Spatiotemporal Representation Learning for Driving Behavior Analysis: A Joint Perspective of Peer and Temporal Dependencies

Submitted by

Atishay Jain (167108)

Ankit Kumar (167107)

Prudhvi Malhotra (167129)

Under the esteemed guidance of

Dr. R.B.V. SUBRAMAANYAM
(Professor)

Department of Computer Science and Engineering

National Institute of Technology Warangal

2019-2020

APPROVAL SHEET

The project work entitled **Spatiotemporal Representation Learning for Driving Behavior Analysis: A Joint Perspective of Peer and Temporal Dependencies** by Atishay Jain, Ankit Kumar, Prudhvi Malhotra is approved for the degree of Bachelor of Technology in Computer Science and Engineering at National Institute of Technology Warangal during the year 2019-20.

Examiners

____

____

____

____

____

Supervisor

____

Dr. R.B.V.

SUBRAMAANYAM

Professor, CSE Dept.

Date: ___________ Place:____________

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included. We have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misinterpreted or fabricated or falsified any ideas/ data / fact / source in our submission. We understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

____

(Signature)
Atishay Jain
167108
Date: ____________

____

(Signature)
Ankit Kumar
167107
Date: ____________

____

(Signature)
Prudhvi Malhotra
167129
Date: ____________

ACKNOWLEDGEMENT

We would like to express heartfelt gratitude and regards to our project guide **Dr. R.B.V. SUBRAMAANYAM,** Department of Computer Science and Engineering, NIT Warangal. We convey a humble thanks to him for his valuable cooperation, support and suggestion throughout the project work which made this project successful. I shall remain indebted throughout my life for his noble help and guidance.

We are thankful to all the faculties of the Department of Computer Science and Engineering, NIT Warangal, for their encouraging words and valuable suggestions towards the project work. Last but not the least we want to acknowledge the contribution of our parents, family members, and friends for their constant and never-ending motivation.

We would like to take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of the project.

Atishay Jain (167108)

Ankit Kumar (167107)

Prudhvi Malhotra (167129)

ABSTRACT

As the world is growing at a great pace, so are humans' dependencies on electronic gadgets and automobiles. Transportation has been one such field which has revolutionized and changed the life of millions. The major changes from walking in past to multi-featured vehicles in present sets our path to autonomous driving in future. As every major change comes with its pros and cons, so to ensure driving remains a safe activity, assessing a driver's behavior in different circumstances and providing improvement measures becomes a necessity.

Our project aims to achieve significant knowledge from various drives conducted in a variety of situations to analyze the driver's behavior. We are using a Peer and Temporal Aware Representation Learning based framework (PATRL) for driving behavior analysis using GPS trajectory data. We first detect the driving operations and states of each driver from their GPS traces. Then, we derive a sequence of multi-view driving state transition graphs from the driving state sequences, in order to characterize a driver’s driving behaviors that vary over time. In addition, we develop a peer and temporal-aware representation learning method to learn a sequence of time-varying yet relational vectorized representations from the driving state transition graphs..

CONTENTS

CHAPTER 1

INTRODUCTION

Driving Behavior Analysis

As driving comprises of various speed and direction related operations, such as acceleration, deceleration, keeping constant speed, turning left, turning right, moving straight. By analyzing driving behaviors, we can try to provide driving assistance, enhance driving comforts, develop intelligent and resilient transportation systems, assessing driver's performances and enhance traffic safety. Our project aims to obtain quantitative representation of driver's behavioral trend from his raw GPS trajectories.

The studies conducted till date have scope of improvements as some of them are based on biased and expensive data sources, others considering data which might have privacy issues e.g. usage of CAN bus data [1], [2], hence GPS data remains the best alternative. However, CAN bus data is more accurate to quantify driving operations, but it receives data from equipment installed in vehicles and even consist of information which driver might not be willing to reveal. Due to the pervasiveness of GPS sensors, GPS data can be collected from location-based apps (e.g. Google Maps), that are granted permissions by user themselves. Thus, we are convinced to use GPS data for our task. It is highly promising to use high-resolution widely-available GPS trajectories with representation learning for the task of driving behavior analysis.

