Applying Machine and Statistical Learning Techniques to Intelligent Transport Systems: Bottleneck Identification and Prediction, Dynamic Travel Time Prediction, Driver Stop-run Behavior Modeling, and Autonomous Vehicle Control at Intersections

Mohammed Mamdouh Zakaria Elhenawy

Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
In
Computer Engineering

Ioannis M. Besieris, (Co-Chair)
Hesham A. Rakha, (Co-Chair)
Amos L. Abbott
Feng Guo,
Mohamed Essam Khedr

May 4th 2015
Blacksburg, Virginia

Keywords: Machine learning, statistical learning, ITS, Bottlenecks Identification and Prediction, Dynamic Travel Time Prediction, Stop-run Driver Behavior Modeling, and uncontrolled intersection

Copyright 2015, Mohammed Elhenawy

Applying Machine and Statistical Learning Techniques to Intelligent Transport Systems: Bottleneck Identification and Prediction, Dynamic Travel Time Prediction, Driver Stop-run Behavior Modeling, and Autonomous Vehicle Control at Intersections

Mohammed Mamdouh Zakaria Elhenawy

Abstract

In this dissertation, new algorithms that address three traffic problems of major importance are developed. First automatic identification and prediction algorithms are developed to identify and predict the occurrence of traffic congestion. The identification algorithms concoct a model to identify speed thresholds by exploiting historical spatiotemporal speed matrices. We employ the speed model to define a cutoff speed separating free-flow from congested traffic. We further enhance our algorithm by utilizing weather and visibility data. To our knowledge, we are the first to include weather and visibility variables in formulating an automatic congestion identification model. We also approach the congestion prediction problem by adopting an algorithm which employs Adaptive Boosting machine learning classifiers again something novel that has not been done previously. The algorithm is promising where it resulted in a true positive rate slightly higher than 0.99 and false positive rate less than 0.001.

We next address the issue of travel time modeling. We propose algorithms to model travel time using various machine learning and statistical learning techniques. We obtain travel time models by employing the historical spatiotemporal speed matrices in conjunction with our algorithms. The algorithms yield pertinent information regarding travel time reliability and prediction of travel times. Our proposed algorithms give better predictions compared to the state of practice algorithms.

Finally we consider driver safety at signalized intersections and uncontrolled intersections in a connected vehicles environment. For signalized intersections, we exploit datasets collected from four controlled experiments to model the stop-run behavior of the driver at the onset of the yellow indicator for various roadway surface conditions and multiple vehicle types. We further propose a new variable (predictor) related to driver aggressiveness which we estimate by monitoring how drivers respond to yellow indications. The performance of the stop-run models shows improvements after adding the new aggressiveness predictor. The proposed models are practical and easy to implement in advanced driver assistance systems. For uncontrolled intersections, we present a game theory based algorithm that models the intersection as a chicken game to solve the conflicts between vehicles crossing the intersection. The simulation results show a 49% saving in travel time on average relative to a stop control when the vehicles obey the Nash equilibrium of the game.

Dedication

“TO MY FAMILY”

Acknowledgements

All Praise is due to Allah, we thank him, seek his guidance and forgiveness.
I would like to thank everybody that contributed to this success directly or indirectly whether a family member, a friend or a colleague.
I would like to express my gratitude to my Co-advisor Dr. Hesham Rakha for his continuous help and support financially and academically. Many thanks go to Dr. Ioannis M Besieris and and Dr. Ihab El-Shawarby for their guidance and contribution to this work.
I would also like to thank all people who ran the field experiments and provide us the datasets we used in this thesis.

Attribution

All of the co-authors of the chapters introduced in this dissertation are working at the center of sustainable mobility (CSM) at the Virginia Tech Transportation Institute and Virginia Polytechnic Institute and State University, Blacksburg campus.

Hesham Rakha is a Samuel Reynolds Pritchard Professor of Engineering, Dept. of Civil & Environmental Engineering and a courtesy Professor, Bradley Dept. of Electrical and Computer Engineering at Virginia Tech and the director of the center for sustainable mobility. He was involved in the early stages of concepts formation and contributed to manuscript edits

Ihab. El-Shawarby is a research scientist at Virginia Tech Transportation Institute. He was involved in managing the approval by the institutional review board at the office of research compliance and the manuscript composition of chapters 13, 14 and 15.

Hao Chen is a research associate at Virginia Tech Transportation Institute. Hao Chen was involved in the manuscript composition of chapters 2, 4,7and 8. He pre-processed the INRIX raw data analyzed in chapters 4, 7 and 8.

Arash Jahangiri is a Ph.D. candidate at the Civil and Environmental Engineering Department at Virginia Tech. Arash Jahangiri was involved in applying support vector machine to the data sets analyzed in chapters 14 and 15 and manuscript composition of both chapters as well.

Mohammed Almannaa is a master student at the Civil and Environmental Engineering Department at Virginia Tech. He was involved in the manuscript composition of chapter 12.

Ahmed Abdelnaeim Elbery is a Ph.D. student at the Computer Science Department at Virginia Tech. He was involved in INTEGRATION simulation and coding of the algorithm proposed in chapter 16 and manuscript composition of chapter 16 as well.

Abdallah Hassan Mahmoud, is a Ph.D. candidate at the Bradley Department of Electrical and Computer Engineering Electrical & Computer Engineering, Virginia Tech. He was involved in the manuscript composition of chapter 11 and chapter 16. Moreover, He participated in Matlab coding of the algorithm in chapter 16.

Table of Contents

List of Figures

List of Tables

Preface

All of the research work introduced in this dissertation was conducted in the center of sustainable mobility (CSM) at the Virginia Tech Transportation Institute and Virginia Polytechnic Institute and State University, Blacksburg campus. All data set collected from controlled experiments involving human subjects were approved by the institutional review board at the office of research compliance.

Chapter 2 is based on Mohammed Elhenawy, Hesham Rakha, and. Hao Chen, "An automated statistically-principled bottleneck identification algorithm (ASBIA)," in Intelligent Transportation Systems - (ITSC), 2013 16th International IEEE Conference on, 2013, pp. 1846-1851.I was responsible for all major areas of concept formation, coding programming and data analysis, as well as manuscript composition. Hesham Rakha was involved in the early stages of concept formation and contributed to manuscript edits. Hao Chen was involved in manuscript composition.

Chapter 3 is based on Mohammed Elhenawy and Hesham Rakha, "Automatic Congestion Identification Using Two-Component Mixture Models," Transportation Research Record: Journal of the Transportation Research Board, 2015. I was responsible for all major areas of concept formation, R coding and data analysis, as well as manuscript composition. Hesham Rakha introduced the idea of using mixture models and contributed to manuscript edits.

Chapter 4 is based on Mohammed Elhenawy, Hao Chen, and Hesham Rakha, "Traffic Congestion Identification Using Mixture of Linear Regression under Different Weather and Visibility Conditions," presented at the Transportation Research Board 94th Annual Meeting, Washington DC, United States, 2015.I was responsible for all major areas of concept formation, Matlab coding, weather data preprocessing and data analysis, as well as manuscript composition. Hao Chen was involved in INRIX data preprocessing and manuscript composition. Hesham Rakha was involved in the early stages of concept formation and contributed to manuscript edits.

Chapter 5 is based on This chapter is based on Mohammed Elhenawy and Hesham Rakha, “A Unified Automatic Congestion Identification Model Considering Visibility and Weather Conditions Using Mixture Linear Regression," under review paper. I was responsible for all major areas of concept formation, Matlab coding and data analysis, as well as manuscript composition. Hesham Rakha proposed the idea of using speed at capacity regime and contributed to manuscript edits.

Chapter 6 is based on This chapter is based on Mohammed Elhenawy and Hesham Rakha, "Congestion Prediction Using Adaptive Boosting Machine Learning Classifiers," in Transportation Research Board 93rd Annual Meeting, 2014. I was responsible for all major areas of concept formation, Matlab coding and data analysis, as well as manuscript composition. Hesham Rakha was involved in the early stages of concept formation and contributed to manuscript edits.

Chapter 7 is based on Mohammed Elhenawy, Hao Chen, and Hesham A. Rakha, "Dynamic travel time prediction using data clustering and genetic programming," Transportation Research Part C: Emerging Technologies, vol. 42, pp. 82-98, 2014. I was responsible for all major areas of concept formation, Matlab coding, and data analysis, as well as manuscript composition. Hao Chen was involved in INRIX data preprocessing and manuscript composition. Hesham Rakha was involved in the early stages of concept formation and contributed to manuscript edits.

Chapter 8 is based on Mohammed Elhenawy, Hao Chen, and Hesham Rakha, "Random Forest Travel Time Prediction Algorithm using Spatiotemporal Speed Measurements," presented at the 2014 World Congress on Intelligent Transport Systems, Detroit, Michigan, United States, 2014. I was responsible for all major areas of concept formation, Matlab coding, and data analysis, as well as manuscript composition. Hao Chen was involved in INRIX data preprocessing and manuscript composition. Hesham Rakha was involved in the early stages of concept formation and contributed to manuscript edits.

Chapter 9 is based on Mohammed Elhenawy and Hesham Rakha, "Speed and Travel Time Prediction Based on Partial Least Squares Regression," under review paper. I was responsible for all major areas of concept formation, Matlab coding, and data analysis, as well as manuscript composition. Hesham Rakha was involved in the early stages of concept formation and contributed to manuscript edits.

Chapter 10 is based on Mohammed Elhenawy and Hesham Rakha, "A Matrix Projection Approach for Predicting Freeway State Evolution and Dynamic Travel Times," presented at the Transportation Research Board 94th Annual Meeting, Washington DC, United States, 2015. I was responsible for all major areas of concept formation, Matlab coding, and data analysis, as well as manuscript composition. Hesham Rakha was involved in the early stages of concept formation and contributed to manuscript edits.

Chapter 11 is based on Mohammed Elhenawy, Abdallah Hassan, and Hesham Rakha, "Travel Time Modeling Using Spatiotemporal Speed Variation and Mixture of Linear Regression," under review paper. I was responsible for all major areas of concept formation, Matlab coding, and data analysis, as well as manuscript composition. Abdallah Hassan was involved in manuscript composition. Hesham Rakha was involved in the early stages of concept formation and contributed to manuscript edits.

Chapter 12 is based on Mohammed Elhenawy, Mohammed Almannaa, and Hesham Rakha, "Travel Time Reliability Modeling using a Mixture of Linear Regressions," under review paper. I was responsible for all major areas of concept formation, Matlab coding, and data analysis, as well as manuscript composition. Mohammed Almannaa was involved in manuscript composition. Hesham Rakha was involved in the early stages of concept formation and contributed to manuscript edits.

Chapter 13 is based on Mohammed Elhenawy, Hesham Rakha, and Ihab El-Shawarby, "Enhanced Modeling of Driver Stop-or-Run Actions at a Yellow Indication," Transportation Research Record: Journal of the Transportation Research Board, vol. 2423, pp. 24-34, 12/01/ 2014. I was responsible for all major areas of concept formation, Matlab coding, and data analysis, as well as manuscript composition. Hesham Rakha was involved in the early stages of concept formation and contributed to manuscript edits. Ihab El-Shawarby was involved in managing the approval by the institutional review board at the office of research compliance and the manuscript composition.

Chapter 14 is based on Mohammed Elhenawy, Arash Jahangiri, Hesham Rakha, and Ihab El-Shawarby, "Modeling Driver Stop/Run Behavior at the Onset of a Yellow Indication Considering Driver Run Tendency and Roadway Surface Conditions," under review journal paper. I was responsible for all major areas of concept formation, Matlab coding, and data analysis, as well as manuscript composition. Arash Jahangiri was involved in applying support vector machine to the data set and manuscript composition .Hesham Rakha was involved in the early stages of concept formation and contributed to manuscript edits. Ihab El-Shawarby was

involved in managing the approval by the institutional review board at the office of research compliance and the manuscript composition.

Chapter 15 is based Mohammed Elhenawy, Arash Jahangiri, Hesham Rakha, and Ihab. El-Shawarby, "Classification of driver stop/run behavior at the onset of a yellow indication for different vehicles and roadway surface conditions using historical behavior," presented at the 6th International Conference on Applied Human Factors and Ergonomics, Las Vegas, Nevada, USA, 2015. I was responsible for all major areas of concept formation, Matlab coding, and data analysis, as well as manuscript composition. Arash Jahangiri was involved in applying support vector machine to the data set and manuscript composition. Hesham Rakha was involved in the early stages of concept formation and contributed to manuscript edits. Ihab El-Shawarby was involved in managing the approval by the institutional review board at the office of research compliance and the manuscript composition.

Chapter 16 is based Mohammed Elhenawy, Ahmed Abdelnaeim Elbery, Abdallah Hassan Mahmoud, and Hesham. Rakha, “A Game-Theory-Based Algorithm for Traffic Control at Uncontrolled Isolated Intersections for Connected Vehicles Environments," under review paper. I was responsible for all major areas of concept formation, Matlab coding, and data analysis, as well as manuscript composition. Ahmed Abdelnaeim Elbery was involved in INTEGRATION simulation and coding and manuscript composition. Abdallah Hassan Mahmoud was involved in Matlab coding and manuscript composition. Hesham Rakha was involved in the early stages of concept formation and contributed to manuscript edits. Ihab El-Shawarby was involved in managing the approval by the institutional review board at the office of research compliance and the manuscript composition.

Chapter 1: Introduction

Nowadays, transportation systems are an important part of human activities. In recent years the dependency of people on the transportation system has increased. Currently, on average 40 percent of the world's population spends at least 1 hour on the road each day. As the dependency of people on transportation systems increases, these systems face several challenges. Congestion is one of the challenges facing transportation systems and has a direct impact on people. Congestion also has an environmental impact where it increases fuel consumption and consequently air pollution [1]. Another challenge is the incident risk which increases with the expansion of the transportation system. In the US, the Department of Transportation (DOT) reported 32,367 fatalities caused by road accidents in 2011 [2]. In China, there were 104,373 fatalities in 2003 and 67,759 fatalities in 2009 [3, 4]. Overall, 75 percent of traffic accidents are due to human errors. The above numbers suggest that there is a real need to find solutions to reduce traffic congestion and enhance transportation safety. The performance of transportation systems does not affect humans and the environment only but also a nation's economy. Another important factor of transportation systems is its capability to handle overloading such as the case of emergency mass evacuation.

Most of the traditional solutions tend to build new infrastructure such as highways and freeways. Traditional solutions need land resources and significant funding and pose social and environmental problems. Transportation engineers realized that optimizing the use of the existing transportation system can be achieved by using Intelligent Transportation Systems (ITSs) given the availability of real-time data. Data can be collected using several types of auxiliary instruments, e.g., cameras, inductive-loop detectors, Global Positioning System (GPS)-based receivers, and microwave detectors.

Definition of ITS and its History in the US

ITSs build an integrated system of people, roads and vehicles by applying and integrating communications, computers and other technologies in the field of transportation. Integrating the information and technologies can establish a large, full-functioning, real-time, accurate and efficient transportation management system.

At the core of ITSs is information that can be static and/or real-time traffic data. Collection, processing, integration and supply of information are done by many ITS tools. ITS enables travelers, transportation engineers, highway authorities and agencies to make better informed and smarter decisions by providing real-time information or on-line information about current conditions on a network. ITSs provide drivers with real-time road information and convenient services to reduce traffic congestion and to increase roadway capacity.

The United States was the first to introduce the concept intelligence into transportation systems in the 20th century [4]. Nowadays, the research and development of ITS has moved to other countries such as Japan, the European Union, South Korea, China, Singapore, the United Arab Emirates, and Qatar. In the 1970's, the United States started the Electronic Route Guidance System (EGRS), which is considered one of the initial ITS applications. Later, in 1991, the Integrated Surface Transportation Efficiency Program (ISTEA) was enacted. The Transportation Equity Act for the 21st Century TEA-21 was the next to ISTEA. An important step towards ITS was the cooperation between Federal and state departments of transportation (DOTs) and vehicle manufacturers to propose the Vehicle Infrastructure Integration (VII). The VII program enabled vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications. Recently, in 2009,

the United States Department of Transportation (USDOT) announced the ITS Strategic Research Plan, 2010-2014, which defines the strategic direction for the USDOT's ITS research program for the next five years.

Understanding Intelligent Transportation Systems

There are many intelligent transportation system applications and they are usually arranged by their primary functional intent into five systems. Figure 1 shows the five systems of the ITS. In the following subsection we will describe each system briefly.

1. Advanced Traveler Information Systems

The most famous ITS application is Advanced Traveler Information Systems (ATISs). It gives drivers real-time travel and traffic information, such as transit routes and schedules, navigation directions, and information about delays due to congestion, accidents, weather conditions, or road repair work. It helps drivers by providing them with real-time information about their precise location and the current traffic conditions on adjacent roadways. At the same time, it helps drivers by providing optimal route selection and navigation instructions. An important point is that, ATIS makes this information available on multiple platforms, both in-vehicle and out.

Recently, ATIS provides drivers with information about vacant parking spots and in some systems it allows drivers to reserve a parking spot in advance. This parking information service helps reducing traffic congestion. Studies have shown that 30 percent or more of urban traffic in large cities consists of drivers circulating as they search for parking.

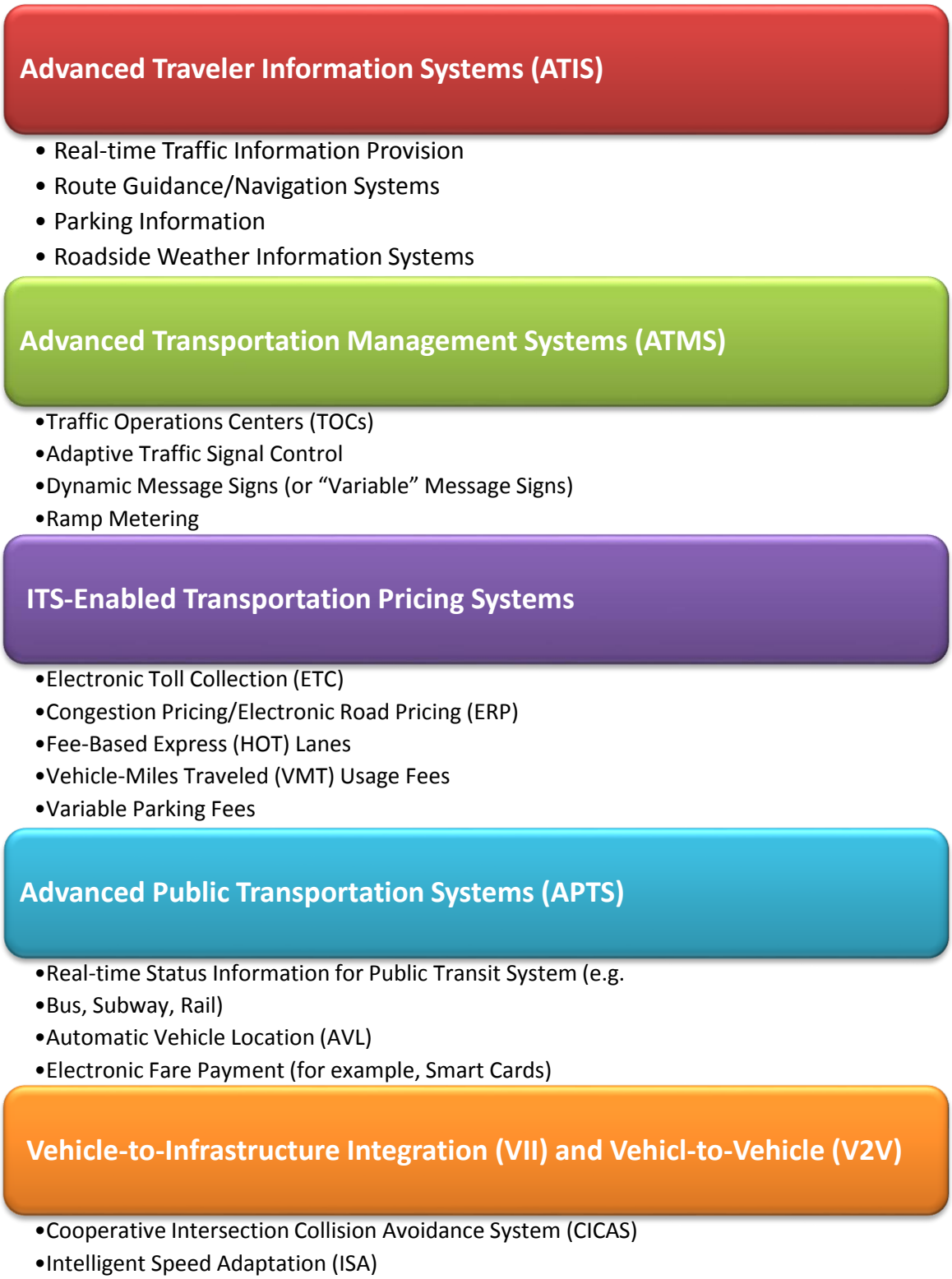


Figure 1: Five ITS Categories

2. Advanced Transportation Management Systems

Advanced Transportation Management Systems (ATMSs) mainly focus on ways to control traffic through the use of traffic control devices, such as traffic signals, ramp metering, and dynamic (or "variable") message signs. ATMSs detect traffic situations and transmit them to Traffic Operations Centers (TOCs). TOCs are centralized traffic management centers run by cities and states worldwide. ATMS creates an integrated view of traffic flow and detects accidents, dangerous weather events, or other roadway hazards by relying on information technologies to connect sensors and roadside equipment, vehicle probes, cameras, message signs, and other devices.

To highlight why ATMS is important from the traffic congestion point of view, we should know that there are 300,000 signalized intersections in the United States. These signalized intersections use static timing plans based on data collected years or decades old. The collected data to define the timing plans may be outdated and the current traffic patterns may be different from the past patterns. Studies estimated 5 to 10 percent of the congestion on major American roadways is attributed to bad signal timing. Giving traffic signals the ability to detect the presence of waiting vehicles, or giving vehicles the ability to communicate that information to a traffic signal, perhaps through DSRC-enabled communication (assuming both the vehicle and traffic signal are DSRC-equipped), could enable improved timing of traffic signals, thereby enhancing traffic flow and reducing congestion.

3. ITS-Enabled Transportation Pricing Systems

ITS plays an important role in both electronic toll collection (ETC), and congestion pricing schemes. Through ETC drivers can pay tolls automatically via a DSRC-enabled on-board device or tag placed on the windshield. Some countries such as Japan have deployed a single national ETC standard. Other countries have various highway operators' ETC systems which mean cars need to carry multiple toll collection tags on cross-country.

Some economists believe that some form of congestion pricing is needed to reduce urban congestion and emissions. Congestion pricing schemes have been applied in many cities throughout the world. Congestion pricing is a means to reduce congestion, to generate needed resources to fund investments in public transportation and to reduce the environmental impact of vehicles.

4. Advanced Public Transportation Systems

Advanced Public Transportation Systems (APTSs) improve the public transportation quality of service and increase the number of customers by applying the ATMS and ATIS technologies. APTS uses automatic vehicle location (AVL) to enable public transportation vehicles to report their current location. The traffic operations managers can build a real-time view of the status of all assets in the public transportation system using the location information. APTS gives customers complete image of the arrival and departure status (and overall timeliness) of buses and trains which makes public transportation more attractive for customers. APTS supports electronic fare payment systems for public transportation systems, where customers pay fares from their smart cards or mobile phones.

5. Vehicle-to-infrastructure Integration (VII) and Vehicle-to-vehicle (V2V) Integration

VII provides a communications link between vehicles running on the road using On-Board Equipment (OBE), and between vehicles and the roadside infrastructure using Roadside

Equipment (RSE). The main goals of VII are safety, efficiency, and convenience of the transportation system. VII uses IEEE 802.11p network protocol over a dedicated short-range communications (DSRC).

Vehicle-to-vehicle (V2V), which is also known as VANETS, is a vehicle technology that allows cars to communicate with each other. V2V offers the opportunity for significant safety improvements. V2V enables each vehicle on the roadway (inclusive of automobiles, trucks, buses, motor coaches, and motorcycles) to exchange anonymous, vehicle-based data regarding position, speed, and location (at a minimum). This information exchange enables a vehicle to identify threats and hazards, calculate risk, issue driver advisories or warnings, or take pre-emptive actions to avoid and mitigate crashes.

Combining both V2V and VII into a consolidated platform enables many ITS applications, such as adaptive signal timing, dynamic re-routing of traffic through variable message signs, lane departure warnings. Another application enabled by vehicle-to-infrastructure integration is intelligent speed adaptation (ISA). ISA helps drivers keep their speed within the speed limit by correlating information about the vehicle's position from GPS with a digital speed limit map. Based on the output of the correlation the vehicle recognizes if it is exceeding the posted speed limit. The ISA could either warn the driver to slow down or automatically slow the vehicle through automatic intervention.

Figure 2: ITS Functions and Services

As shown in Figure 2 our research in travel time prediction, congestion identification and prediction fits into driving information during travel and pre-trip information services. Our research on stop-run driver modeling and managing vehicles crossing at uncontrolled intersection fits into intersection collision prevention.

Research Objectives

The main goal of this research effort is to develop innovative solutions to address traffic congestion and driver safety issues. The developed algorithms fit into many ITS services. In particular, this study has the following five objectives:

1. Develop automatic congestion identification and prediction algorithms that are suitable for real-time applications.
2. Integrate weather and visibility conditions into the automatic congestion identification and prediction algorithms.
3. Develop travel time models using machine learning and statistical learning techniques that are suitable for real-time applications.
4. Use machine learning techniques to model driver stop-run behavior at the onset of a traffic signal yellow indication to enhance the safety of traffic signal timings.
5. Develop real-time algorithms to control the movement of autonomous/automated vehicles at intersections.

Research Contributions

The research work in this dissertation provides many significant contributions to the field of transportation. Our contributions include novel and innovative algorithms and applications of machine learning and statistical learning techniques in the field of transportation for the development automatic algorithms suitable for real-time applications. Specifically, the contributions of this research effort can be summarized as follows:

1- We develop and formulate a novel algorithm that predicts the speed of all segments of a roadway stretch (stretch-wide). The proposed algorithm determines two spaces for the multivariate predictors and multivariate responses, respectively, such that the projections of predictors (scores) and responses are equal. These new spaces are obtained through a series of matrix factorizations and multiplications. The predicted speed is used to construct the traveler trajectory and then calculate the predicted travel time. Our novel algorithm is compared with the partial least square regression (PLSR) the well-known multivariate algorithm and the outputs are quite similar, however; our novel algorithm is 4509 times faster than the PLSR in the case of the large parameters values.

2- We proposed a new measure for driver aggressiveness at traffic signalized intersections. The new measure (predictor) is based on the fact that at the onset of a yellow light, the decision whether to run the light or to stop depends on the yellow time and the time to intersection. It is clear that at the onset of a yellow light stopping is not always the best decision. Our proposed aggressiveness parameter is not simply the percentage of time a driver stops/runs during the yellow interval. The new measure is based on the number of runs the driver makes when both the time-to-intersection at the onset of the yellow indication is greater than the yellow time and the vehicular speed is equal to or greater than the posted speed limit. This number is incorporated into a Bernoulli-Beta model to estimate the driver aggressiveness.

3- We considered driver aggressiveness as a predictor to develop driver stop-run models. Adding this predictor increases the classification accuracy and reduces the false alarm which will help increase user's acceptance of the advanced driver assistance system.

4- We developed a novel game theory framework algorithm for connected vehicles equipped with Cooperative Adaptive Cruise Control (CACC) to control vehicles passing through uncontrolled intersections. Our algorithm is inspired by the chicken game to

model the intersection and to define the players, strategies and payoffs relevant to the intersection dynamics.

5- To the best of our knowledge, we developed the first automatic congestion identification algorithm that incorporates weather conditions and visibility levels.
6- We developed a unified model for congestion identification applicable to any highway road system in the United States.
7- We exploited the Adaptive Boosting machine learning algorithm (AdaBoost) to develop an automatic congestion prediction algorithm which is considered one of a few research efforts to tackle this important problem.
8- We modeled travel time using a mixture of linear regression and historical data. The resultant model can be used to predict travel time and travel time reliability. Other algorithms divide the day into different time slots and fit the data of these slots to a mixture of components. Our resultant model includes means and proportional parameters that are functions of the speeds inside a window from the departure time and backwards.
9- Machine learning techniques are exploited in modeling the stop-run behavior of the driver at the onset of a yellow indication for various roadway surface conditions and different vehicle types.

Dissertation Layout

This dissertation consists of three main parts which are organized into 17 chapters. Each part is assigned to one of the three problems we addressed. Chapter 1 briefly introduces ITSs, its services and the mapping between our algorithms and the ITS services. The first part consists of the resulting five papers from the congestion identification and prediction research. Each paper constitutes one chapter. The chapters are organized so as to reflect the advances and development of our research with time. The second part is assigned to the travel time modeling, which spans from the seventh chapter to twelfth chapter. The chapters propose different algorithms to predict travel time, travel time interval and travel time reliability. The third part of the dissertation addresses the driver safety at intersections. It consists of four papers each represented in a chapter. Chapters 13, 14 and 15 show the work related to the driver stop-run modeling at the onset of the yellow indicator. The last chapter in this part showcases the proposed game control algorithm to resolve conflicts between crossing vehicles at uncontrolled intersections. Chapter 17 provides a summary of the research done in this dissertation and recommended future work.

Publications

The research presented in this dissertation has resulted in the following publications:

1. Journal Publications

1- Mohammed Elhenawy, Hao Chen, and Hesham Rakha, "Dynamic travel time prediction using data clustering and genetic programming," *Transportation Research Part C: Emerging Technologies*, vol. 42, pp. 82-98, 2014.
2- Mohammed Elhenawy, Hesham. Rakha, and Ihab El-Shawarby, "Enhanced Modeling of Driver Stop-or-Run Actions at a Yellow Indication," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2423, pp. 24-34, 12/01/ 2014.

3- Mohammed Elhenawy, and Hesham Rakha, "Automatic Congestion Identification Using Two-Component Mixture Models," *Transportation Research Record: Journal of the Transportation Research Board*, 2015 (in press).

2. Conference Publications

1- Mohammed Elhenawy, Hesham Rakha, and Hao Chen, "An automated statistically-principled bottleneck identification algorithm (ASBIA)," in Intelligent Transportation Systems - (ITSC), 2013 16th International IEEE Conference on, 2013, pp. 1846-1851.
2- Mohammed Elhenawy, and Hesham Rakha, " Automatic Congestion Identification Using Two-Component Mixture Models," presented at the Transportation Research Board 94th Annual Meeting, Washington DC, United States, 2015.
3- Mohammed Elhenawy, Hao Chen, and Hesham Rakha, "Traffic Congestion Identification Using Mixture of Linear Regression for Different Weather and Visibility Conditions," presented at the Transportation Research Board 94th Annual Meeting, Washington DC, United States, 2015.
4- Mohammed Elhenawy and Hesham Rakha, "Congestion Prediction Using Adaptive Boosting Machine Learning Classifiers," in Transportation Research Board 93rd Annual Meeting, 2014.
5- Mohammed Elhenawy, Hao Chen, and Hesham Rakha, " Dynamic Travel Time Prediction Using Genetic Programming," presented at the Transportation Research Board 93th Annual Meeting, Washington DC, United States, 2014.
6- Mohammed Elhenawy, Hao Chen, and Hesham Rakha, "Random Forest Travel Time Prediction Algorithm using Spatiotemporal Speed Measurements," presented at the the 2014 World Congress on Intelligent Transport Systems, Detroit, Michigan, USA, 2014.
7- Mohammed Elhenawy and Hesham Rakha, "A Matrix Projection Approach for Predicting Freeway State Evolution and Dynamic Travel Times," presented at the Transportation Research Board 94th Annual Meeting, Washington DC, United States, 2015.
8- Mohammed Elhenawy, Hesham Rakha, and Ihab El-Shawarby, "Enhancing Driver Behavior Modeling at Signalized Intersections using a Driver Aggressiveness Measure and Machine Learning Techniques," presented at the Conference on Agent-Based Modeling in Transportation Planning and Operations, 2013.
9- Mohammed Elhenawy, Arash Jahangiri, Hesham Rakha, and Ihab El-Shawarby, "Classification of driver stop/run behavior at the onset of a yellow indication for different vehicles and roadway surface conditions using historical behavior," presented at the 6th International Conference on Applied Human Factors and Ergonomics, Las Vegas, Nevada, USA, 2015.

3. Under review Papers

1- Mohammed Elhenawy, Arash Jahangiri, Hesham Rakha, and Ihab El-Shawarby, "Modeling Driver Stop/Run Behavior at the Onset of a Yellow Indication Considering Driver Run Tendency and Roadway Surface Conditions," submitted to Accident Analysis & Prevention Journal (Second round of review).
2- Mohammed. Elhenawy and Hesham Rakha, "Speed and Travel Time Prediction Based On Partial Least Squares Regression,".

3- Mohammed. Elhenawy, Abdallah Hassan, and Hesham Rakha, "Travel Time Modeling Using Spatiotemporal Speed Matrix and Mixture of Linear Regression,".
4- Mohammed. Elhenawy, Mohammed Almannaa, and Hesham Rakha, "Travel Time Modeling and Reliability Using Mixture of Linear Regressions,".
5- Mohammed Elhenawy, Ahmed Abdelnaeim Elbery, Abdallah Hassan Mahmoud, and Hesham. Rakha, " A Game-Theory-Based Algorithm for Traffic Control at Uncontrolled Isolated Intersections for Connected Vehicles Environments,".
6- Mohammed. Elhenawy and Hesham Rakha, "A unified model for Automatic Congestion Identification at Different Levels of Visibility and Weather Conditions Using Mixture of three Linear Regression,".

References

[1] J. Shawe-Taylor, T. De Bie, and N. Cristianini, "Data mining, data fusion and information management," *Intelligent Transport Systems, IEE Proceedings*, vol. 153, pp. 221-229, 2006.

[2] Available: http://www-nrd.nhtsa.dot.gov/Pubs/811701.pdf

[3] Z. Junping, W. Fei-Yue, W. Kunfeng, L. Wei-Hua, X. Xin, and C. Cheng, "Data-Driven Intelligent Transportation Systems: A Survey," *Intelligent Transportation Systems, IEEE Transactions on,* vol. 12, pp. 1624-1639, 2011.

[4] A. Sheng-hai, L. Byung-Hyug, and S. Dong-Ryeol, "A Survey of Intelligent Transportation Systems," in *Computational Intelligence, Communication Systems and Networks (CICSyN), 2011 Third International Conference on*, 2011, pp. 332-337.

Part I

Chapter 2: An Automated Statistically-principled Bottleneck Identification Algorithm (ASBIA)

This chapter is based on
Mohammed Elhenawy, Hesham Rakha, and. Hao Chen, "An automated statistically-principled bottleneck identification algorithm (ASBIA)," in Intelligent Transportation Systems - (ITSC), 2013 16th International IEEE Conference on, 2013, pp. 1846-1851.

