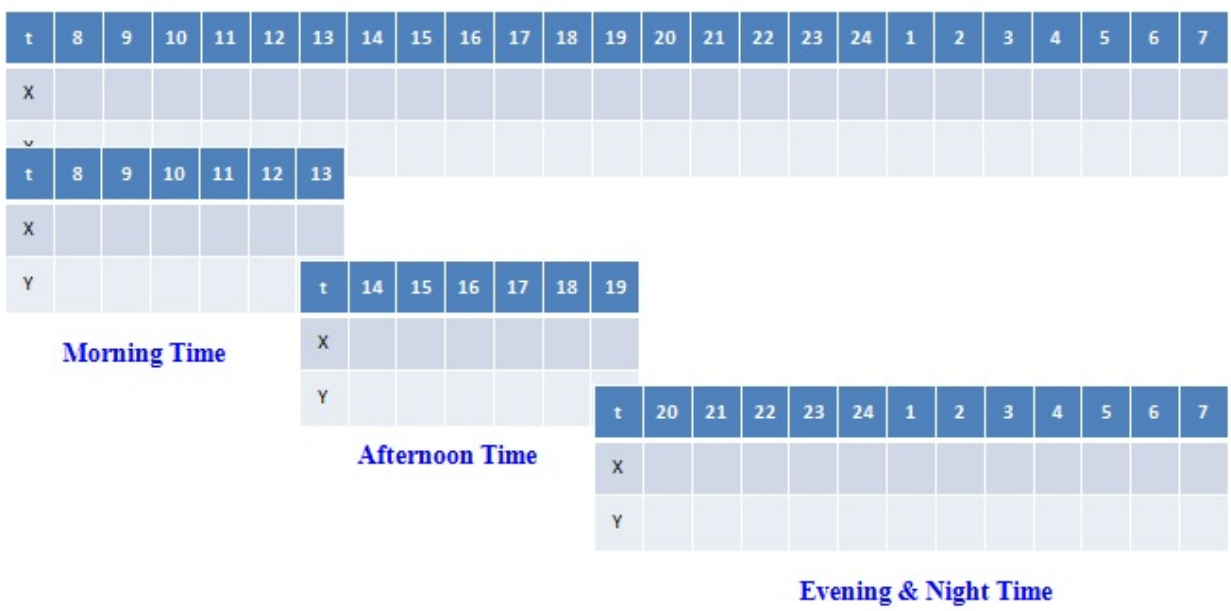
i) Mobility pattern is a continuous output Hidden Markov Model
ii) Human mobility pattern is not stationary so that we assume there are 3 different transition matrices and initial state probabilities with respect to time (see 3. Training Data Set Relaxation)
iii) The measurements drawn from each state is the Gaussian Distribution
iv) Time step is 1hr and measurements at same time of different day is time synchronized
v) The measurements are not a mixture Gaussian Model

2.4 Challenges

Practically, we have to collect measurement data set that corresponds exactly to our model and learn parameters and apply parameters to maximize optimal state sequence probability. There is a publicly released real measurement data set from Microsoft, the data set is stored in a relational database and we processed to extract data from db file using mkSQLite which is MATLAB based SQL query, but after looking up the data, we cannot use the data set in its form directly because not every movement recorded on each day occurred at exactly the same time. For example, the time that a person's GPS location gets recorded can occur at 8:00 on one day, and at 8:01 on another day, and so on. Therefore, we have to relax and modify the training data set to be able to use this data set with our model.

