DESTINATION POINT ESTIMATION: A BAYESIAN INFERENCE BASED APPROACH

Rami Abousleiman
Electrical and Computer Engineering
Oakland University
Rochester Hill, MI, USA
email: abousleiman.rami@gmail.com

Romi Boimer
Amazon Web Services
East Palo Alto, CA, USA
email: br.romi@gmail.com

ABSTRACT
Estimating the destination point or the intended travel destination is imperative for an improved passenger-vehicle experience. For example, this feature helps in improving power train efficiencies and electric range estimations. It can also help with focused content through targeted ads, or assist with road hazard alerts that enable preemptive situation mitigation. In this work, we present a Bayesian Inference based approach that learns from driving history. The model utilizes only few features such as time and current location to predict the destination point. To mitigate the passengers' behavioral changes, we add an evaporation rate factor so that older recordings have less influence on the model than newer ones. The evaporation rate also allows for a more evolved model that avoids getting stuck in local minimums. Results show that with less than 200 recordings we are able to successfully predict more than 85% of the travel destination points even with the incorporation of passengers' habitual changes.

KEY WORDS
Bayesian Inference, Electric Vehicles, Optimization, Routing

1 Introduction

The increased adaptation of vehicle connectivity Siegel et al. (2018) allowed for the addition of numerous features that were not possible few years back. Passenger and driver expectations are increasing and are demanding more customizable and adaptable mobility services. Determining the travel destination is an important implementation that has numerous positive implications on many vehicle systems. In this work, we will present a Bayesian inference based approach for determining the travel destination, we will illustrate the developed model on a toy example and then expand the results and apply it on real collected data to test the validity of the results. Modeling the problem as a Markov Chain Model Ames (1989) the following are assumed:

1. Independent of time i.e. time invariant.

2. Ergodic chain i.e. irreducible.

3. Absorbed i.e. once a location is reached we can't leave it anymore. From a practical perspective this means that the prediction criterium is complete.

Common Machine Learning (ML) methods require a large amount of data to perform well, in our example as well as in many others this might not be possible. Models built on small data generally suffer from over-fitting. Furthermore, outliers and high leverage points can induce big problems on the developed models. Li et al. (2018) presented a review on several learning methods dealing with small data. The authors described the behavior of different ML methods when applied to small data. Support Vector Machines, K-Nearest Neighborhoods, Neural Networks, and Bayesian methods are some of the examples that the authors evaluated on small data. Some practical examples of work done on small data covers topics in biomedical engineering such as the work done by Shaikhina & Khovanova (2017). Shaikhina mitigated the abundance of data by evaluating a large set of models. These models where in the excess of 2000 but where able to perform well and produce acceptable results when compared to models dealing with larger data.

The rest of this paper is organized as follows: the next section will present a toy example and elaborate on the features recorded to establish the problem. Section 3, will discuss feature characterization, Section 4 will introduce the evaporation rate and describe its implications. Section 5 will show the results and discuss them. Finally, we will end the paper with suggested future work and a conclusion.

2 Problem Formulation and Bayesian Inference Representation

Trip travel destination is a random and stochastic process that is highly dependent on uncertain factors aggravated with the uncertainty of the human behavior. Predicting the travel point estimation thus seems to be an impossible task to accomplish 100% of the time. This is not the goal of this work. Instead, we are looking for trends and habits in the vehicle travel patterns that will allow us to establish a linkage that increases predictability. To establish this linkage, few features are tracked and recorded. These features are listed as follows:

1. Time of the day

2. Current location
3. Holiday or Weekend
4. Day of the week
5. Number of people sitting in the car

Figure 1 shows a simple toy example, that represents three habitual travel locations defined as: 'home', 'work', 'gym' and one generic location named as 'other' that defines every other location that cannot be determined or estimated. The probability of traveling from one location to another is defined as shown in Table 1. The goal of this work is to estimate these probabilities given the set of features listed eariler. Each feature is defined as X_i and the said features are defined in the same sequence listed above. For example, 'Time of the day' is feature X_1, 'Holiday or Weekend' is feature X_3, etc. The probability of heading to work given that the vehicle is at home is then expressed as shown in equation 1.

$$P_1 = P(W^{[t]}|X_1^{[t-1]}, X_2^{[t-1]}, X_3^{[t-1]}, X_4^{[t-1]}) \quad (1)$$

Equation 1 defines the probability of being at work at time $[t]$ given that all the events defined by $X_n \forall n \in [1, 6]$ have happened at time $[t-1]$. The same logic applies in calculating the other probabilities $P_2 \to P_{12}$.