Abstract

Bottlenecks are key features of any freeway system. The deployment of stationary sensors and proliferation of mobile vehicle probes provides researchers with a wealth of data that can be used for the automatic identification of active freeway bottlenecks. In this paper we introduce an automated statistically principled algorithm to characterize traffic into two states: free-flow or congested and subsequently identify the spatiotemporal activation of bottlenecks. The proposed algorithm uses speed measurements over short temporal and spatial intervals and segments, respectively to identify the status of a segment while accounting for spatiotemporal correlations and interactions. The outputs of the algorithm are the status of the roadway segment (free-flow or congested) and the confidence level of the test (p-value). The experimental results based on archived data from the northbound Interstate 5 (I-5) corridor in the Portland, Oregon, metropolitan region demonstrates significant improvements over state-of-the-art bottleneck identification algorithms.

Introduction

Traffic congestion has increased globally as a result of increased motorization, population growth, and changes in population density. Congestion may cause various social, environmental, and economic problems. According to the FHWA publication FHWA-HOP-11-034, 40 percent of all congestion nationwide can be attributed to recurring congestion. Recurring congestion is classified into “mega” where the traffic demand overwhelms entire regions or large facilities (e.g., interchanges or corridors). Some of it periodically overwhelms “subordinate” – locations on the highway system by temporarily being loaded by huge traffic demand exceeds the physical capacity of the roadway. The physical capacities of the subordinate locations are sufficient during the off-peak hours. The later congestion type is the recurring “localized” bottlenecks from which commuters suffer every day. The cause, location, time of day, and approximate duration of localized” bottlenecks can be predicted with good accuracy. On the other hand, congestion which is caused by random events, such as crashes, is nonrecurring and is hard to predict.

Traffic congestion reduces the utilization of the transportation infrastructure and increases traveler travel times, air pollution, and fuel consumption levels. In the prevalent literature and practice the terms “congestion” and “bottlenecks” are often used interchangeably. A definition of a bottleneck is that, it is subordinate locations along a highway that need to be

This work was jointly supported by the Mid-Atlantic University Transportation Center and the Virginia Department of Transportation.

Mohammed Elhenawy is a Ph.D. candidate with Bradley Department of Electrical and Computer Engineering Electrical & Computer Engineering MC 0111,1185 Perry St.,Blacksburg, VA 24061Phone: (540) 231-6646 Fax: (540) 231-3362(e-mail: mohame1@vt.edu)

Hao Chen is a Ph.D. candidate with the Charles E. Via, Jr. Department of Civil and Environmental Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA 24060 USA (e-mail: haochen@vt.edu).

Hesham A. Rakha is a Professor with the Charles E. Via, Jr. Department of Civil and Environmental Engineering and Director of the Center for Sustainable Mobility, Virginia Polytechnic Institute and State University, Blacksburg, VA 24060 USA (Correspondence author, phone: 540-231-1505, e-mail: hrakha @vt.edu).

fixed, and not necessarily the knee-jerk expectation to rebuild the entire facility. Identifying bottlenecks and applying one or two cost wise corrections to inefficient subordinate locations on a facility may be all that is needed to improve the condition. A bottleneck is distinguished from a systemic congestion problem in that it occurs at a specific location, and not pervasively along the entire corridor. In summary, recurring congestion is made up of localized congestion/bottlenecks and systemic congestion. A great portion of the recurring congestion problem can be solved by bottleneck improvements that handle the additional traffic that is now freed up.

Freeway bottlenecks are important contributors to congestion. The congestion formed upstream of active freeway bottlenecks is a complex spatiotemporal nonlinear dynamic process. One goal of Advanced Traffic Management Systems (ATMSs) within the Intelligent Transportation System (ITS) framework is to identify bottlenecks within the transportation system. Correctly identifying traffic bottlenecks is critical in understanding traffic dynamics and characterizing the spatiotemporal interactions and correlations that exist along a roadway segment. Once these spatiotemporal interactions have been characterized, transportation engineers can develop appropriate solutions to alleviate congestion and improve the performance of the freeway network. Dynamically managed lanes, speed harmonization (variable speed limits), and ramp modifications are among the bottleneck solution approaches which are cost effective.

Active freeway bottlenecks are not only difficult to identify but more importantly the spatiotemporal evolution of congestion upstream of these bottlenecks is extremely difficult to quantify. Furthermore, the spatiotemporal evolution of congestion upstream of these bottlenecks varies from one day to another. For example, the spatiotemporal location of active traffic bottlenecks on workdays may differ significantly from active traffic bottlenecks on weekends and bottlenecks during the morning peak are not the same as traffic bottlenecks during the evening peak. Consequently, without automatic bottleneck identification algorithms, transportation engineers might drive along the freeway for several days and at different times, noting locations downstream of which traffic is free-flowing, but upstream of which traffic is significantly slower. This is a time-consuming approach to identify bottlenecks and asses their impact.

The deployment of sensors has laid the foundation for researchers to develop algorithms to identify freeway bottlenecks. These algorithms automatically analyze archived loop detector data and identify potential bottlenecks in terms of spatial location and time of activation so that solutions can be developed to eliminate or alleviate them. Automatic algorithms are used in the initial screening of bottlenecks over a large freeway network to save time and resources. These automatic algorithms will become very important in the near future when connected vehicles become commercially available. We expect developing and running congestion avoidance and control algorithms on a new generation of road controllers. It uses the automatic bottleneck identification algorithms to detect congestion regions. Then it communicates with vehicles approaching congestion and provides them with strategies to eliminate or reduce these bottlenecks. Some of these systems are known as variable speed limit or speed harmonization systems. The state-of-the-art automatic bottleneck identification algorithms include rule-based, contour-map-based, and simulation-based methods.

Several rules are developed in [1] to automatically identify the location and time of bottleneck activation, deactivation, and delay caused by the bottleneck. The algorithm was validated using three months' worth of loop detector data in San Diego area. Additional follow-up studies focused on the field implementation of the algorithm [2-4]. Research was also

conducted on the real-time identification and display of active bottlenecks using graphical tools and then used in developing travel time reliability measures. The bottleneck identification algorithms were further used to predict the propagation of congestion using archived data. In addition, new approaches were developed in [5]to identify bottlenecks and calibration of traffic simulation software. The identification process was conducted using percentile speeds from archived data over multiple days. The bottleneck identification model calibration included a three-step process of visual assessment, bottleneck area matching and detailed speed calibration. The experimental results demonstrated the accuracy of the method in bottleneck identification and the improvement of calibration procedure for traffic simulation model calibration. However, the above mentioned approaches either require some pre-defined parameters or a complicated computation process. A simple method is still desired that can be implemented in real-time with minimum calibrated parameters.

Considering the research need, a simple statistical approach is proposed in this paper to identify freeway bottlenecks. In the proposed approach no pre-defined parameters are required. The speed measurement at every segment at each time interval and the neighboring data in the spatiotemporal domain are used to identify the binary status of the roadway segment (either free-flow or congested) and the corresponding confidence level (p-value). The field data from the Interstate 5 corridor in the Portland, Oregon are used to evaluate the performance of the proposed algorithm in the identification of congestion and bottlenecks.

The remainder of this paper is organized as follows. First, the proposed statistically-principled bottleneck identification algorithm is presented. Subsequently, the field data used to validate the algorithm and compare its performance to the state-of-the-art Chen algorithm is described [1]. Finally, conclusions and recommendations for future work are presented.

The Proposed Algorithm (ASBIA)

A t-test is used to test hypotheses about the population mean when the population standard deviation is unknown. Its basic assumption is that the population distribution is normal. The t-test has the advantages that it is robust with departures from normality assumption and that it can deal with small sample sizes[6]. The proposed method is mainly based on the well-known one sample t-test. The one sample t-testis used to examine the mean difference between the sample μ_{obs} (the observed mean) and the known value of the population mean μ_0 (the hypothesized mean). In a one sample t-test, we know the population mean. We draw a random sample from the population and then compare the sample mean with the population mean and make a statistical decision as to whether or not the sample mean is different from the population mean. The formula for the t-test is a ratio that follows a Student's t distribution if the null hypothesis is supported. The numerator is the difference between the observed mean and the hypothesized mean. The denominator is a measure of the variability or dispersion of the scores. The t-ratio is a kind of signal-to-noise ratio that follows the Student’s t distribution with n-1 degrees of freedom where n is sample size.

1. ASBIA

ASBIA makes use of the correlation between a point in the spatiotemporal domain and its neighbors. This correlation exists along both the time and space axes. The proposed algorithm uses speed measurements at each point x and its neighbors in the time-space domain to evaluate the status of speeds at x. The output of the algorithm is the status of the roadway segment (free-flow or congested) and the confidence level of the test (p-value). Specifically, the ASBIA makes use of the correlation between a point in the time-space domain and its neighbors. This

correlation exists along both the time and space domains (i.e. temporal and spatial correlations). The algorithm uses the fact that points in close proximity (both temporally and spatially) to any point x provide additional information about point x and makes the following two assumptions:

1. The algorithm assumes a two-phase traffic flow theory, where traffic states are either free-flow or congested.
2. The speed data are assumed to be sampled from two component Gaussian distributions and modeled as a mixture model. The first component represents the congested regime and all speed measurements in the congested regime are drawn from this component. The second component represents the uncongested regime and all speed measurements within this regime are drawn from free-flow conditions.

The multi-state model can be based on various component distributions. The two-component multi-state models based on the normal distribution is presented below. The parameter λ is the mixture proportion, i.e., the probability of the traffic stream speed is in the first state; and u is the traffic stream space-mean-speed.

$$f(u|\lambda,\mu_1,\mu_2,\sigma_1,\sigma_2) = \lambda\frac{1}{\sqrt{2\pi}\sigma_1}e^{\frac{(u-\mu_1)^2}{2{\sigma_1}^2}} + (1-\lambda)\frac{1}{\sqrt{2\pi}\sigma_2}e^{\frac{(u-\mu_2)^2}{2{\sigma_2}^2}} \quad (1)$$

where, (μ_1,σ_1) and (μ_2,σ_2) are the mean and standard deviation of the first and second component distributions, respectively.

Unlike the Chen et al. heuristic algorithm, ASBIA is a statistically-principled algorithm. ASBIA uses a δ1xδ2 window to select a sample from the speed matrix and identify the status of the point at the center of the window as shown in Figure 3. The algorithm simply consists of the following steps:

1. Move a window $\delta_1 x \delta_2$ over the time-space domain such that the window scans all the data points in the domain.
2. If the speed measurements within the spatiotemporal window are equal then the subject point (point 5 in Figure 3) is identified as free-flow if the speed is greater than the speed-at-capacity (u_c), otherwise it is identified as congested.
3. If the speed values within the spatiotemporal window are not constant, then we use the t-test to characterize the status of window center point (point 5 in Figure 3). The null hypotheses H_0: $\mu_{obs} \leq u_c$ and the alternative hypothesis is $\mu_{obs} > u_c$. The center point is congested if the test fails to reject the null hypothesis; otherwise it is considered free-flow if we reject the null hypothesis.

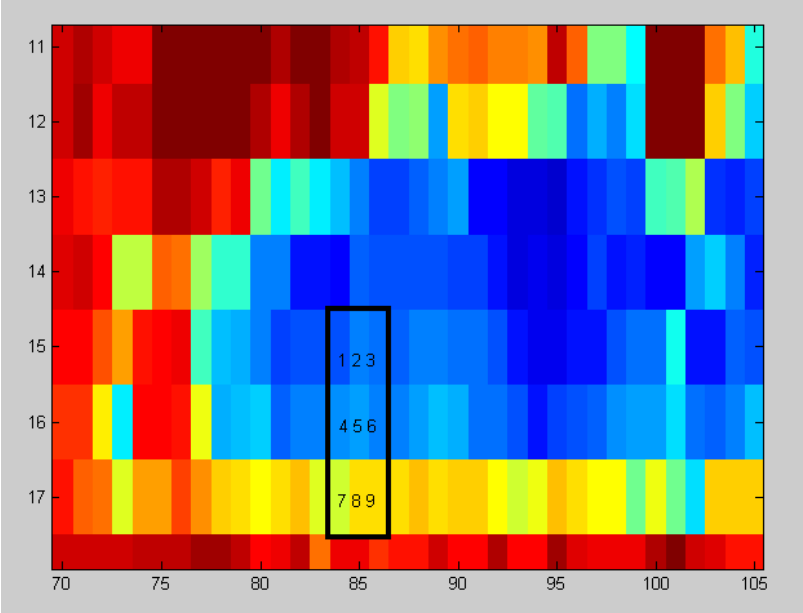


Figure 3: Example Illustration of ASBIA Algorithm

Experimental Work

In this paper we use the pair method to compare the performance of the proposed algorithm with the well-known Chen et al. algorithm [1]. The Chen et al. algorithm uses the presence of a sustained speed gradient between a pair of upstream-downstream detectors to identify bottlenecks. The Chen et al. algorithm compares the speed reading at adjacent detectors: x_i (upstream) and x_j (downstream) each 5-minute time step to identify bottlenecks. The algorithm identifies a bottleneck between these two points when the speed at the upstream detector is below the maximum speed threshold u_i, and the difference in the speeds between the downstream and upstream detectors is above a speed differential threshold Δu. In this setting Chen et al. assume that a bottleneck separates congested and uncongested conditions. Chen et al. set the speed threshold at 64 km/h (40 mi/h), and the speed differential at 32 km/h (20 mi/h) using data aggregated at 5 minute intervals.

In order to test the performance of ASBIA and compare it to the Chen et al. method, we used a dataset obtained from an earlier study to test and validate the algorithm [4]. The dataset consists of 24 days' worth of data. The data consists of high-data-quality, midweek non-holiday days between February and December 2008. The data were collected from archived data from the northbound Interstate 5 (I-5) corridor in the Portland, Oregon, metropolitan region. This road segment is 22-mi (35-km) in length. Along the segment there are 22 detectors, two of them are ignored because of their poor data quality. Each day included readings from 20 detectors, at the lowest available resolution of 20 s between 5:00 a.m. and 10:00 p.m. The baseline, or the ground truth that was used to evaluate the ASBIA algorithm was defined using a manual procedure. During this manual procedure, the activation and deactivation times of each candidate bottleneck were carefully diagnosed and verified using oblique curves of cumulative vehicle arrival versus time and cumulative occupancy (or speed) versus time constructed from data measured at neighboring freeway loop detectors [7-10].

In the experimental work that has been done we tested several window shapes and sizes. The window size should be large enough to include the largest possible number of speed reading

to conduct the t-test. On the other hand the window should be small to include speed readings from the close neighbors only. In the experimental work we have used three window shapes with different sizes. There are three shapes square, a vector along the time axis, and a vector along the spatial dimension.

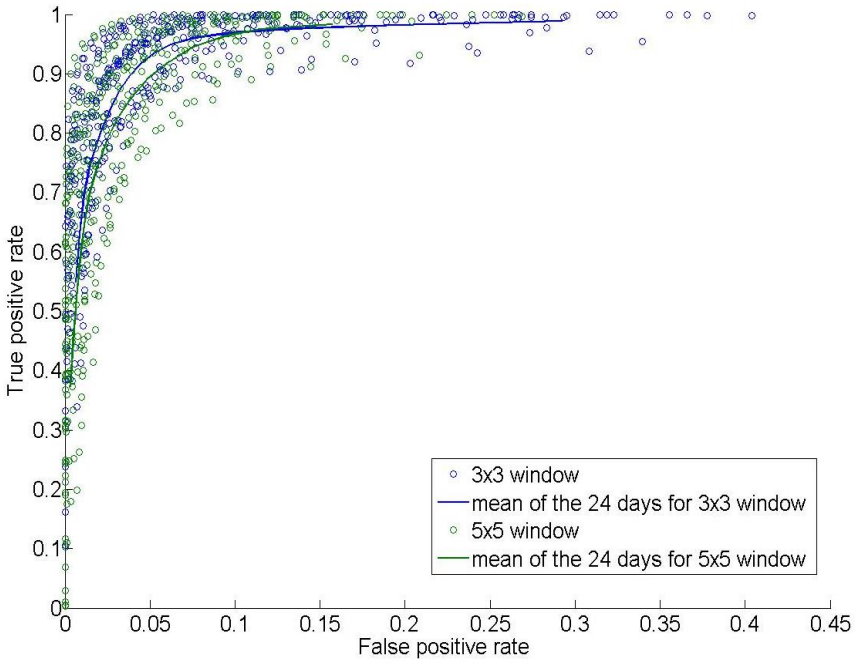
1. First Experimental Set

In the initial experiments the performance of the algorithm using two square windows of sizes 3x3 and 5x5 were compared in an attempt to identify the optimum parameters for experimentation using a0.05 α level. The receiver operating characteristic (ROC) plot was also generated for comparison purposes[11]. The ROC is created by plotting the true positive rate (TPR) vs. the false positive rate (FPR) at various νsettings. The true positive rate (TPR) and false positive rate (FPR) are computed using Equations (2) and (3).

$$\text{TPR} = \frac{\text{True detected positive}}{\text{Actual positive}} \quad (2)$$

$$\text{FPR} = \frac{\text{False detected positive}}{\text{Actual negative}}, \quad (3)$$

Where positive in our context is the correct identification of a congested or free-flow state and negative is the false identification of a congested or free-flow state. The experiment above was repeated for the five time aggregations and the ROC curves are plotted for each time aggregation. From the curves we see that the use of a 3x3 window results in an enhanced algorithm performance compared to the use of a 5x5 window. The 3x3 window has higher TPR compared to the large window at the same FPR for the different temporal aggregations considered. The figure shows only two different temporal aggregations because of the limited space.



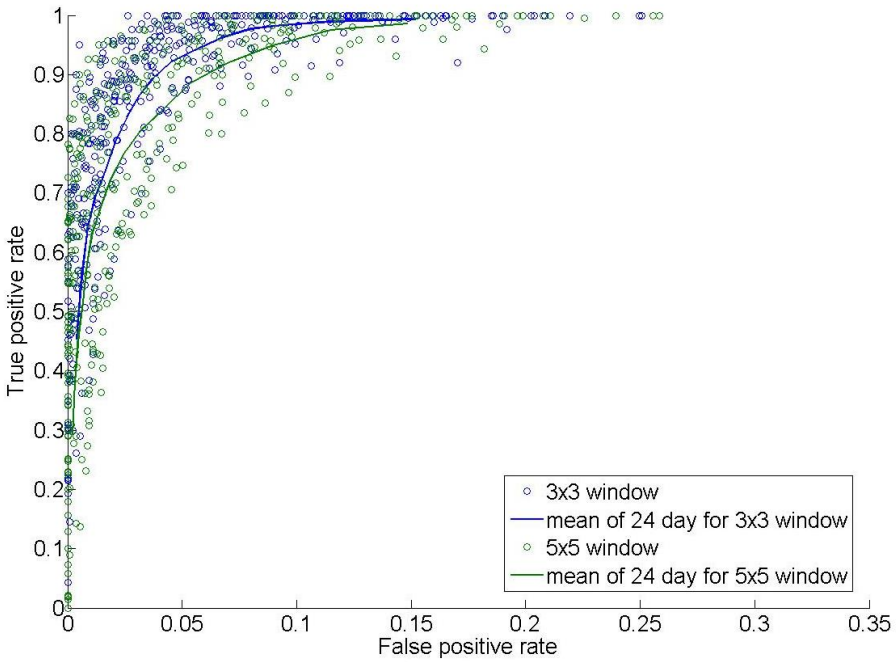


Figure 4: ROC Curve for Square Windows for 1-Minute Aggregation (top) and 5-Minute Aggregation (bottom)

2. Second Experimental Set

In the first experiment set different 2D (square) window sizes were considered. In the second set of experiments other window shapes were considered. The first window shape is a 1D window along the temporal domain. Specifically, different window sizes were considered including: a 3x1, 5x1, and 7x1 window. The result of this set of experiments is shown in Figure 5. The results show that the use of a 3x1 window produces the worst performance for all temporal aggregation levels. While the use of a 7x1 window produces the best algorithm performance.

3. Third Experimental Set

In this set of single dimension windows were considered; however in this case the windows were spatial windows. As demonstrated in Figure 6 the use of a 1x5 window results in the best algorithm performance and the use of 1x3 window results in the worst performance. After identifying the optimum window size for each of the shapes, the next step entails comparing the ROC curves using all window shapes (2D and 1D) and Chen et al. algorithm.

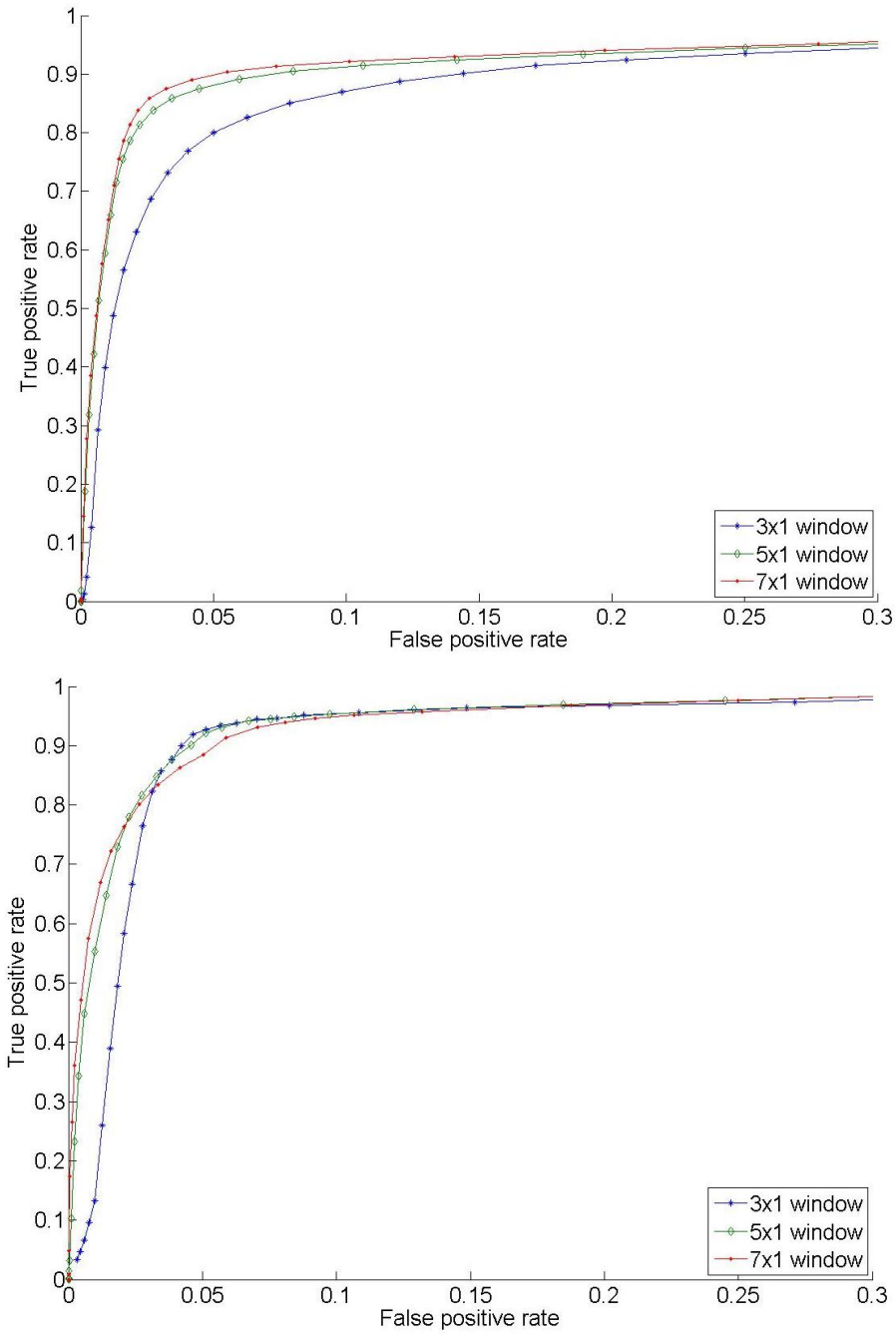


Figure 5: ROC Curves for Temporal Windows for 1-Minute Aggregation (top) and 5-Minute Aggregation (bottom)

As is the case with any heuristic algorithm, the Chen et al. method has several parameters that require calibration to achieve good performance. Wieczorek et al. [4] studied large combinations of different settings of the Chen et al. method and identified the optimum parameter settings. The best parameter settings for this dataset are u_i= 35 mph, Δu = 20 mph, with data aggregated at 3-minute intervals, as described in [4]. We compared the ROC curves produced by our proposed algorithm using different window shapes to the Chen et al. algorithm results. We examined the ROC visually to identify the best congestion identification models.

Figure 7shows that the proposed ASBIA model is the best model when using 1-D temporal window or 2-D spatiotemporal window. Alternatively, just using the spatial dimension does to produce adequate results; however it still operates more efficiently when compared to the Chen et al. algorithm, if our application allows for a false positive rate greater than or equal to 0.06. Examining the figure we can conclude that if we are interested in a false positive rate less than or equal to 0.04 and high true positive rate the best choice is the 7x1 temporal window. Another important conclusion is that if our traffic application allows the false positive rate to be greater than or equal to 0.04, 3x3 window produces optimum algorithm performance.

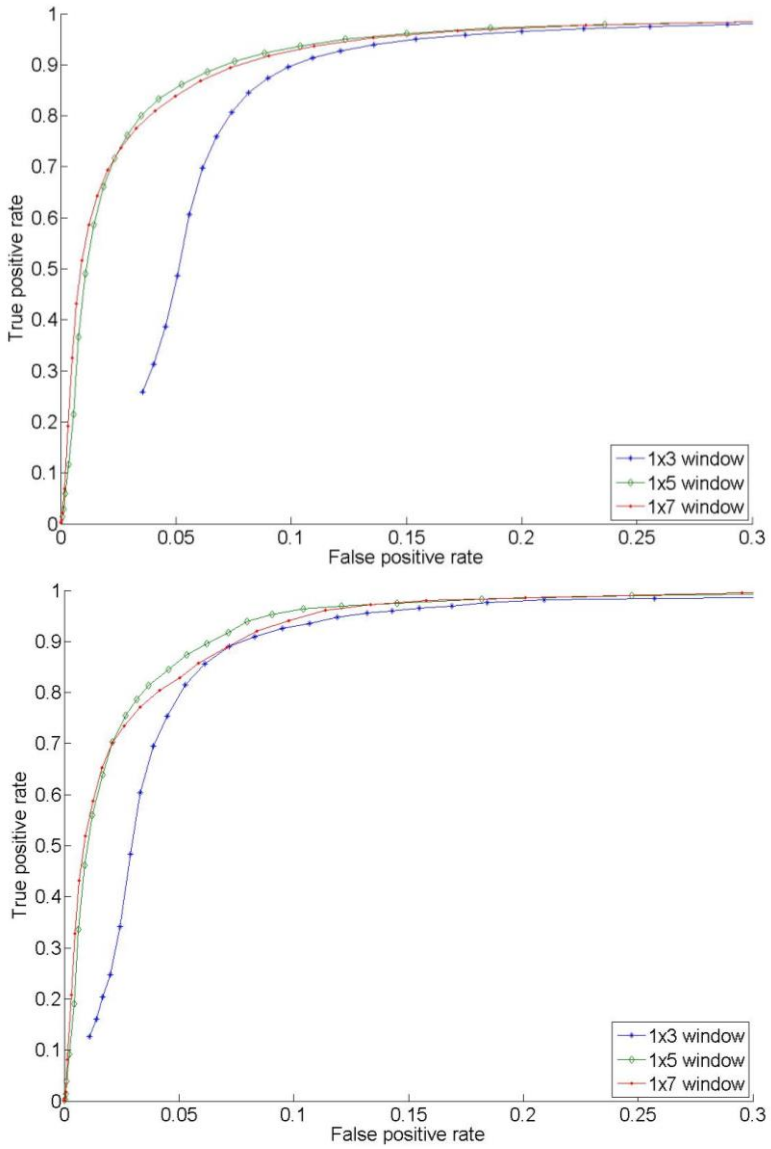


Figure 6: ROC Curve of the ASIBA for 1 Minute Aggregation Data and the other Plot Is For 5 Minute Aggregation Data

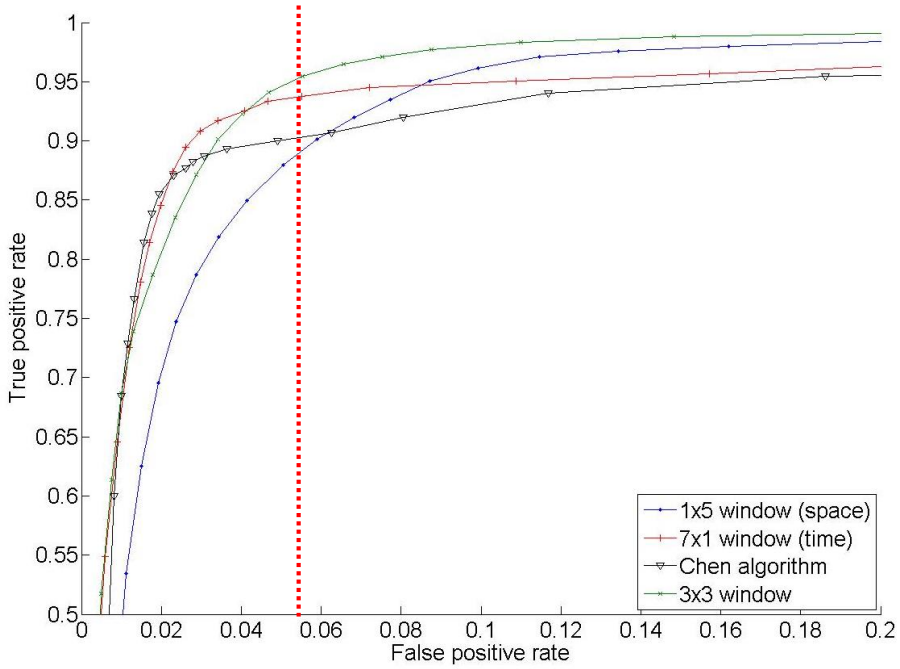


Figure 7: Comparison of ASBIA and Optimized Chen et al. Algorithm Results

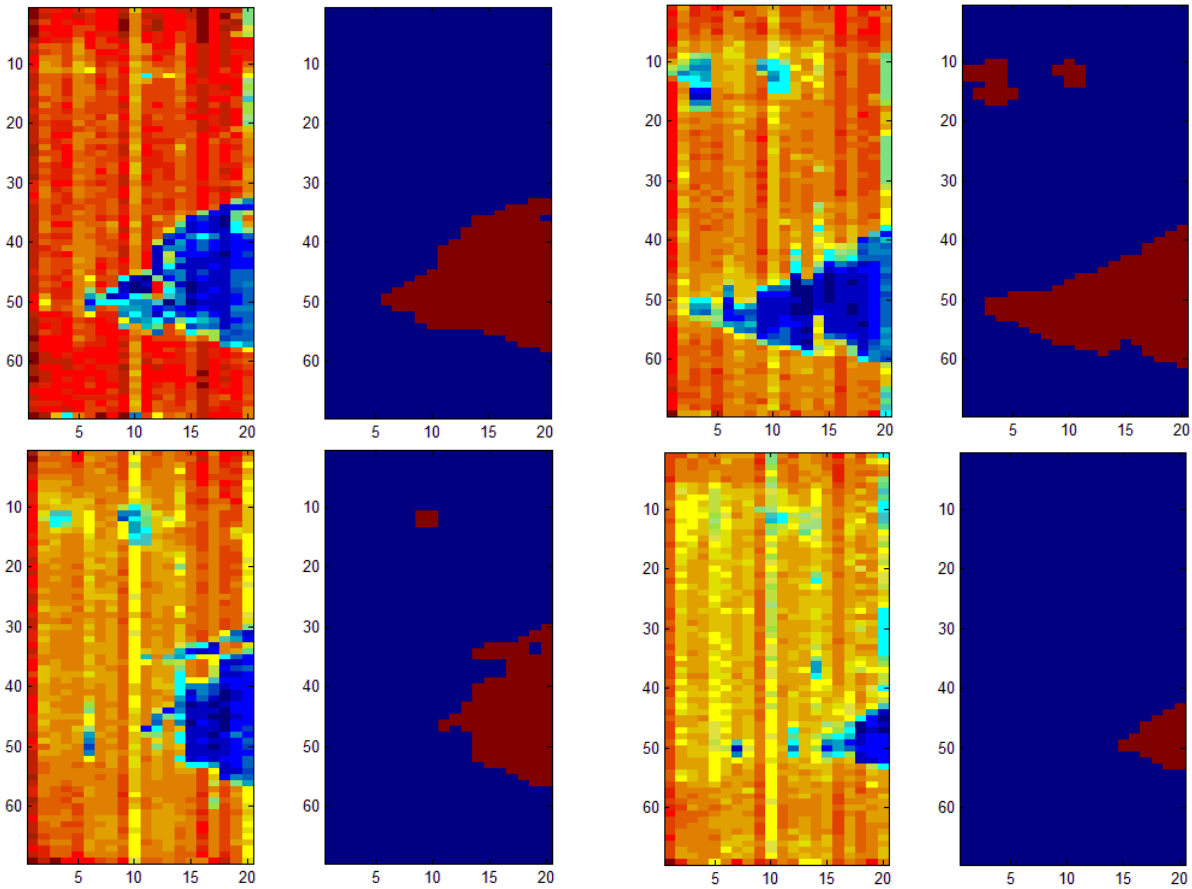


Figure 8: Spatiotemporal Congestion Regions Identified by ASBIA

Conclusion and Future Work

In this paper we introduced a novel automated statistically principled algorithm to characterize traffic into two states: free-flow or congested and subsequently identify the spatiotemporal activation of bottlenecks. The proposed algorithm uses speed measurements over short temporal and spatial intervals/segments to identify the status of a segment while accounting for spatiotemporal correlations and interactions. The outputs of the algorithm are the status of the roadway segment (free-flow or congested) and the confidence level of the test (p-value). The experimental results based on archived data from the northbound Interstate 5 (I-5) corridor in the Portland, Oregon, metropolitan region demonstrates significant improvements over the Chen et al. bottleneck identification algorithms. It should be noted that in some cases the data deviated from the normality assumption, which may affect the performance of the proposed ASBIA algorithm. Consequently, further work is recommended to attempt to improve the performance of the ASBIA algorithm by using data transformations to ensure that the normality assumption is satisfied. We will also try using other nonparametric hypothesis tests that do not require the normality condition. Finally the ASBIA is currently being used to enhance travel time prediction models.

Acknowledgment

The authors thank Mr. Wieczorek and Dr. Robert Bertini for providing and sharing the field data and its ground truth that were used in this study.

References

[1] C. Chen, A. Skabardonis, and P. Varaiya, "Systematic identification of freeway bottlenecks," *Freeway Operations and Traffic Signal Systems 2004*, pp. 46-52, 2004.

