Discovery of Driving Patterns by Trajectory Segmentation

Sobhan Moosavi, Arnab Nandi, and Rajiv Ramnath

Department of Computer Science and Engineering, Ohio State University
{moosavinejaddaryakenari.1,nandi.9,ramnath.6}@osu.edu

ABSTRACT

Telematics data is becoming increasingly available due to the ubiquity of devices that collect data during drives, for different purposes, such as usage based insurance (UBI), fleet management, navigation of connected vehicles, etc. Consequently, a variety of data-analytic applications have become feasible that extract valuable insights from the data. In this paper, we address the especially challenging problem of discovering behavior-based driving patterns from only externally observable phenomena (e.g. vehicle's speed). We present a trajectory segmentation approach capable of discovering driving patterns as separate segments, based on the behavior of drivers. This segmentation approach includes a novel transformation of trajectories along with a dynamic programming approach for segmentation. We apply the segmentation approach on a real-word, rich dataset of personal car trajectories provided by a major insurance company based in Columbus, Ohio. Analysis and preliminary results show the applicability of approach for finding significant driving patterns.

Categories and Subject Descriptors

F.2.2 [**Theory of computation**]: Mathematical optimization; H.2.8 [**Information systems**]: Spatial-temporal systems

Keywords

Driving Patterns, Trajectory, Segmentation

1. INTRODUCTION

The amount of telematics data has drastically increased thanks to the ubiquity of various types of devices and mobile apps to collect data during drive. Some instances of such transportation data are the New York taxi cab[1] with 1.1 billion taxi trips and T-Drive [12] with trajectories of 10,357 Beijing taxi cabs for one week. Given the availability of these large transportation data sources, various analysis applications have been implemented to gain insights from this data. Trajectory segmentation is one of the applications which tries into break a trajectory to several partitions or segments based on a set of optimization goals (e.g., minimizing the number of segments, maximizing homogeneity within segments, etc.), where each segment may represent a specific kind of movement pattern, phase, or behavior. In this paper, we propose a trajectory segmentation approach which is capable of discovering driving behavior patterns. Some examples of driving pattern are *make turn*, *change lane*, *merge highway*, etc. We use the following example to describe the goal of current research in more detail.

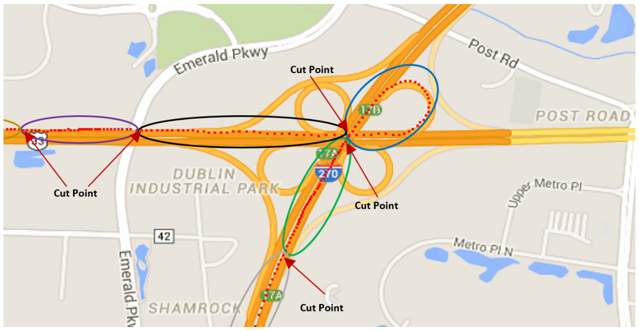


Figure 1: A sample trajectory with several behavior-based driving patterns specified by ovals. Transitions between patterns are pointed by arrows.

Example 1. *Consider the trajectory in Figure 1. Red dots show the location of the car for every second of the trip. The trajectory begins at the bottom center and continues to the left after a clock-wise turn. Different parts of the trip exhibit different driving behavior-based patterns marked out by ovals. For instance, the green oval shows* slow movement*, where the captured locations are close to each other. Another pattern occurs when the car* enters the ramp and merges into a highway *(blue oval).*

Example 1 is intended to illustrate that driving patterns are portions of a trajectory where there is homogeneity of driving behavior. The problem of finding significant driving patterns, as described by Example 1, is a challenging one for following reasons. First, unlike studies such as [7, 11] which collected data using a fully monitored environment (for example, with cameras placed inside the car monitoring the driver's every move and expression), and with a small set of drivers and routes, our dataset is the result of collecting data by observing only externally visible phenomena (e.g. vehicle's speed) with no additional intrusive monitoring. In addition, because of the size of the dataset of trajectories, and the potentially wide range of identifiable and useful driving patterns, a supervised approach is not viable. Thus, finding significant set of driving patterns is a challenging problem, worthy of our study.