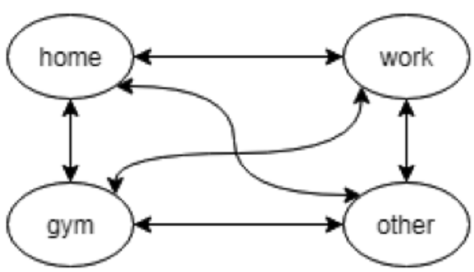

Figure 1: Simple Toy Example

Table 1: Travel probability names

TRAVEL PATH	PROBABILITY
HOME → WORK	P_1
HOME → GYM	P_5
HOME → OTHER	P_{12}
WORK → HOME	P_2
WORK → GYM	P_8
WORK → OTHER	P_{10}
GYM → OTHER	P_6
GYM → HOME	P_4
GYM → WORK	P_9
OTHER → WORK	P_{11}
OTHER → GYM	P_7
OTHER → HOME	P_3

Table 2: Sample training data for toy example

ID	START	END	DAY	TIME
1	WORK	HOME	MONDAY	7:00 PM
2	WORK	GYM	MONDAY	7:00 PM
3	WORK	HOME	MONDAY	6:00 PM
4	WORK	OTHER	MONDAY	7:00 PM
5	WORK	HOME	TUESDAY	7:00 PM
6	WORK	GYM	MONDAY	6:00 PM
7	WORK	HOME	MONDAY	7:00 PM

The joint probability $P(X_1, X_2, X_3, X_4)$ can be represented as shown in Equation 2 because these events can be assumed statistically independent. h is the number of features, assumed to be 4 in the toy example.

$$P(X_1, ..., X_h) = \prod_{n=1}^{h} P(X_n) \quad (2)$$

To use the available data, a generalized formula can be used to calculate the transition matrix factor, this can be expressed as shown in Equation 3 $\forall k \in [1, 3]$. Where 3 is the total number of nodes that are used minus 1. It is observed that the $P(X_n)$ is constant $\forall k$ and thus it can be dropped from the equation. Equation 3 can then be written as shown in Equation 4.

$$\omega_k = \prod_{n=1}^{h} (Destination|X_n) P(X_n) \quad (3)$$

$$\omega_k^* = \prod_{n=1}^{h} (Destination|X_n) \quad (4)$$

As an example, let's consider the below toy example, with the assumed training data shown in Table 2. For simplicity, we will only be using two features: 'Time of the Day' and 'Day of the Week'.

Given that we have the above history as shown in Table 2. We can then calculate the transition matrix factor for going 'home', 'gym', or 'other' using Equation 4.

2.1 Example 1: {Monday, 7 pm}

$\omega^*(home) = P(home|Monday) \times P(home|7pm)$
$\omega^*(gym) = P(gym|Monday) \times P(gym|7pm)$
$\omega^*(other) = P(other|Monday) \times P(other|7pm)$
Therefore:
$\omega^*(home) = 3/6 \times 3/5 = 3/10$
$\omega^*(gym) = 2/6 \times 1/5 = 1/15$
$\omega^*(other) = 1/6 \times 1/5 = 1/30$

This means that given it's a Monday at 7:00pm, the probability of going 'home' is 4.5 more likely than that of going to the 'gym' and 9 times more likely than going to an undetermined location. Using the same approach, the following calculations can be done as well.