[2] R. L. Bertini, H. Li, J. Wieczorek, and R. J. Fernández-Moctezuma, "Using Archived Data to Systematically Identify and Prioritize Freeway Bottlenecks," in *10th International Conference on Application of Advanced Technologies in Transportation, Athens, Greece*, 2008.

[3] H. Li and R. L. Bertini, "Comparison of algorithms for systematic tracking of patterns of traffic congestion on freeways in Portland, Oregon," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2178, pp. 101-110, 2010.

[4] J. Wieczorek, R. J. Fernández-Moctezuma, and R. L. Bertini, "Techniques for Validating an Automatic Bottleneck Detection Tool Using Archived Freeway Sensor Data," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2160, pp. 87-95, 2010.

[5] X. Ban, L. Chu, and H. Benouar, "Bottleneck identification and calibration for corridor management planning," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1999, pp. 40-53, 2007.

[6] L. Ott and M. Longnecker, *An introduction to statistical methods and data analysis*, 6th ed. Australia ; United States: Brooks/Cole Cengage Learning, 2010.

[7] R. L. Bertini and A. M. Myton, "Use of performance measurement system data to diagnose freeway bottleneck locations empirically in Orange County, California," *Freeway Operations, High-Occupancy Vehicle Systems, Traffic Signal Systems, and Regional Transportation Systems Management 2005*, pp. 48-57, 2005.

[8] Cassidy, M. J., and R. L. Bertini., "Observations at a Freeway Bottleneck," presented at the 14th International Symposium on Transportation and Traffic Theory, Jerusalem, Israel, 1999.

[9] Cassidy, M. J., and J. R. Windover, "Methodology for Assessing Dynamics of Freeway Traffic Flow," *Transportation Research Record*, vol. 1484, pp. 73–79, 1995.
[10] M. J. Cassidy and R. L. Bertini, "Some traffic features at freeway bottlenecks," *Transportation Research Part B-Methodological*, vol. 33, pp. 25-42, Jan 1999.
[11] V. Bewick, L. Cheek, and J. Ball, "Statistics review 13: receiver operating characteristic curves," *Crit Care*, vol. 8, pp. 508-12, Dec 2004.

Chapter 3: Automatic Congestion Identification Using Two-Component Mixture Models

This chapter is based on
Mohammed Elhenawy and Hesham Rakha, "Automatic Congestion Identification Using Two-Component Mixture Models," Transportation Research Record: Journal of the Transportation Research Board, 2015.

Abstract

Automatic identification of traffic congestion is a very important component of any Intelligent Transportation System. These systems need computer algorithms to identify current congestion and predict future congestion evolution. The output of the congestion identification algorithm enables various users to be better informed and make safer, more coordinated, and smarter use of transportation networks. This paper proposes a new automatic congestion identification algorithm that assumes the speed data are drawn from a two-component mixture model. The first component represents the speed distribution in congestion, while the second component is the free-flow speed distribution. The proposed algorithm is first calibrated using historical speed data using a two-component mixture model. A free-flow speed threshold, based on the estimated parameters of the free-flow speed distribution, is set. Subsequently, a road segment is identified as free-flow if its speed is greater than the threshold and congested if its speed is less than the threshold. The mixture components consider lognormal and gamma skewed distributions and normal symmetric distributions. The proposed algorithm is tested using two real datasets collected from two different roadways and is demonstrated to produce good performance.

Introduction

Traffic congestion has increased globally as a result of increased motorization, population growth, and changes in population density. Congestion may cause various social, environmental, and economic problems. According to the FHWA publication An Agency Guide on How to Establish Localized Congestion Mitigation Programs, 40% of all congestion nationwide can be attributed to recurring congestion [1]. “Mega” recurring congestion occurs when the traffic demand overwhelms entire regions or large facilities (e.g., interchanges or corridors). Some of this congestion periodically overwhelms subordinate locations on the highway system by temporarily loading them with huge traffic demands that exceed their physical capacities, which are sufficient during the off-peak hours. Another congestion type is the recurring, localized bottleneck that commuters suffer every day. The cause, location, time of day, and approximate duration of localized bottlenecks can be accurately predicted. However, congestion caused by random events, such as crashes, is nonrecurring and hard to predict.

Traffic congestion reduces the utilization of the transportation infrastructure and increases travel times, air pollution, and fuel consumption levels. In the prevalent literature and practice, the terms “congestion” and “bottleneck” are often used interchangeably. However, a bottleneck is defined as a subordinate location along a highway that needs to be fixed. Unlike a systemic congestion problem, bottlenecks occur at specific locations, not pervasively along the entire corridor. Applying one or two cost-conscious corrections to inefficient subordinate locations in a facility may be all that is needed to improve a bottleneck. Recurring congestion encompasses localized congestion, or bottlenecks, and systemic congestion. A great portion of

the recurring congestion problem can be solved by bottleneck improvements that handle the additional traffic.

Freeway bottlenecks are important contributors to congestion. The congestion formed upstream of active freeway bottlenecks is a complex spatiotemporal nonlinear dynamic process. One goal of Advanced Traffic Management Systems (ATMSs) within the Intelligent Transportation System (ITS) framework is to identify bottlenecks within the transportation system [2]. Correctly identifying traffic bottlenecks is critical to understanding traffic dynamics and characterizing the spatiotemporal interactions and correlations that exist along a roadway segment. Once these spatiotemporal interactions have been characterized, transportation engineers can develop appropriate solutions to alleviate congestion and improve the performance of the freeway network. Dynamically managed lanes, speed harmonization (variable speed limits), and ramp modifications are among the cost-effective bottleneck solution approaches.

Active freeway bottlenecks are difficult to identify, but more importantly the spatiotemporal evolution of upstream congestion is extremely difficult to quantify, and it varies from one day to another. For example, the spatiotemporal location of active traffic bottlenecks may differ significantly between workdays and weekends, and bottlenecks during the morning peak are not the same as traffic bottlenecks during the evening peak. Consequently, without automatic bottleneck identification algorithms, transportation engineers might need to identify bottlenecks by driving along the freeway, which is a time-consuming approach.

The deployment of sensors has laid the foundation for researchers to develop algorithms to identify congested segments of roadway. These algorithms automatically analyze archived loop detector data and identify the time and location of potential congestion. Congestion prediction is another important research area for ITS, which would allow solutions to be developed to eliminate or alleviate congestion conditions. These automatic algorithms will become very important in the near future when connected vehicles become commercially available. Once developed, congestion avoidance and control algorithms are expected to be run by a new generation of road controllers that uses the automatic congestion identification algorithms to detect congestion regions. The controllers would then communicate with vehicles approaching congestion and provide drivers strategies to eliminate or reduce the congestion. Some of these control systems are known as variable speed limit or speed harmonization systems.

Considering the research need, this paper proposes a simple statistical approach to identifying freeway congestion. The proposed approach requires no pre-defined or fine-tuned parameters. The speed measurement at every segment is compared with a speed threshold estimated from a two-component mixture model. Based on this comparison, a binary status of the roadway segment is assigned: free-flow or congested. The corresponding confidence level (p-value) can be calculated using the component's estimated parameters. This paper uses two datasets to justify the usage of the mixture models. The first dataset, field data from the Interstate 5 corridor in the Portland, Oregon, provides the ground truth against which the performance of the proposed algorithm in the identification of congestion can be evaluated. The second dataset is collected mainly using GPS-equipped probe vehicles. It represents the average space-mean speed of a roadway segment over a 5-min interval along I-64 and I-264. The dataset consists of travel time data from the three months of the summer of 2010.

The remainder of this paper is organized as follows. First, a previously developed algorithm at VTTI called the automated statistically-principled bottleneck identification algorithm (ASBIA) is presented. Second, the proposed algorithm is introduced. Third, the field

datasets used to validate the two components' assumptions and test the algorithm are described. Fourth, the experimental results are shown and discussed. Finally, the paper conclusions are presented.

Automated Statistically-principled Bottleneck Identification Algorithm (ASBIA)

ASBIA uses the fact that points in close proximity (both temporally and spatially) to any point x provide additional information about point x [3]. The algorithm makes the following two assumptions:

1. A two-phase traffic flow theory is assumed, where traffic states are either free-flow or congested.
2. The speed data are assumed to be sampled from two component Gaussian distributions and modeled as a mixture model. The first component represents the congested regime, and all speed measurements in the congested regime are drawn from this component. The second component represents the free-flow regime, and all speed measurements within this regime are drawn from free-flow conditions.

ASBIA uses a δ1xδ2 window to select a sample from the speed matrix and identify the status of the point at the center of the window, as shown in Figure 9. The algorithm consists of the following steps:

1. Move δ1xδ2 window over the time-space domain such that the window scans all the data points in the domain.
2. If the speed measurements within the spatiotemporal window are equal, then the subject point (point 5 in Figure 9) is identified as free-flow if the speed is greater than the congestion break point, which is assumed to be the speed-at-capacity (ν_c); otherwise it is identified as congested.
3. If the speed values within the spatiotemporal window are not constant, then the t-test can be used to characterize the status of window center point (point 5 in Figure 9). The null hypothesis is $H_0: \mu_{obs} \leq \nu_c$, and the alternative hypothesis is $H_a: \mu_{obs} > \nu_c$. The center point is congested if the test fails to reject the null hypothesis; otherwise it is considered free-flow.

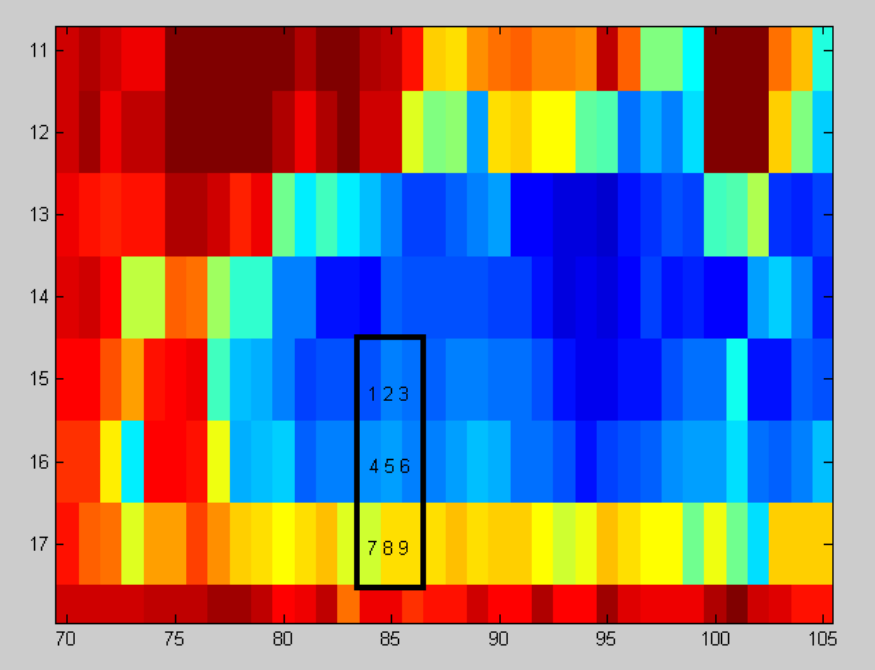


Figure 9: Illustration of the ASBIA Algorithm Where the x-axis is the Time and the y-axis is the Segment Number.

Proposed Algorithm

1. Model Deficiencies

This section summarizes the drawbacks of the ASBIA modeling framework and presents the motivation for the new proposed algorithm. ASBIA is based on the one-sample t-test. The t-test is a parametric test that uses the Student's t-distribution, which assumes the speeds in the congested and the free-flow regimes are normally distributed. An analysis of the speed histogram for both the free-flow and congested regimes reveal that neither distribution is normally distributed: both distributions are skewed and have long tails, as shown in Figure 10 This deviation from normality affects the accuracy of the ASBIA model, increasing the false positive rate.

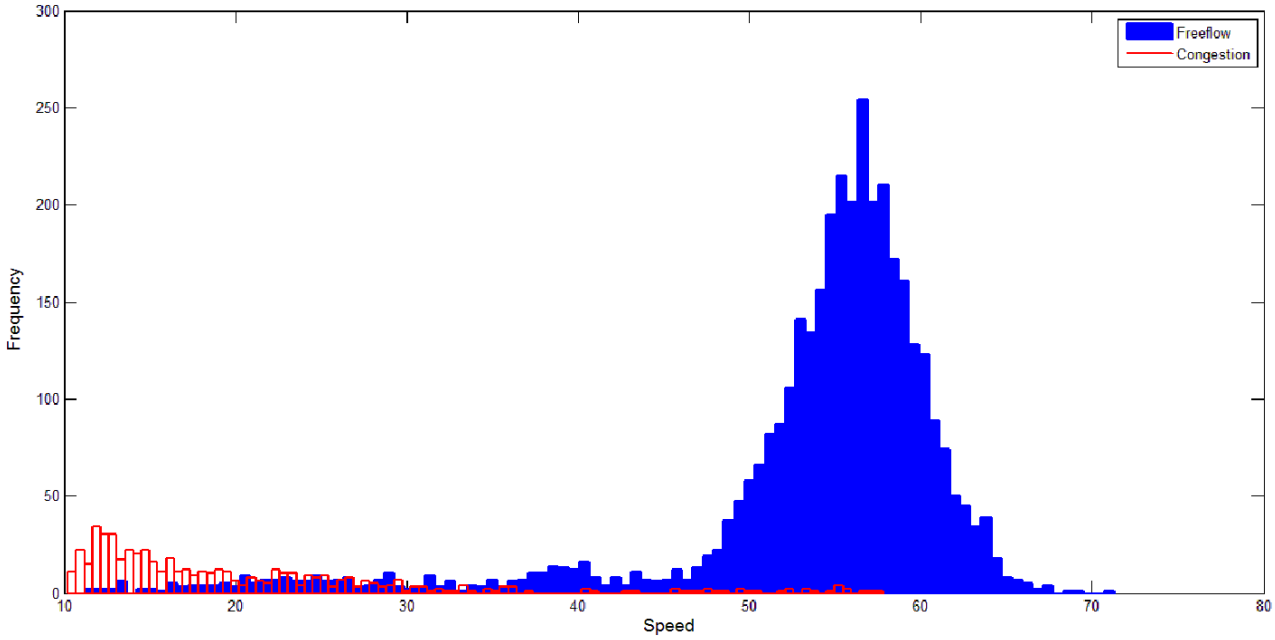


Figure 10: Speed Histogram for a Single Day (Blue Represents Free-flow Speeds, Red Represents Congested Speeds).

The ASBIA model also requires the identification of the v_c to categorize congested speeds. This parameter, which needs calibration, can have a significant impact on the speed classifications. The ASBIA model also assumes that the speed limits of contiguous segments are the same, which is not always true. ASBIA uses speed measurements at each point x(s,t) and at its neighbors as shown in Figure 11, where s is the space axis , t is the time axis with the cell colors reflecting the different speeds.

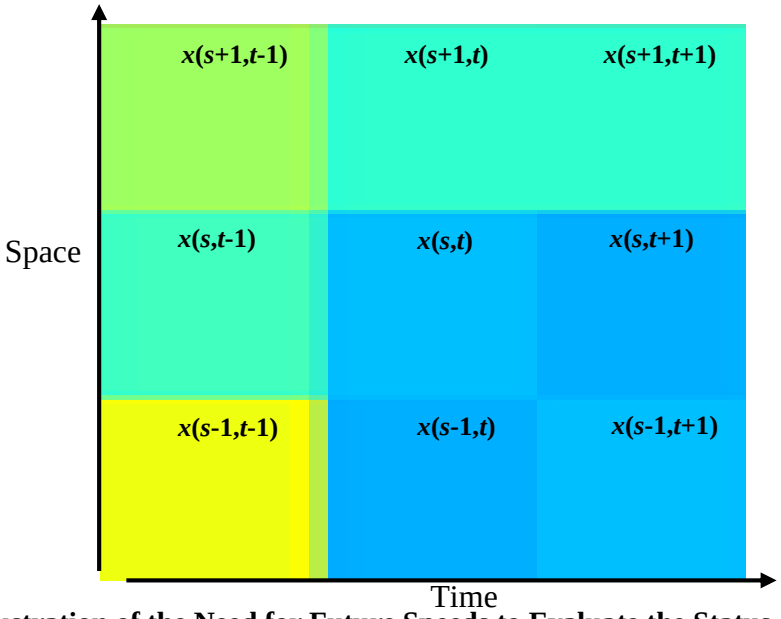


Figure 11: Illustration of the Need for Future Speeds to Evaluate the Status of x(s,t).

In real-time applications, the points $x(s+1,t+1)$, $x(s,t+1)$, and $x(s-1,t+1)$ are in the future relative to $x(s,t)$ and at time t are not known. Consequently, ASBIA is suitable for offline applications only and needs modification for use in real-time applications.

2. Background

The above limitations of the ASBIA motivated the development of an algorithm that does not require the identification of the v_c or require future speed readings. In addition, the proposed algorithm can easily deal with skewed data by choosing a skewed mixture component such as lognormal or gamma distributions. This paper evaluates three basic components: the normal, lognormal, and gamma distributions. The three distributions are used in two-component mixtures to model speed distributions. The speed dataset was fitted to three models: a normal mixture model, a lognormal mixture model, and a gamma mixture model. Both the gamma and the lognormal models are used to model skewed data, while the normal distribution is symmetric about its mean.

The two-component mixture models based on normal, gamma, and lognormal distributions are presented below. The three models use the common notation λ, which is the mixture proportion, in other words, the probability that the drawn speed is from the low-speed (congested) distribution.

Normal Mixture Model

$$f(v|\lambda,\mu_1,\mu_2,\sigma_1,\sigma_2) = \lambda\frac{1}{\sqrt{2\pi}\sigma_1}e^{\frac{(v-\mu_1)^2}{2{\sigma_1}^2}} + (1-\lambda)\frac{1}{\sqrt{2\pi}\sigma_2}e^{\frac{(v-\mu_2)^2}{2{\sigma_2}^2}} \quad (1)$$

Where (μ_1,σ_1) and (μ_2,σ_2) are the mean and standard deviation of the first and second component distribution, respectively.

Lognormal Mixture Model

$$f(v|\lambda,\mu_1,\mu_2,\sigma_1,\sigma_2) = \lambda\frac{1}{\sqrt{2\pi}\,v\sigma_1}e^{\frac{(\ln v-\mu_1)^2}{2{\sigma_1}^2}} + (1-\lambda)\frac{1}{\sqrt{2\pi}\,v\sigma_2}e^{\frac{(\ln v-\mu_2)^2}{2{\sigma_2}^2}} \quad (2)$$

Where (μ_1,σ_1) and (μ_2,σ_2) are the location and scale parameters of the first and second component distributions, respectively. The mean and the standard deviation are obtained using the location and scale parameters, respectively, as shown below in Equation (3).

$$\text{mean} = e^{\mu+2\sigma^2}$$
$$\text{SD} = \sqrt{e^{2\sigma^2}-1}e^{\mu+2\sigma^2} \quad (3)$$

Gamma Mixture Model

$$f(v|\lambda,\alpha_1,\alpha_2,\beta_1,\beta_2) = \lambda\frac{{\beta_1}^{\alpha_1}}{\Gamma(\alpha_1)}v^{\alpha_1-1}e^{-\beta_1 v} + (1-\lambda)\frac{{\beta_2}^{\alpha_2}}{\Gamma(\alpha_2)}v^{\alpha_2-1}e^{-\beta_2 v} \quad (4)$$

Where (α_1,β_1) and (α_2,β_2) are the shape and scale parameters of the first and second gamma distribution component, respectively. The mean and the standard deviation are obtained using the location and scale parameters, respectively, as shown below in Equation (5).

$$\text{mean} = \alpha/\beta \quad \text{and} \quad \text{SD} = \sqrt{\alpha}/\beta \quad (5)$$

3. Proposed Mixture Model

The speeds across the road segments have an underlying fundamental diagram trend with randomness associated with the data. The variability of speeds is substantial in congestion. Due to this random nature of speed, stochastic models are the best choice for speed modeling. Stochastic models have been proven effective in travel-time reliability modeling [4, 5]. We can model the traffic stream speed using only one standard distribution, which is appropriate when there is only one traffic state. The problem of multistate traffic can be overcome by using a mixture of distributions where each component corresponds to a specific traffic state. The proposed mixture model assumes a two-phase traffic theory: traffic is either free-flow or congested. This section describes the proposed algorithm to identify congested segments using a spatiotemporal speed matrix. The model does not require any parameter setting or parameter fine-tuning in advance, which makes it simpler and more applicable than the state-of-the-art Chen algorithm [6] or ASBIA. The proposed algorithm can be described as follows:

1. The proposed algorithm fits two component distributions to the training dataset, as demonstrated in Equation (6).
2. $f(u|\lambda, \mu_1, \mu_2, \sigma_1, \sigma_2) = \lambda\varphi(\mu_1, \sigma_1) + (1-\lambda)\varphi(\mu_2, \sigma_2),$ (6)
3. Where (μ_1, σ_1) and (μ_2, σ_2) are the location and spread of the first and second component distributions, and λ is the mixture parameter. Expectation-maximization (EM) algorithm is used to calibrate the five parameters to the data.
4. Calculate the 0.001 quantile of the free-flow-state speed distribution (the distribution with the higher estimated location parameter).
5. Use the 0.001 quantile as a threshold to classify the state of each segment along the road. All segments with speeds greater than the threshold are classified as free-flow segments, and other segments are classified as congested segments.

The output of the above algorithm is a spatiotemporal binary matrix that is of the same dimensions as the spatiotemporal speed matrix. In the binary matrix, a 1 identifies a segment as congested and a 0 represents free-flow.

One application of the proposed model is to identify bottlenecks. Bottlenecks do not exist as a single, isolated congested segment. The long queues that form behind bottlenecks create a larger set of congested, contiguous segments. To identify bottlenecks, single congested segments must first be removed from the spatiotemporal binary matrix. ASBIA can be used to smooth the proposed algorithm's binary matrix output and remove single congested segments. This case assumes the speed limit of the road segments are the same. In the standard ASBIA, the $\delta 1 x \delta 2$ window is moved over the time-space domain so that the window scans all the data points in the spatiotemporal speed matrix.

The points identified as congested in the spatiotemporal speed matrix by the mixture-model algorithm were tested using ASBIA. The tested sample included the tested point and the points surrounding it within the $\delta 1 x \delta 2$ window. If the average speed was statistically smaller than the congestion break point, which is assumed to be the mean of the free-flow speed distribution, then the tested point was identified as congested. Otherwise it was identified as free-flow. The t-test was used to characterize the status of point of interest. For the t-test, the null hypothesis was H_0: $\mu_{obs} \leq \mu_2$ and the alternative hypothesis was H_a: $\mu_{obs} > \mu_2$, where μ_2 is the mean of the second component (the free-flow component). If the test failed to reject the null hypothesis, the point was considered congested; otherwise it was considered free-flow.

Datasets

1. Portland

The Portland dataset consisted of 24 days' worth of high-quality data: midweek, non-holiday days between February and December 2008. The data were collected from archived data from the northbound Interstate 5 (I-5) corridor in the Portland, Oregon, metropolitan region. This road segment is 22 mi (35 km) in length. Along the segment there were 22 detectors, two of which were ignored because of their poor data quality. Each day included readings from 20 detectors at the lowest available resolution of 20 s, between 5:00 a.m. and 10:00 p.m.

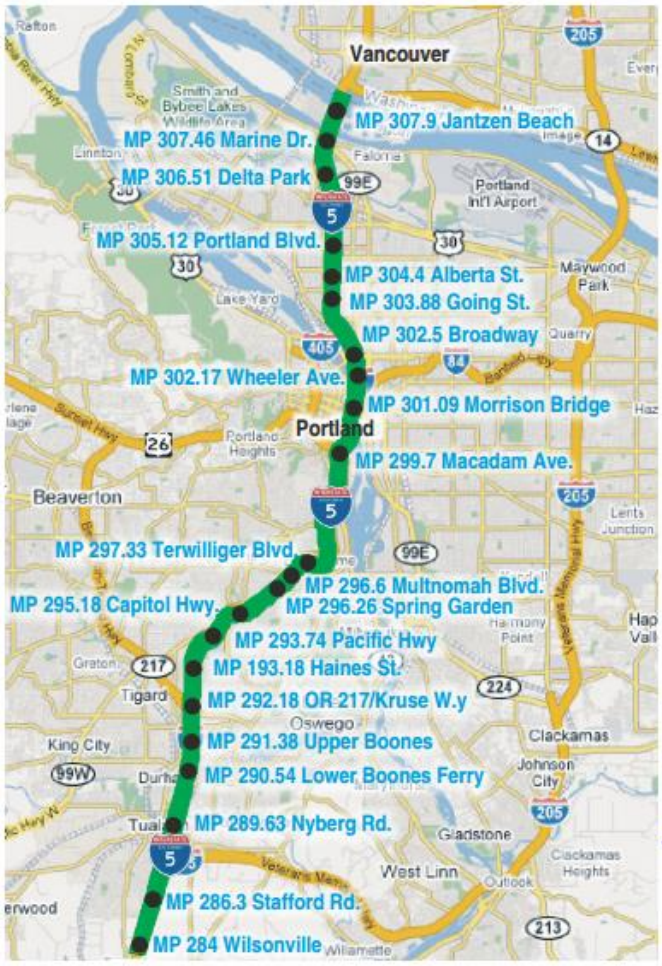


Figure 12: Northbound Interstate 5 Corridor in the Portland Region where the Black Dots Indicate the Location of Sensors.

2. INRIX

The INRIX traffic data are collected mainly using GPS-equipped probe vehicles. The collected probe data are supplemented with traditional road sensor data, as well as mobile devices and other sources [7]. As a result, the traffic data represent the average space-mean speed of a roadway segment over a 5-min interval. The 2010 INRIX data along I-64 and I-264 were used to construct the travel database. A 37-mile freeway stretch includes most of the congested areas heading toward Virginia Beach from Richmond. The selected stretch of the freeway goes from Newport News to Virginia Beach along I-64 and I-264 and includes 58 sections, as shown in

Figure 13. The average section length is 0.65 mi, and the longest section is 3.7 mi and is located at the Hampton Roads Bridge-Tunnel (HRBT). The speed matrix is a 58 by 288 matrix, where 58 is the number of road sections and 288 is the number of time intervals per day. The dataset consists of travel data from the three months of summer 2010.

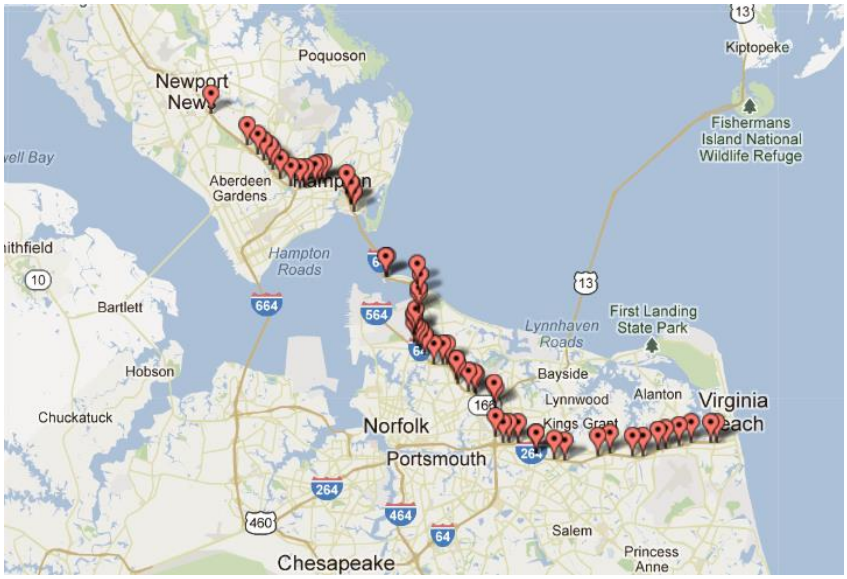


Figure 13: Selected I-66 and I-264 Freeway Stretch.

Experimental Work

1. Justification of the Two-Component Mixture Model

This experiment set used the two datasets to justify the two-component mixture model. The EM algorithm was used to fit each dataset to the mixture model, and the bimodal probability distributions were drawn over the histogram of each dataset, as shown in Figure 14. It is obvious that the histograms for both datasets are bimodal, which means that the assumption of the two-component model is realistic. The figure also shows that the fitted models approximate the true dataset distributions well, but it is not visually obvious which one is the best. To find the best model, the log-likelihood for each model was calculated, and the best model has the maximum log-likelihood. The calculated log-likelihoods for each model are shown in Table 1.

Table 1: The Log-Likelihood for Each Model

	Log-likelihood	
	Portland dataset	INRIX dataset
Normal	-111119.1	-2002257
Lognormal	-105443.6	-2111074
Gamma	-107076.0	-2067934

The table shows that the lognormal has the highest log-likelihood in the Portland dataset and the normal has the highest log-likelihood in the INRIX dataset. In general, the lognormal distribution is preferred to the normal because of its capability to model skewed data. The lognormal distribution is also better than the gamma distribution because the estimation of gamma parameters takes more time than for the lognormal distribution.

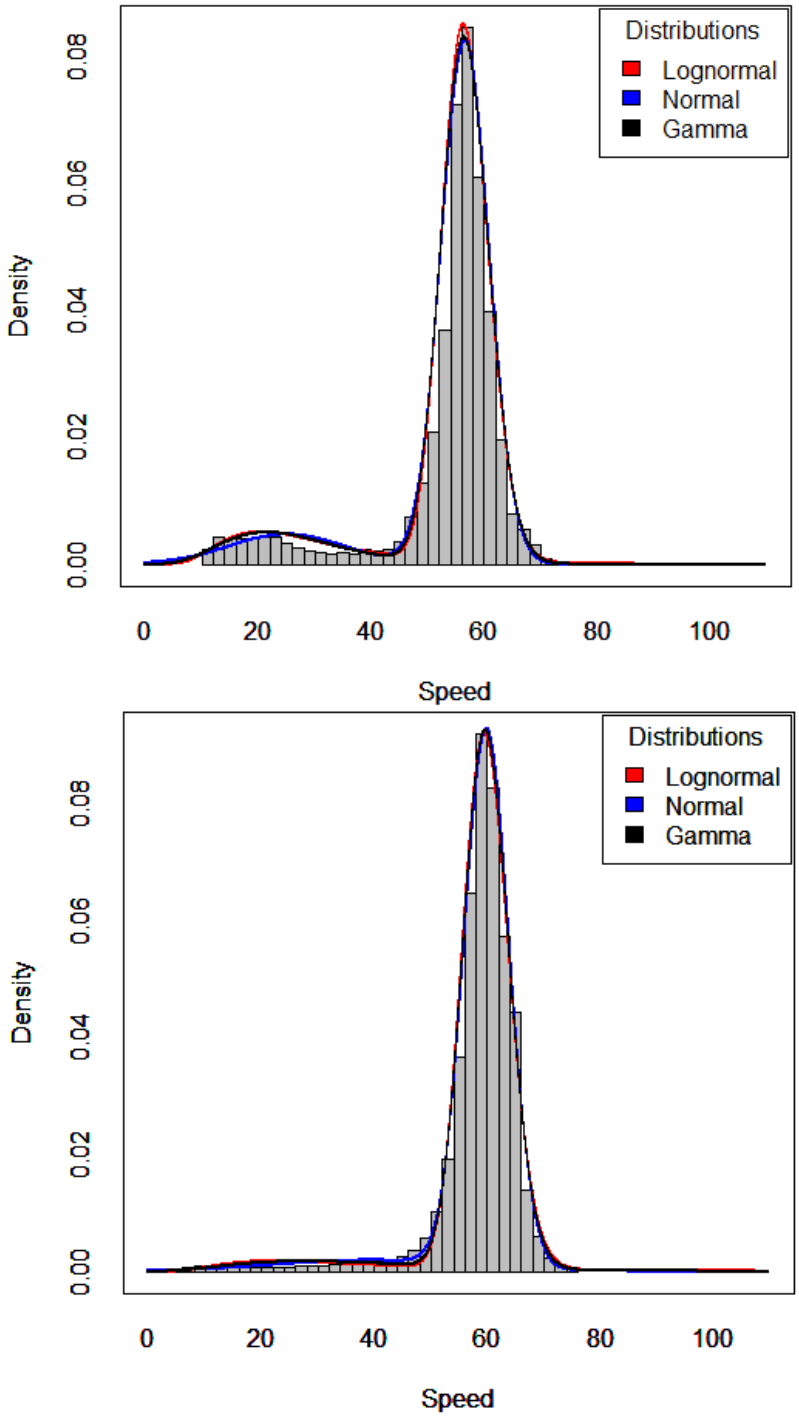


Figure 14: The Histogram of the Dataset and the Probability Density Function of the Three Models, Where the Top Figure is for The First Dataset and the Bottom Figure is for the Second Dataset.

2. Evaluating the Performance of the Proposed Algorithm

In this set of experiments, the performance of the algorithm was tested using the Portland dataset because it contains the ground truth, which was defined using a manual procedure. During this manual procedure, the activation and deactivation times of each candidate bottleneck were carefully diagnosed and verified using oblique curves of cumulative vehicle arrival versus time

and cumulative occupancy (or speed) versus time, constructed from data measured at neighboring freeway loop detectors [8-10]. The ground truth dataset for each day is a binary matrix, where 1 represents a congestion segment and 0 represents a free-flow segment. Though the available ground truth is not accurate because it depends on the estimated free-flow speed, it was used because it is the only available reference.

The true positive rate (TPR) and false positive rate (FPR) are computed using Equations (7) and (8).

$$\text{TPR} = \frac{\text{True detected positive}}{\text{Actual positive}} \tag{7}$$

$$\text{FPR} = \frac{\text{False detected positive}}{\text{Actual negative}} \tag{8}$$

Where positive in this context is a congested segment and negative is a free-flow state segment. The TPR for the proposed algorithm is calculated as the average of the TPRs of both sets of trials, using the leave-one-out (LOO) cross-validation method [11]. In the LOO approach, the two-component mixture is built using 23 days only. The remaining day is used as the unseen test day, and the TPR is calculated for that day. The entire process is repeated for each day, which is used once as a test day and compared to the average TPR calculated across all 23 remaining days. This procedure was repeated to calculate the FPR for the proposed algorithm. The proposed algorithm achieves an average TPR of 0.9097 and an average FPR of 0.0420. Table 2 shows the TPR and FPR for each day. Some days have high TPR and low FPR, but other days are not as good. The spatiotemporal matrix visually compared to both the ground truth and the algorithm's output, as shown in Figure 15. We found the ground truth of days having low TPR is not accurate. The regions indicated in Figure 15 show segments that the ground truth did not classify as congested but the proposed algorithm did. The promising performance of the algorithm is noticeable when visually comparing the spatiotemporal matrix and the algorithm's output.