Discovery of behavior-based driving patterns is a part of a more generic framework for analysis of behavior of drivers to reveal how risky or safe are their driving habits. The result of such studies can be used for *usage based insurance*, *driver coaching*, *risk man-*

[1] http://toddwschneider.com/posts/analyzing-1-1-billion-nyc-taxi-and-uber-trips-with-a-vengeance/

3rd ACM SIGSPATIAL PhD Workshop'16, October 31-November 03 2016, Burlingame, CA, USA
© 2016 ACM. ISBN 978-1-4503-4584-2/16/10 ...$15.00
DOI: http://dx.doi.org/10.1145/3003819.3003824.

agement, and other related purposes. The main contribution of this paper is a novel trajectory segmentation approach to find driving patterns, based on the behavior of drivers. The rest of this paper is structured as follows: Section 2 provides the formal problem statement and required definitions. Detail of trajectory segmentation approach is addressed in Section 3. Next, the evaluation protocol and preliminary results are presented in Section 4. We provide a summary of related work in Section 5. Section 6 concludes our study and describes potential future work.

2. PROBLEM STATEMENT

Assume we are given a transportation database $\mathcal{D}$ of the form $\langle \Upsilon, \Gamma \rangle$ where Υ and Γ are the set of vehicles and trajectories, respectively. Each trajectory $\gamma \in \Gamma$ is sequence of $|\gamma|$ data points $\langle \rho_1, \rho_2, \ldots, \rho_{|\gamma|} \rangle$. Each data point ρ is a tuple of the form $\{t, lat, lng, s, acc, h\}$ which captures a vehicle's status at time t as its latitude and longitude are $\langle lat, lng \rangle$, with speed s (km/h), acceleration acc (m/s^2), and heading h (degrees). All time is assumed to be in seconds. Also, the heading is the direction of the moving vehicle, described by a degree-value between 0 and 359, where 0 means the north.

A segmentation for a trajectory γ into n segments, denoted as seg_γ, is a set of cutting indexes $seg_\gamma = \langle I_1, I_2 \ldots, I_n \rangle$ that mark the beginning points of the segments within a trajectory. Thus, we can define a set of cutting data points for the segmented trajectory γ as $\langle p_{I_1}, p_{I_2} \ldots, p_{I_n} \rangle$. Note that $p_{I_1} = \rho_1$. All data points between indexes I_i and I_{i+1}, including point ρ_{I_i} and excluding point $\rho_{I_{i+1}}$, belong to the i^{th} segment. We denote the i^{th} segment of seg_γ as seg^i_γ and its size as $|seg^i_\gamma|$. Note that segments are non-overlapping. Each segment represents a *driving pattern* and each cutting point $p_{I_i}, I_i \in seg_\gamma$, represents a *transition between patterns*. Figure 1 demonstrates segments (by ovals) and cutting points (by arrows) for a given trajectory. We define the optimization objectives for segmentation task as i) maximizing homogeneity within segments, ii) minimizing homogeneity between neighboring segments, and iii) minimizing the number of created segments.

3. SEGMENTATION APPROACH

We propose a novel approach to intelligently partition a trajectory, such that each resulting homogeneous segment corresponds to a specific driving pattern. Our trajectory segmentation approach includes following steps:

i. Preprocessing of the trajectory dataset.

ii. Creating a memory-less Markov Model based on behavior of population of drivers in trajectory dataset.

iii. Using the Markov Model to transform a trajectory to a *signal* in *Probabilistic Movement Dissimilarity (PMD)* space.

iv. Segmenting a signal by using a Dynamic Programming Segmentation approach and finding the best number of segments by Minimum Descriptor Length (MDL).

We next describe each step in more detail.

3.1 Preprocessing the Dataset

Regarding the description of the data model in section 2, the data set is a collection of trajectories, where each trajectory has a sequence of data points. The main steps for preprocessing the dataset are as follows:

– Remove data points with missing or noisy (out of range) GPS records.

– Normalize the values of $Acceleration$ and $Heading$ to be divisible by 0.25 and 5 respectively. This step helps to simplify the Markov Model, by reducing the number of possible states.

– Create training and test sets: We use the training set for creating the Markov Model and the test set for experiments.

3.2 Creating the Markov Model

We create a memory-less Markov model $M = \{\Phi, \Delta, \Pi\}$, where Φ is the set of states, Δ is the set of transition between states (along with the frequency of each transition), and Π is the set of probabilities of transition between the states. We use the following guidelines to create the M:

- State: We define a state $\phi \in \Phi$ as $\phi = \langle Speed, Acceleration, Heading \rangle$.