Table 3: Data type for each of the features

Feature	Data Type
Time of day	quantitative
Current location	qualitative
Holiday or Weekend	qualitative
Day	qualitative
Number of people sitting in the car	qualitative

Table 4: Extended sample training data for toy example

ID	Start	End	Day	Time
1	work	home	Monday	7:00 pm
⋮	⋮	⋮	⋮	⋮
8	work	home	Monday	6:11 pm
9	work	other	Monday	7:07 pm

2.2 Example 2: {Monday, 6 pm}

$\omega^*(home) = 3/6 \times 1/2 = 1/4$
$\omega^*(gym) = 2/6 \times 1/2 = 1/6$
$\omega^*(other) = 1/6 \times 0/2 = 0$

2.3 Example 3: {Tuesday, 7 pm}

$\omega^*(home) = 1/1 \times 3/5 = 3/5$
$\omega^*(gym) = \omega(other) = 0$

3 Feature Characterization

The four features listed in Section 2 need to be characterized as either qualitative or quantitative. This is required as different approaches must be developed for each data type. Table 3 shows the data types for each of the features.

The approach used in Section 2 works well for qualitative features but will fail to generate reasonable numbers for quantitative features. To solve this, two approaches can be implemented:

1. Split the quantitative features into segments. For example, split the time into 15-minute intervals so we will end up with 96 segments of time.

2. Rearrange the recorded quantitative features as a Gaussian distribution with the mean being the recorded value of the data with a tuneable variance. For example, a recording at 6pm will be expressed as a Gaussian distribution with mean 18 and a tuneable variance.

Both approaches require tuning a parameter, the first requires tuning the size of the segment, the later requires adjusting the variance of the Gaussian plot. To illustrate the two methods, let's again consider the toy example used in Table 2 but with two added measurements as shown in Table 4.

Using approach one, the 'Time' for Drive ID 8 and 9 can be bucketed as 6:15 pm and 7:00 pm respectively. On the other hand, using approach two, all the time column values will have to change and will then be represented as shown in Figure 2.

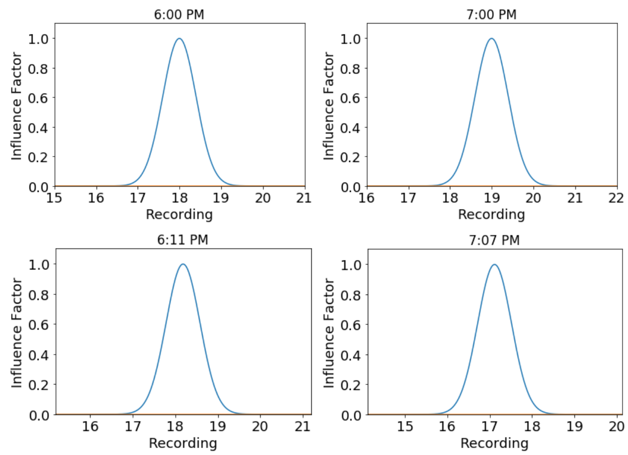

Figure 2: Gaussian Distribution of the 'Time' feature values

The probability P at any time t is then calculated as shown in Equation 5

$$P(t) = \frac{1}{\sigma}\sum_{i=1}^{\sigma} Y_i(t) \tag{5}$$

Where σ is the total number of recorded data points. Y is the Gaussian distribution value at time t. As an example, the conditional probability of heading home given that it is some time t is then calculated as shown in Equation 6.

$$P(home|t) = \frac{1}{\mu}\sum_{i=1}^{\mu} Y_i(t) \tag{6}$$

Where μ is the total number of recorded data points that show 'home' as a destination point irrespective of time. Again Y is the Gaussian distribution value at time t.

4 Evaporation Rate

The evaporation rate is inspired by the Ant Colony Optimization (ACO) method Dorigo & Birattari (2017) where traditionally it was used to help prevent local minimum solutions Maniezzo et al. (1991). The same idea will be used here by which older recorded data will have a lower effect on the calculations. The diminishing effect of older data is proportional to the evaporation coefficient (ρ). The evaporation coefficient in this example is expressed as shown in Equation 7. ρ is the evaporation coefficient and is a tuneable parameter that is bounded between $[0, 1]$. ρ ultimately controls how fast you want to forget or 'evaporate' old data. $\rho = 0$ means that the influence factor is always 1 and no evaporation is assumed, while $\rho = 1$ means that evaporation is extreme in such a way that no data is recorded. k is the vector representing the number of recordings between 0 and the current recording. Equation 7 is then bounded by 0 and 1 and is shown in Figure 3. Figure 3, shows that as the number of recordings increase, the evaporation rate decreases for older data until it completely diminishes. In this example, given a 600 recording sample and an evaporation coefficient $\rho = 0.0019$, the recorded data influence diminishes after 364 recordings.