Table 2: TPR and FPR for Each Day in Dataset #1

	TPR	FPR		TPR	FPR
Day1	0.9465	0.0936	Day13	0.7625	0.0236
Day2	**0.9805**	**0.0339**	Day14	0.8247	0.0141
Day3	0.8908	0.0493	Day15	**0.9716**	**0.0293**
Day4	0.9320	0.0397	Day16	0.7389	0.0241
Day5	**0.9268**	**0.0324**	Day17	1.0000	0.0640
Day6	**0.9752**	**0.0306**	Day18	0.9803	0.0667
Day7	0.9146	0.0325	Day19	0.8733	0.0243
Day8	0.9884	0.0557	Day20	**0.9147**	**0.0360**
Day9	0.9710	0.0664	Day21	0.9787	0.0668
Day10	0.8212	0.0345	Day22	0.8925	0.0453
Day11	0.8919	0.0290	Day23	0.9171	0.0533
Day12	0.8548	0.0504	Day24	0.8860	0.0133

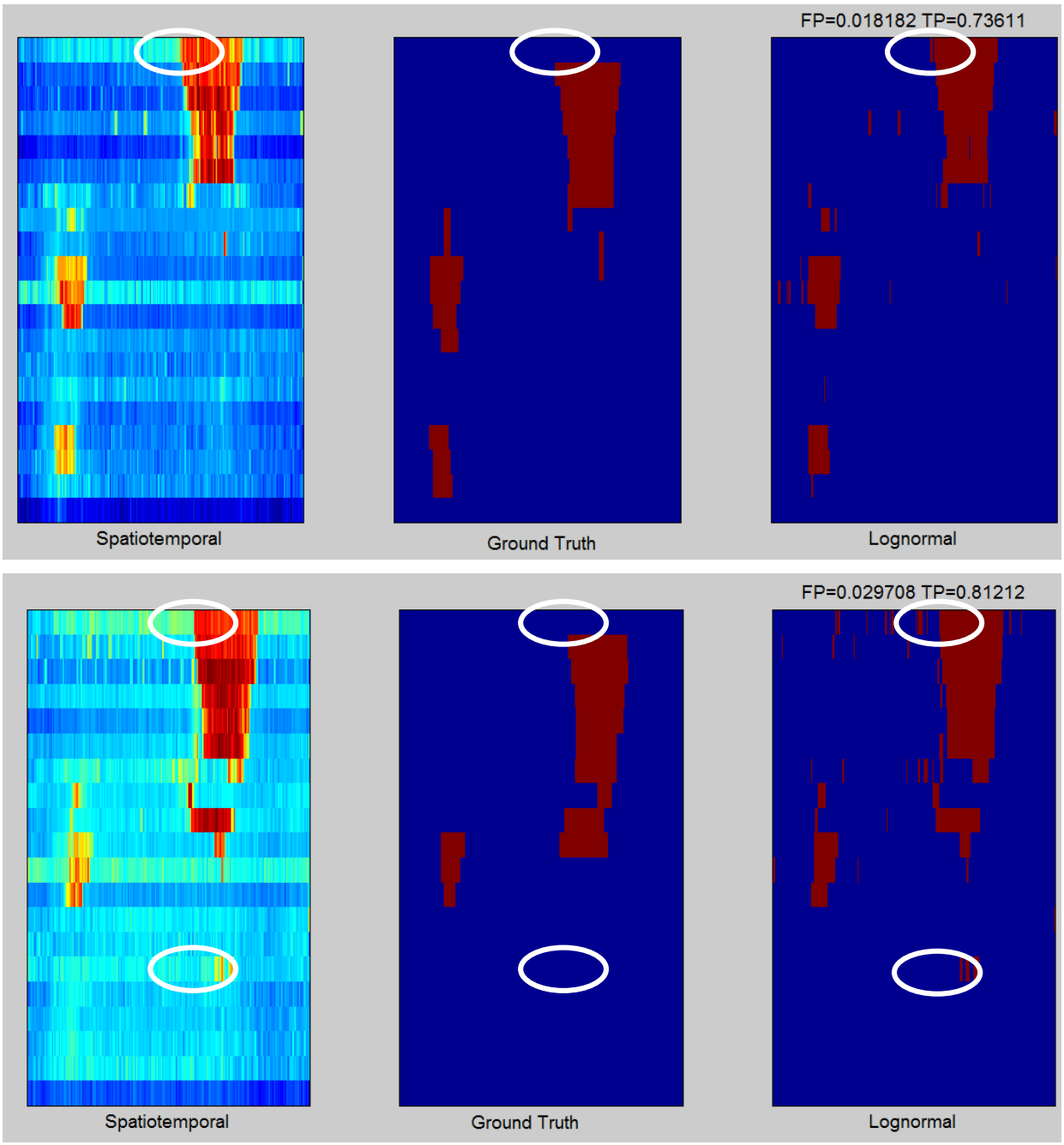


Figure 15: Comparison Between Spatiotemporal Matrix and Both the Ground Truth and the Algorithm's Output for 2 Days With Low TPR Where x-axis is the Time and y-axis is Spatial.

The next experiment set tested the smoothing using the ASBIA to remove the identified isolated congested segments. In this version of ASBIA, three different windows were used. The first window is square window of size 2x2. The other two windows are rectangular, one along the temporal dimension and the other along the spatial dimension. The used window is designed such that at time t_o only the neighbor segments are used and that before time t_o no future (unknown) speed measurement is needed. In the case of the 2x2 window, the neighbors were used at time t_o and at $t_o - 1$. The rectangular window along the temporal dimension is of size

3x1, and the rectangular window along the spatial dimension is of size 1x3. Figure 16 shows the output of the proposed algorithm and the smoothed versions of this output. The figure marks the removal of most of the identified single, congested segments. The smoothed binary matrix was visually inspected using the three different windows. In terms of TPR and FPR, the best smoothed result is obtained when using the 2x2 window. The smoothed version of the binary matrix is appropriate for identifying bottlenecks or lowering the FPR. Table 3 shows the TPR and FPR of the 24 days for the different windows.

Table 3: TPR and FPR for Each Day in Dataset #1 Using the Different Windows

	Lognormal		Lognormal+2x2 window		Lognormal+1x3 window		Lognormal+3x1 window	
	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
Day1	0.9465	0.0936	0.9410	0.0658	0.8801	0.0377	0.9354	0.0821
Day2	0.9805	0.0339	0.9416	0.0125	0.7908	0.0054	0.9489	0.0260
Day3	0.8908	0.0493	0.8649	0.0237	0.6264	0.0053	0.8678	0.0360
Day4	0.9320	0.0397	0.8435	0.0157	0.6327	0.0061	0.8639	0.0299
Day5	0.9268	0.0324	0.8862	0.0118	0.6829	0.0018	0.8780	0.0234
Day6	0.9752	0.0306	0.9471	0.0155	0.8264	0.0069	0.9537	0.0237
Day7	0.9146	0.0325	0.8537	0.0163	0.6366	0.0062	0.8780	0.0225
Day8	0.9884	0.0557	0.9567	0.0383	0.8253	0.0154	0.9634	0.0429
Day9	0.9710	0.0664	0.9382	0.0508	0.8205	0.0302	0.9614	0.0519
Day10	0.8212	0.0345	0.7697	0.0233	0.5303	0.0172	0.7909	0.0260
Day11	0.8919	0.0290	0.8459	0.0153	0.6541	0.0011	0.8649	0.0223
Day12	0.8548	0.0504	0.8150	0.0392	0.6183	0.0250	0.8220	0.0422
Day13	0.7625	0.0236	0.6544	0.0073	0.4987	0.0032	0.7282	0.0134
Day14	0.8247	0.0141	0.6959	0.0044	0.3351	0.0008	0.7629	0.0072
Day15	0.9716	0.0293	0.9113	0.0165	0.6738	0.0092	0.9291	0.0196
Day16	0.7389	0.0241	0.7167	0.0158	0.5056	0.0136	0.7139	0.0168
Day17	1.0000	0.0640	1.0000	0.0478	0.7212	0.0282	0.9939	0.0506
Day18	0.9803	0.0667	0.9721	0.0455	0.8177	0.0309	0.9589	0.0484
Day19	0.8733	0.0243	0.8388	0.0131	0.6065	0.0064	0.8445	0.0154
Day20	0.9147	0.0360	0.8942	0.0213	0.6587	0.0137	0.8840	0.0250
Day21	0.9787	0.0668	0.9547	0.0513	0.7040	0.0279	0.9520	0.0548
Day22	0.8925	0.0453	0.8832	0.0327	0.7383	0.0185	0.8832	0.0345
Day23	0.9171	0.0533	0.8756	0.0382	0.7047	0.0197	0.8731	0.0425
Day24	0.8860	0.0133	0.8538	0.0048	0.6404	0.0027	0.8509	0.0064

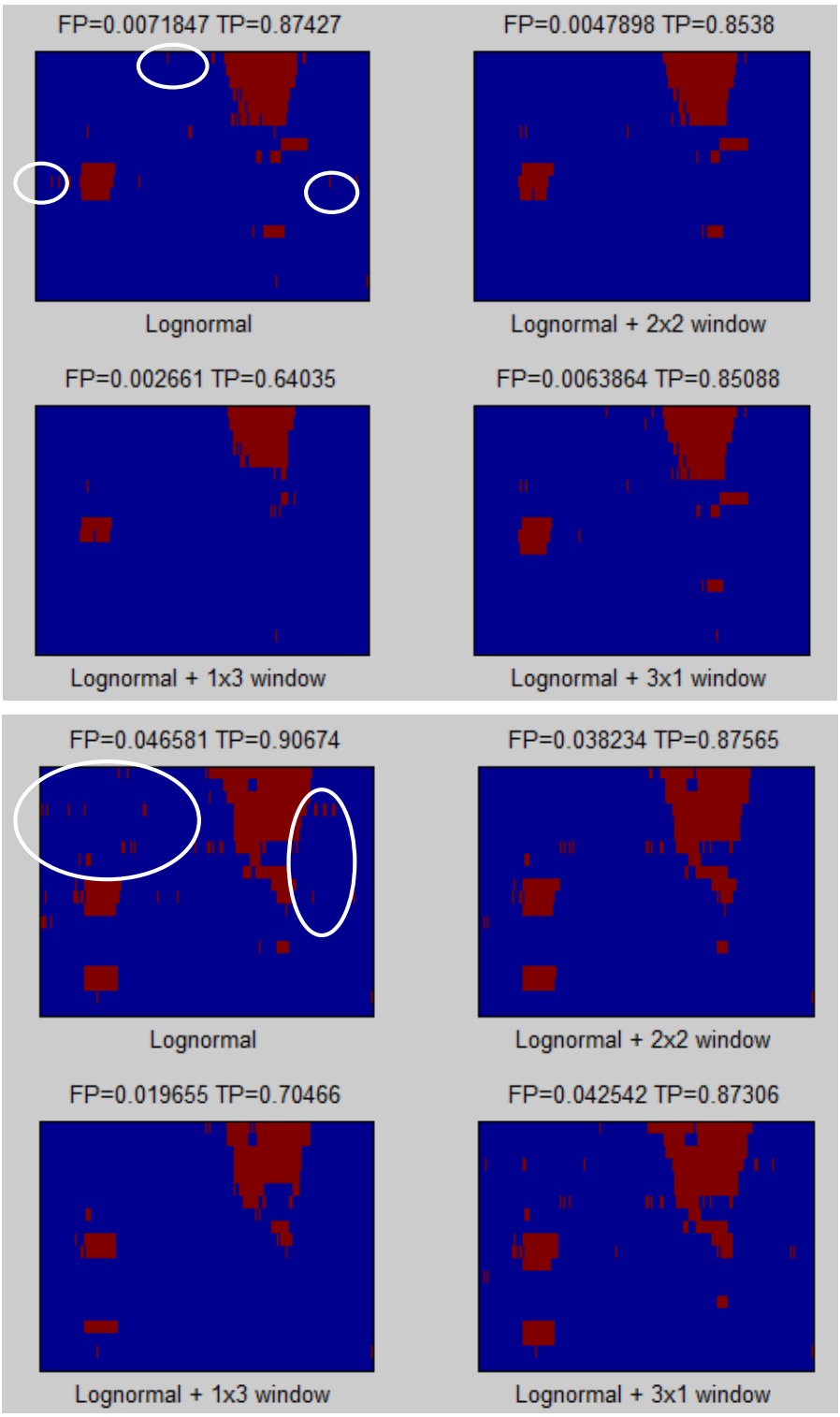


Figure 16: Comparison Between the Proposed Algorithm's Output for 2 Days Before and After Smoothing Using Different Windows.

Conclusions

This paper addresses the traffic congestion identification problem by proposing a new algorithm based on a two-component mixture model. Two datasets were used to justify the two-component assumption and to compare between the different models using normal, lognormal, and gamma distributions. The experimental results show that the lognormal mixture model is the best model if the data is skewed. The proposed algorithm overcomes the model deficiencies of ASBIA. The proposed algorithm overcomes the normality problem by using the lognormal distribution to model skewed data. It is suitable for online (real-time) application because it does not require knowledge of future speed readings. In some applications, transportation engineers are interested only in contiguous, congested segments, not single congested segments. To remove the unwanted, individual congested segments, the output of the proposed algorithm was smoothed using a modified version of ASBIA that uses different window shapes and sizes. The smoothed version of the output is suitable for bottleneck identification and has less FPR and slightly less TPR.

Because the roadway surface condition changes with weather conditions, the distributions of speed under both regimes are expected to change as well. Future work includes considering a three-component mixture model, integrating the weather conditions into the traffic congestion identification algorithm and studying its effect on the identification process.

Acknowledgements

This effort was funded by the Mid-Atlantic University Transportation Center (MAUTC) and the Virginia Department of Transportation (VDOT). The authors also thank Dr. Robert Bertini for providing us the Portland data that was used in the analysis.

References

[1] (March 2011). *An Agency Guide on How to Establish Localized Congestion Mitigation Programs* Available: http://ops.fhwa.dot.gov/publications/fhwahop11009/fhwahop11009.pdf

[2] S. EZELL. (JANUARY 9, 2010). *Explaining International IT Application Leadership: Intelligent Transportation Systems*. Available: http://www.itif.org/publications/explaining-international-it-application-leadership-intelligent-transportation-systems

[3] M. Elhenawy, H. A. Rakha, and H. Chen, "An Automated Statistically-Principled Bottleneck Identification Algorithm (ASBIA)," presented at the 16th International IEEE Annual Conference on Intelligent Transportation Systems, The Hague, The Netherlands, 2013.

[4] F. Guo, H. Rakha, and S. Park, "Multistate Model for Travel Time Reliability," *Transportation Research Record: Journal of the Transportation Research Board* 2010.

[5] F. Guo, Q. Li, and H. Rakha, "Multistate Travel Time Reliability Models with Skewed Component Distributions," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2315, pp. 47-53, 12/01/ 2012.

[6] C. Chen, A. Skabardonis, and P. Varaiya, "Systematic identification of freeway bottlenecks," *Freeway Operations and Traffic Signal Systems 2004*, pp. 46-52, 2004.

[7] INRIX. (2012). *Traffic Information*. Available: http://www.inrix.com/trafficinformation.asp

[8] R. L. Bertini and A. M. Myton, "Use of performance measurement system data to diagnose freeway bottleneck locations empirically in Orange County, California," *Freeway Operations, High-Occupancy Vehicle Systems, Traffic Signal Systems, and Regional Transportation Systems Management 2005*, pp. 48-57, 2005.

[9] Cassidy, M. J., and R. L. Bertini., "Observations at a Freeway Bottleneck," presented at the 14th International Symposium on Transportation and Traffic Theory, Jerusalem, Israel, 1999.

[10] Cassidy, M. J., and J. R. Windover, "Methodology for Assessing Dynamics of Freeway Traffic Flow," *Transportation Research Record,* vol. 1484, pp. 73–79, 1995.

[11] T. Evgeniou, M. Pontil, and A. Elisseeff, "Leave One Out Error, Stability, and Generalization of Voting Combinations of Classifiers," *Machine Learning*, vol. 55, pp. 71-97, 2004/04/01 2004.

Chapter 4: Traffic Congestion Identification Using Mixture of Linear Regression under Different Weather and Visibility Conditions

This chapter is based on
Mohammed Elhenawy, Hao Chen, and Hesham Rakha, "Traffic Congestion Identification Using Mixture of Linear Regression under Different Weather and Visibility Conditions," presented at the Transportation Research Board 94th Annual Meeting, Washington DC, United States, 2015.

Abstract

Automatic identification of traffic congestion is an important task for transportation planning and traffic operations. Transportation engineers use congestion identification as a preprocessing step for the downstream analysis to identify and rank traffic bottlenecks. Real time automatic congestion identification is one of the important routines of intelligent transportation systems (ITS). In the near future conveying congestion information to connected vehicles can help them making better route choice and avoid congested road segments. Previous efforts usually use traffic state measurements (speed, flow, occupancy) to develop congestion identification algorithms. However, the impacts of weather conditions to identify congestion have not been investigated in the existing studies. In this paper the impacts of weather conditions and visibility levels on the congestion identification algorithm are investigated by modeling the speed distribution using a mixture of linear regression. In this approach, the speed of each traffic regime is modeled using a normal distribution and its mean is defined using a linear regression model, which is a function of different weather conditions and visibility levels. Therefore, the mode parameters at any weather condition and visibility level can be calculated by fitting the speed model to the training dataset. Then using the calculated model parameters we can calculate the speed cut off between congestion and free-flow which minimize either the Bayesian classification error or the false positive (congestion) rate. A freeway stretch along I-66 to connect I-81 and Washington D.C. is selected as the study site. The proposed algorithm is evaluated by three years of traffic data and the corresponding weather information. The test results demonstrate the proposed method produces promising congestion identification output by considering weather condition and visibility level.

Introduction

Traffic congestion has become one of the modern life problems in many metropolitan areas. This growing problem has environmental effects. During congestion time cars cannot run efficiently so air pollution, carbon dioxide (CO2) emissions and fuel use increases. In 2007, Americans lost $87.2 billion in wasted fuel and lost productivity. This waste reached $115 billion in 2009 [1]. Congestion increases travel time, for example back in 1993 driving under congested condition causes a delay of about six-tenths of a minute per kilometer of travel on expressways and 1.2 minutes delay per kilometer of travel in arterials[2]. The congestion problem becomes worse as reported by Texas Transportation Institute where the number of Americans' wasted hours in traffic congestion becomes fivefold between 1982 and 2005. Moreover, congestion has its economic effect where studies shows that congestion slow metropolitan growth, inhibits agglomeration economies, and shape economic geographies [3]. Traffic Congestion could result by obstruction, or lack of road capacity which is a kind of inefficient use of the roads. This

problem can be relaxed by increasing the road-building budgets to build more infrastructures. But adding more road capacity is costly and budget is limited, and the construction itself takes long time. With the continuous increase in traffic volumes, managing traffic, particularly at times of peak demand, is a good and inexpensive solution to congestion. Advanced traffic management systems (ATMS) use various applications of intelligent transportation systems (ITS) to manage traffic and reduce congestion problems. Recently, the advancement in communication and computers greatly improve ITS and make it more capable of identifying and reducing congestion. ITS is an effective solution to traffic problem where it improves the dynamic capacity of the road system without building extra expensive infrastructure [4]. Accurate and real-time traffic information is the foundation of ITS, which uses traffic measurement data from various traffic sensing techniques for freeway traffic surveillance and control [5, 6].

Congestion usually starts from a road bottleneck, and then spills over the neighbor road segments. It takes time until this congestion to be disappeared. Depending on the frequency of congestion occurrence, traffic congestion can be divided into two categories [7]. The first is recurrent traffic congestion, and the second is accidental traffic congestion. Recurrent traffic congestion, which usually results from exceeding the road capacity, is easier to identify and predict. The accidental traffic congestion usually results from traffic incident. Traffic congestion is different at different location, time periods, and different weather condition.

During the last few years, many automatic congestion identification algorithms are proposed. An Automated Statistically-principled Bottleneck Identification Algorithm (ASBIA) uses speed measurements over short temporal and spatial intervals and segments, respectively to identify the status of a segment using t-test [8]. The outputs of the algorithm are the status of the roadway segment (free-flow or congested) and the confidence level of the test (p-value). Another algorithm uses vehicle trajectories in intelligent vehicle infrastructure co-operation system (IVICS) [4]. Then the spatial–temporal trajectories are considered as an image to extract the propagation speed of congestion wave and construct congestion template. Finally correlation is evaluated between the template and the spatial–temporal velocity image to identify the congestion. Parallel SVM is used in [9] to identify traffic congestion. The authors propose Parallel SVM instead of SVM because the training computation cost of SVM is expensive and congestion identification is a real-time task.

Floating car data is used in [10] to find meaningful congestion patterns. The analysis of the floating car data is done using a method based on data cube and spatial-temporal related relationship of slow-speed road segment to identify the traffic congestion. The research team at the center for sustainable mobility (CSM) at the Virginia Tech Transportation institute (VTTI) developed an algorithm to identify congested segments using a spatiotemporal speed matrix [11]. The proposed algorithm fits two lognormal (or normal) distributions to the training dataset.

To the best of our knowledge no research addresses the impacts of both visibility and weather conditions on congestion identification. In this paper the impacts of weather conditions and visibility levels on the congestion identification algorithm are investigated by modeling the speed distribution as mixture of two normal components whose means consist of the linear function of weather condition and visibility level. So that based on these factors the two normal components may get close or apart and the cut-off speed is changed. The proposed algorithm is tested using three years of historical data of the I-66 eastbound, and produces promising and reasonable results where, for example, the cut-off speed increases as the visibility level increases.

The remainder of this paper is organized as follows. First, a brief background of the method used in this work is given. After that, the proposed algorithm is introduced. The dataset

used in the case study is described. Subsequently the result of the experimental work is explained and an illustrative example is given to show how to implement the proposed model. Finally, conclusions and recommendations for future work are presented.

Methodology

In this section a brief introduction to the modeling techniques used in this paper are presented to familiarize readers with these emerging techniques. The strengths of each modeling technique are presented given that the models will be compared later on the same dataset. The modeling techniques represent a variety of machine learning techniques. They range from very simple algorithms to complicated and high computational demand algorithms. The used algorithms demonstrate the wide variety of algorithms that can be used by transportation practitioners.

1. Mixture of Linear Regressions[12, 13]

Finite mixture models are powerful tools in modeling a wide variety of random phenomena. It is used to model random phenomena in many fields such as agriculture, biology, economics, medicine and genetics. A mixture of linear regressions is one of the mixture families that is studied carefully and can be used to model the traffic speed for different regimes of traffic conditions. The mixture of linear regression can be written as

$$p(y|X) = \sum_{j=1}^{m} \frac{\lambda_j}{\sigma_j\sqrt{2\pi}} e^{-\frac{\left(y-x^T\beta_j\right)^2}{2\sigma_j^2}} \quad (1)$$

or as

$$y_i = \begin{cases} x_i^T\beta_1 + \epsilon_{i1} & \text{with probability } \lambda_1 \\ x_i^T\beta_2 + \epsilon_{i2} & \text{with probability } \lambda_2 \\ \vdots & \\ x_i^T\beta_m + \epsilon_{im} & \text{with probability } 1 - \sum_{q=1}^{m-1}\lambda_q \end{cases} \quad (2)$$

where y_i is the response corresponding to a vector p of predictors x_i^T ,β_j is the vector of regression coefficients for the j^{th} component, λ_j is the mixing probability of the j^{th} component and ϵ_{ij} are normal random errors. The model parameters $\psi = \{\beta_1, \beta_2, \ldots, \beta_m, \sigma_1^2, \sigma_2^2, \ldots, \sigma_m^2, \lambda_1, \lambda_2, \ldots, \lambda_m\}$ can be estimated by maximizing the log-likelihood of Equation (1) given a set of response predictors pairs $(y_1, x_1), (y_2, x_2), \ldots, (y_n, x_n)$ using Expectation-Maximization algorithm (EM).

EM algorithm

The EM algorithm iteratively finds the maximum likelihood estimates by alternating the E-step and M-step. Let $\psi^{(k)}$ the parameter estimates after the k^{th} iteration. On the E-step, the posterior probability of the i^{th} observation comes from component j is computed as shown in Equation (3).

$$w_{ij}^{(k+1)} = \frac{\lambda_j^{(k)}\phi_j\left(y_i|x_i, \psi^{(k)}\right)}{\sum_{j=1}^{m}\lambda_j^{(k)}\phi_j\left(y_i|x_i, \psi^{(k)}\right)} \quad (3)$$

where $\phi_j\left(y_i|x_i, \psi^{(k)}\right)$ is the probability density function of the j^{th} component.

On the M-step the new parameter estimates $\psi^{(k+1)}$ that maximizes the log-likelihood function in Equation (1) is calculated, as shown in Equations (4-6)

$$\lambda_j^{(k+1)} = \frac{\sum_{i=1}^{n} w_{ij}^{(k+1)}}{n} \quad (4)$$

$$\hat{\beta}_j^{(k+1)} = (X^T W_j X)^{-1} X^T W_j Y \quad (5)$$

where X is $nx(p+1)$ predictor matrix, Y is the corresponding nx1 response vector, and W is nxn diagonal martrix which has $w_{ij}^{(k+1)}$ along its diagonal.

$$\hat{\sigma}_j^{2(k+1)} = \frac{\sum_{i=1}^{n} w_{ij}^{(k+1)} (y_i - x_i^T \hat{\beta}_j^{(k+1)})^2}{\sum_{i=1}^{n} w_{ij}^{(k+1)}} \quad (6)$$

The E-step and M-step are alternated repeatedly until the change in the incomplete log-likelihood is arbitrarily small, as shown in Equation (7).

$$\left| \prod_{i=1}^{n} \sum_{j=1}^{m} \lambda_j^{(k+1)} \phi_j(y_i | x_i, \psi^{(k+1)}) - \prod_{i=1}^{n} \sum_{j=1}^{m} \lambda_j^{(k)} \phi_j(y_i | x_i, \psi^{(k)}) \right| < \xi \quad (7)$$

where ξ is a small number.

2. Bayesian Approach to Identify the Threshold

In the above algorithm, the threshold to identify congestion by speed data is based only on the free-flow distribution, and the threshold is chosen to guarantee the false positive rate is around 0.001 for the 0.001 quantile. Another approach to calibrate the threshold after fitting the speed data to the mixture model is to choose the threshold that minimizes the misclassification error based on Bayes rule. Given the probability distribution for congestion and free-flow speeds, Bayes rule is used for a given

$$p(\text{congestion}|\text{speed reading}) = p(\text{speed reading}|\text{congestion}) p(\text{congestion}) \quad (8)$$

$$p(\text{freeflow}|\text{speed reading}) = p(\text{speed reading}|\text{freeflow}) p(\text{freeflow}) \quad (9)$$

The optimum threshold is the value of the speed at which p(congestion|speed reading) = p(free-flow|speed reading). Assume this threshold can be calculated, any speed above it is classified as free-flow and any speed below it is identified as congested. The threshold can be derived for normal distributions. Assume that the $p(\text{congestion}) = 1 - \lambda$ and $p(\text{freeflow}) = \lambda$ then the y_{th} is shown in equation (10) can be calculated as follow

$$y_{th} = \min \left\{ \frac{-\mu_1 \sigma_2^2 + \mu_2 \sigma_1^2 \mp \sqrt{(\mu_1 \sigma_2^2 - \mu_2 \sigma_1^2)^2 - (\sigma_1^2 - \sigma_2^2)\left(2\sigma_2^2 \sigma_1^2 \ln\left(\frac{\lambda \ \sigma_2}{(1-\lambda)\sigma_1}\right) - \mu_1^2 \sigma_2^2 + \mu_2^2 \sigma_1^2\right)}}{(\sigma_1^2 - \sigma_2^2)} \right\} \quad (10)$$

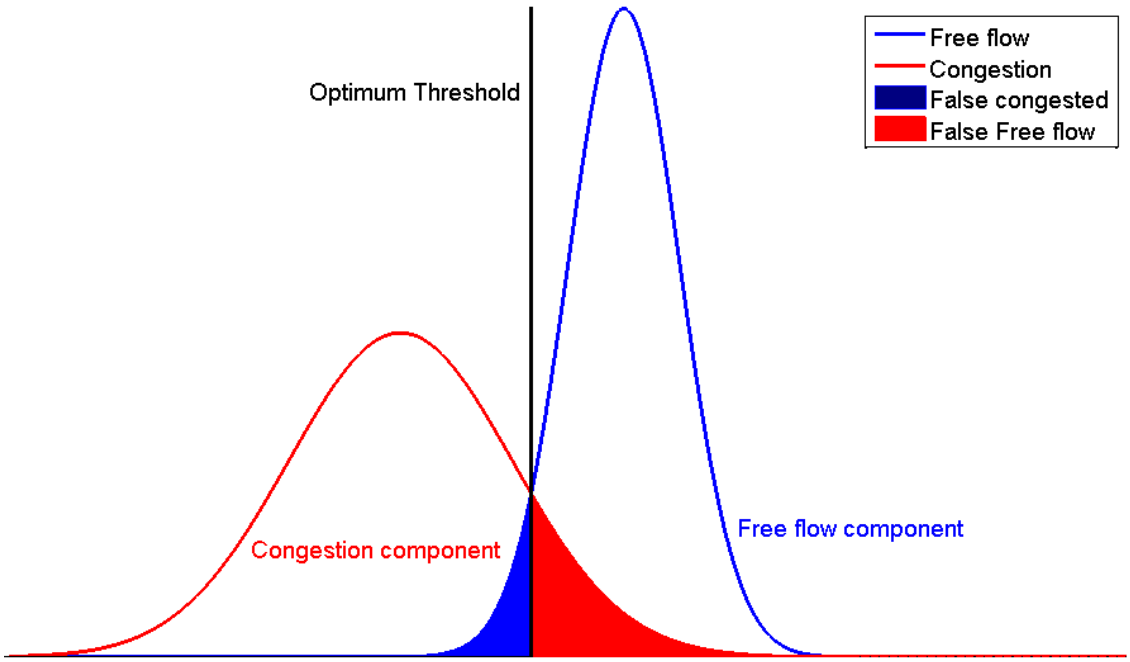


Figure 17: Illustration of the Bayesian Threshold.

Proposed Congestion Identification Algorithm

The speeds across the road segments have an underlying fundamental diagram trend with randomness associated with the data. The variability of speeds is substantial in congestion. Due to this random nature of speed, stochastic models are the best choice for speed modeling. Stochastic models have been proven to be really good tools in travel time reliability modeling [14, 15]. The traffic stream speed can be modeled by using only one standard distribution, which is good if only one traffic state is provided. An alternative to overcome the problem of multistate traffic is to use a mixture of distributions where each component corresponds to a specific traffic state. In this study, a two-phase traffic theory is assumed where the traffic is either free-flow or congested. The research team at the center for sustainable and mobility (CSM) at the Virginia Tech Transportation institute (VTTI) developed a simple algorithm to identify congested segments using a spatiotemporal speed matrix [11]. The model does not required any parameter setting in advance which makes it simpler and more applied than the state-of-the-art Chen algorithm [16]. The proposed algorithm fits two lognormal (or normal) distributions to the training dataset, as demonstrated in Equation (11).

$$f(y|\lambda,\mu_1,\mu_2,\sigma_1,\sigma_2) = \lambda\frac{1}{\sqrt{2\pi}\sigma_1}e^{\frac{(y-\mu_1)^2}{2\sigma_1^2}} + (1-\lambda)\frac{1}{\sqrt{2\pi}\sigma_2}e^{\frac{(y-\mu_2)^2}{2\sigma_2^2}} \quad (11)$$

Here (μ_1,σ_1) and (μ_2,σ_2) are the mean and standard deviation of the first and second component distributions and λ is the mixture parameter.

A Maximum Likelihood algorithm is used to calibrate the five parameters to the training dataset. The next step after fitting the model is calculating the 0.001 quintile of the free-flow state speed distribution. Then we use the 0.001 quintile as a threshold to classify the state of each segment along the road. All segments with speeds greater than the threshold are classified as free-flow segments and other segments are classified as congested segments. The above algorithm does not take in consideration the weather which would change road surface condition

and hence the means of the components. Moreover it did make use of the visibility level which is very important factor.

A straight forward solution to integrate the weather conditions and visibility levels is grouping data into bins based on weather condition and visibility level. Each bin will have only one weather condition and only on visibility level. Then we apply the above algorithm for each bin separately. This approach has a serious drawback where the number of bins will be very large and there will be many bins with very little data or no data at all. Consequently, we develop a mathematical modeling technique that pools the data and attempts to estimate the different cut-off speeds without dividing the dataset into clusters (bins).

In this paper, traffic speed is modeled by using a mixture of two linear regressions where each linear equation describes the relationship between the independent variables (visibility and weather conditions) and the dependent variable which is the speed. In other words instead of using a mixture of two components, which has only unchanged means, we model the speed using a mixture of two components where their means are a function of the weather condition and visibility level. The proposed algorithm is shown in Table 4.

Table 4: The Proposed Algorithm.

1. Use the EM algorithm described in the above section to fit two component distributions to the training dataset, as demonstrated in Equation (12).

$$f(y|\lambda,\beta_1,\beta_2,\sigma_1,\sigma_2) = \lambda\frac{1}{\sqrt{2\pi}\sigma_1}e^{\frac{\left(y-X^T\beta_1\right)^2}{2\sigma_1^2}} + (1-\lambda)\frac{1}{\sqrt{2\pi}\sigma_2}e^{\frac{\left(y-X^T\beta_2\right)^2}{2\sigma_2^2}}, \quad (12)$$

Where vector x is the vector of weather conditions and visibility predictors.

Here $(X^T\beta_1,\sigma_1)$ and $(X^T\beta_2,\sigma_2)$ are the location and spread of the first and second component distributions and λ is the mixture parameter.

2. For unseen data use the weather condition, the visibility level and the equations of the means ($X^T\beta_1$ and $X^T\beta_2$) to calculate the locations (means) of the two components.
3. Calculate the cut-off speed. We have two options to calculate the cut-off speed and we can use any of them.
   - 3.1. Calculate the 0.001 quintile of the free-flow state speed distribution (the distribution with the higher estimated location parameter).
   - 3.2. Calculate the cut-off speed using equation (10) which minimizes the classification error.
4. Use the cut-off speed as a threshold to classify the state of each segment along the road.

All segments with speeds greater than the threshold are classified as free-flow segments and other segments are classified as congested segments. The output of the above algorithm is a spatiotemporal binary matrix that is of the same dimensions as the spatiotemporal speed matrix. The one of the binary matrix identifies a segment as congested and a zero represents free-flow.

Case Study

1. Data Description

The freeway stretch of I-66 eastbound to connect I-81 and Washington D.C. is selected as the test site in this study. High traffic volumes are usually observed during morning and afternoon peak hours on I-66 heading towards Washington D.C., therefore various types or scales of traffic bottleneck can be located on the selected study site. Furthermore, a wide variety of weather conditions can be observed on I-66 considering the geographic location. Therefore, the study site

provides a great environment to test the proposed congestion identification algorithm and investigate the impacts of different weather conditions.