CHAPTER 2

RELATED WORK

Prior studies in driving behaviors analysis can be categorized into:

(i) **Causal Analysis**: Researchers identify the causal factors of driving behaviors and explain how these factors influence road safety [3].

(ii) **Predictive Analysis**: Researchers mine the patterns from driving data and apply machine learning models (e.g., SVM, naive Bayesian, etc.) to predict risky scores [4].

(iii) **Descriptive Analysis**: Transportation experts define measurements (e.g., harsh or frequent acceleration/braking, sharp turn, acceleration before turn) based on transportation theory to describe driving behaviors [5].

CHAPTER 3

PROBLEM STATEMENT

In our project, we explore the problem of automated driving behavior profiling with GPS traces. Formally, given a driver (a vehicle) and corresponding GPS trajectories, we aim to find a mapping function that takes the GPS trajectories as inputs, and outputs a sequence of time-varying yet relational vectorized representations in order to quantify the dynamics of the driver's driving behavior. We formulate this problem as a task of spatio-temporal representation learning. Essentially, we first construct a sequence of driving state transition graphs from GPS trajectories, and then learn the latent representations of driving behavior from the graphs.

CHAPTER 4

ATTRIBUTES DEFINITION

1. *Driving State*: It is a tuple with two values, the first being speed status and second as direction status. E.g. <acceleration, turning-left> [6].

2. *State Transition Graphs*: Driver's movement from one driving state to another is depicted as state transition graph which inherently is a directed graph [6].

TABLE 1: Summary of notations.

Symbol	Discription
ϕ_t	The latitude at time t
λ_t	The longitude at time t
G_i^τ	The driving behavior transition graph sequence of driver i at the time slot τ
$\mathbf{x}_i^\tau$	The original original vector representation of G_i^τ
$\mathbf{z}_i^\tau$	The learned representation for the driver i at the time slot τ
$(\mathbf{y}_i^o)^\tau$	The latent feature representations of the driver i at hidden layers o at the time τ in the encode process
$(\hat{\mathbf{y}_i^o})^\tau$	The latent feature representations of the driver i at hidden layers o at the time τ in the decode process
$\mathbf{W}$ $\mathbf{b}$	Weights and biases in the encode process
$\hat{\mathbf{W}}$ $\hat{\mathbf{b}}$	Weights and biases in the decode process process
$\mathcal{H}(\star)$	Loss function
A, B	The hyperparameters to control the weight of the representation learning loss and regression loss

CHAPTER 5

MODEL FRAMEWORK

To address the problem of consideration of both peer and temporal dependencies simultaneously, a framework called Peer and Temporal Aware Representation Learning Framework (PTARL) [6] is being used in this project. Raw GPS trajectories are converted into state transition graphs and are used with aforementioned framework which can learn a sequence of time-varying yet relational vectorized representations.

PTARL framework consists of following essential steps:

1. **Construction of Multi-View Driving State Transition Graphs**: To obtain driving state sequence and then driving state transition graph from GPS trajectories, we first detect different speed and direction related operations for each driver and then identify driving states and obtain driving state sequence. Later, the driving state sequence is segmented into small subsequences with a fixed time window, and then it is converted into a driving state transition graph. For each driving state subsequence, we get a driving state transition graph and hence it can be used to characterize the time-varying driving behavior of a driver.