$$er = -exp^{\rho k} + 2 \tag{7}$$

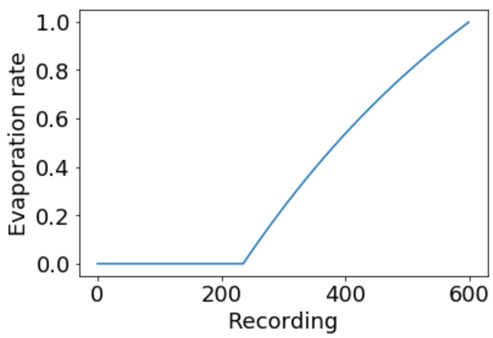

Figure 3: Evaporation Rate of Recorded Data

Evaporation rate is critical for the algorithm to work given that people change their behavior over time. This implies that that algorithm presented in this work will be able to adapt with the changing habits of drivers. Introducing the evaporation rate has effects on the equations presented earlier. One way to approach this is to drop older data, in the case of $\rho = 0.0019$ everything older than 364$^{\text{th}}$ recording is dropped from the model. To put more emphasis on newer data we weight the newer data in a way that is correlated to equation 7. For example, given 600 recordings, we first drop the oldest 236 recordings. This means that the Markov Model needs to be generated using the remaining 364 readings with the newest data having more influence on the updated model than the oldest ones. To quantify this influence, Equation 7 is used to weight recordings such that newer recordings are weighed more aggressively than older recordings. The process can be expressed as shown in Equation 8, where `fix` is a function that rounds down to the nearest integer, and the `numRecordings` is the total number of recordings used to generate the model (364 in this case). Figure 4 shows the plot of Z.

$$Z = \texttt{fix}(er \times \texttt{numRecordings}) \tag{8}$$

Now that the data has been weighted, there is no need to modify Equations 1, 2, and 3 and they can be used as is. For the toy example we will set $\rho = 0.02$. The weighing factor can then be expressed as shown in Figure 5. This means that the first four recordings are weighted seven times while the last four recording are weighted eight times more

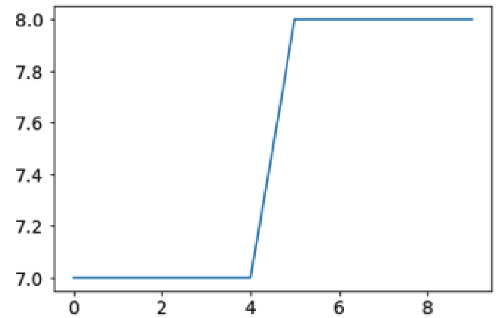

Figure 4: Weighing Factors for the Toy Example

and thus have more influence when the transition matrix is calculated. A summary of the proposed algorithm is shown in Algorithm 1.

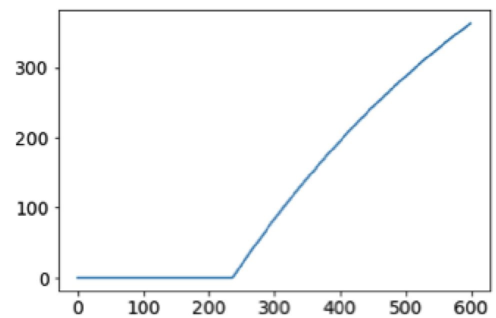

Figure 5: Weighing Factors for the Recorded Data

5 Results and Discussions

The developed model was tested on 50 data points with a pre-known destination location. The trained model was fed the data minus the results. Once the model estimated the results, these results are then compared to the pre-known destination points. This gave us a reference in order to quantify the performance of the model. Figure 6 shows a flowchart describing how the model was tested on the test data. The logic starts by only one test drive from the train data, the model is then generated using that one drive, and then tested on all of the 50 test data points. The logic proceeds by generating a model based on 2 test drives and then again test the model on all 50 test data points. The process is repeated until all of the test drives from the train data are embodied in the model. This process allows us to see the progression of the model along the fed data. The generated results are shown in Figures 7 through 10. These figures represent the same train and test data but with different tuning parameters. Figure 7 shows the results with $\rho = 0$, which indicates a zero-evaporation coefficient. t is set to 30 where t is the feature characterization tuneable parameter