The weather data on the study site including weather condition and visibility level were collected from the weather station at Reagan Airport. That's almost 5 miles away from the easternmost point of the roadway segment on the study site. Besides weather data, traffic speed is the key input to identify congestion by using the proposed algorithm. In this study, the traffic data are provided by INRIX, which mainly collects probe data by GPS-equipped vehicles and supplemented with traditional road sensor data, as well as mobile devices and other sources [17]. The probe data on the test site covers 64 freeway segments with a total length of 74.4 miles. The average segment length is 1.16 miles long, and the length of each segment is unevenly divided in the raw data from 0.1 to 8.22 miles. The location of the study site and deployment of roadway segments are presented in Figure 18. Three years of INRIX probe data from 2011 to 2013 on the study site and the corresponding weather information are used in the case study to evaluate the proposed algorithm. The raw data provides the average speed for each roadway segment and is collected at one-minute intervals.

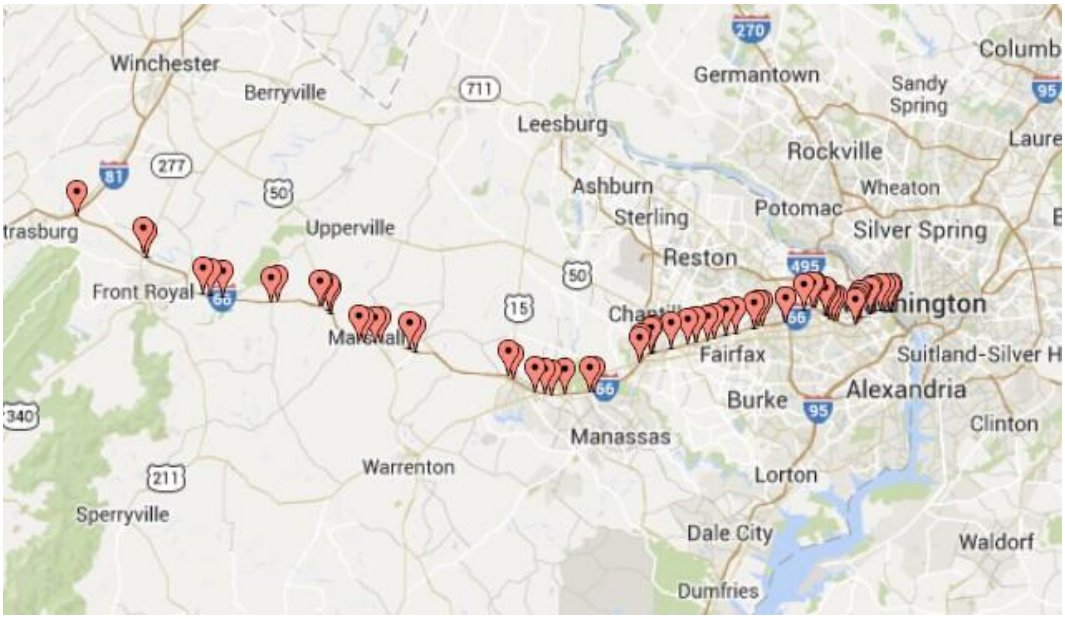


Figure 18: The Study Site on I-66 Eastbound (Source: Google Maps).

In order to use the collected traffic data in the proposed algorithm, data reduction is needed to transfer the raw measurement data into the required format of input data. In this paper, spatiotemporal traffic state matrix is the main component of input data for the proposed approach. The INRIX data is collected by each roadway segment on different time interval. Each roadway segment represents a Traffic Management Center (TMC) station, and the geographic information of TMC station is also provided. The average speed for each TMC station can be used to derive spatiotemporal traffic state matrix. However, the raw INRIX data includes several problems, such as geographically inconsistent sections, irregular time intervals of data collection, and missing data [18-20]. Considering these problems, the data reduction process is illustrated in Figure 19. Note that the data reduction is not constrained to the INRIX probe data, other types of traffic measurement data from various sensing techniques (e.g. loop detector, GPS and etc.) can also be used as the input to generate spatiotemporal traffic state matrix in the data reduction process.

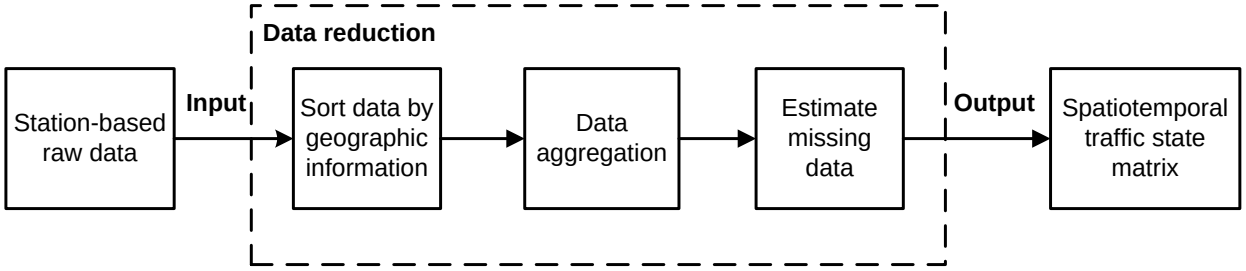
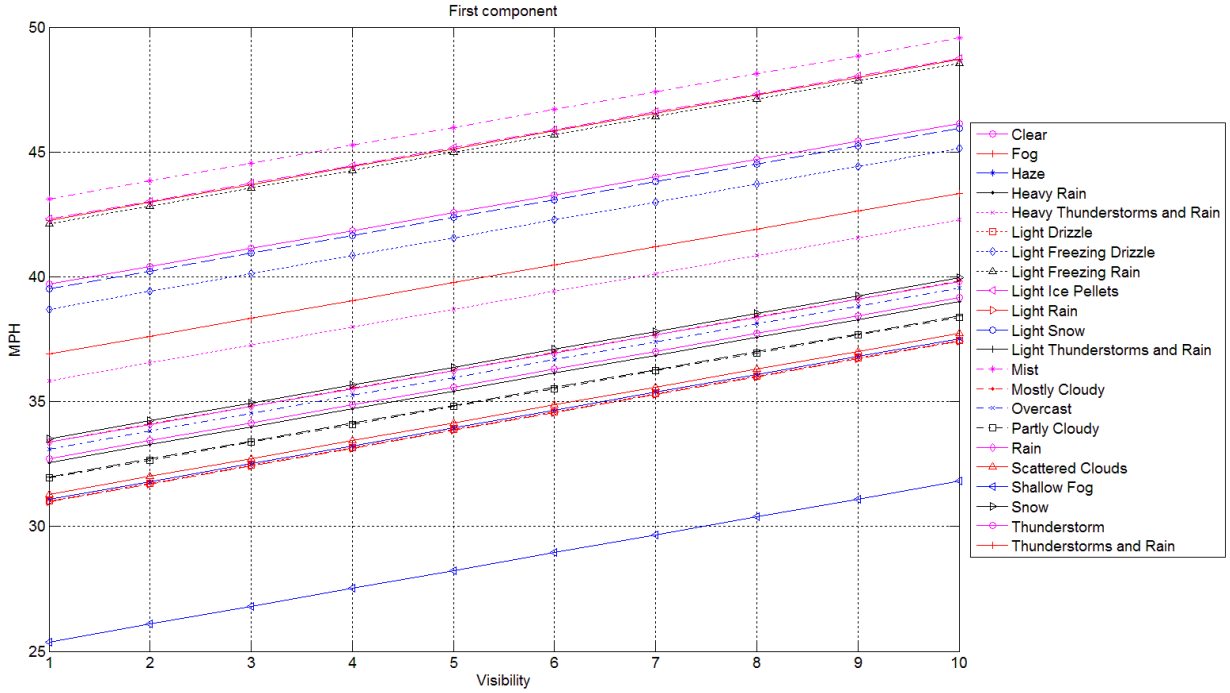


Figure 19: Data Reduction of INRIX Probe Data.

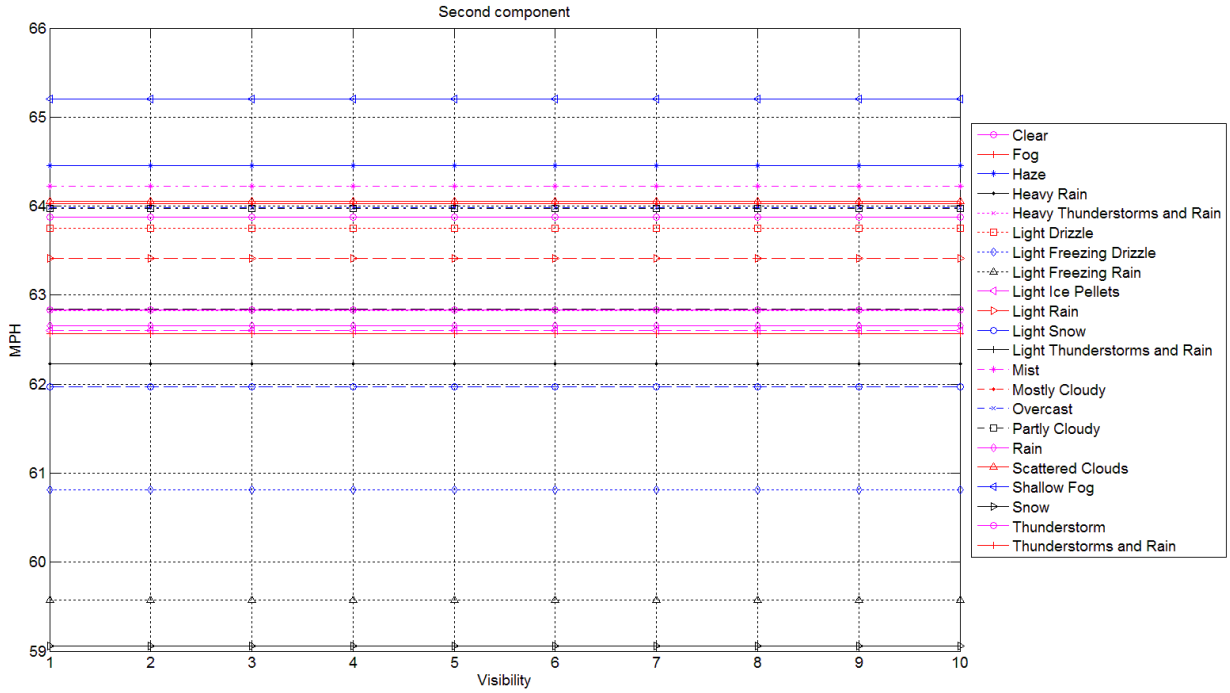
According to the geographic information of each TMC station, the raw data is sorted along the roadway direction (e.g. towards eastbound of I-66). An examination should be adopted to check any overlapping or inconsistent stations along the direction. Afterward, the speed data are aggregated by time intervals (e.g. 5 minutes in this study) to reduce the noise and smooth measurement error in this study. In this way, the raw data is aggregated to the form of daily data matrix along spatial and temporal intervals. It should be noted that missing data usually exist on the developed data matrix, therefore data imputation methods should be conducted to estimate the missing data by values of neighboring cells [20, 21]. Consequently, the daily spatiotemporal traffic state matrix can be generated for congestion identification algorithm. The weather data are transformed into the same format of daily spatiotemporal matrix to the speed data to investigate the impacts of different weather conditions and visibility levels on congestion identification.

2. Results of Appling the Mixture of Two Linear Regressions

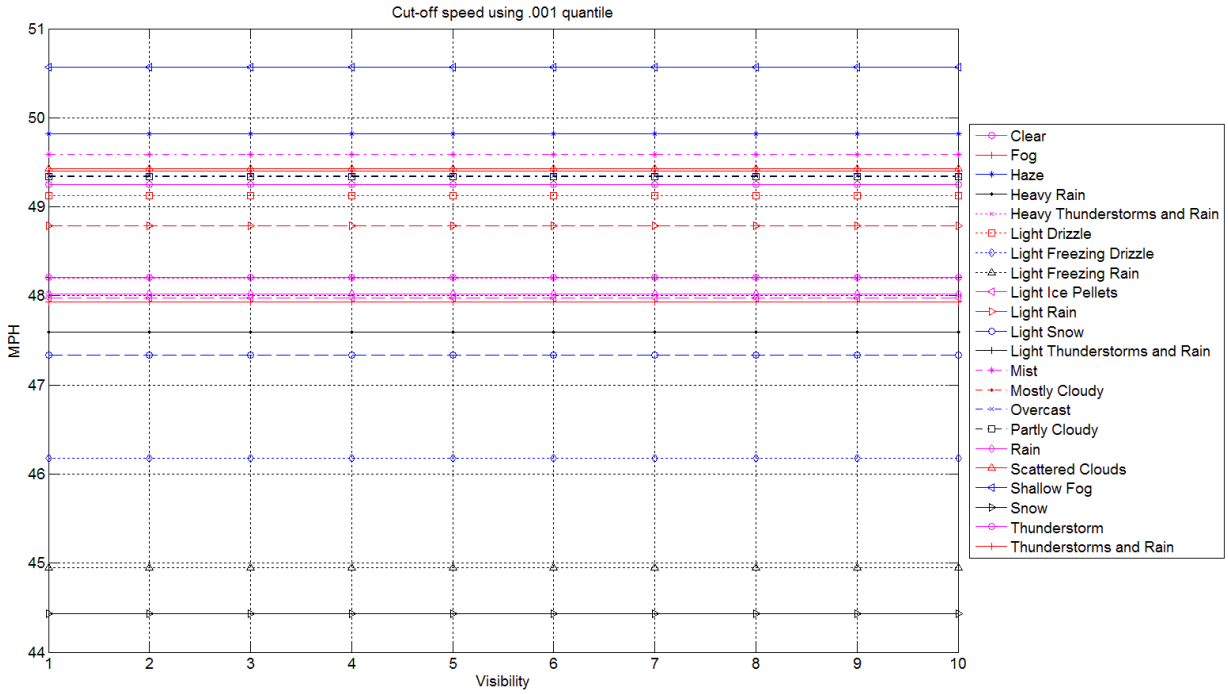
By applying the above model, the data were grouped by weather condition. Three weather conditions are removed because they have very little data. A total of 1000 realizations were then drawn randomly from each weather group to construct a random sample. Each random sample included the speed and visibility level together with indicator variables for the weather condition (categorical). The speeds were used as the response variable and the weather code and visibility as the explanatory variables (predictors). The coefficients of the predictors, variance of each component (σ_1^2, σ_2^2) and the proportions (λ_1, λ_2) of each component were estimated using the above iterative EM algorithm (Equations 3-7). This procedure was repeated many times by bootstrapping the sample construction without repetition. The final model parameters are the mean or the median of all model coefficients. Once the final model was derived, the mean of the speed distribution changed with weather condition and visibility level for both regimes (free-flow and congested). Given any combination of weather and visibility, the final model computes the mean speeds for the free-flow and congested regimes. Furthermore, using the ($\sigma_1^2, \sigma_2^2, \lambda_1, \lambda_2$) the model computes the Bayesian cut-off speed or the 0.001 quantile cut-off speed.


First component
MPH
Visibility
50
45
40
35
30
25
1
2
3
4
5
6
7
8
9
10
Clear
Fog
Haze
Heavy Rain
Heavy Thunderstorms and Rain
Light Drizzle
Light Freezing Drizzle
Light Freezing Rain
Light Ice Pellets
Light Rain
Light Snow
Light Thunderstorms and Rain
Mist
Mostly Cloudy
Overcast
Partly Cloudy
Rain
Scattered Clouds
Shallow Fog
Snow
Thunderstorm
Thunderstorms and Rain

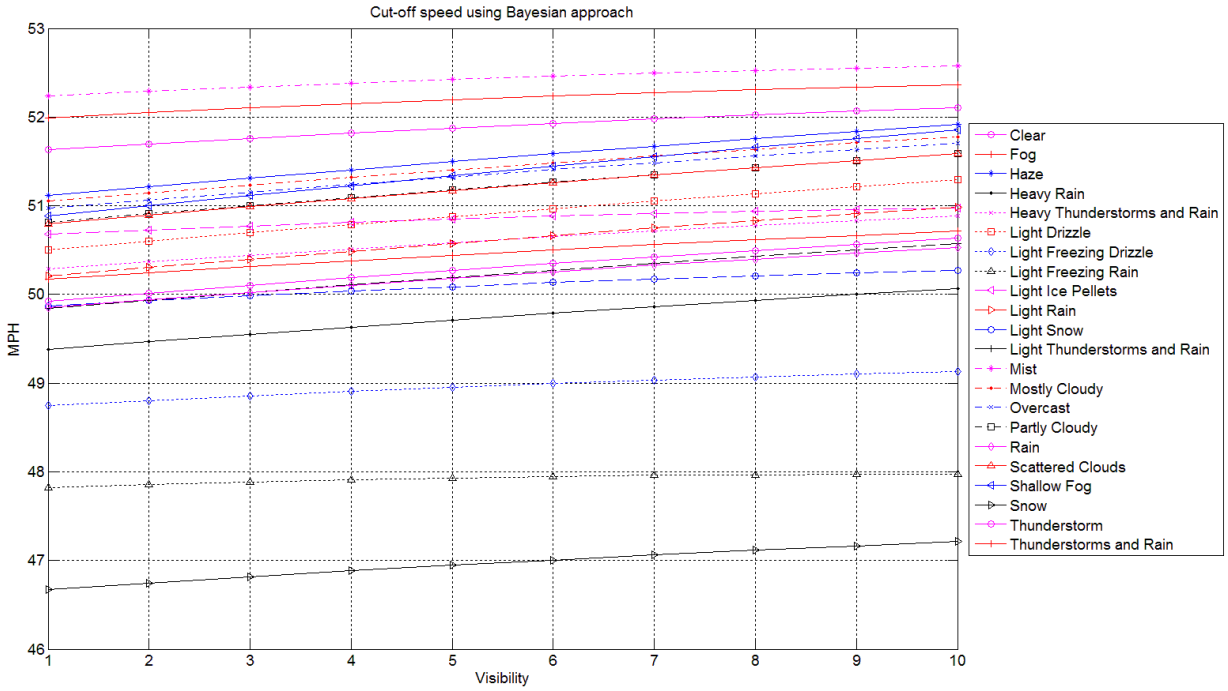
(a)


Second component
MPH
Visibility
66
65
64
63
62
61
60
59
1
2
3
4
5
6
7
8
9
10
Clear
Fog
Haze
Heavy Rain
Heavy Thunderstorms and Rain
Light Drizzle
Light Freezing Drizzle
Light Freezing Rain
Light Ice Pellets
Light Rain
Light Snow
Light Thunderstorms and Rain
Mist
Mostly Cloudy
Overcast
Partly Cloudy
Rain
Scattered Clouds
Shallow Fog
Snow
Thunderstorm
Thunderstorms and Rain

(b)



(c)



(d)

Figure 20: Results of the Mean and Cut-off Speeds Using the Medians of Models Coefficients. (A) Variation of the Mean Speed by The First Component (Congested); (B) Variation of the Mean Speed by the Second Component (Free-flow); (C) The Cut-off Speed by 0.001 Quantile; (D) The Cut-off Speed by the Bayesian Method.

(d)

Figure 20 shows the calculated means and cut-off speed using the above model when the final model parameters are the median of all models coefficients. The y-axis of all panels in Figure 20 is the speed measured in mph and the x-axis is the visibility level. Panel (a) shows the mean of the congested component (first component) at different weather conditions. The mean of congested component increases as the visibility get better. Panel (b) shows the mean of the free-flow component (second component) at the different weather condition. The mean of the free-flow does not remarkably affect the visibility level. Panel (c) shows the .001 cut-off speed for the different weather condition and visibility levels. Because this threshold based only on the free-flow distribution, it is not remarkably affected by the visibility level. This threshold suffers from increasing the increase of false negative as the visibility level increases. In order to explain this disadvantage let's call the distribution of the free-flow is our null hypotheses distribution and the distribution of the congestion speed is our alternative hypotheses. As the visibility level increases the means of the alternative and the null get closer and type II error, which is the failure to reject a false null hypothesis, is increased. In other terms, the false negatives increases where negative here is choosing the null hypothesis. At the same time this threshold has the advantage of keeping the type I error, which is the incorrect rejection of a true null hypothesis, close to .001. Panel (d) shows the change of the Bayesian cut-off speed with both weather conditions and visibility levels. This cut-off speed is affected by both visibility and weather condition because it takes into account both free-flow and congestion distributions. The common between both methods of calculating the cut-off speed is that it is low for bad weather such as snow and light freezing rain because the roadway surface conditions. We expected some weather conditions have low cut-off speed, but they are not because the traffic demand becomes less at these conditions.

Table 5: Estimated Coefficients for Linear Regression Using the Medians of Model Parameters.

	Free-flow	Congested
'Clear' (Intercept)	63.8794	38.9904
'Visibility'	0.0004	0.7174
'Fog'	0.1542	2.5626
'Haze'	0.5731	-8.6185
'Heavy Rain'	-1.6477	-7.1347
'Heavy Thunderstorms and Rain'	-1.0508	-3.8735
'Light Drizzle'	-0.1228	-8.6846
'Light Freezing Drizzle'	-3.0676	-0.9973
'Light Freezing Rain'	-4.2996	2.4235
'Light Ice Pellets'	-1.2715	2.6105
'Light Rain'	-0.4571	-8.7240
'Light Snow'	-1.9111	-0.1932
'Light Thunderstorms and Rain'	-1.0374	-7.7132
'Mist'	0.3444	3.4138
'Mostly Cloudy'	0.1563	-6.3225
'Overcast'	0.1029	-6.6054
'Partly Cloudy'	0.0953	-7.7646
'Rain'	-1.2181	-6.3281
'Scattered Clouds'	0.1766	-8.4237
'Shallow Fog'	1.3234	-14.3274
'Snow'	-4.8163	-6.1945
'Thunderstorm'	-1.0417	-6.9934
'Thunderstorms and Rain'	-1.3095	-2.7989
σ	4.7370	14.4225
λ_j	0.8688	0.1312

3. An Illustrative Example to Use the Proposed Mixture Linear Regression Model

In this subsection, an example is provided to demonstrate how to use the above fitted mixture of linear regression to get the cut-off speed in both cases .001 quantile and Bayesian. The above model answers the question that what is the cut-off speed at certain weather condition and certain visibility level. For example, what is the cut-off speed at "Mist" weather condition and visibility level equals 5. Given Table 6, the mean of the free-flow distribution can be calculated by multiplying predictors' vector by the coefficient vector of the free-flow speed distribution as shown in equation

$$\mu_F|_{(\text{visibility=5,mist})} = [1\ 5\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0] \begin{bmatrix} 63.3720 \\ 0.0606 \\ 0.5998 \\ 0.6981 \\ -1.3275 \\ -0.7416 \\ 0.0705 \\ -2.8769 \\ -4.1322 \\ -1.2637 \\ -0.4331 \\ -1.7594 \\ -1.0436 \\ 0.7302 \\ 0.1508 \\ 0.1192 \\ 0.0962 \\ -1.0285 \\ 0.1768 \\ 1.3678 \\ -4.4279 \\ -1.0412 \\ -1.0488 \end{bmatrix} \quad (13)$$

$$\mu_F|_{(\text{visibility=5,mist})} = 63.372 + 5 * 0.0606 + 1 * 0.7302 = 64.4052 \quad (14)$$

Where the first term in the predictors' vector is corresponding to the intercept coefficient and the number 5 is the visibility level and the other predictors are the code for mist weather condition.

using the $\mu_F = 64.4052$ and $\sigma_F = 4.7386$ we can calculate the .001 quantile cut-off speed by finding the value of speed at which the cumulative distribution function of free-flow distribution equals .001.

$$\text{cut} - \text{off speed}_{\ .001\ \text{quantile}} = 49.7618$$

If we wanted to calculate the Bayesian threshold, in this case we need to calculate the mean of the speed distribution at congestion. The multiplication process for getting the $\mu_{\text{congestion}}$ is shown in equation (15-16)

$$\mu_C|_{(\text{visibility=5,mist})} = [1\ 5\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0] \begin{bmatrix} 39.1063 \\ 0.7031 \\ 2.1454 \\ -8.7354 \\ -7.0658 \\ -4.0504 \\ -8.6757 \\ -1.1802 \\ 2.0802 \\ 2.5874 \\ -8.7218 \\ -0.5385 \\ -7.3123 \\ 3.0476 \\ -6.3016 \\ -6.6682 \\ -8.1741 \\ -6.3963 \\ -8.3979 \\ -14.2329 \\ -6.3436 \\ -6.7387 \\ -2.7412 \end{bmatrix} \quad (15)$$

$$\mu_C|_{(\text{visibility=5,mist})} = 39.1063 + 5 * 0.7031 + 1 * 3.0476 = 45.6694 \quad (16)$$

Then using equation (15) which is driven in the previous report we can get the Bayesian cut-off speed

$$\text{cut} - \text{off speed}_{\text{Bayesian}} = \min\left\{\frac{-\mu_F\sigma_C{}^2+\mu_C\sigma_F{}^2 \mp \sqrt{(\mu_F\sigma_C{}^2-\mu_C\sigma_F{}^2)^2-(\sigma_F{}^2-\sigma_C{}^2)\left(2\sigma_C{}^2\sigma_F{}^2\ln\left(\frac{\lambda_F\ \sigma_C}{(1-\lambda_F)\sigma_F}\right)-\mu_F{}^2\sigma_C{}^2+\mu_C{}^2\sigma_F{}^2\right)}}{(\sigma_F{}^2-\sigma_C{}^2)}\right\} \quad (17)$$

where the means in equation (17) are calculated above and the other parameters are obtained during the model fitting. All parameters are recalled again below

$$\mu_F = 64.4052$$
$$\sigma_F = 4.7386$$
$$\lambda_F = 0.8685$$
$$\mu_C = 45.6694$$
$$\sigma_C = 14.3988$$
$$\lambda_C = 0.1315$$

Substituting these parameters in equation (17), we get

$$\text{cutoff speed}_{\text{Bayesian}} \approx 52.6$$

Study Conclusions and Future Work

In this paper, two components mixture of linear regression is used in the proposed method to model the speed distributions at different traffic conditions including free-flow and congested traffic. To the best of our knowledge we are the first to integrate the weather conditions and the visibility level into the automatic congestion algorithm. The speed of each regime is modeled as

a normal distribution its mean is function of weather condition and visibility. The proposed model overcomes the problem of limited data for some weather condition and visibility level by pooling the data during fitting the model. We expect our proposed algorithm becomes the state of art tool that is used for congestion identification during both planning and operation.

As is the case with any research effort further enhancements to the model are required to model speed using more weather information like wind direction and speed. The model will be validated using other datasets from other road stretches. Another future enhancement of the model is replacing the symmetric normal distributions with other distribution such as log-normal or gamma distribution that can model the skew of the data.

References

[1] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking," *IEEE Transactions on Signal Processing*, vol. 50, pp. 174-188, 2002.

[2] R. Arnott and K. Small, "The Economics of Traffic Congestion," *American Scientist*, vol. 82, pp. 446-455, 1994.

[3] M. Sweet, "Does Traffic Congestion Slow the Economy?," *Journal of Planning Literature*, vol. 26, pp. 391-404, November 1, 2011 2011.

[4] H. Jianming, M. Qiang, W. Qi, Z. Jiajie, and Z. Yi, "Traffic congestion identification based on image processing," *Intelligent Transport Systems, IET*, vol. 6, pp. 153-160, 2012.

[5] H. Chen, H. A. Rakha, S. A. Sadek, and B. J. Katz, "A Particle Filter Approach for Real-time Freeway Traffic State Prediction," in *Transportation Research Board 91st Annual Meeting*, Washington D.C., 2012.

[6] H. Chen, H. A. Rakha, and S. A. Sadek, "Real-time Freeway Traffic State Prediction: A Particle Filter Approach," in *14th International IEEE Conference on Intelligent Transportation Systems*, Washington, DC, USA, 2011, pp. 626-631.

[7] J. Guiyan, N. Shifeng, C. Ande, M. Zhiqiang, and Z. Chunqin, "The method of traffic congestion identification and spatial and temporal dispersion range estimation," in *Informatics in Control, Automation and Robotics (CAR), 2010 2nd International Asia Conference on*, 2010, pp. 36-39.

[8] M. Elhenawy, H. A. Rakha, and C. Hao, "An automated statistically-principled bottleneck identification algorithm (ASBIA)," in *Intelligent Transportation Systems - (ITSC), 2013 16th International IEEE Conference on*, 2013, pp. 1846-1851.

[9] Z.-q. Sun, J.-q. Feng, W. Liu, and X.-m. Zhu, "Traffic congestion identification based on parallel SVM," in *Natural Computation (ICNC), 2012 Eighth International Conference on*, 2012, pp. 286-289.

[10] L. Xu, Y. Yue, and Q. Li, "Identifying Urban Traffic Congestion Pattern from Historical Floating Car Data," *Procedia - Social and Behavioral Sciences*, vol. 96, pp. 2084-2095, 11/6/ 2013.

[11] M. Elhenawy and H. Rakha, "Title," unpublished|.

[12] R. D. De Veaux, "Mixtures of linear regressions," *Computational Statistics & Data Analysis*, vol. 8, pp. 227-245, 11// 1989.

[13] S. Faria and G. Soromenho, "Fitting mixtures of linear regressions," *Journal of Statistical Computation and Simulation*, vol. 80, pp. 201-225, 2010/02/01 2009.

[14] F. Guo, H. Rakha, and S. Park, "Multistate Model for Travel Time Reliability," *Transportation Research Record: Journal of the Transportation Research Board* 2010.
[15] F. Guo, Q. Li, and H. Rakha, "Multistate Travel Time Reliability Models with Skewed Component Distributions," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2315, pp. 47-53, 12/01/ 2012.
[16] C. Chen, A. Skabardonis, and P. Varaiya, "Systematic identification of freeway bottlenecks," *Freeway Operations and Traffic Signal Systems 2004*, pp. 46-52, 2004.
[17] INRIX. (2012). *http://www.inrix.com/trafficinformation.asp*. Available: http://www.inrix.com/trafficinformation.asp
[18] M. Elhenawy, H. Chen, and H. A. Rakha, "Dynamic travel time prediction using data clustering and genetic programming," *Transportation Research Part C: Emerging Technologies*, vol. 42, pp. 82-98, 2014.
[19] H. Chen and H. A. Rakha, "Real-time travel time prediction using particle filtering with a non-explicit state-transition model," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 112-126, 2014.
[20] H. Rakha, H. Chen, A. Haghani, and K. Farokhi, "Assessment of Data Quality Needs for use in Transportation Applications," MAUTC-2011-01, 2013.
[21] H. Chen, H. A. Rakha, and C. C. McGhee, "Dynamic Travel Time Prediction using Pattern Recognition," in *20th World Congress on Intelligent Transportation Systems*, Tokyo, Japan, 2013.