- Transition: Given a trajectory $\gamma = \langle \rho_1, \rho_2, \ldots, \rho_n \rangle$, for each pair of consecutive data points ρ_i and ρ_{i+1} of γ, where $1 \leq i < n$, we create two states $\phi_i = \langle s_i, acc_i, h_i \rangle$ and $\phi_{i+1} = \langle s_{i+1}, acc_{i+1}, h_{i+1} \rangle$ for ρ_i and ρ_{i+1} respectively. We denote a transition from state ϕ_i to ϕ_{i+1} as $\phi_i \rightarrow \phi_{i+1}$. If Δ doesn't contain transition $\phi_i \rightarrow \phi_{i+1}$, then we insert $\langle \phi_i \rightarrow \phi_{i+1}, 1 \rangle$ into Δ. Otherwise, we increase the frequency of transition $\phi_i \rightarrow \phi_{i+1}$ by 1.

- Probability of Transition: For a specific state ϕ, let us assume there is a $\delta \subseteq \Delta$ where $\delta = \{\langle \phi \rightarrow \phi_1, n_1 \rangle, \ldots, \langle \phi \rightarrow \phi_k, n_k \rangle\}$, and where n_i is the number of observed transitions from ϕ to ϕ_i in the dataset, we update Π by inserting the probability of each transition $\phi \rightarrow \phi_i$, $1 \leq i \leq k$, using Equation 1:

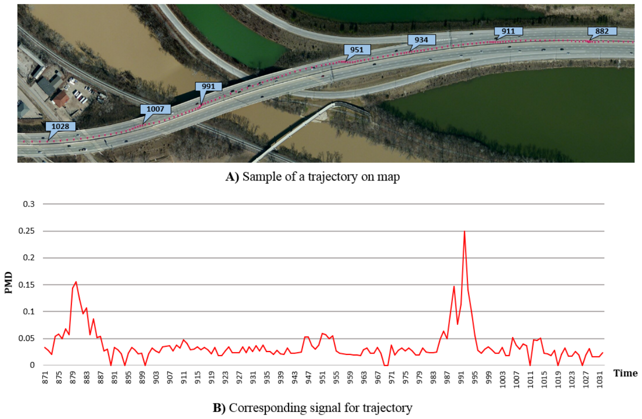
$$prob_{\phi \rightarrow \phi_i} = \frac{n_i}{\sum_{j=1}^{k} n_j} \qquad (1)$$

3.3 Transforming Trajectories

The aim of our segmentation approach is to provide a segmentation of trajectories based on behavior of drivers. Hence, an important step is to transform an input trajectory to a signal in Probabilistic Movement Dissimilarity (PMD) space. Suppose we have a trajectory $\gamma = \langle \rho_1, \rho_2, \ldots, \rho_n \rangle$ and a Markov Model $M = \{\Phi, \Delta, \Pi\}$, we propose Algorithm 1 to map γ to a signal S_γ in PMD space. Given consecutive data points $\rho_i, \rho_{i+1} \in \gamma$, Algorithm 1 first maps them to states ϕ and ϕ' respectively. Then, it calculates how $unlikely$ is the transition $\phi \rightarrow \phi'$, based on M.

Algorithm 1: `Trajectory Transformation`

```
Input: $\gamma$, $M$
Output: $S_\gamma$                              ▷ $S_\gamma$ is transformed version (signal) of $\gamma$
$S_\gamma \leftarrow \langle\rangle$
for $i = 1$ to $n$-1 do
    $\phi \leftarrow ReturnState(M, \rho_i)$
    $\phi' \leftarrow ReturnState(M, \rho_{i+1})$
    $v = 0$
    if $\phi \neq \phi'$ then
        $prob_{\phi \rightarrow \phi'} = ReturnProb(M, \phi, \phi')$
        $R \leftarrow TransitionFrom(M, \phi)$
                                    ▷ $R = \{r | (\phi \rightarrow r) \in \Delta\}$
        for $r \in R$ do
            $prob_{\phi \rightarrow r} = ReturnProb(M, \phi, r)$
            $v$ += $Euclidean(\phi', r) \times prob_{\phi \rightarrow r}$
        end
        $v = \frac{v}{|R|}$
    end
    $S_\gamma \leftarrow Append(S_\gamma, v)$          ▷ Appending $v$ at the end of $S_\gamma$
end
```