Procedure 1 Destination Point Estimation

Input: driving data, test data
Output: estimated destination point
initialization define tuneable parameters and set all transition values to zero
data cleaning delete data points older than x readings where x is calculated using the evaporation coefficient of equation (7)
data scaling scale the training data based on equation (8)

repeat
capture the inputs listed in Table (2)
calculate the transition matrix using equation (4)
update transition matrix
until looped through the recorded test drives from 1 to k
generate probability array for the test data destination points
return destination point with highest probability

as described in Section 3. Specifically, this means that the time vector is split into 48 classes with 30 minutes each. Figure 8 shows the same t but with evaporation rate enabled because ρ is set to 0.0032. The performance of Figure 8 is significant over the ones shown in Figure 7. The same can be concluded from observing Figures 9 and 10. Figure 10, shows the best performance when $t = 240$ and $\rho = 0.0032$. As future work, it will be desired for a search method to be implemented to determine the optimal values of ρ and t that result in the highest success rate.

This algorithms was tested using Python 3.6 running a 3 GHz Intel Core i7 on macOS Mojave. Average test results took approximately 430ms per test datapoint. This implies we can run the logic in an onboard embedded system that is capable of supporting runtime applications.

6 Future Work and Conclusions

In this paper, we presented a novel Bayesian Inference based approach that was used to generate a prediction model for the destination point. Driving records were used as training data to establish the transition matrix factors that are key to the destination point estimation success. Although, the number of driving data records were insignificant when compared to traditional machine learning algorithms, the presented logic was able to converge fast and provide good results with more than 50% and 80% success rates at 25 and 180 data points respectively. The presented algorithm, can be further improved by optimizing the evaporation coefficient in reference to the total number of recordings captured. Evaporation coefficient can also be optimized in a way to maximize the success rate given the change in data points due to behavioral changes. Another way to optimized the presented method is to optimize the process of converting quantitative features into qualitative. In our example, time was used as a feature and the split was tested at 15 and