3. Training and Test Data Set Relaxation

Human's mobility pattern is not stationary, but time dependent. Specifically, people are usually stationary at home during sleeping time until they are heading for workplace. During day time, people are usually staying in their workplace. Sometimes, they go out for lunch or shopping, and they go to gym after work. Based on 3 divisions of human activity pattern, Hence, we split into 3 blocks of sequences specifically, based on general human activity, we divide 24 hours into 8am to 1pm with length of 6 , 2pm to 7pm with length of 6, 8pm to 7am with length of 12 (Figure 4). We used a training data set that is publicly released by Microsoft. However, each geographic data in the data set had a very granular time distribution in the unit of seconds. For the purpose of simplicity to be used in our model, we performed data set relaxation to an hour interval from the seconds. Hence, the movement of one user in a single hour will be funneled down into a single point. Total number of each block's sequence can be varied by the user. Therefore, we defined three different transition matrices Q with respect to time division. We have defined total of four states in this data set. From this data set we can compute three different initial probability π_i as well. Lastly, we also have to obtain mean of each state and corresponding covariance matrix because the measurement is two dimensional, so multivariate Gaussian distribution. We call this initial model $\lambda = (Q, \mu_i, \sum, \pi_i)$. After obtaining relaxed output measurements, we applied measurements to our mobility pattern prediction algorithm and adjusted result model λ^* which maximize the probability of state sequence and observation sequence, in order to validate that the algorithm is a good estimator of our reference model.



- Length of each sequence per day are 6, 6, 12
- Total number of each sequence is 300 days
- Total number of training set are 600, 600, 1200

Figure 4

Total data sequence is 300 days and 75% of data is training data set and 25% of data is test data set to validate the performance of the algorithm. While relaxing measurement data, we also

figured out the state variable simultaneously, in order to compare how many state sequence variable is differ from the state sequence variable produced by viterbi algorithm later.

4. Algorithm Implementation

The features of our training data set is 2 dimensional position coordinate and time. From this data set, we estimate all parameters we need for maximizing optimal sequence probability we want to know. Our algorithm starts from this point.

4.1 Algorithm Procedure

i) Given training data set, calculate the initial probabilities and transition probabilities

- The method to identify the hidden state of each measurements is using K-means algorithm.
- K-means algorithm clusters data points in order to make neighbor data points clustered same state variable
- The initial model($\widehat{\lambda}$) parameter equations are given below :

$$P(x_o) = \hat{\pi}_i = \frac{Number\ of\ occurences\ of \{y_0 \in i\}}{Total\ \ Number\ of\ occurences\ of\ y_0} \qquad i : \text{state}, 1 \leq i \leq 4$$

$$\hat{q}_{ij} = \frac{Number\ of\ occurences\ of \{y_t \in i, y_{t+1} \in j\} for\ all\ t}{Total\ \ Number\ of\ occurences\ of\ \{y_t \in i\} for\ all\ t}$$

$$\hat{\mu}_i = \frac{1}{N_i} \sum_{y_t \in i} y_t$$

$$\hat{\Sigma}_i = \frac{1}{N_i} \sum_{y_t \in i} (y_t - \hat{\mu}_i)^T (y_t - \hat{\mu}_i)$$

ii) Now we shall assume that an initial Hidden Markov Model is known from step i) and maximize $P(x_0, x_1, ..., x_T; \widehat{\lambda})$, $P(y, y_1, ..., y_T; \widehat{\lambda})$ by adjusting the parameters of $\widehat{\lambda} = (\widehat{Q}, \widehat{\mu_i}, \widehat{\Sigma}, \widehat{\pi_i})$ through EM(Expectation and Maximization) algorithm.

- This EM phase uses dynamic programming which is forward-backward procedure and the Baum-Welch re-estimation formula [1], [2]
- This maximization criterion is called maximum likelihood criterion and the function $P(y, y_1, ..., y_T; \widehat{\lambda})$ is called likelihood function.
- Baum-welch re-estimation Formula is given below :

- *Forward – Backward Procedure*

$\alpha_t(i) = P(y_1 y_2 \cdots y_T, x_t = i; \hat{\lambda})$: prob. of the partial observation sequence up to time t and the state i at time t **(Forward procedure)**

$\beta_t(i) = P(y_{t+1} y_{t+2} \cdots y_T \mid x_t = i; \hat{\lambda})$: prob. of the observation sequence from t+1 to T given the state i at time t **(Backward procedure)**

$$P(\bar{y}; \hat{\lambda}) = \sum_{i=1}^{N} \alpha_T(i)$$

$$\gamma_t(i) = P(x_t = i \mid y_1 y_2 \cdots y_T; \hat{\lambda}) = \frac{P(x_t = i, y_1 y_2 \cdots y_T; \hat{\lambda})}{P(y_1 y_2 \cdots y_T; \hat{\lambda})} = \frac{\alpha_t(i)\beta_t(i)}{P(y_1 y_2 \cdots y_T; \hat{\lambda})}$$