A) Sample of a trajectory on map

B) Corresponding signal for trajectory

Figure 2: A) Sample trajectory on map with numbers in call-outs indicate timestamps B) The sample trajectory mapped to a signal in PMD space

In Algorithm 1, $ReturnState$ returns a state corresponding to input data point ρ_i, and $ReturnProb$ returns transition probability from ϕ to ϕ'. $TransitionFrom$ returns a list of all states r given an input state ϕ, such that transition $(\phi \rightarrow r) \in \Delta$. Also, note that if ϕ and ϕ' represent the same state, then the transition is quite likely. Based on this algorithm, we map a test trajectory to a signal in PMD space. The signal of a trajectory demonstrates the unlikelihood of behavior of driver during the trip. An unlikelihood score is calculated based on the transition probabilities in the Markov Model M. Lines 7 to 14 in Algorithm 1 measure how far the observed transition $\phi \rightarrow \phi'$ is from our expectation regarding the Markov Model M.

Figure 2 depicts a part of a sample trajectory and it's corresponding signal in PMD space. The numbers in rectangular call-outs in Figure 2.A show time stamps which can be matched with *Time* axis in Figure 2.B. The more unlikely the behavior of driver be, the larger the value of PMD is. For instance, a large PMD value is observable for time stamp 991 in Figure 2.B, where the actual trip in Figure 2.A shows an unexpected reduction in speed and also a lane change.

The main takeaway from this step is that we use a signal in PMD space as a representation of the behavior of a driver for a given trip, in comparison with the rest of the population of drivers and trajectories.

3.4 Dynamic Programming Trajectory Segmentation

Once the signal for a trajectory has been created, the trajectory segmentation problem reduces to a *Signal Segmentation* problem. For segmenting a signal, we use an existing approach which has been successfully applied for segmenting electrical signals [6]. This approach is a dynamic programming algorithm that uses the Maximum Likelihood principle for segmenting one dimensional signals. Given an input signal $S = \langle x_1, x_2, \ldots, x_N \rangle$, the Maximum Likelihood for S can be defined by Equation 2.

$$ML(\theta; x_1, x_2, \ldots, x_N) = f(x_1, x_2, \ldots, x_N|\theta) = \prod_{i=1}^{N} f(x_i|\theta) \quad (2)$$

In this formula, θ is the set of parameters for a probability density function (PDF) f, which can be estimated based on data points of signal S. As in [6], we leverage the *Gaussian distribution* to find the parameters of the PDF f, thus, $\theta = \langle \mu, \sigma \rangle$, where μ and $sigma$ are the sample mean and standard deviation respectively.

Note that the goal of segmenting a trajectory γ and it's signal $S_\gamma = \langle x_1, x_2, \ldots, x_N \rangle$ (see section 2), is to find a set of cutting indexes $seg_\gamma = \langle I_1, I_2 \ldots, I_n \rangle$, where $n \leq N$ is the best number of existing segments (i.e. with the greatest maximum likelihood). The recurrence relation for segmenting the signal S_γ is defined below:

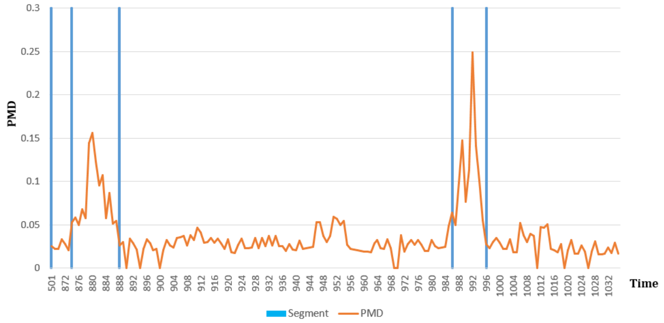


Figure 3: Segmentation of a sample trajectory, where the best number of segments is 5. One can observe the homogeneous pattern of behavior within each segment.

$$SSC(S_\gamma, 1, n) = \underset{2 \leq i \leq N}{\operatorname{argmax}}(ML(S_\gamma, 1, i) + SSC(S_\gamma, i+1, n-1)) \quad (3)$$

In Equation 3, $SSC(S_\gamma, i, \nu)$ gives the best Segmentation Score for a sub-sequence of signal S_γ which starts at index i, with the goal being to find ν segments. Also, $ML(S_\gamma, i, j)$ gives the maximum likelihood score for sub-sequence $\langle x_i, x_{i+1}, \ldots, x_j \rangle$ of S_γ. Note that we assume the minimum length of a segment to be 2. More details of this algorithm may be found in [6].