Appendix A

The derivation of the Bayesian threshold (cut-off) assuming that the $p(\text{congestion}) = 1 - \lambda$ and $p(\text{freeflow}) = \lambda$

$$\lambda \frac{1}{\sqrt{2\pi}\sigma_1} e^{\frac{(v_{th}-\mu_1)^2}{2{\sigma_1}^2}} = (1-\lambda) \frac{1}{\sqrt{2\pi}\sigma_2} e^{\frac{(v_{th}-\mu_2)^2}{2{\sigma_2}^2}} \quad (18)$$

$$\frac{\lambda}{(1-\lambda)} \frac{\sigma_2}{\sigma_1} e^{\frac{(v_{th}-\mu_1)^2}{2{\sigma_1}^2}} = e^{\frac{(v_{th}-\mu_2)^2}{2{\sigma_2}^2}} \quad (19)$$

$$\ln \frac{\lambda}{(1-\lambda)} \frac{\sigma_2}{\sigma_1} - \frac{(v_{th}-\mu_1)^2}{2{\sigma_1}^2} = -\frac{(v_{th}-\mu_2)^2}{2{\sigma_2}^2} \quad (20)$$

$$\ln \frac{\lambda}{(1-\lambda)} \frac{\sigma_2}{\sigma_1} - \frac{({v_{th}}^2+{\mu_1}^2-2\mu_1 v_{th})}{2{\sigma_1}^2} + \frac{({v_{th}}^2+{\mu_2}^2-2\mu_2 v_{th})}{2{\sigma_2}^2} = 0 \quad (21)$$

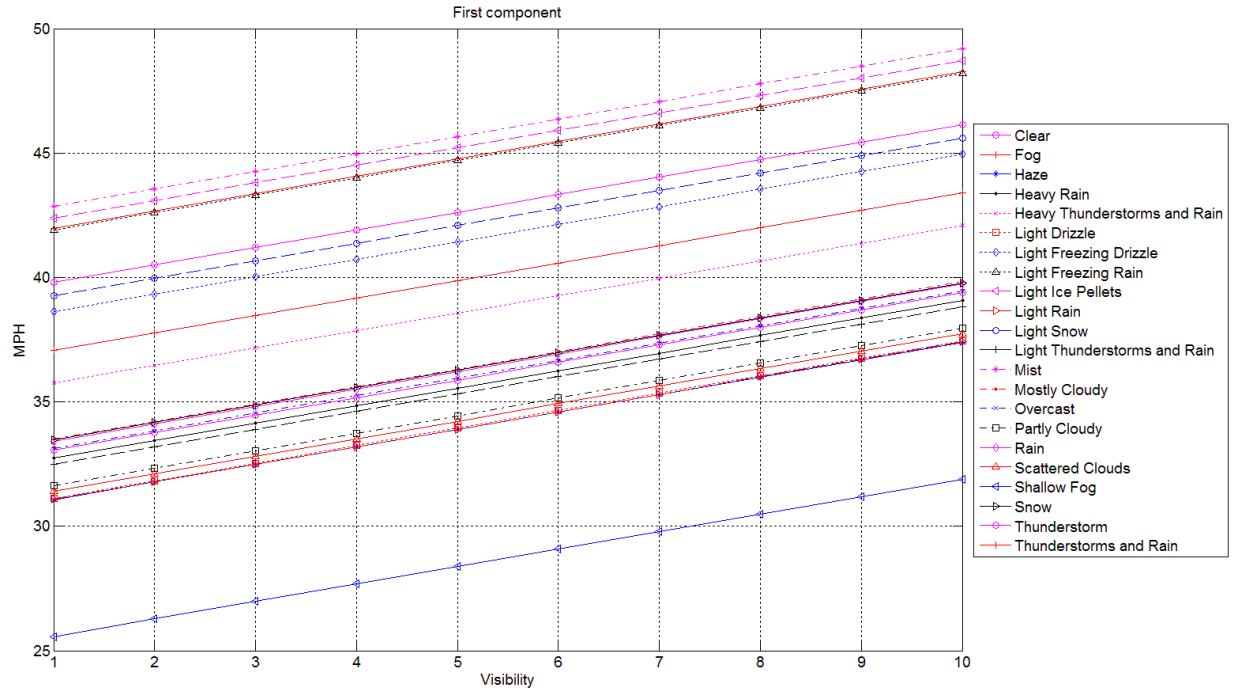
$$({\sigma_1}^2-{\sigma_2}^2){v_{th}}^2 + (2\mu_1{\sigma_2}^2 - 2\mu_2{\sigma_1}^2)v_{th} + 2{\sigma_2}^2{\sigma_1}^2 \ln\left(\frac{\lambda}{(1-\lambda)}\frac{\sigma_2}{\sigma_1}\right) - {\mu_1}^2{\sigma_2}^2 + {\mu_2}^2{\sigma_1}^2 = 0 \quad (22)$$

The above equation is quadratic and can be solved using the following general formula

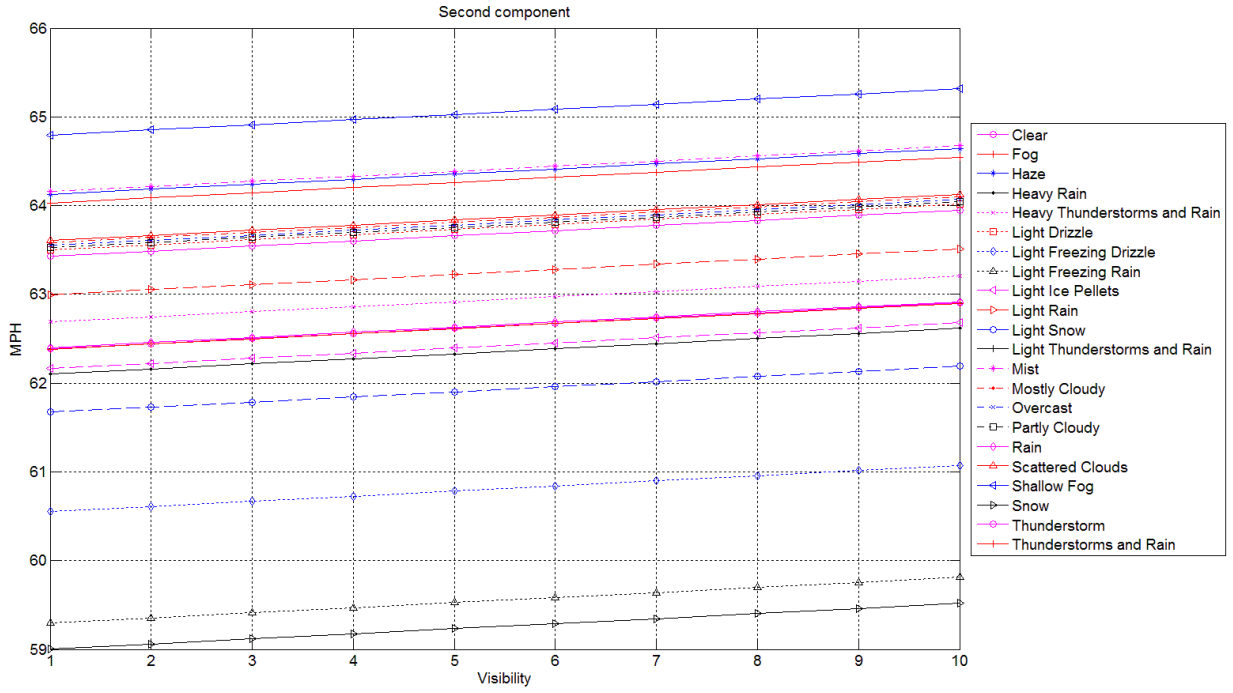
$$x = \frac{-b \pm \sqrt{b^2-4ac}}{2a} \quad (23)$$

$$v_{th} = \min\left\{\frac{-\mu_1{\sigma_2}^2+\mu_2{\sigma_1}^2 \mp \sqrt{(\mu_1{\sigma_2}^2-\mu_2{\sigma_1}^2)^2-({\sigma_1}^2-{\sigma_2}^2)\left(2{\sigma_2}^2{\sigma_1}^2\ln\left(\frac{\lambda}{(1-\lambda)}\frac{\sigma_2}{\sigma_1}\right)-{\mu_1}^2{\sigma_2}^2+{\mu_2}^2{\sigma_1}^2\right)}}{({\sigma_1}^2-{\sigma_2}^2)}\right\} \quad (24)$$

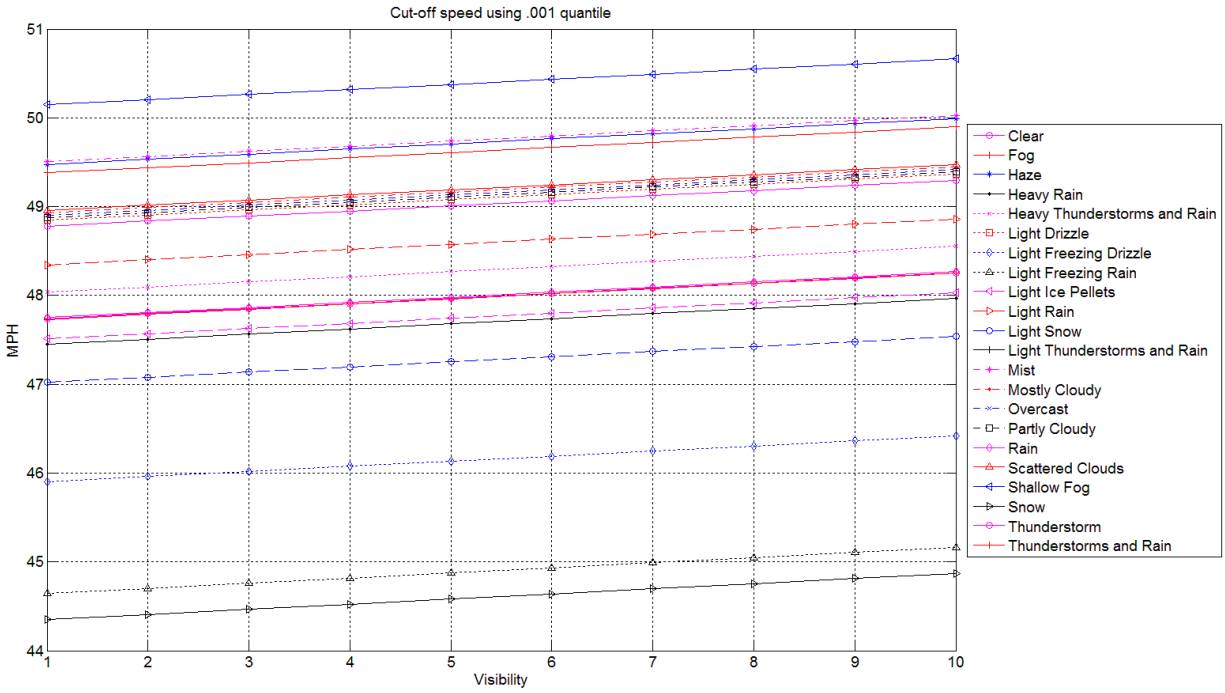
Appendix B



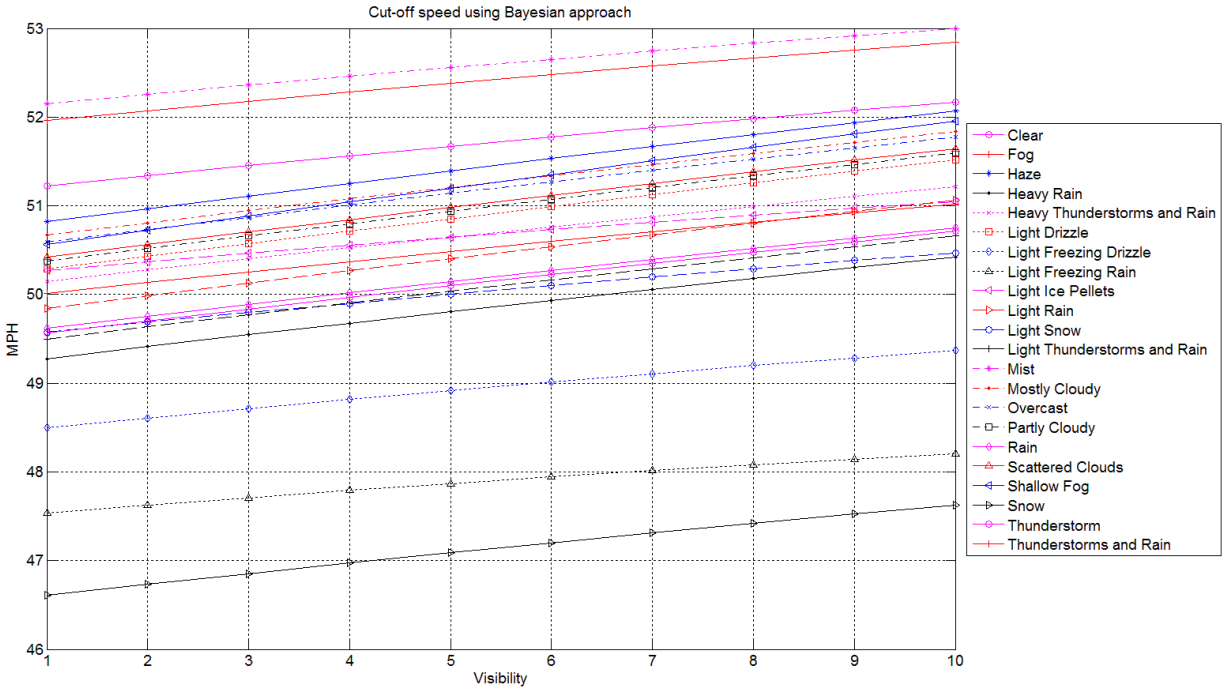
(a)



(b)



(c)



(d)

Figure 21: Results of the Mean and Cut-off Speeds Using the Means of Models Coefficients. (A) Variation of the Mean Speed by the First Component (Congested); (B) Variation of the Mean Speed by the Second Component (Free-flow); (C) The Cut-off Speed by 0.001 Quantile; (D) The Cut-off Speed by The Bayesian Method.

Figure 21 shows the calculated means and cut-of threshold using the above model but this time the final model parameters are the mean of all models coefficients. It's clear that the cut-off speeds in both Figure 20 and Figure 21 are almost the same.

Table 6: The Coefficients of the Mixture Two-component Linear Regression Using the Mean to Obtain the Final Model Parameters.

	Free-flow	Congested
'Clear' (Intercept)	63.3720	39.1063
'Visibility'	0.0606	0.7031
'Fog'	0.5998	2.1454
'Haze'	0.6981	-8.7354
'Heavy Rain'	-1.3275	-7.0658
'Heavy Thunderstorms and Rain'	-0.7416	-4.0504
'Light Drizzle'	0.0705	-8.6757
'Light Freezing Drizzle'	-2.8769	-1.1802
'Light Freezing Rain'	-4.1322	2.0802
'Light Ice Pellets'	-1.2637	2.5874
'Light Rain'	-0.4331	-8.7218
'Light Snow'	-1.7594	-0.5385
'Light Thunderstorms and Rain'	-1.0436	-7.3123
'Mist'	0.7302	3.0476
'Mostly Cloudy'	0.1508	-6.3016
'Overcast'	0.1192	-6.6682
'Partly Cloudy'	0.0962	-8.1741
'Rain'	-1.0285	-6.3963
'Scattered Clouds'	0.1768	-8.3979
'Shallow Fog'	1.3678	-14.2329
'Snow'	-4.4279	-6.3436
'Thunderstorm'	-1.0412	-6.7387
'Thunderstorms and Rain'	-1.0488	-2.7412
σ	4.7386	14.3988
λ_j	0.8685	0.1315

Chapter 5: A Unified Automatic Congestion Identification Model Considering Visibility and Weather Conditions Using Mixture Linear Regression

This chapter is based on
Mohammed Elhenawy and Hesham Rakha, "A Unified Automatic Congestion Identification Model Considering Visibility and Weather Conditions Using Mixture Linear Regression," under review paper.

Abstract

Automatic identification of traffic congestion is an important task for transportation planning and traffic operations. Transportation engineers use congestion identification as a preprocessing step for the downstream analysis to identify and rank traffic bottlenecks. Real time automatic congestion identification is one of the important routines of intelligent transportation systems (ITS). In the near future conveying congestion information to connected vehicles can help them making better route choice and avoid congested road segments. Previous efforts usually use traffic state measurements (speed, flow, occupancy) to develop congestion identification algorithms. However, the impacts of weather conditions to identify congestion have not been investigated in the existing studies. In this paper, the impacts of weather and visibility on the congestion identification algorithm are investigated. The proposed algorithm uses the speed probe data and the corresponding weather and visibility to build a transferable model that can be used on any road stretch. Our algorithm assumes traffic states can be classified into three regimes: congestion, speed at capacity, and free-flow. Moreover, the speed distribution follows three components mixture components its means are functions in weather and visibility. The mean of each component is defined using a linear regression model its predictors are the different weather conditions and visibility levels. We used three datasets from VA, CA, and TX and the corresponding weather information to estimate the model parameters. The fitted model is used to calculate the speed cut-off between congestion and speed at capacity which minimize either the Bayesian classification error or the false positive (congestion) rate. The test results demonstrate the proposed method produces promising congestion identification output by considering weather condition and visibility level and we expect it becomes the state of art automatic congestion identification algorithm.

Introduction

Traffic congestion has become one of the modern life problems in many metropolitan areas. This growing problem has environmental effects. During congestion time, cars cannot run efficiently so air pollution, carbon dioxide (CO2) emissions, and fuel use increases. In 2007, Americans lost $87.2 billion in wasted fuel and lost productivity. This waste reached $115 billion in 2009 [1]. Congestion increases travel time, for example back in 1993 driving under congested condition causes a delay of about six-tenths of a minute per kilometer of travel on expressways and 1.2 minutes delay per kilometer of travel in arterials[2]. The congestion problem becomes worse as reported by Texas Transportation Institute where the number of Americans' wasted hours in traffic congestion becomes fivefold between 1982 and 2005 Moreover, congestion has its economic effect where studies show that congestion slow metropolitan growth, inhibits agglomeration economies and shape economic geographies[3]. Traffic Congestion could result

by obstruction or lack of road capacity which is a kind of inefficient use of the roads. This problem can be relaxed by increasing the road-building budgets to build more infrastructures. But adding more road capacity is costly and budget is limited, and the construction itself takes long time. With the continuous increase in traffic volumes, managing traffic, particularly at times of peak demand, is a good and inexpensive solution to congestion. Advanced traffic management systems (ATMS) use various applications of intelligent transportation systems (ITS) to manage traffic and reduce congestion problems. Recently, the advancement in communication and computers greatly improve ITS and make it more capable of identifying and reducing congestion. ITS is an effective solution to traffic problem where it improves the dynamic capacity of the road system without building extra expensive infrastructure [4]. Accurate and real-time traffic information is the foundation of ITS.

Congestion usually starts from a road bottleneck, and then spills over the neighbor road segments. It takes time until this congestion to disappear. Depending on the frequency of congestion occurrence, traffic congestion can be divided into two categories [5]. The first is recurrent traffic congestion, and the second is accidental (non-recurring) traffic congestion. Recurrent traffic congestion, which usually results from exceeding the road capacity, is easier to identify and predict. The accidental traffic congestion usually results from traffic incident or severe weather conditions. Traffic congestion is different at different location, time periods, and different weather condition.

The impact of weather on the freeway traffic operations is a big concern for roadway management agencies, however, there is little research work done to link weather and congestion in a quantitative sense. Two groups at the University of Washington correlated weather and traffic phenomena using the Traffic Data Acquisition and Distribution (TDAD) data mine and the Doppler radar data mine [6]. Their basic idea is that, moving weather sells can be tracked and predicted using weather radar then they can find the correlation between the properties of the weather cell and observed traffic states. Nookala studied the traffic congestion caused by weather conditions and its effect on traffic volume and travel time. Nookala observed an increase in the traffic congestion at inclement weather conditions due to drop in the freeway capacity while the traffic demand does not drop significantly[7].Chung et al. used traffic data collected over a 2 year period from 1 July 2002 to 30 June 2004 at Tokyo Metropolitan Expressway (MEX) and showed a decrease in free-flow speed and in capacity with increasing amount of rainfall[8]. Brilon and Ponzlet used three years of historical data for 15 freeway sites in Germany to investigated impacts of several factors including weather on speed-flow relationships [9]. They found that wet roadway conditions cause different speed reduction at highways with different lane number. Agarwal et al. highlighted that due to the different roadway and driver characteristics results obtained from studies outside the United States can't applied to the United States. Moreover, the result obtained from rural freeway segments within the United States may be different from urban freeway[10]. Ibrahim and Hall used limited historical dataset and multiple regression analysis to study the impact of rain and snow on speed [11]. Their results showed that light rain and snow causes similar reductions in speeds (3%–5%), while 14%–15% and 30%–40% reduction in speed is caused by heavy rain and heavy snow respectively. Rakha et al used weather data (precipitation and visibility) and loop detector data (speed, flow, and occupancy) obtained from Baltimore, Twin Cities, and Seattle in the USA to quantify the impact of inclement weather on traffic stream behavior and key traffic stream parameters, including free-flow speed, speed-at-capacity, capacity, and jam density. For more detailed discussion of the Rakha's result readers are referred to [12].

During the last few years, many automatic congestion identification algorithms are proposed. ASBIA is an algorithm that uses speed measurements over short temporal and spatial intervals and segments, respectively to identify the status of a segment using t-test[13]. The outputs of the algorithm are the status of the roadway segment (free-flow or congested) and the confidence level of the test (p-value). Another algorithm uses vehicle trajectories in intelligent vehicle infrastructure co-operation system (IVICS)[4]. Then the spatial–temporal trajectories are considered as an image to extract the propagation speed of congestion wave and construct congestion template. Finally correlation is evaluated between the template and the spatial–temporal velocity image to identify the congestion. Parallel SVM is used in [14] to identify traffic congestion. The authors propose Parallel SVM instead of SVM because the training computation cost of SVM is expensive and congestion identification is a real-time task.

Floating car data is used in [15] to find meaningful congestion patterns. The analysis of the floating car data is done using a method based on data cube and the spatial-temporal related relationship of slow-speed road segment to identify the traffic congestion. The research team at the center for sustainable mobility (CSM) at the Virginia Tech Transportation institute (VTTI) developed an algorithm to identify congested segments using a spatiotemporal speed matrix [16]. The proposed algorithm fits two lognormal (or normal) distributions to the training dataset.

To the best of our knowledge, no research addresses the impacts of both visibility and weather conditions on congestion identification. In this paper the impacts of weather conditions and visibility levels on the congestion identification algorithm are investigated by modeling the speed distribution as mixture of two normal components whose means are linear functions of weather condition and visibility level. So that based on these factors the two normal components may get close or apart and the cut-off speed is changed. The proposed algorithm is built using three different datasets from three different states (VA, TX, and CA). The results of our proposed model are promising and reasonable where, for example, the cut-off speed increases as the visibility level increases.

The remainder of this paper is organized as follows. First, a brief background of the method used in this work is given. After that, the proposed algorithm is introduced. The datasets used in the case study is described. Subsequently the result of the experimental work is explained and an illustrative example is given to show how to implement the proposed model. Finally, conclusions and recommendations for future work are presented.

Mixture of Linear Regressions[17, 18]

Finite mixture models are powerful tools in analyzing a wide variety of random phenomena. They are used to model random phenomena in many fields including agriculture, biology, economics, medicine, and genetics. A mixture of linear regressions is one of the mixture families studied carefully in the literature. It can be used to model the travel time for different traffic regimes.

The mixture of linear regression can be written as:

$$f(y|X) = \sum_{j=1}^{m} \frac{\lambda_j}{\sigma_j\sqrt{2\pi}} e^{-\frac{\left(y-x^T\beta_j\right)^2}{2\sigma_j^2}} \quad (1)$$

Or as

$$y_i = \begin{cases} x_i^T\beta_1 + \epsilon_{i1} & \text{with probability } \lambda_1 \\ x_i^T\beta_2 + \epsilon_{i2} & \text{with probability } \lambda_2 \\ \vdots & \\ x_i^T\beta_m + \epsilon_{im} & \text{with probability } 1 - \sum_{q=1}^{m-1}\lambda_q \end{cases} \quad (2)$$

Where y_i is a response corresponding to a predictors' vector x_i^T , β_j is a vector of regression coefficients for the j^{th} mixture component, λ_j is a mixing probability of the j^{th} mixture component, ϵ_{ij} are normal random errors, and m is the number of components in mixture model. Model parameters $\psi = \{\beta_1, \beta_2, \ldots, \beta_m, \sigma_1^2, \sigma_2^2, \ldots, \sigma_m^2, \lambda_1, \lambda_2, \ldots, \lambda_m\}$ can be estimated by maximizing the log-likelihood of Equation (1); given a set of response predictor pairs$(y_1, x_1), (y_2, x_2), \ldots, (y_n, x_n)$, and using the Expectation-Maximization algorithm (EM).

1. EM Algorithm

The EM algorithm iteratively finds maximum likelihood estimates by alternating the E-step and M-step. Let $\psi^{(k)}$ be parameter estimates after the k^{th} iteration. On the E-step, the posterior probability of the i^{th} observation comes from component j and is computed as shown in Equation (3).

$$w_{ij}^{(k+1)} = \frac{\lambda_j^{(k)}\phi_j\left(y_i|x_i, \psi^{(k)}\right)}{\sum_{j=1}^{m}\lambda_j^{(k)}\phi_j\left(y_i|x_i, \psi^{(k)}\right)}$$

(3)

Where $\phi_j\left(y_i|x_i, \psi^{(k)}\right)$ is the probability density function of the j^{th} component

On the M-step, new parameter estimates $\psi^{(k+1)}$ maximizing the log-likelihood function in Equation (1) are calculated, as shown in Equations (4-5).

$$\lambda_j^{(k+1)} = \frac{\sum_{i=1}^{n} w_{ij}^{(k+1)}}{n} \quad (4)$$

$$\hat{\beta}_j^{(k+1)} = (X^T W_j X)^{-1} X^T W_j Y \quad (5)$$

Where X is an $nx(p+1)$ predictor matrix, Y is the corresponding nx1 response vector, and W is an nxn diagonal martrix having $w_{ij}^{(k+1)}$ along its diagonal

$$\hat{\sigma}_j^{2(k+1)} = \frac{\sum_{i=1}^{n} w_{ij}^{(k+1)}(y_i - x_i^T\hat{\beta}_j^{(k+1)})^2}{\sum_{i=1}^{n} w_{ij}^{(k+1)}} \quad (6)$$

The E-step and M-step are alternated repeatedly until the incomplete log-likelihood change is arbitrarily small, as shown in Equation (7).

$$\left|\prod_{i=1}^{n}\sum_{j=1}^{m}\lambda_j^{(k+1)}\phi_j\left(y_i|x_i, \psi^{(k+1)}\right) - \prod_{i=1}^{n}\sum_{j=1}^{m}\lambda_j^{(k)}\phi_j\left(y_i|x_i, \psi^{(k)}\right)\right| < \xi \quad (7)$$

Where ξ is a small number

The Proposed congestion identification algorithm

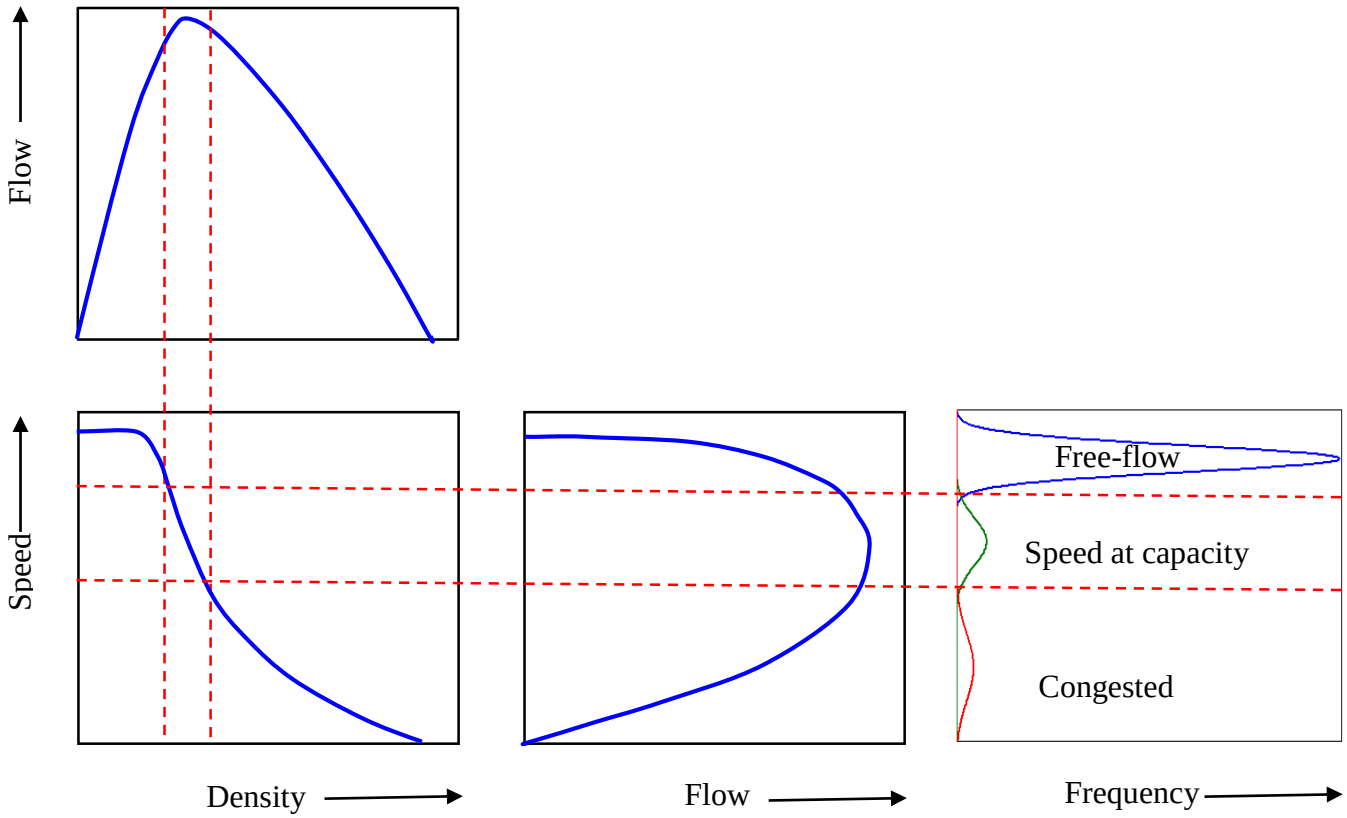


Figure 22: Illustration of Link Between the Fundamental Diagrams and the Three Components Mixture

As shown in Figure 22, we divide the traffic states of a road segment into three traffic regimes where the speed of each regime can be modeled by a lognormal distribution. So that the overall speed distribution can be represented as a mixture of three log-normal components· First regime is the free-flow which has the speed distribution with the highest mean. At free-flow regime, the density lies below the capacity density. The second regime is the Congested flow which has the speed distribution with the lowest mean. The congested flow is characterized by the traffic that has density lies between the capacity density and the jam density. The third regime is the capacity flow which separates the free-flow from the congested flow and it speed distribution has a mean that between the means of the other two regimes. As shown in several studies the flow fundamental diagram is affected by the weather conditions [12, 19, 20]. So that we expect the mean of the speed distribution corresponding to each regime changes with weather and visibility. The proposed algorithm uses the mixture of three linear regression and real datasets to learn the means of the distribution as a function of weather and visibility and find the boundary between the three regimes. The proposed algorithm is shown below.

Table 7: The proposed Algorithm

1. Use the EM algorithm described earlier to fit three component distributions to locally-collected data, as demonstrated in Equation (8).

$$f(\log(y)|\,\lambda_1,\lambda_2,\beta_1,\beta_2,\beta_3,\sigma_1,\sigma_2,\sigma_3) = \lambda_1 \frac{1}{\sqrt{2\pi}\sigma_1} e^{\frac{\left(\log(y)-X^T\beta_1\right)^2}{2{\sigma_1}^2}} + \lambda_2 \frac{1}{\sqrt{2\pi}\sigma_2} e^{\frac{\left(\log(y)-X^T\beta_2\right)^2}{2{\sigma_2}^2}} + (1-\lambda_2-\lambda_1)\frac{1}{\sqrt{2\pi}\sigma_3} e^{\frac{\left(\log(y)-X^T\beta_3\right)^2}{2{\sigma_3}^2}}, \quad (8)$$

Where vector x is a vector of weather conditions and visibility predictors.

Here $(X^T\beta_1,\sigma_1)$, $(X^T\beta_2,\sigma_2)$, and $(X^T\beta_3,\sigma_3)$ are the locations and spreads of the mixture components, and (λ_1,λ_2) are the mixture parameters.

2. For unseen data use the weather condition, visibility level, and equations of the means ($X^T\beta_1$,$X^T\beta_2$, and $X^T\beta_3$) to calculate locations (means) of three components.
3. Calculate the cut-off speed. We have two options to calculate the cut-off speed and we can use either of them.
   3.1. Calculate 0.001 quintile of the speed at capacity (the middle distribution).
   3.2. Calculate cut-off speed using the Bayesian approach; which finds the intersection point (between congestion and speed at capacity) that minimizes the classification error.
4. Use cut-off speed as a threshold to classify the state of each road segment.

All segments with speeds greater than the threshold are classified as free-flow segments, and other segments are classified as congested segments. The output of the above algorithm is a spatiotemporal binary matrix with dimensions identical to the spatiotemporal speed matrix. A '1' in the binary matrix identifies a segment as congested, and a '0' represents free-flow conditions.

Experimental Work

1. Data Reduction

In order to use collected traffic data in the proposed algorithm, data reduction was an important process for transferring raw measured data into required input data formats. In general, the spatiotemporal traffic state matrix is a fundamental attribute of input data. Reduction of INRIX probe data is one example, and a similar process can be applied to other types of measured data (e.g. loop detector). INRIX data are collected for each roadway segment and time interval. Each roadway segment represents a TMC station. Geographic TMC station information is also provided. The average speed for each TMC station can be used to derive a spatiotemporal traffic state matrix. However, raw INRIX data includes geographically inconsistent sections, irregular time intervals of data collection, and missing data. Considering these problems, the data reduction process is illustrated in
Figure 23.

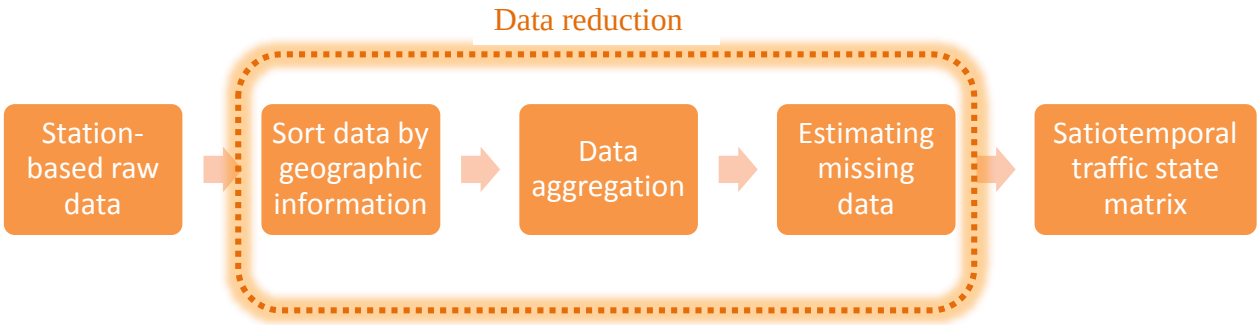


Figure 23: Data Reduction of INRIX Probe Data

Based on the geographic information of each TMC station, raw data are sorted along the roadway direction (e.g. towards eastbound or westbound). An examination should be adopted to check any overlapping or inconsistent stations along the direction. Afterward, speed data should be aggregated by time intervals (e.g. 5 minutes), according to the algorithm's resolution requirement. In this way, raw data can be aggregated into a daily matrix format, along spatial and temporal intervals. It should be noted that missing data usually exist in the developed data matrix. Therefore, data imputation methods should be conducted, to estimate the missing data by neighboring cell values. Consequently, the daily spatiotemporal traffic state matrix can be generated for congestion and bottleneck identification.

2. Study Sites

INRIX traffic data in three states (Virginia, Texas and California) were used to develop the proposed automatic congestion identification algorithm. Specifically, the study included 2011~2013 data along I-66 eastbound, 2012 data along US-75 northbound and 2012 data along I-15 southbound. The selected freeway corridor on I-66 is presented in Figure 24, which includes 36 freeway segments along 30.7 miles. Average speeds (or travel times) for each roadway segment are provided in the raw data, which were collected every minute. In order to reduce the stochastic noise and measurement error, raw speed data were aggregated by five-minute intervals. Therefore, the traffic speed matrix over spatial (upstream to downstream) and temporal (from 0:00 AM to 23:55 PM) domains could be obtained for each day. For the other two locations, daily speed matrices were obtained using the same procedure. Selected freeway corridors on US-75 and I-15 are presented in Figure 25 and Figure 26; including 81 segments across 38 miles, and 30 segments across 15.6 miles, respectively.

In addition to the traffic data, weather and visibility conditions were collected at weather stations closest to the three sites. Weather and visibility data were translated into data matrices having identical dimensions to the speed data. These data will be used to investigate the impacts of weather and visibility conditions on automatic congestion identification algorithm.