    a. *Detection of speed-related operations*: Let $\Delta\phi_{1,2}$ be the difference of ϕ_1 and ϕ_2, $\Delta\phi_{2,3}$ be the difference of ϕ_2 and ϕ_3, $\Delta\lambda_{1,2}$ be the difference of λ_1 and λ_2, $\Delta\lambda_{2,3}$ be the difference of λ_2 and λ_3, and R be the radius of the earth. Then, the distance $d_{1,2}$ between the two GPS points $<\phi_1, \lambda_1>$ and $<\phi_2, \lambda_2>$ is given by following equation:

$$2R.a\tan 2\left(\sqrt{\frac{\sin^2(\Delta\varphi_{1,2})}{2} + \cos\phi 1 \cdot \cos\phi 2.\sin^2\left(\frac{\Delta\lambda_{1,2}}{2}\right)}, \sqrt{1 - \sin^2\left(\frac{\Delta\varphi_{1,2}}{2}\right) - \cos\phi 1 \cdot \cos\phi 2.\sin^2\left(\frac{\Delta\lambda_{1,2}}{2}\right)}\right)$$

Similarly, the distance $d_{2,3}$ between the two GPS points $<\phi_2, \lambda_2>$ and $<\phi_3, \lambda_3>$ can also be calculated. Then, given the time stamps of the three GPS points, denoted by t_1, t_2 and t_3, the speed s_2 at t_2 is given by $s_2 = d_{1,2}/(t_2 - t_1)$, the speed s_3 at t_3 is given by $s_3 = d_{2,3}/(t_3 - t_2)$. For t_3, if $s_3 > s_2$, the operation is detected as acceleration; if $s_3 < s_2$, the operation is deceleration; otherwise, the operation is “constant speed”. In practice, due to the noise caused by GPS devices, we introduce a loosing boundary $€_s$ into the calculation. For t_3, if $s_3 > s_2$ and $| s_3 - s_2| > €_s$, the operation is detected as acceleration; if s3 < s2 and |s3 −s2| > $€_s$, the operation is deceleration; otherwise, the operation is “constant speed”.

b. *Detection of direction-related operations:* To detect the direction-related operations, we calculate the bearing $\theta_{1,2}$ between the two GPS points <ϕ_1, λ_1> and <ϕ_2, λ_2> by the following equation:

$$\theta_{1,2} = a\tan 2\left(\sin\Delta\lambda_{1,2} - \cos\varphi_2, \cos\varphi_1 . \sin\phi_2 - \sin\phi_1 . \cos\varphi_2 . \cos\Delta\lambda_{1,2} \right)$$

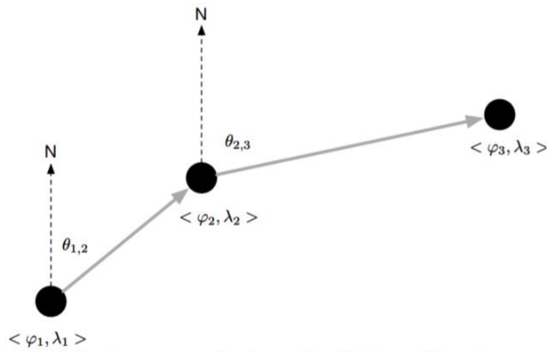
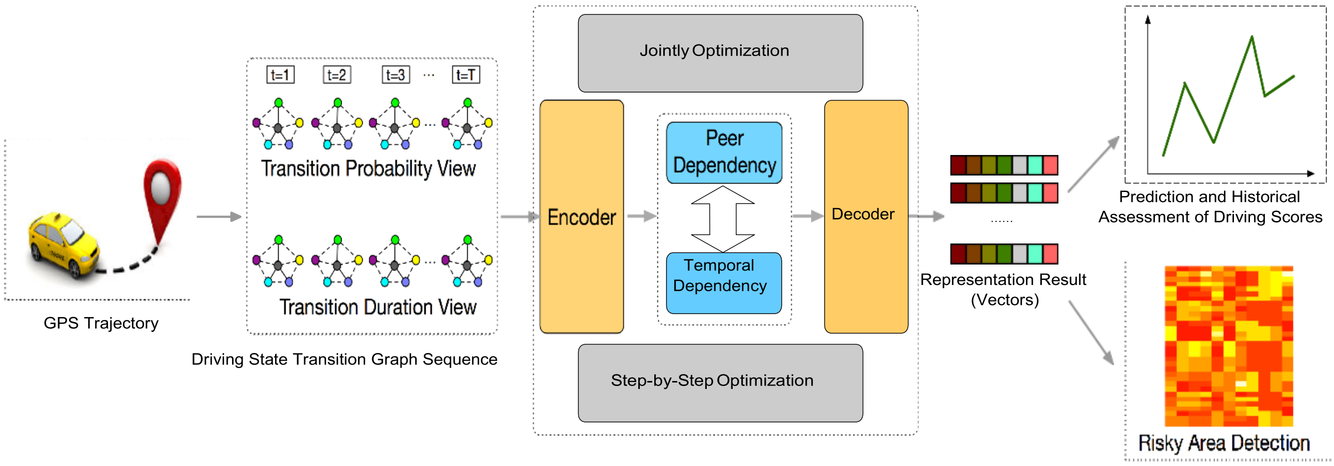