The last question in this sub-section is: how to find the best number of existing segments within a signal? We use the Minimum Descriptor Length (MDL) [10] for this purpose, which has been applied in [6] as well. MDL tries to minimize the Equation 4 for $n = 1, 2, \ldots, K$, where n is the number of segments and K is the maximum possible number of segments (chosen by the user):

$$MDL(n) = -ln \prod_{i=1}^{n} f(x_{I_i}, x_{I_i+1}, \ldots, x_{I_{i+1}-1}, |\theta_i) + \frac{r_n}{2} lnN \quad (4)$$

In Equation 4, θ_i is the parameter set of the corresponding PDF, r_n is the number of estimated parameters (where n is the number of segments), and N is the length of the signal. Figure 3 shows a part of a segmented signal which is related to the sample trajectory in figure 2.A. The blue lines in figure 3 show the starting points of segments (i.e. the cutting points). The best number of segments which has been found by our MDL algorithm is 5. Note that we can observe the homogeneity of driving behavior patterns *within* segments and the heterogeneity of the driving patterns *between* segments.

As an example of driving behavior pattern which is captured by our trajectory segmentation approach, we point to the segment which starts at time stamp 986 in Figure 3. Regarding the actual trip in 2.A, we see this segment is related to a part of driving behavior where driver reduces speed and changes the lanes.

4. EVALUATION

We first describe the dataset which is used in this study. Then, we provide experimental settings and some statistics as earlier results of trajectory segmentation approach which is applied on our real-world dataset.

4.1 Trajectory Dataset

We used a real-world dataset of 100,000 personal car trajectories provided by a major insurance company based in Columbus,

Table 1: Summary of trajectory test set and segmentation result

Route	#Trajectories	Avg. Length	Avg. #Segment	Std. #Segment
315 Fwy	426	705	8	7
I-270	701	389	4.9	3.8
I-670	443	392	7.4	6.4
I-70	1,572	324	5.4	4.9
I-71	1,320	549	7.5	6.8

Ohio. These trajectories were collected during 2011 to 2015. We used approximately 95% of trajectories for training (i.e. creating the Markov model) and 5% as the test set (for evaluation). The test dataset contains about 4,500 trajectories of 92 drivers for 5 different, popular routes in the city. Routes and number of trajectories for each is summarized in Table 1.

4.2 Segmentation results

We used the process which is described in Section 3 to segment trajectories in the test set. To find the the upper bound on the number of existing segments K (Section 3), we used a heuristic as follows: for a given trajectory γ of length N, we set $K = \frac{N}{10}$. Based on the segmentation result which is illustrated in Table 1, this is a reasonable upper bound. Note that the best number of segments is likely a result of the length of the trips in test set. Table 1 summarizes the segmentation results by providing the average and standard deviation for the number of segment for trajectories in different routes of the test set.

5. RELATED WORK

Trajectory Segmentation, as described in Section 2, has been addressed in the literature in several studies like [4, 1, 5, 3]. In [4], a greedy segmentation algorithm exploits a set of monotonic spatio-temporal criteria (e.g., defining relative thresholds for some feature values) on features like speed, heading, etc. Alewijnse et al. extended the previous work to both monotonic and non-monotonic criteria [1]. However, criteria-based methods need human input for tuning parameters. Moreover, they are *context-agnostic* in that they only consider the input trajectory and not the whole dataset. Therefore, the optimization process is a local one, where we propose a global optimization for segmentation.

Our segmentation approach is a context-aware one by building a Markov Model for the whole dataset prior to segmentation. Similarly, some context-aware approaches are proposed in the literature including [8, 2]. Alewijnse et al. [2] present a context-aware approach which builds a Brownian Bridge model and uses a dynamic programming algorithm to capture the best set of segments of animal movements. While our solution bears some similarities with [2], it exploits a normal distribution model instead, which we find it more suitable for car transportation data.