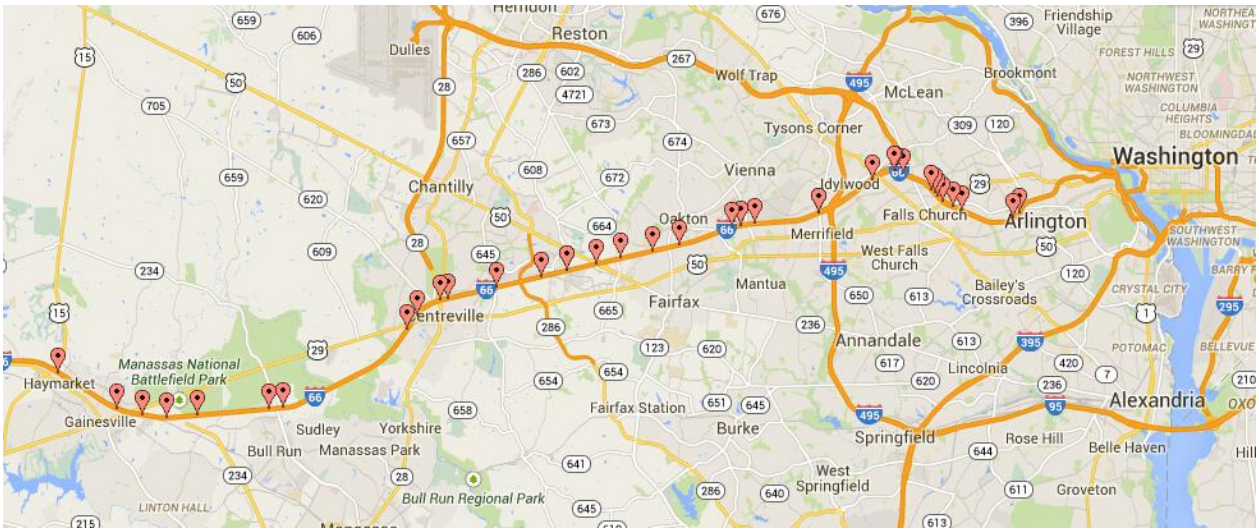


Figure 24: Layout of the Selected Freeway Stretch on I-66. (Source: Google Maps)

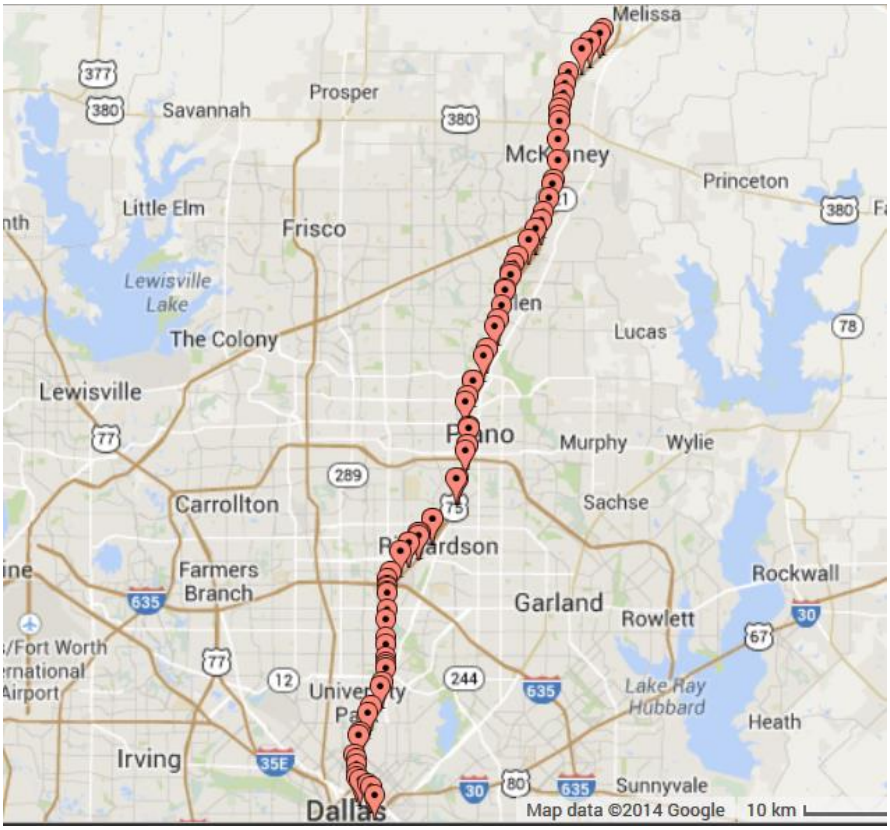


Figure 25: Layout of the Selected Freeway Stretch on US-75. (Source: Google Maps)

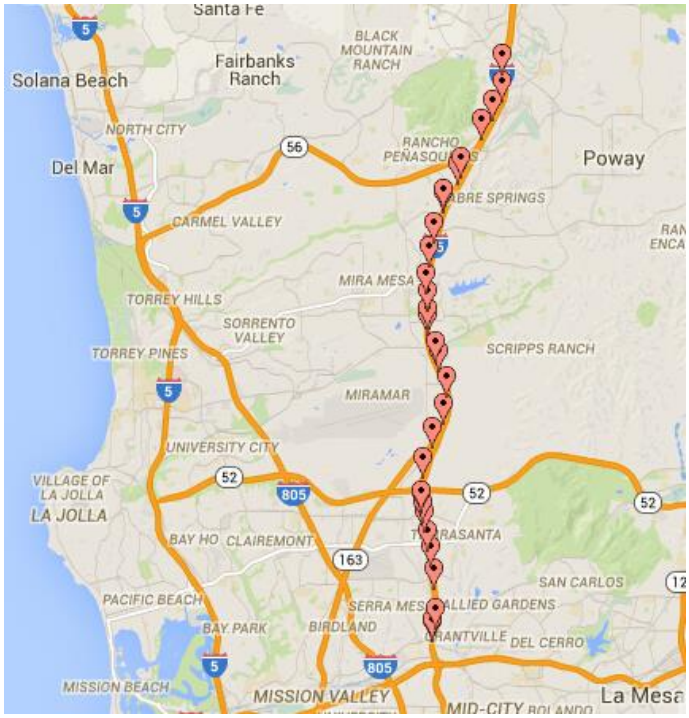


Figure 26: Layout of the Selected Freeway Stretch on I-15. (Source: Google Maps)

3. Effect of Visibility and Weather

This subsection describes the investigation of weather and visibility impacts on the cut-off speed (threshold) that is used to define the congested condition. The investigation was limited by the fact that data could not be divided into bins containing each weather condition and visibility level. Moreover many bins had small amounts of data or no data at all. With this in mind, the mixture of linear regressions is proposed to pool data and estimate cut-off speeds, without sorting the data into clusters. In this subsection, we describe a speed model, featuring a mix of three linear regressions. Each linear equation describes a relationship between independent variables (visibility and weather) and the dependent variable, which is speed. In other words, instead of mixing three components with unchanged means, the speed model mixed three components whose means were a function of weather and visibility.

4. Unified Model for all Three Datasets

In order to get a unified model that is independent of the location or the speed limit, we did the following:

1. Weather conditions for the three datasets were consolidated based on precipitation, as shown in Appendix A. Weather conditions from all three datasets were then mapped into these weather groups.
2. We put all the three datasets in one pool and did not include indicator variables that show the id of the dataset.
3. The speed is normalized by dividing the speed at each road segment by the posted maximum speed at this segment.

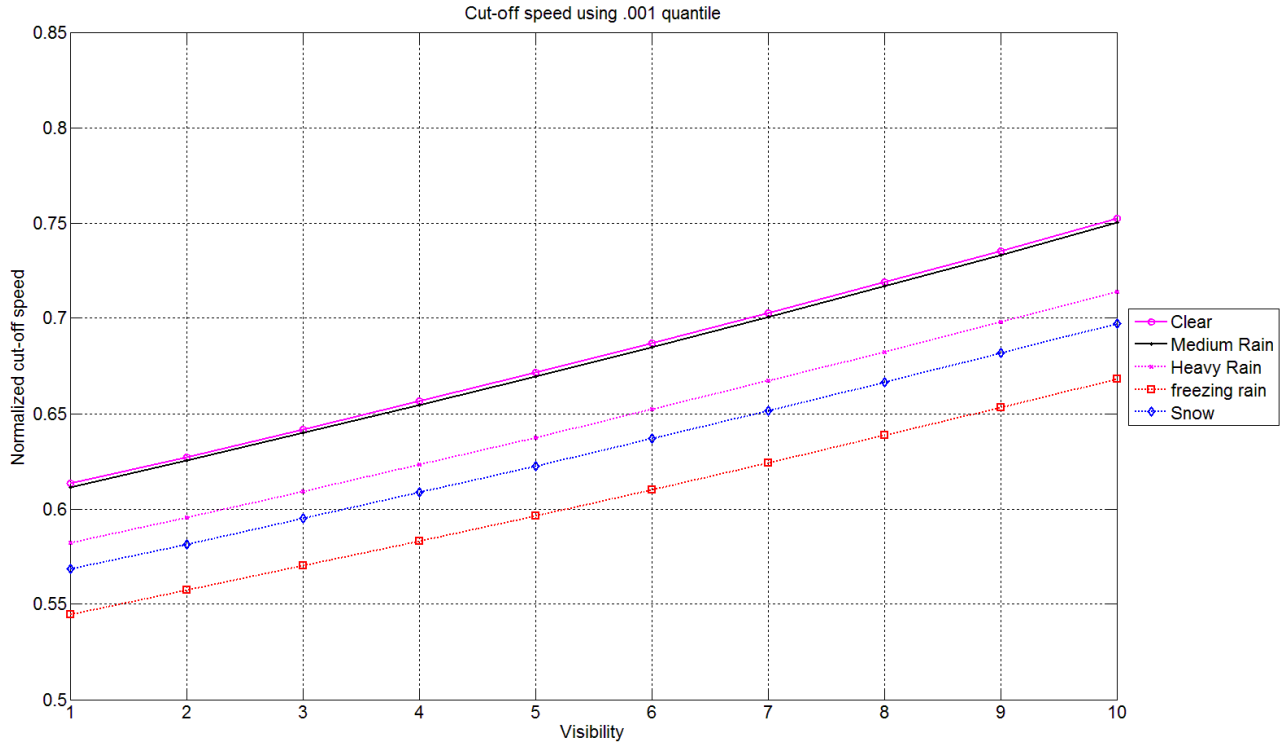
The unified model has a response which is the normalized speed come from the three dataset and the predictors are the indicator variables for the weather groups and the visibility level.

In applying the mixture of three linear regression model, speed and visibility data were grouped by weather. Because the dataset is huge and we cannot estimate the model parameters using the whole dataset at once due to memory issues, A total of 7000 random sample were then drawn randomly from each weather group, to construct a realization (dataset). Each random sample includes the speed and visibility level, together with indicator variables for the weather. Because speed distributions are skewed, the log-normal distribution is preferred to the normal distribution. Log speed was used as the response variable. Weather code and visibility were the explanatory variables (predictors). Coefficients of the predictors$(\beta_1, \beta_2, \beta_3)$, variance of each component $(\sigma_1^2, \sigma_2^2, \sigma_3^2)$ and proportions $(\lambda_1, \lambda_2, \lambda_3)$ of each component were estimated using the above iterative EM algorithm (Equations 3-6). This procedure was repeated 300 times by bootstrapping the sample construction without replacement. Final model parameters were the mean or median of all model coefficients. Once the final model was derived, we can observe the shift of the distribution mean with weather condition and visibility level in the three regimes (free-flow, speed at capacity and congested). Given any combination of weather and visibility, the final model computes mean speeds for the three regimes. Furthermore, using the estimated model's parameters, the model computes Bayesian and 0.001 quantile cut-off speeds.

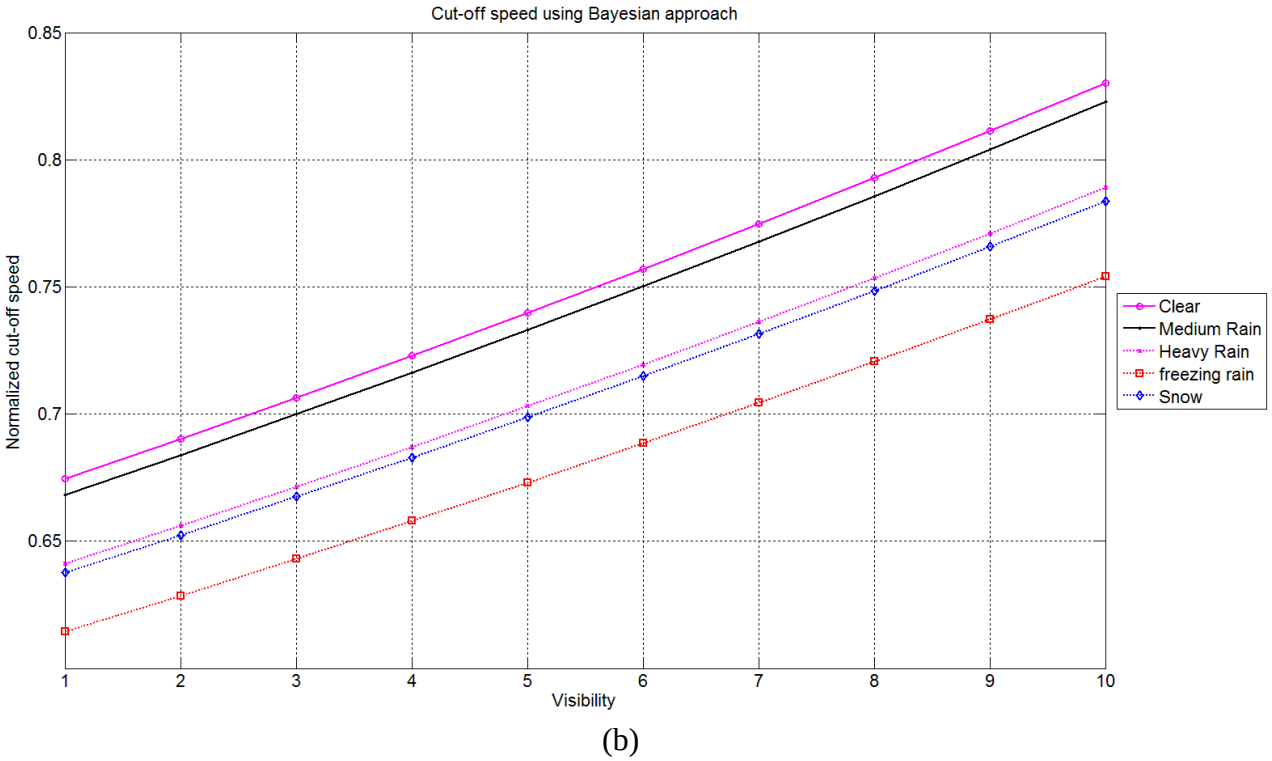
The estimated general model's parameters are shown in Table 8. As shown Figure 27 the results are sensible because all weather groups have cut-off speeds lower or equal to the clear group. Moreover, the cut-off speed increases as visibility increases. We should mention that the cut-off speeds for clear and light rain are very close so we can apply the cut-off speed of the clear condition at light rain as well. Appendix B shows the speed matrix and the corresponding binary matrix after applying the proposed algorithm.

Table 8: Unified Model Parameters

	Congestion	Speed at capacity	Free-flow
'Clear' (Intercept)	-0.9025	-0.1947	0.0335
'Visibility'	0.0260	0.0229	0.0026
'Medium Rain'	-0.0722	-0.0024	-0.0238
'Heavy Rain'	-0.0398	-0.0465	-0.0308
'freezing rain'	0.2809	-0.1134	-0.0018
'Snow'	0.1754	-0.0740	-0.0149
σ	0.4881	0.1027	0.0680
λ_j	0.0846	0.1123	0.8028



(a)



(b)

Figure 27: The General Model's Cut-off Speeds (a) Quantile, (b) Bayesian.

5. An Example Illustrating the Unified Model

Recall that, he model that explain the variation in normalized speed using the weather and visibility is shown in Equation (9)

$$f(\log(y)|\,\lambda_1,\lambda_2,\beta_1,\beta_2,\beta_3,\sigma_1,\sigma_2,\sigma_3) = \lambda_1 \frac{1}{\sqrt{2\pi}\sigma_1} e^{\frac{\left(\log(y)-X^T\beta_1\right)^2}{2{\sigma_1}^2}} + \lambda_2 \frac{1}{\sqrt{2\pi}\sigma_2} e^{\frac{\left(\log(y)-X^T\beta_2\right)^2}{2{\sigma_2}^2}} + (1-\lambda_2-\lambda_1)\frac{1}{\sqrt{2\pi}\sigma_3} e^{\frac{\left(\log(y)-X^T\beta_3\right)^2}{2{\sigma_3}^2}}, \quad (9)$$

Where vector x is the vector of weather conditions and visibility predictors, and y is the normalized speed.

Here $(X^T\beta_1,\sigma_1)$, $(X^T\beta_2,\sigma_2)$, and $(X^T\beta_3,\sigma_3)$ are the locations and spreads of the mixture components and (λ_1,λ_2) are the mixture parameter.

The above table shows equation that govern the locations of the three components are:

$$\mu_{\text{Congestion}} = -0.9025 + 0.0260 * \text{Visibility} - 0.0722 * \text{Medium Rain} - 0.0398 * \text{Heavy Rain} + 0.2809 * \text{freezing rain} + 0.1754 * \text{Snow} \quad (10)$$

$$\mu_{\text{Speed at capacity}} = -0.1947 + 0.0229 * \text{Visibility} - 0.0024 * \text{Medium Rain} - 0.0465 * \text{Heavy Rain} - 0.1134 * \text{freezing rain} - 0.0740 * \text{Snow} \quad (11)$$

$$\mu_{\text{Free-flow}} = 0.0335 + 0.0026 * \text{Visibility} - 0.0238 * \text{Medium Rain} - 0.0308 * \text{Heavy Rain} - 0.0018 * \text{freezing rain} - 0.0149 * \text{Snow} \quad (12)$$

Let's give an example to show how to come up with the Q quantile cut-off speed for given weather group and visibility level. .Based on the model the predictors' vector is as shown in Equation (13)

$$X^T = [1\ \text{Visibility}\ \ \text{Medium Rain}\ \ \text{Heavy Rain}\ \ \text{freezing rain}\ \ \text{Snow}] \quad (13)$$

Assume the weather is "freezing rain" and the visibility is "2", what is the Q quantile cut-off speed. Given the previous information, the predictors' vector is shown in equation (14),

$$X^T = [1\ 2\ \ 0\ 0\ 1\ 0] \quad (14)$$

Then the mean of speed at capacity component is calculated as shown in equation (15)

$$\mu_{\text{Speed at capacity}} = -0.1947 + 0.0229 * 2 - 0.0024 * 0 - 0.0465 * 0 - 0.1134 * 1 - 0.0740 * 0 \quad (15)$$

Manipulating the above equation, we get -0.2623 as the mean of the speed at capacity component. Then using the Matlab command "norminv(Q, -0.2623, 0.1123)" we get the Q quantile cut-off speed where 0.1123 is the standard deviation for the speed at capacity component. We should highlights that the standard deviation and the proportion parameters are constant and do not depend on the weather group or visibility.

Now, let assume we are interested in the .001 quantile at the "freezing rain" and the visibility is "2". Using the Matlab command "norminv(.001, -0.2623, 0.1123)" we get the .001 quantile cut-off speed which is -0.6093. -0.6093 is the cut-off speed on the log scale and the cut-off speed used to get the binary matrix is exp(-0.6093)= 0.5437.In the previous example, the .001 quantile cut-off speed is 0.5437 of the posted maximum speed. In other words, the cut-off speed is 0.5437*65= 35.3405 MPH if the posted maximum speed is 65 MPH.

Conclusions

This study developed models of speed distributions in free-flow, speed at capacity and congested traffic states. To the best of our knowledge, this is the first methodology integrates the

impact of weather and visibility into automated congestion identification. Moreover, this methodology is expected to be more portable because it is based on three different dataset covers three various regions that hopefully represent the US. The proposed algorithm is expected to be the state of practice at many DOTs because of its simplicity, promising result and suitability to run in real time scenarios. Because our algorithm precisely identify the traffic congestion, both spatially and temporally, it is recommended as an important first step towards identifying and ranking bottlenecks.

References

[1] *Managing Congestion Problems with Intelligent Transportation Systems*. Available: http://www.dot.state.fl.us/research-center/Program_Information/RS-Fall2012.pdf

[2] R. Arnott and K. Small, "The Economics of Traffic Congestion," *American Scientist*, vol. 82, pp. 446-455, 1994.

[3] M. Sweet, "Does Traffic Congestion Slow the Economy?," *Journal of Planning Literature*, vol. 26, pp. 391-404, November 1, 2011 2011.

[4] H. Jianming, M. Qiang, W. Qi, Z. Jiajie, and Z. Yi, "Traffic congestion identification based on image processing," *Intelligent Transport Systems, IET*, vol. 6, pp. 153-160, 2012.

[5] J. Guiyan, N. Shifeng, C. Ande, M. Zhiqiang, and Z. Chunqin, "The method of traffic congestion identification and spatial and temporal dispersion range estimation," in *Informatics in Control, Automation and Robotics (CAR), 2010 2nd International Asia Conference on*, 2010, pp. 36-39.

[6] D. J. Dailey, "The use of weather data to predict non-recurring traffic congestion," 2006.

[7] L. S. Nookala, "Weather impact on traffic conditions and travel time prediction," University of Minnesota Duluth, 2006.

[8] E. Chung, O. Ohtani, H. Warita, M. Kuwahara, and H. Morita, "Does weather affect highway capacity."

[9] W. Brilon and M. Ponzlet, "Variability of speed-flow relationships on German autobahns," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1555, pp. 91-98, 1996.

[10] M. Agarwal, T. H. Maze, and R. Souleyrette, "Impacts of weather on urban freeway traffic flow characteristics and facility capacity," in *Proceedings of the 2005 mid-continent transportation research symposium*, 2005.

[11] A. T. Ibrahim and F. L. Hall, *Effect of adverse weather conditions on speed-flow-occupancy relationships*, 1994.

[12] H. Rakha, M. Farzaneh, M. Arafeh, R. Hranac, E. Sterzin, and D. Krechmer, *Empirical studies on traffic flow in inclement weather*, 2007.

[13] M. Elhenawy, H. A. Rakha, and C. Hao, "An automated statistically-principled bottleneck identification algorithm (ASBIA)," in *Intelligent Transportation Systems - (ITSC), 2013 16th International IEEE Conference on*, 2013, pp. 1846-1851.

[14] Z.-q. Sun, J.-q. Feng, W. Liu, and X.-m. Zhu, "Traffic congestion identification based on parallel SVM," in *Natural Computation (ICNC), 2012 Eighth International Conference on*, 2012, pp. 286-289.

[15] L. Xu, Y. Yue, and Q. Li, "Identifying Urban Traffic Congestion Pattern from Historical Floating Car Data," *Procedia - Social and Behavioral Sciences*, vol. 96, pp. 2084-2095, 11/6/ 2013.

[16] Mohammed.Elhenawy, and Hesham A. Rakha. "Automatic Congestion Identification Using Two-Component Mixture Models." Transportation Research Board 94th Annual Meeting. No. 15-1440. 2015.
[17] R. D. De Veaux, "Mixtures of linear regressions," *Computational Statistics & Data Analysis*, vol. 8, pp. 227-245, 11// 1989.
[18] S. Faria and G. Soromenho, "Fitting mixtures of linear regressions," *Journal of Statistical Computation and Simulation*, vol. 80, pp. 201-225, 2010/02/01 2009.
[19] K. Meead Saberi and R. L. Bertini, "Empirical Analysis of the Effects of Rain on Measured Freeway Traffic Parameters."
[20] B. L. Smith, K. G. Byrne, R. B. Copperman, S. M. Hennessy, and N. J. Goodall, "AN INVESTIGATION INTO THE IMPACT OF RAINFALL ON FREEWAY TRAFFIC FLOW," 2003.

Appendix C

Table 9: Six Weather Groups

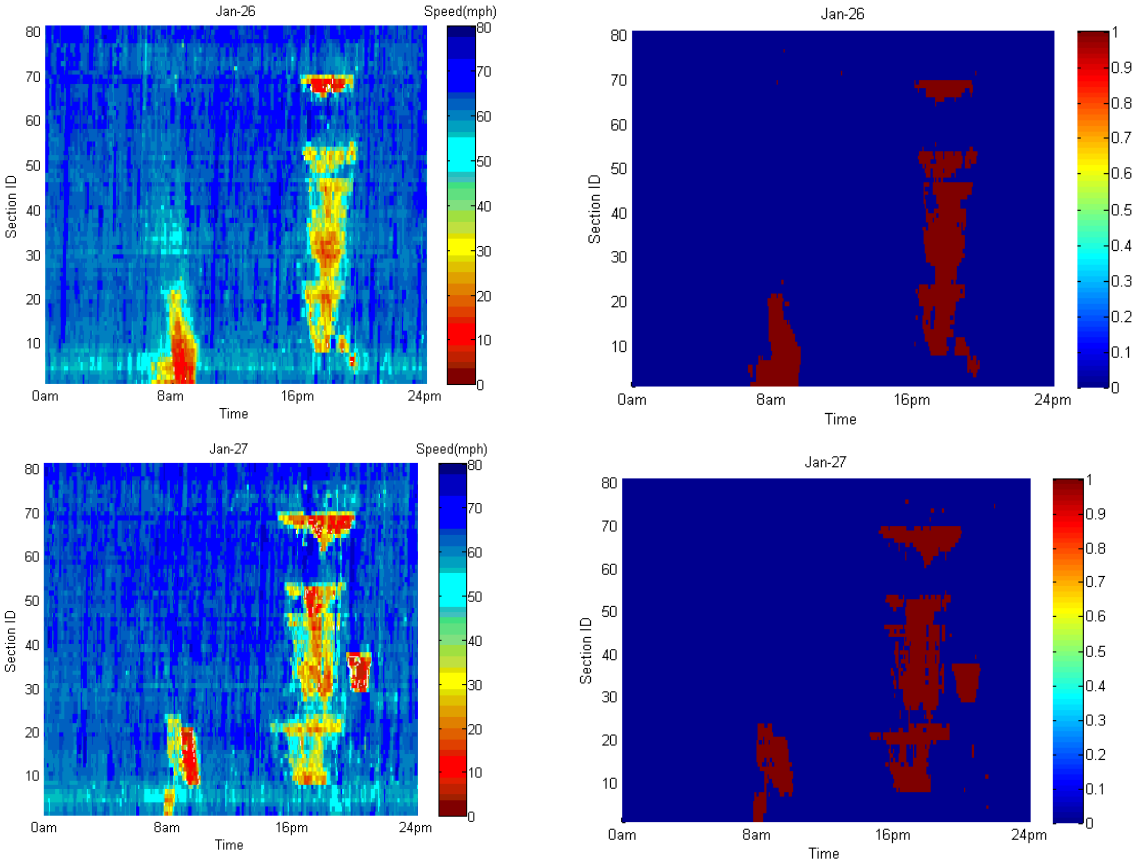
Groups	#
Clear	1
Light Rain	2
Rain	3
Heavy rain	4
Freezing rain	5
Snow	6

Table 10: Mapping Between Weather Conditions and Weather Groups

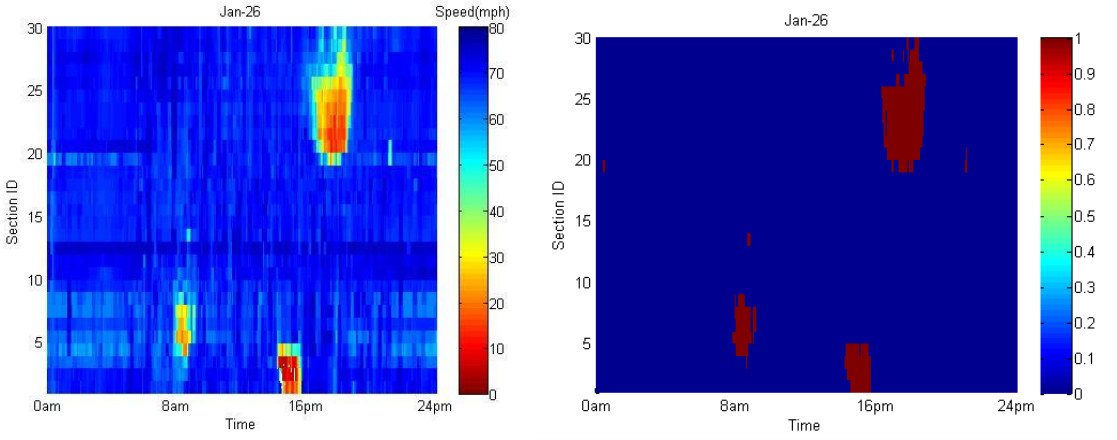
Weather Condition	Group #
'Blowing Snow'	6
'Clear'	1
'Drizzle'	2
'Fog'	1
'Freezing Rain'	5
'Haze'	1
'Heavy Drizzle'	2
'Heavy Rain'	4
'Heavy Rain Showers'	4
'Heavy Snow'	6
'Heavy Thunderstorms and Rain'	3
'Ice Pellets'	5
'Light Drizzle'	2
'Light Freezing Drizzle'	5
'Light Freezing Rain'	5
'Light Ice Pellets'	5
'Light Rain'	2
'Light Rain Showers'	2
'Light Snow'	6
'Light Thunderstorms and Rain'	3
'Light Thunderstorms and Snow'	6
'Mist'	1
'Mostly Cloudy'	1
'Overcast'	1
'Partly Cloudy'	1
'Patches of Fog'	1
'Rain'	3
'Scattered Clouds'	1
'Shallow Fog'	1
'Smoke'	1
'Snow'	6
'Thunderstorm'	1
'Thunderstorms and Rain'	3
'Thunderstorms and Snow'	6

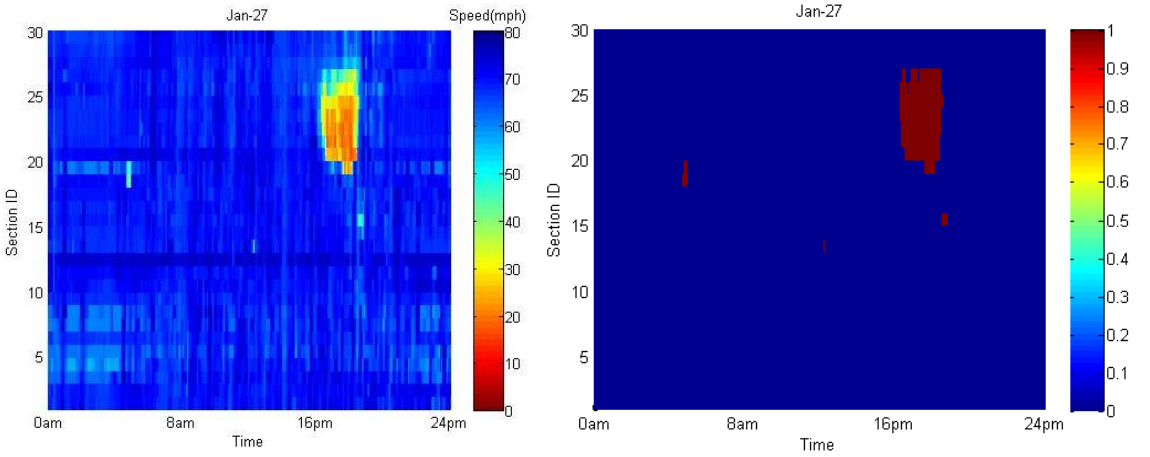
Appendix D

Figure 28 shows the speed matrix and the corresponding binary matrix after applying the proposed algorithm. The binary matrix will be further filtered to fill gaps and remove noise using image processing techniques.

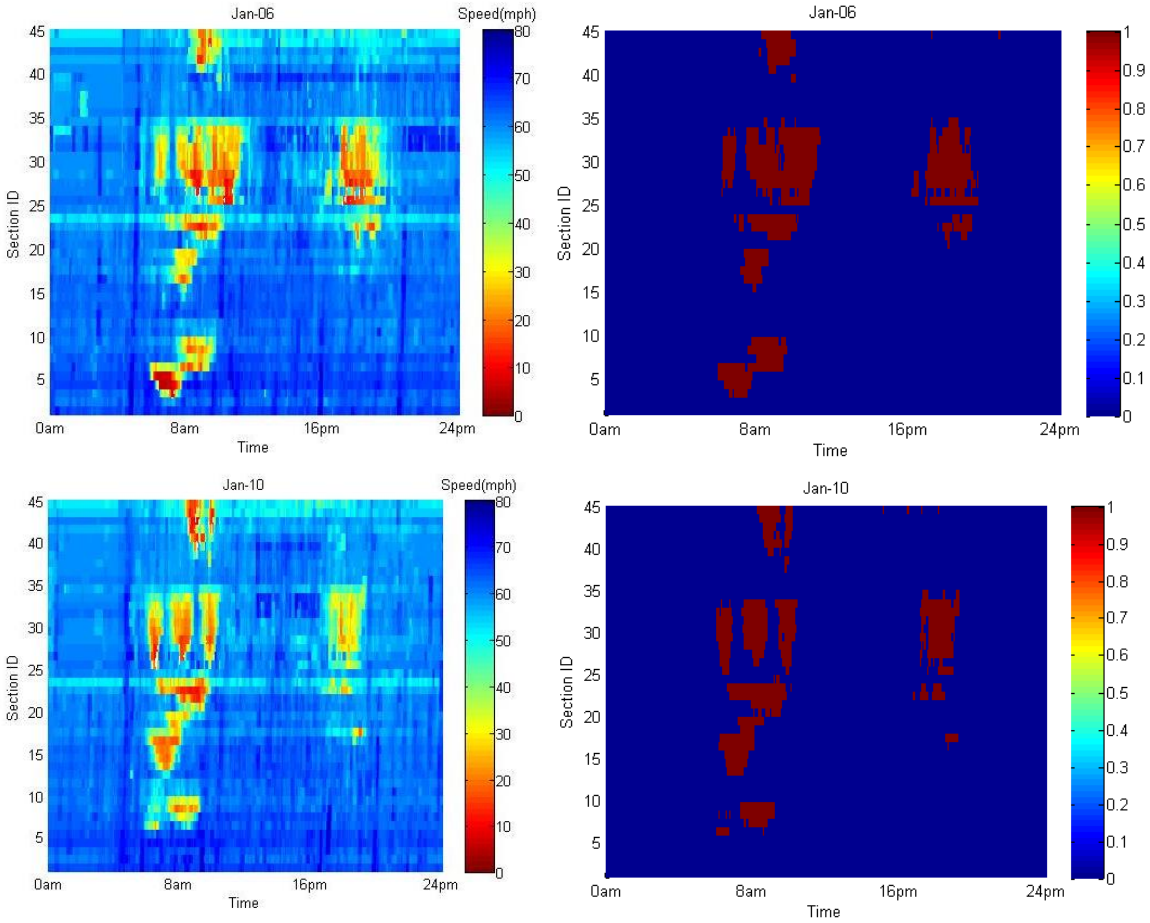


(a)





(b)



(c)

Figure 28: Speed (Left) and Binary Matrix After Applying Algorithm (Right); (a) TX; (b) CA; (c) VA.

Chapter 6: Congestion Prediction using Adaptive Boosting Machine Learning Classifiers

This chapter is based on
Mohammed Elhenawy and Hesham Rakha, "Congestion Prediction Using Adaptive Boosting Machine Learning Classifiers," in Transportation Research Board 93rd Annual Meeting, 2014.

Abstract

Congestion is a challenge that commuters have to deal with on a daily basis. Consequently, predicting the future status of a roadway is valuable for travelers in making better travel decisions. The deployment of stationary sensors and the proliferation of mobile vehicle probes provide researchers with a wealth of historical and real-time data that can be used for the automatic prediction of congestion along freeway segments. In this paper we introduce a new algorithm for the automatic prediction of congestion using Adaptive Boosting machine learning classifiers. The proposed algorithm creates the learning dataset by identifying congested sections using a skewed distribution mixture model of speed data to create a binary congestion matrix. The elements of this binary matrix are then used as responses in the training of the classifiers. The predictors for the classifier during the training phase are windows (slots) of the historical spatiotemporal speed matrix. In the real-time running phase, the classifiers use the most recent spatiotemporal speed matrix window to predict the short- and medium-term (up to 100 minutes into the future) status of the roadway. Experimental results using archived data from a 22-mile section of the northbound Interstate 5 (I-5) corridor in the Portland, Oregon, metropolitan region demonstrates promising high true positive and low false positive rates. Specifically, using a relatively large number of weak learners (between 20 and 30 learners) the achieved true positive prediction rate is slightly greater than 0.99 and the false positive rate is less than 0.0001.