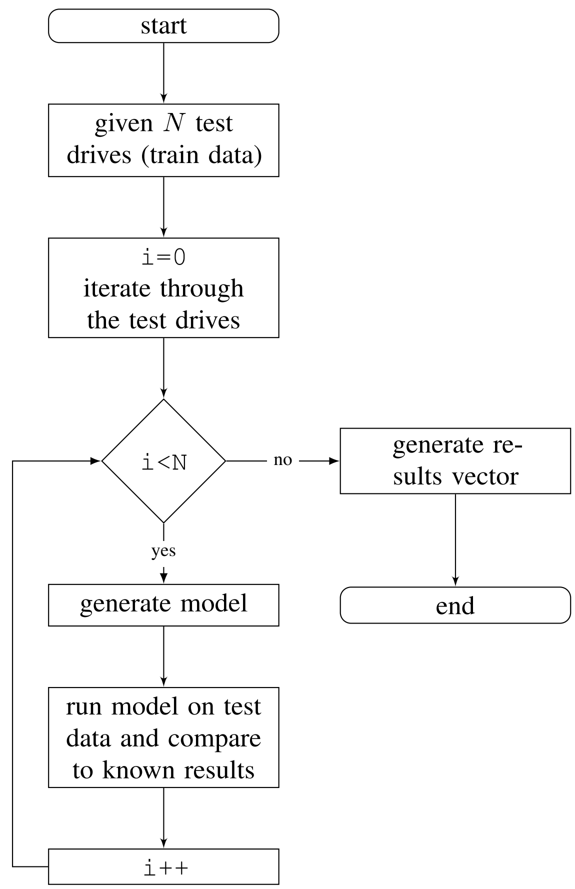

Figure 6: Model Testing Workflow

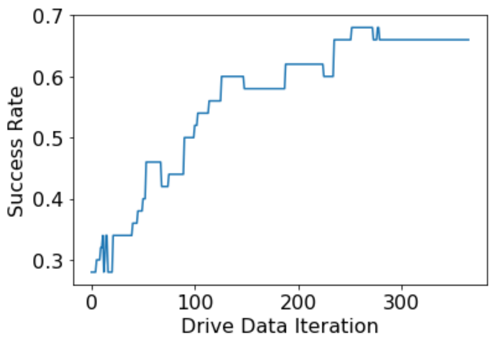

Figure 7: $\rho = 0.0, t = 30$

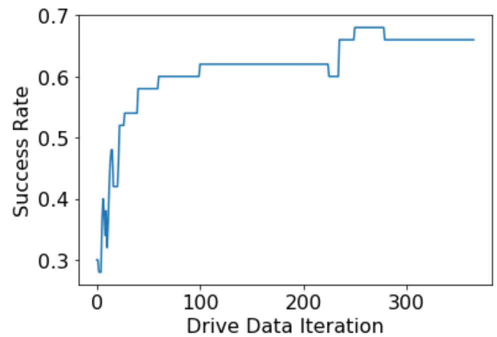

Figure 9: $\rho = 0.0, t = 240$

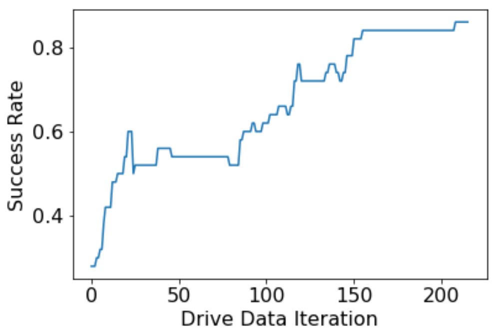

Figure 8: $\rho = 0.0032, t = 30$

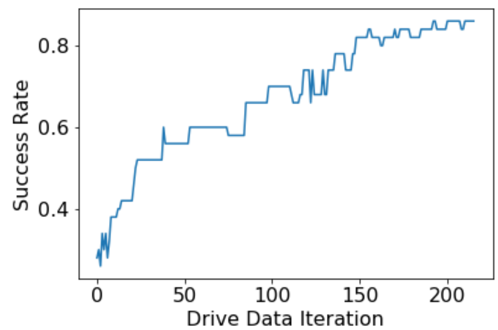

Figure 10: $\rho = 0.0032, t = 240$

30 minutes. More advanced options should be considered given that there might be hundreds or even thousands of quantitative features that behave in an uncorrelated manner.

7 Software and Data

Source code to reproduce the results is available as Supporting Information on Github.

References

Ames, C. The markov process as compositional model: A survey and tutorial. In *Leonardo*, volume 22, pp. 175–187, 01 1989. doi: 10.2307/1575226.

Dorigo, M. and Birattari, M. *Ant Colony Optimization*, pp. 56–59. Springer US, Boston, MA, 2017. ISBN 978-1-4899-7687-1. doi: 10.1007/978-1-4899-7687-1_22. URL `https://doi.org/10.1007/978-1-4899-7687-1_22`.

Li, X., Deng, S., Wang, S., Lv, Z., and Wu, L. Review of small data learning methods. In *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, volume 02, pp. 106–109, July 2018. doi: 10.1109/COMPSAC.2018.10212.

Maniezzo, D., Dorigo, M., Maniezzo, V., and Colorni, A. Ant system: An autocatalytic optimizing process, 1991.

Shaikhina, T. and Khovanova, N. A. Handling limited datasets with neural networks in medical applications: A small-data approach. *Artificial intelligence in medicine*, 75:51–63, 2017.

Siegel, J. E., Erb, D. C., and Sarma, S. E. A survey of the connected vehicle landscape - architectures, enabling technologies, applications, and development areas. *IEEE Transactions on Intelligent Transportation Systems*, 19 (8):2391–2406, Aug 2018. ISSN 1524-9050. doi: 10.1109/TITS.2017.2749459.