: prob. of being in state i at time t given the observation sequence **(Forward-Backward procedure)**

$\xi_t(i,j) = P(x_t = i, x_{t+1} = j \mid y_1 y_2 \cdots y_T; \hat{\lambda})$: prob. of being in state i at time t and making a transition to state j at time t+1

$$= \frac{P(x_t = i, x_{t+1} = j \mid y_1 y_2 \cdots y_T; \hat{\lambda})}{P(y_1 y_2 \cdots y_T; \hat{\lambda})}$$

$$= \frac{\alpha_t(i) q_{ij} b_j(y_{t+1}) \beta_{t+1}(j)}{P(y_1 y_2 \cdots y_T; \hat{\lambda})}$$

$$\sum_{t=1}^{T} \gamma_t(i) = \textit{Expected number of transitions from state } i$$

$$\sum_{t=1}^{T-1} \xi_t(i,j) = \textit{Expected number of transitions from state } i \textit{ to state } j$$

- The Baum-Welch re-estimation Formula for parameters
  - *Initial State Probability*: $\hat{\pi}_i = \gamma_0(i)$
  - *Transition matrix*:

$$\hat{q}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

  - *Emission Statistics*: $\hat{b}_i(y_t) \sim N(y_t, \hat{\mu}_i, \hat{\Sigma}_i)$

$$\hat{\mu}_i = \frac{\sum_{t=1}^{T} \gamma_t(i) y_t}{\sum_{t=1}^{T} \gamma_t(i)} \qquad \hat{\Sigma}_i = \frac{\sum_{t=1}^{T} \gamma_t(i)(y_t - \hat{\mu}_i)(y_t - \hat{\mu}_i)^T}{\sum_{t=1}^{T} \gamma_t(i)}$$

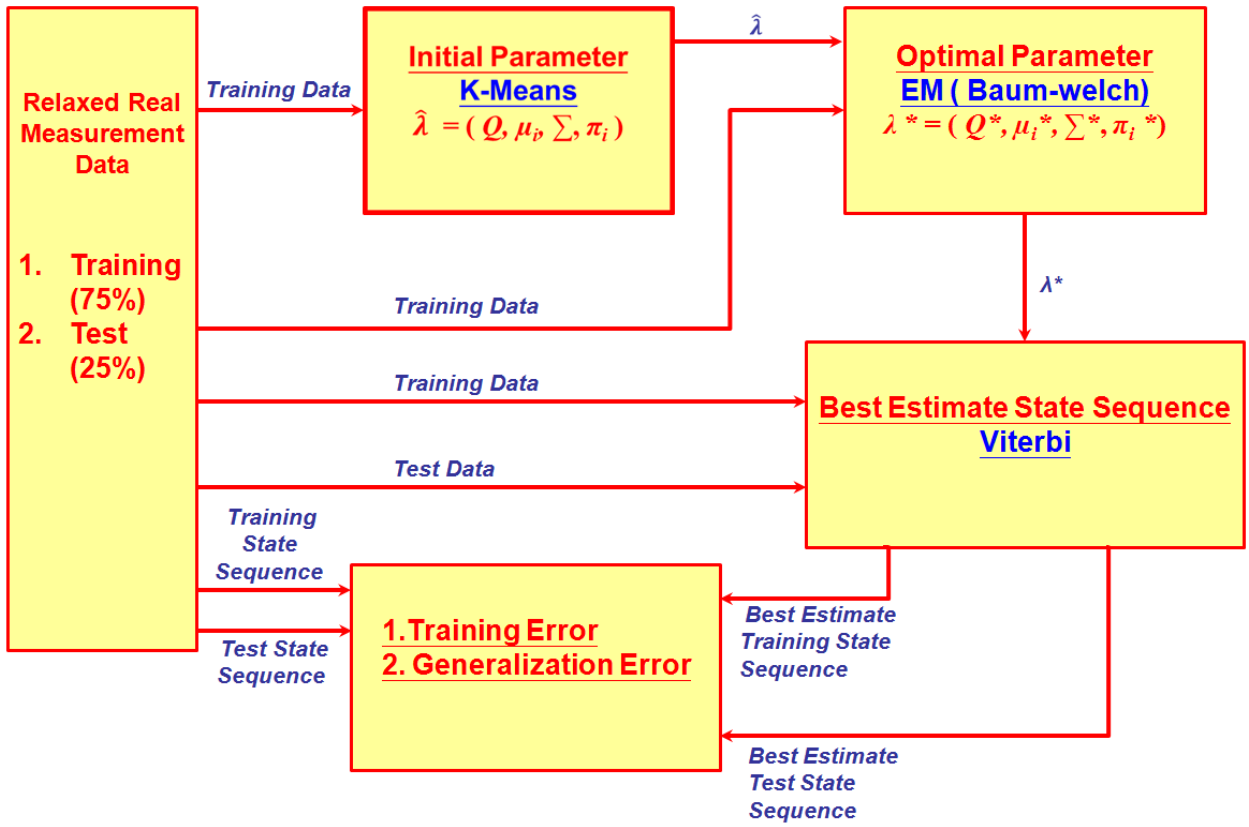
iii) algorithm iterates until log likelihood function will converges to get optimal parameters $\lambda*$ such that $P(y, y_1, ..., y_T; \lambda*) \geq P(y, y_1, ..., y_T; \widehat{\lambda})$