Fig. 2: An example for calculating directions.

Similarly, we can obtain the bearing $\theta_{2,3}$ between the two GPS points <ϕ_2, λ_2> and <ϕ_3, λ_3>.Therefore, as shown in Figure2, at t_3, if $\theta_{2,3} > \theta_{1,2}$, then the operation is “turning right”; if $\theta_{2,3} < \theta_{1,2}$, then the operation is “turning left”; otherwise, the operation is “moving straight”. In practice, we also introduce a loosing boundary $€_d$ to estimate directions. At t_3, if $\theta_{2,3} > \theta_{1,2}$ and $|\theta_3 - \theta 2| > €_d$, then the operation is “turning right”; if $\theta_{2,3} < \theta_{1,2}$ and $|\theta_3 - \theta_2| > €_d$, then the operation is “turning left”; otherwise, the operation is “moving straight”.



An overview of the proposed analytic framework

2. **Extraction of Driving State Sequences**: Based on the two speed-related operations and the three direction-related operations, we can define the following driving states as per the table:

TABLE 2: Driving states.

(1)	acceleration while turning right
(2)	acceleration while turning left
(3)	acceleration while straightforward
(4)	deceleration while turning right
(5)	deceleration while turning left
(6)	deceleration while straightforward
(7)	constant speed while turning right
(8)	constant speed while turning left
(9)	constant speed while straightforward

With the above definitions, we can identify the driving state of a driver at each time stamp. In other words, each trajectory is associated with a driving state sequence, which is denoted by $\{(ID, t_n, S_n)\}^{N}_{n=1}$, where ID is the identity of the driver, N is the size of the driving state sequence, t_n is the n^{th} time stamp, and S_n is the driving state at t_n.

3. **Peer and Temporal-Aware Representation Learning**: There are peer and temporal dependencies among driving behavior. Therefore, in this project, we model the representation of driving behavior based on the following intuitions.

- Intuition 1: *Structural Reservation*: After reducing driving behavior into graphs, we need a representation learning based method to transform graphs into vectors in a latent feature space for automated quantification and profiling. Consequently, the method should be able to project graphs into lower-dimensional vectors while reserving corresponding characteristics and structures.
- Intuition 2: *Peer Dependency*: If two drivers exhibit similar driving habits, and the vehicle operation patterns of two corresponding trajectories are similar, then the driving state transition graphs of these two trajectories share a lot in terms of structures and characteristics. As a result, the learned representations of driving behavior should be close to each other. Consequently, the method should be able to model the graph-graph peer dependency in representation learning.
- Intuition 3: *Temporal Dependency*: The driving operations of the current time slot have autocorrelation with previous driving states. For example, if a driver decelerates while straightforward at t, and if $\Delta(t, t+1)$ is small enough, then he is likely to accelerate at $t + 1$. Consequently, the method should be able to model the current-past temporal dependency in representation learning.

4. **Base Model**: We utilize the deep Auto-Encoder model [3] as our base model. The motivation of using Auto-Encoder is that we aim to model the structural information of the driving state transition graph. Auto-Encoder shows the good performance of modeling structural information [1]. In addition, previous studies [18], [27] show that autoencoder is effective in modeling human mobility data, which fits the scenario in our work.