In [9], a trajectory-to-signal transformation is performed prior to segmentation using similarity values between each line segment of input trajectory and the rest of the line segments in the dataset, using global voting. Then, segmentation discovery is done using a sliding-window approach. Our approach, in contrast, performs a behavior likelihood-based transformation to provide a behavior based segmentation and to find the segments which are representatives for driving behavior patterns. Essentially, our solution is a global optimization-based segmentation approach that builds up a model on the entire dataset. Note also that here is no need for human intervention in our solution as in [4, 1].

6. CONCLUSION AND FUTURE WORK

In this paper, we proposed a Trajectory Segmentation approach to detect behavior based driving patterns for a given trajectory, based on externally observable phenomena. Our approach is a context aware solution which considers the behavior of the entire population of drivers to detect driving patterns. Our preliminary analysis based on existing use cases demonstrate the interpretability of segmentation results, as one of them described in Section 3 for instance (Figures 2 and 3).

We use the current study as a part of a more generic framework for analyzing the behavior of drivers to reveal how risky or safe their driving habits are. Other parts of this framework can be outlined as follows and they also will be considered as extensions of current study. In order to get more insight about extracted patterns by segmentation approach, we will design a supervised learning approach to learn and then predict true labels for patterns. Potential labels may be *making a turn*, *changing the lane*, *merging to a highway*, etc. Moreover, by having true labels for extracted patterns, we will apply sequential pattern mining techniques to extract significant sequences of driving patterns for a single driver or a population of drivers. Finally, by having human experts in the loop, we will identify the safe or risky sequences of driving patterns. In this way, we can formulate the problem of finding safe or risky drivers, based on their driving habits, as an end-to-end solution.

7. REFERENCES

[1] S. Alewijnse, K. Buchin, M. Buchin, A. Kölzsch, H. Kruckenberg, and M. A. Westenberg. A framework for trajectory segmentation by stable criteria. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 351–360. ACM, 2014.

[2] S. P. Alewijnse, K. Buchin, M. Buchin, S. Sijben, and M. A. Westenberg. Model-based segmentation and classification of trajectories. In *Dead Sea, Israel: Proceedings of the 30th European Workshop on Computational Geometry March*, pages 3–5, 2014.

[3] A. Anagnostopoulos, M. Vlachos, M. Hadjieleftheriou, E. Keogh, and P. S. Yu. Global distance-based segmentation of trajectories. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 34–43. ACM, 2006.

[4] M. Buchin, A. Driemel, M. van Kreveld, and V. Sacristán. An algorithmic framework for segmenting trajectories based on spatio-temporal criteria. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 202–211. ACM, 2010.

[5] C. Chen, H. Su, Q. Huang, L. Zhang, and L. Guibas. Pathlet learning for compressing and planning trajectories. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 392–395. ACM, 2013.

[6] T. X. Han, S. Kay, and T. S. Huang. Optimal segmentation of signals and its application to image denoising and boundary feature extraction. In *Image Processing, 2004. ICIP'04. 2004 International Conference on*, volume 4, pages 2693–2696. IEEE, 2004.

[7] A. Liu and D. Salvucci. Modeling and prediction of human driver behavior. In *Intl. Conference on HCI*, 2001.

[8] R. Mann, A. D. Jepson, and T. El-Maraghi. Trajectory segmentation using dynamic programming. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 1, pages 331–334. IEEE, 2002.

[9] C. Panagiotakis, N. Pelekis, I. Kopanakis, E. Ramasso, and Y. Theodoridis. Segmentation and sampling of moving object trajectories based on representativeness. *IEEE Transactions on Knowledge and Data Engineering*, 24(7):1328–1343, 2012.

[10] J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

[11] A. Sathyanarayana, P. Boyraz, and J. H. Hansen. Driver behavior analysis and route recognition by hidden markov models. In *Vehicular Electronics and Safety, 2008. ICVES 2008. IEEE International Conference on*, pages 276–281. IEEE, 2008.

[12] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang. T-drive: driving directions based on taxi trajectories. In *Proceedings of the 18th SIGSPATIAL International conference on advances in geographic information systems*, pages 99–108. ACM, 2010.