iv) With the optimal parameters of model $\lambda*$, we can verify that $\lambda*$ is the best estimate of the reference model we defined.

- We applied the training data set and the test data set which are only location and time data with optimal parameters to the viterbi algorithm which produces the best estimate state sequence $\lambda*$

v) Evaluate the error rate of by comparing the state variable sequence produced by viterbi algorithm and the state variable sequence of relaxed real measurement data.

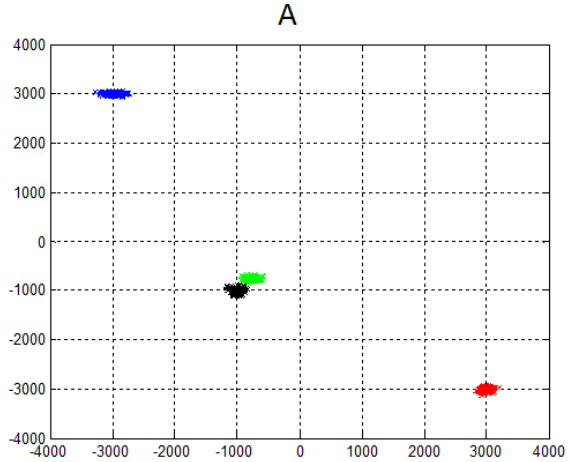
- We calculate training data set error and test data set error.



5. Result and Analysis

Our results are shown below in the Figure 6. Figure 6A shows 4 states for 8am to 1pm state sequence found by the K-means algorithm, and Figure 6B shows the error rate under these 4 states. Error rates are well below 1%, which indicates that less than 5 out of 450 sequence resulted in error in the time slot in 8am to 1pm and 2pm to 7pm, and approximately 9 out of 900 error in 8pm to 7am.

A



B

8am to 1pm	2pm to 7pm	8pm to 7am
0.1 (%)	0.52 (%)	0.11 (%)

8am to 1pm	2pm to 7pm	8pm to 7am
0.22 (%)	0.44 (%)	0.11 (%)

Figure 6. A. Measurements and States of Data **B.** Error Rates. Top is the error rate for training data, and bottom table is the error rate for test data.

6. Reference

[1] A tutorial on hidden markov models,
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.27.3772&rep=rep1&type=pdf

[2] Hidden Markov Model(HMM) Toolbox for Matlab written by Kevin Murphy
http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html