   Auto-Encoder is an unsupervised neural network model, which projects the instances in original feature representations into a lower-dimensional feature space via a series of non-linear mappings. The Auto-Encoder model involves two steps: encode and decode. The encode part projects the original feature vector to the objective feature space, while the decode step recovers the latent feature representation to a reconstruction space. In the auto-encoder model, we need to ensure that the original feature representation of instances should be as similar to the reconstructed feature representation as possible. Formally, let xi be the original feature representation of the i^{th} driver, and y^1 , y^2 , · · · , y^o be the latent feature representations of the diver at hidden layers 1, 2, · · · , o in the encode step respectively, the encoding result in the objective lower-dimension feature space can be represented as $z_i \in R^d$ with dimension d. Formally, the relationship between these vector variables is denoted by:

$$\begin{cases} y_i^{\,1} & = \sigma\left(W^1 x_i + b^1 \right), \\ y_i^{\,k} & = \sigma\left(W^k y_i^{\,k-1} + b^k \right), \qquad \forall k \in \{2,3,4,\ldots,o\}, \\ z_i & = \sigma\left(W^{o+1} y_i^{\,o} + b^{o+1} \right). \end{cases} \tag{3}$$

Meanwhile, in the decode step, the input will be the latent feature vector z^i (i.e., the output of the encode step), and the final output will be the reconstructed vector $\widehat{X}_i$. The latent feature vectors at each hidden layers can be represented as $\widehat{y}_i^{\,o}, \widehat{y}_i^{\,o-1}, \widehat{y}_i^{\,o-2}, \ldots, \widehat{y}_i^{\,1}$. The relationship between these vector variables is denoted by:

$$\begin{cases} \widehat{y}_i^{\,o} & = \sigma\left(\widehat{W}^{o+1} z_i + \widehat{b}^{o+1} \right), \\ \widehat{y}_i^{\,k-1} & = \sigma\left(\widehat{W}^k y_i^{\,k} + \widehat{b}^k \right), \quad \forall k \in \{2,3,4,\ldots,o\}, \\ \widehat{x}_i & = \sigma\left(\widehat{W}^1 \widehat{y}_i^{\,1} + \widehat{b}^1 \right). \end{cases} \tag{4}$$

where Ws and bs are the weight matrices and bias terms to be learned in the model. The objective of the auto-encoder model is to minimize the loss between the original feature vector x and the reconstructed feature vector xˆ. Formally, the loss function is

$$\mathbb{H}_c(G^T) \;=\; \sum_{u_i \in \mathfrak{U}} \; \sum_{u_j \in \mathfrak{U},\; u_i \neq u_j} S^T{}_{i,j} \cdot \| Z_i^{\,T} - Z_i^{\;T} \|^2{}_2$$

where u_i denotes the i^{th} driver and U denotes the driver set.

CHAPTER 6

EXPERIMENTAL RESULTS

Data Preprocessing:

Before using trajectory data, we need to deal with a number of issues, such as *noise-filtering*, *segmentation* [7]. The goal of noise filtering is to remove from a trajectory some noise points that may be caused by the poor signal of location positioning systems (e.g., when traveling in a city canyon). Trajectory segmentation divides a trajectory into fragments by time interval, spatial shape, or semantic meanings, for a further process like clustering and classification.

TABLE 3: Statistics of the experimental data.

Properties	**Statistics**
Number of drivers	10,357
Time range	Feb.2 - Feb.8
Sampling Interval	117 Seconds
Sampling Distance	623 Meters
Total Trajectories	15 Million
Total Distance	9 million KM
Records are within a time window	83 in Average
City	Beijing

Data Description:

Table 3 shows the statistics of our real-world data sets T-Drive trajectory dataset [29], [30]. This dataset contains the GPS trajectories of 10,357 taxis during the period of Feb. 2 to Feb. 8, 2008 within Beijing. The total number of points in this dataset is about 15 million and the total distance of the trajectories reaches to 9 million kilometers. The average sampling interval is about 177 seconds with a distance of about 623 meters. Each GPS point contains the information of corresponding driver ID, latitude, longitude, and time stamp.

CHAPTER 7

CONCLUSION

Driving behavior analysis can be very useful in assessing driver performances, improving traffic safety, and development of an intelligent and resilient transportation systems. In this project, we tried to explore the idea to investigate driving behavior analysis from the perspective of representation learning. We used the formulation of problem of driving behavior profiling and scoring as a task of spatial and temporal embedding and labelling with driving state transition graphs. We used the studies of large-scale driving behavior data, and identified the peer and temporal dependencies.

To improve the performance of automated behavior profiling, we used an analytic framework that jointly modelled the peer and temporal dependencies, discussed in this paper [6]. Specifically, we used the idea to first construct multi-view driving state transition graphs from GPS traces to characterize driving behavior. Besides, the idea of gated recurrent unit is also incorporated to model both the graph-graph peer dependency and integrate graph-graph peer penalties to capture the current-past temporal dependency in two optimization strategies, i.e.(i) jointly optimization and (ii) step-by-step optimization. The empirical experiments on real-world data demonstrated the effectiveness of spatio-temporal representation learning for profiling driving behavior.

CHAPTER 8

REFERENCES

[1] Hailong Liu, Tadahiro Taniguchi, Yusuke Tanaka, Kazuhito Takenaka, and Takashi Bando. Visualization of driving behavior based on hidden feature extraction by using deep learning. IEEE Transactions on Intelligent Transportation Systems, 18(9):2477–2489, 2017.

[2] Hideaki Misawa, Kazuhito Takenaka, Tomoya Sugihara, Hailong Liu, Tadahiro Taniguchi, and Takashi Bando. Prediction of driving behavior based on sequence to sequence model with parametric bias. In 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), pages 1–6. IEEE, 2017.

[3] Sarah M Simmons, Anne Hicks, and Jeff K Caird. Safety-critical event risk associated with cell phone tasks as measured in naturalistic driving studies: A systematic review and meta-analysis. Accident Analysis & Prevention, 87:161–169, 2016.

[4] Xiaoyu Zhu, Yifei Yuan, Xianbiao Hu, Yi-Chang Chiu, and YuLuen Ma. A bayesian network model for contextual versus noncontextual driving behavior assessment. Transportation Research Part C: Emerging Technologies, 81:172–187, 2017.

[5] Adrian B Ellison, Michiel CJ Bliemer, and Stephen P Greaves. Evaluating changes in driver behaviour: a risk profiling approach. Accident Analysis & Prevention, 75:298–309, 2015.

[6] P. Wang, X. Li, Y. Zheng, C. Aggarwal and Y. Fu, "Spatiotemporal Representation Learning for Driving Behavior Analysis: A Joint Perspective of Peer and Temporal Dependencies,"

in *IEEE Transactions on Knowledge and Data Engineering*.
doi: 10.1109/TKDE.2019.2935203
[7] Yu Zheng. Trajectory Data Mining: An Overview
ACM Transactions on Intelligent Systems and Technology (TIST) - Survey Paper, Regular Papers and Special Section on Participatory Sensing and Crowd Intelligence archieve
Volume 6 Issue 3, May 2015 Article No. 29 ACM New York, NY, USA doi: 10.1145/2743025

[8] Jing Yuan, Yu Zheng, Xing Xie, and Guangzhong Sun. Driving with knowledge from the physical world. In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 316–324. ACM, 2011.
[9] Jing Yuan, Yu Zheng, Chengyang Zhang, Wenlei Xie, Xing Xie, Guangzhong Sun, and Yan Huang. T-drive: driving directions based on taxi trajectories. In Proceedings of the 18th SIGSPATIAL Internationalconferenceonadvancesingeographicinformationsystems, pages 99–108. ACM, 2010.