Introduction

Over the past decade many fields and industries are benefiting from the advances in information technology (IT).Nowadays, IT is used in education, health care, government, and is transforming transportation profession. Using IT, the future of transportation does not rely solely on building new infrastructure; but more importantly relies on making the current infrastructure more efficient by adding new IT techniques to the operations and management of transportation systems. The current electronic and communication technologies transform the transportation system elements such as vehicles, traffic controllers to intelligent elements. By integrating microchips and sensors into transportation elements, these elements can communicate with each other and exchange information. Several countries deployed Intelligent Transportation Systems (ITSs) that depend on IT. ITSs improve the transportation system performance by reducing congestion and increasing road safety. ITS are used to improve safety, operational performance by reducing congestion, mobility and convenience, environmental benefits, and productivity[1].Reducing and avoiding traffic congestion is one of the key benefits that ITSs are capable of achieving. Reducing and eliminating congestion is still a major challenge to ITS. Congestion prediction using real-time information is critical to eliminating or reducing congestion by allowing travelers to avoid congested locations by either altering the mode of travel, departure time, or route of travel.

Traffic congestion has increased globally as a result of increased motorization, population growth, and changes in population density. Congestion may cause various social, environmental,

and economic problems. According to the Federal Highway Administration's (FHWA) publication FHWA-HOP-11-034, 40 percent of all congestion nationwide can be attributed to recurring congestion[2]. Recurring congestion is classified into "mega" where the traffic demand overwhelms entire regions or large facilities (e.g., interchanges or corridors). Some of it periodically overwhelms "subordinate" – locations on the highway system by temporarily being loaded by high traffic demand levels that exceed the physical capacity of the roadway. The physical capacities of the subordinate locations are sufficient during the off-peak hours. The latter congestion type is the recurring "localized" bottlenecks from which commuters suffer every day. Currently, a large number of research efforts are being conducted to predict the cause, location, time-of-day, and approximate duration of localized bottlenecks. On the other hand, congestion caused by random events, such as incidents, is nonrecurring and is hard to predict.

Traffic congestion reduces the utilization of the transportation infrastructure and increases traveler travel times, air pollution, and fuel consumption levels. In the prevalent literature and practice the terms "congestion" and "bottlenecks" are often used interchangeably. A definition of a bottleneck is that, it is subordinate locations along highways that cause the congestion and need to be fixed, and not necessarily the knee-jerk expectation to rebuild the entire facility. Consequently bottlenecks are subsets of congestion and identifying and predicting congestion is required in order to identify the location of bottlenecks. Once congestion is identified bottlenecks can be identified as the downstream front of any spatiotemporal congestion. One of the transportation engineer tasks is identifying bottlenecks and applying fixes to inefficient subordinate locations on a facility. These solutions may take time to be deployed and remove the bottleneck and the subsequent congestion. A more realistic solution is predicting if there was a chance of recurring congestion. Such prediction algorithm allows for improvements in traffic control strategies such as ramp metering that prevent or delay the activation of the bottleneck and thus reduce the congestion resulting from the bottleneck. It will also provide support for lane closure decisions, as well as predictions useful for traveler information.

The availability of fast powerful computing power assists data mining and Machine Learning researchers to develop algorithms that achieve remarkable success in many knowledge engineering areas including classification and prediction. There are several applications for Machine Learning in various fields. Machine Learning algorithms establish relationships between multiple features that are very difficult for humans especially in high dimensional space. Establishing such relationships between features helps in identifying solutions to certain problems. Machine Learning can often be successfully applied to several problems including transportation problems. Every instance in any dataset used by Machine Learning algorithms has the same number of features. Machine Learning algorithms can be trained using continuous, categorical or binary features. Learning is called supervised if training instances are labeled, in contrast to unsupervised learning, where instances are unlabeled. Supervised machine learning algorithms have received great attention from researchers. Supervised classification algorithms partition predictor (feature) space into non overlapping partitions each with its label during the training phase. The learning algorithms do these partitioning using externally supplied labeled instances. The developed algorithms then make predictions about future instances. In other words, the goal of supervised learning is to train a classifier the distribution of class labels using labeled predictor features. The built classifier can be then be used to predict the class labels for incoming new unseen instances.

Advances in computers and Machine Learning algorithms over the past two decades have resulted in the development of various traffic prediction algorithms. Accurate prediction of future traffic congestion could decrease travel times by improving vehicle navigation systems and enhancing traffic flows. Recently, a methodology was developed to predict future automated traffic recorder (ATR) readings using the real-time and historical data from local ATRs[3]. This methodology is based on the well-known Machine Learning algorithm Random Forests[4]. Ant colony is used in another algorithm to predict congestion where cars are modeled as ants. Cars equipped with sensors would deposit multiple digital pheromone on the basis of sensed traffic information[5]. Other cars that follow their route would avoid traffic congestion by checking the intensity of pheromone. Recurrent Jordan networks, which are popular in the modeling of time series, were used for prediction of road traffic[6]. Specifically, the algorithm uses traffic volume measurements to forecast future volumes. Simulation results showed a good generalization ability of this algorithm.

Considering the research need, an automatic congestion prediction algorithm is proposed in this paper to accurately predict congestion for use in any ITS system. In the proposed approach a bank of Adaptive Boosting (AdaBoost) classifiers are trained to predict the future status of road segments (either free-flow or congested). The proposed algorithm uses only the speed spatiotemporal matrix. The field data from the I-5 corridor in the Portland, Oregon area are used to test the performance of the proposed algorithm in the predication of congestion.

The remainder of this paper is organized as follows. First, a brief background of the AdaBoost is given and the Center for Sustainable Mobility's (CSM's) congestion identification algorithm based on skewed component distributions is described briefly. After that, the proposed algorithm is introduced .Subsequently field testing of the algorithm is presented. Finally, conclusions and recommendations for future work are presented.

Methods

The AdaBoost is a Machine Learning algorithm[7]is based on the idea of incremental contribution. AdaBoost is introduced as an answer to the question of whether a group of "weak" learner algorithms that each has low accuracy can be grouped together and boosted into an arbitrarily accurate "strong" learning algorithm. Before introducing the idea of AdaBoost, it should be noted that the traditional way of thinking in Machine Learning entails selecting the most possible class of discriminating features. It is needed to be as class discriminatory as possible. Then, using these features we try to find the most discriminating learning algorithm. For example, if we want to predict the class label for an unseen data instance, first, we collect labeled data to use in training the algorithm. Afterwards, we go through feature selection. If the number of features is larger than the number of instances, then we suffer from the curse of dimensionality[8]. We can use the Principal Components Analysis (PCA)[9]or independent component analysis (ICA)[10] to reduce the space dimensions and overcome the curse of dimensionality problem. The last step after defining the feature space is choosing an algorithm such as K-Nearest Neighbor (K-NN)[11], Support Vector Machine(SVM)[12], or parametric models that give the highest prediction accuracy.

AdaBoost does not use one classifier; instead it uses a set of weak classifiers each is trained using the same training dataset but with a different weight distribution. Each weak learner focuses on the instances that are misclassified by the previous learner. The output of AdaBoost is the weighted average of all weak learners' output. AdaBoost is proved to have smaller misclassification error as you add weak learners; also it has a bound on the

generalization error. To describe the AdaBoost algorithm, let us assume the training set consists of n instances $T = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ where, $x_i$is the vector of predictors which can be represented by a point in the multidimensional feature (predictor) space. The y_i is the corresponding label. Because our algorithm predicts whether a segment is congested or free-flow, we will focus on binary classification problems. In binary classification problems $y_i = \{-1, +1\}$. The pseudo-code of the classic AdaBoost is described in Table 11.

Table 11: Proposed AdaBoost Algorithm Pseudo Code

Set a probability distribution$P_t(x_i)$ over all the training samples.Initially, $P_t(x_i)$ is set to be uniform and then is modified iteratively with each selection of a weak classifier.
for iteration *t* do

1. Train the weak learner $L_t$with weighted sample.
2. Test the weak learner $L_t$on all data and obtain the predicted label $L_t(x_i)$for each x_i.
3. Compare the predicted labels $L_t(x_i)$ with y_i for i=1,…,n and calculate the classification error ϵ_t
4. Calculate the trustiness level $\propto_t$ of the L_t using the following equation
   $\propto_t = \frac{1}{2} \ln \frac{1 - \epsilon_t}{\epsilon_t}$
5. Update F_t such that misclassified instances weights are increased using,
   $P_{t+1}(x_i) = \frac{P_t(x_i) e^{-\propto_t y_i L_t(x_i)}}{Z_t}$, where Z_t normalizes the weight such that $\sum_{i=1}^{n} P_{t+1}(x_i) = 1$

end for

After training T weak learners the model is ready to predict the label for test instances (unseen) x_{test}. The label of the test instance is defined using Equation(1).

$$\text{sign}\left(\sum_{t=1}^{T} \propto_t L_t(x_{test})\right) \quad (1)$$

Where the label is 1 if the output of Equation (1) is positive and -1 if the output is negative.

Congestion Identification Using a Mixture Skewed Distribution Model

The speeds across the road segments have an underlying fundamental diagram trend with randomness associated with the data. The variability of speeds is substantial in congestion. Due to this random nature of speed, stochastic models are the best choice for speed modeling. Stochastic models have been proven to be really good tools in travel time reliability modeling[13, 14]. We can model the traffic stream speed using only one standard distribution which is good if we only have one traffic state. We can overcome the problem of multistate traffic by using a mixture of distributions where each component corresponds to a specific traffic state. In our case we assume a two-phase traffic theory where the traffic is either free-flow or congested. The research team at the CSM at the Virginia Tech Transportation institute (VTTI) developed a simple algorithm to identify congested segments using a spatiotemporal speed matrix[15]. The model does not required any parameter setting in advance which makes it simpler and more applied than the state-of-the-art Chen algorithm [16]. The proposed algorithm fits two lognormal distributions to the training dataset, as demonstrated in Equation (2).

$$f(u|\lambda, \mu_1, \mu_2, \sigma_1, \sigma_2) = \lambda \frac{1}{\sqrt{2\pi} u \sigma_1} e^{\frac{(\ln u - \mu_1)^2}{2\sigma_1^2}} + (1 - \lambda) \frac{1}{\sqrt{2\pi} u \sigma_2} e^{\frac{(\ln u - \mu_2)^2}{2\sigma_2^2}}, \quad (2)$$

Here (μ_1, σ_1) and (μ_2, σ_2) are the mean and standard deviation of the first and second component distributions and λ is the mixture parameter.

A Maximum Likelihood algorithm is used to calibrate the five parameters to the data. The next step after fitting the model is calculating the 0.01 quintile of the free-flow state speed distribution. Then we use the 0.01 quintile as a threshold to classify the state of each segment along the road. All segments with speeds greater than the threshold are classified as free-flow segments and other segments are classified as congested segments. The output of the above algorithm is a spatiotemporal binary matrix that is of the same dimensions as the spatiotemporal speed matrix. The one of the binary matrix identifies a segment as congested and a zero represents free-flow. Figure 29shows the spatiotemporal speed matrices for several days and the corresponding spatiotemporal binary congestion matrix after applying the mixture congestion identification model to the data.

The Proposed Algorithm

The proposed algorithm uses the most recent spatiotemporal speeds to predict the future status of the road segments. This can be done by building (training) an AdaBoost model for each segment. This means that if the road is divided into (K) segments, the speed data is (A) minutes aggregated, and we want to predict the status of all segment during for next (F) minutes then the bank of AdaBoost consists of $\frac{K*F}{A}$AdaBoost models. Each AdaBoost model, which consists of many weak learners, is built independently as will be described in the following sections.

1. Training Phase

In order to train the AdaBoost model we need first to define the predictors and the response. The predictors come from the spatiotemporal speed matrix $V_{s,t}$where s is the segment number and t is the time-of-day. The responses come from the binary matrix $\beta_{s,t} = \Psi(V_{s,t})$where Ψ is the congestion identification mapping function derived from the spatiotemporal speed matrix $V_{s,t}$ to the spatiotemporal binary matrix $\beta_{s,t}$. In other terms, $\beta_{s,t}$ is the output generated by the congestion identification algorithm based on the mixture skewed component distribution model that was described earlier.

Now assume that we are at time t_0 and we want train the AdaBoost model using only one day of the historical dataset. The trained AdaBoost model will be used to predict the traffic status of the segment $SEG^{k}_{t_0+\Delta t}$, where k is the segment number. In other words, we want to train the AdaBoost model to predict the status of segment number k, Δt minutes into the future considering we are at time t0. The predictors should include the speed distribution along the road segments at the times [t-m+1, t-m+2, …, t-1, t0]where m is the parameter that indicates how far back we need to look in order to predict the future. The training predictor vector is the concatenation of all the above speeds as shown in Equation(3).

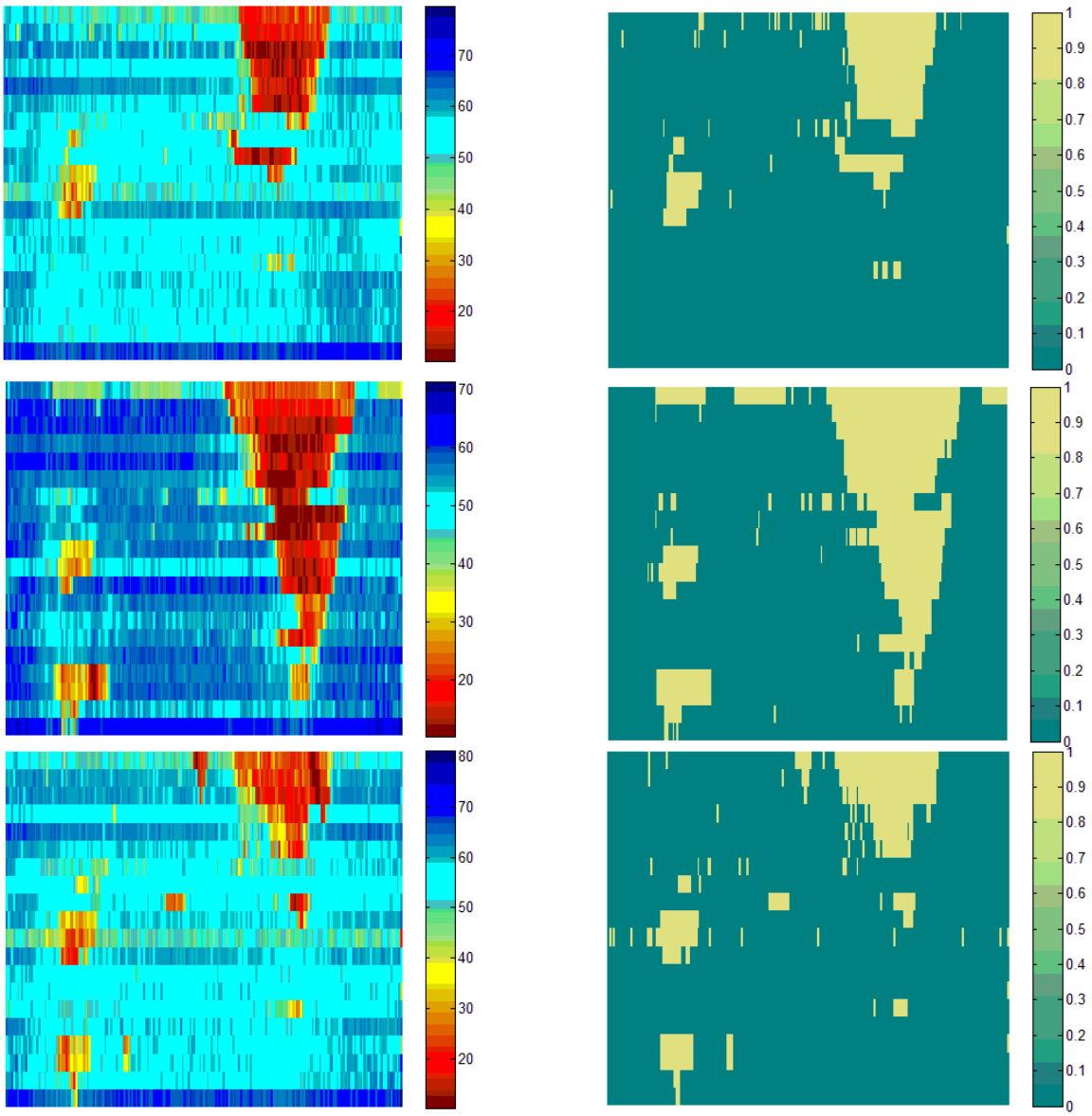


Figure 29: Spatiotemporal Speed and Corresponding Congestion Matrices for 3 Days

$$X_{t_0} = [\grave{V}_{.,t}\big|_{t=t_{-m+1}} \grave{V}_{.,t}\big|_{t=t_{-m+2}} \dots \grave{V}_{.,t}\big|_{t=t_0}], \quad (3)$$

Here X_{t_0} is a row vector of the predictors at time t_0, and $\grave{V}_{.,t}\big|_{t=t_{-m+1}}$ is the transpose of $V_{.,t}\big|_{t=t_{-m+1}}$. The corresponding response $Y_{t_0}\big|_{k,t_0+\Delta t}$ for X_{t_0} is $\beta_{s,t}\big|_{s=k,t=t_0+\Delta t}$ which is either one or zero depending on the status of the segment.

The training dataset is the collection of all $(X_{t_0}, Y_{t_0}\big|_{k,t_0+\Delta t})$at different values of t_0. This training dataset can be used to build the AdaBoost model which is used to predict the status of the segment $SEG^{k}_{t_0+\Delta t}$ when a new unseen predictor vector arrives. By varying the values of k and Δt in $\beta_{s,t}\big|_{s=k,t=t_0+\Delta t}$ we obtain several training datasets all having the same predictor matrix but with different responses. Each of this dataset is used to train one AdaBoost in the AdaBoost Bank. Each trained AdaBoost using the dataset $(X_{t_0}, Y_{t_0}\big|_{k,t_0+\Delta t})$can be used only, to predict the status of a segment number k at time $t_0 + \Delta t$.

2. Reducing the Prediction Time

As we mentioned before the AdaBoost bank consists of a larger number of AdaBoost models (classifiers) each classifier is responsible for predicting the traffic status for a specific segment at a specific Δtin the future. We can imagine that the bank is used to create a binary image for the next (h) hours in the future and each classifier is responsible for one pixel of this image. So the question is do we need to predict the status of each pixel? Can we use the priori information and decide whether this particular segment at this particular time will not be congested. To answer these questions, we can use the available historical dataset during the training phase to build a spatiotemporal probability of congestion matrix for the road segments. This matrix is built once during the training phase and then is used during the testing phase. We assume that the probability distribution of congestion at each segment follows a Bernoulli distribution with mean π, as demonstrated in Equation(4).

$$p(Z/z) = (1-\pi)^{1-z}(\pi)^{z} \quad (4)$$

$p(Z/z)$ is a discrete probability, which takes value 1 with success probability π and value 0 with failure probability$(1-\pi)$. In other terms, $p(Z/z)$ has a probability mass function which has only two values (i.e. 0 or1) .Here z equals one if the segment is congested and zero otherwise. The variable π is not constant, but instead varies as a function of the segment and time interval$\pi_{s,t} = \varphi(\text{segment}, t)$. It may change from a segment to another segment and from a time period to another time period. To estimate $\pi_{s,t}$we use the skewed component congestion identification algorithm to process each day's spatiotemporal speed matrix$V_{s,t}$. The output of the congestion identification algorithm is a binary spatiotemporal congestion matrix $\beta_{s,t}$where 1 indicates a congested segment and 0 a free-flow segment. After processing all historical days, $\pi_{s,t}$ is simply the frequency of ones at each spatiotemporal element divided by the number of days, as shown in Equation(5).

$$\pi_{s,t} = \sum_{i=1}^{N} \frac{\Psi(V_{s,t}^{(i)})}{N}, \quad (5)$$

Where N is the number of days in the historical dataset. The output $\pi_{s,t}$ is shown in Figure 30.

Figure 30 shows congested segments are localized in both time and space. For example, in the morning hours we can use the AdaBoost classifiers that correspond to segments from 1 to 4 and from 2 to 8 because the other segments have probabilities in the congestion map less than a minimum threshold ($th_{\pi_{s,t}}$). Those that are below the threshold are considered a priory with AdaBoost labels of zeroes (freeflow). The threshold $th_{\pi_{s,t}}$ depends on the used dataset and the tradeoff between false negative rate and computation speed. As $th_{\pi_{s,t}}$ increases as the false negative rate increases. Using a probability of congestion map reduces the computation time during the testing phase when the road has a large number of segments and there is a need to predict several hours into the future.

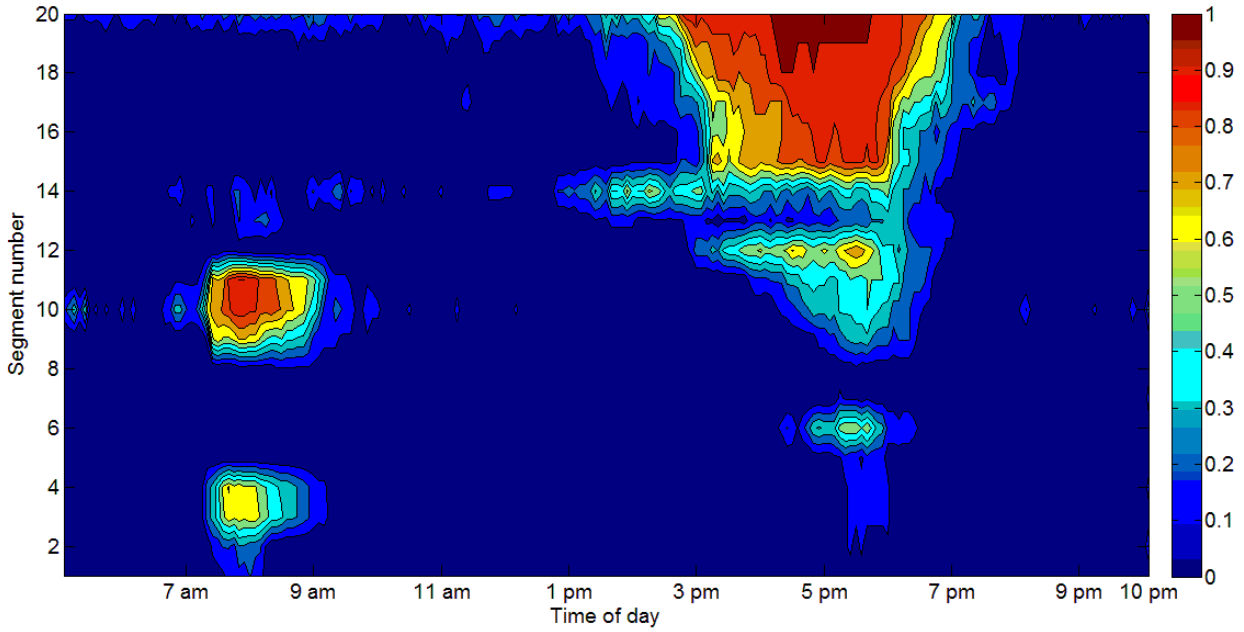


Figure 30: Spatiotemporal Congestion Probability Matrix

3. Testing Phase

This phase is simpler than the training phase and is much faster regarding the computational cost. In this phase we need to predict the status of each segment along the road for up to (h) hours into the future. Given the current time is t_0, the first step is constructing the predictor vector. It is constructed by concatenating the speed distribution along the road segments at the times $\{t_{-m+1}, t_{-m+2}, \ldots, t_{-1}, t_0\}$ of the current day as shown in Equation(6).

$$X_{t_0}^{(test)} = [\hat{V}_{.,t}|_{t=t_{-m+1}} \hat{V}_{.,t}|_{t=t_{-m+2}} \ldots \hat{V}_{.,t}|_{t=t_0}], \quad (6)$$

Then analyzing the spatiotemporal probability of congestion matrix starting at $t_0 + 1$ and ending at $t_0 + h$ to decide which AdaBoost classifiers in the bank should be used for prediction purposes. This is done by comparing each single element in column number $t_0 + 1$ to column number $t_0 + h$ against $th_{\pi_{s,t}}$. $th_{\pi_{s,t}}$ is set up zero in our experimental work. For example, if $\pi_{1,t_0+1} > th_{\pi_{s,t}}$ then the AdaBoost classifier which is responsible for predicting the status of segment number 1 at time $t_0 + 1$ will be used. We consider the outputs are zeroes (free-flow) for all other AdaBoost classifiers that are not used. The last step entails classifying the predictor vector using the chosen AdaBoost classifiers to complete the binary image for the next (h) hours in the future.

Case Study

In order to test the performance of the proposed algorithm and compare its performance for different number of weak learners, we used a real dataset obtained from an earlier study for congestion identification [17]. The dataset consists of 24 days' worth of data. The data consists of high-data-quality, midweek non-holiday days between February and December 2008. The data were collected from archived data from the northbound Interstate 5 (I-5) corridor in the Portland, Oregon, metropolitan region. This section of freeway is 22 miles (35-km) in length. Along the chosen section there are 22 detectors, two of them are ignored because of their poor

data quality. Each day included readings from 20 detectors, at the lowest available resolution of 20 s between 5:00 a.m. and 10:00 p.m.

1. Model Parameter Sensitivity Analysis

The proposed algorithm has three important parameters that need setting:

1- The number of weak learners of the AdaBoost model. It is an important parameter in AdaBoost and it is application dependent.
2- How far back in time should the model consider from the current time t_0 through the speed matrix$V_{s,t}$. This parameter affects the number of predictors used to predict the future status of a roadway segment.
3- How far ahead in time should the model. The number of AdaBoost classifiers in the Bank is affected by this parameter.

For the sake of simplicity we set the look-back and look-ahead parameters to be equal to m (i.e. parameters 2 and 3 are set equal).We used the Classification trees and regression trees (CART) algorithm as a weak learner. The well-known machine learning technique CART is one of the common decision trees used in AdaBoost. CART is a greedy and recursive top-down binary partitioning that divides the feature space into sets of disjoint regions. In the reminder of the paper when we use the word tree we mean CART.

The proposed algorithm was tested by varying the number of trees and them parameter. At each run the performance of the proposed algorithm was evaluated. To evaluate the performance of the proposed algorithm we use the true positive rate (TPR) and the false positive rate (FPR) [18]. The true positive rate (TPR) and false positive rate (FPR) are computed using Equations(7) and(8).

$$\text{TPR} = \frac{\text{True detected positive}}{\text{Actual positive}} \tag{7}$$

$$\text{FPR} = \frac{\text{False detected positive}}{\text{Actual negative}}, \tag{8}$$

Where positive in our context refers to a congested segment and negative refers to a free-flow segment. The TPR and FPR are calculated as an average of TPR and FPR of each day. The daily TPR and FPR is computed using the leave one (day) out cross validation (LOO) approach [19]. In LOO, we build the classifier using labeled data from 23 days and leave one day out for testing. The left day is used as an unseen test day and both the TPR and FPR are calculated for that day. The entire process is repeated where each day is used once as a test day and the average TPR and FPR for all 24 days is computed. The temporal resolution of the data used in the analysis was set at five minutes. We ran the proposed algorithm several times and varied from 2 to 20 (i.e. from 10 to 100 minutes) and varied the number of trees from 5 to 30. The average TPR and FPR are shown in Figure 31andFigure 32.

2. Impact of Model Parameters on the True Positive Rate

The first test entailed quantifying the effect of the number of weak learners on the algorithm TPR. As shown in Figure 31as the number of weak learners (trees) increases the algorithm TPR increases. This is true for a small number of weak learners, but as shown in the figure the improvement in the TPR becomes marginal as the number of weak learners exceeds 20. We can see that the TRP rates are almost identical (no significant improvement) for AdaBoost models with 20 weak learners and above. Consequently, the use of 20 weak learners in each AdaBoost appears to be sufficient for the proposed application. Any other addition of weak learners will

not improve the performance of the algorithm significantly but will add a computational burden especially during the training phase and some delay in the running phase. The figure also demonstrates that the curves are all almost parallel, with no intersection, implying that there is interaction between the two factors, the number of weak learners and look back and look ahead parameter m. Each TPR curve indicates that as the m parameter increases the TPR increases, but this is at the cost of adding more predictors and making the AdaBoost model more complicated and time consuming. The positive side of increasing m is that our proposed algorithm achieves very high TPR when predicting a long time into the future. The figure shows that using 5-minute aggregated speed data, the algorithm is able to predict up to 100 minutes into future with almost a 0.99 TPR, which is very promising.

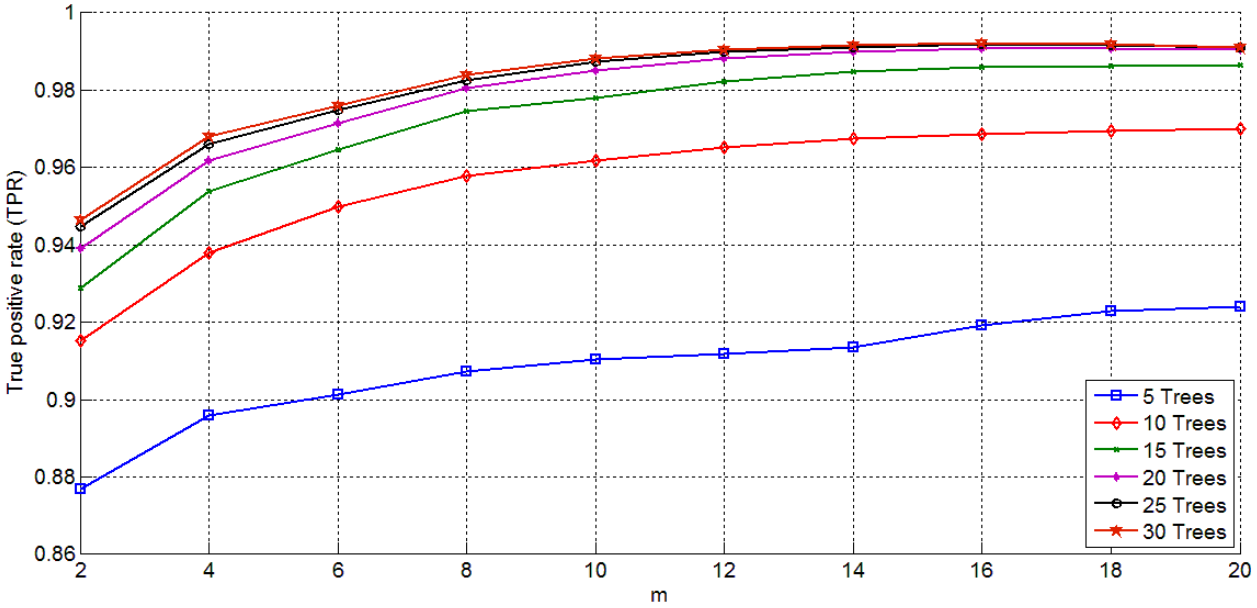


Figure 31: Variation in TPR as a Function of Number of Weak Learners and m Parameter

3. Impact of Model Parameters on the False Positive Rate

The results indicate that as the number of weak learners increases the algorithm FPR decreases, as illustrated in Figure 32. The figure demonstrates that the addition of weak learners results in a continuous decrease in the FPR, but at the same time increases the computational load needed to train and run the algorithm in real-time. The use of 20 weak learners appears to be a good compromise between model efficacy and computational load. At 20 weak learners, the FPR is always less than 0.001.

Maintaining a fixed number of weak learners the figure clearly demonstrates that as the look back and ahead time parameter m increases, the FPR increases slightly at low number of weak learners. It is clear that, FPR decreases as we increase the m parameter for AdaBoost models when the number of weak learners is greater than 20.

In conclusion, the proposed algorithm appears to be very promising for predicting congestion producing high TPRs and very low FPRs.

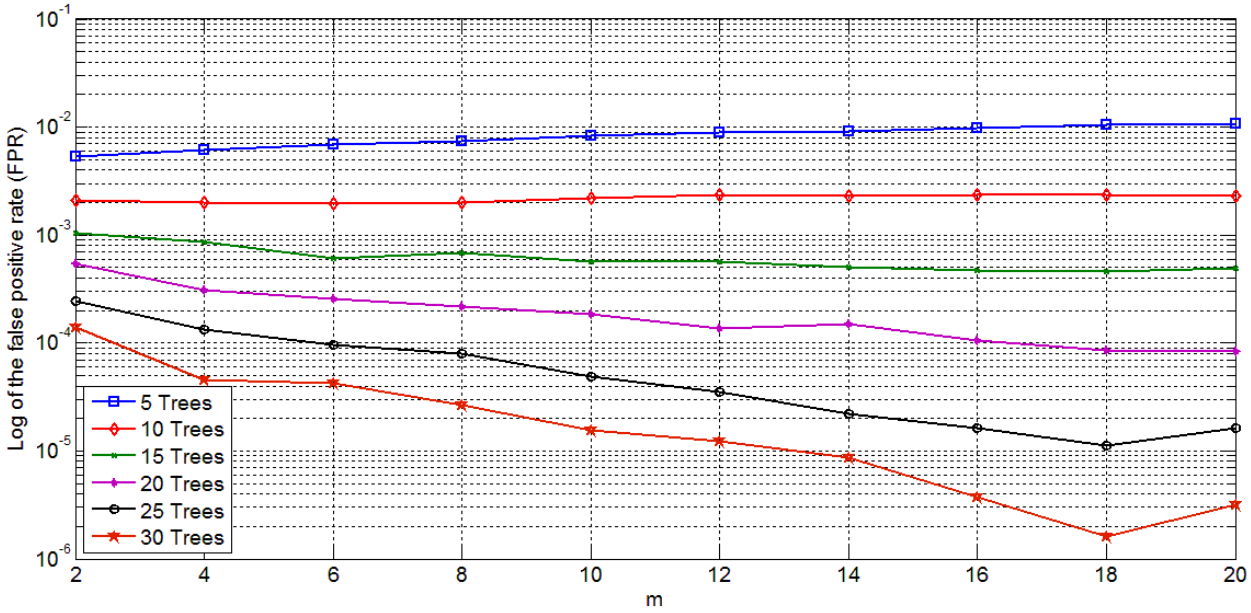


Figure 32: Variation in FPR as a Function of Number of Weak Learners and m Parameter

Conclusions and Future Work

The research presented in this paper develops an Adaptive Boosting (AdaBoost) machine learning algorithm that predicts the status of a roadway as either congested or free-flow. The proposed algorithm uses a previously developed congestion identification algorithm based on a skewed distribution mixture model to develop a congestion matrix that is used to train the proposed AdaBoost algorithm. Subsequently, the congestion and its corresponding spatiotemporal speed matrices are used to build a bank of AdaBoost classifiers. Given the speeds along the various road segments in the time interval $[t_0, t_0 - \Delta t - 1]$this bank is utilized to construct a binary image of the future status of all roadway segments – congested or free-flow (1 or 0). The proposed algorithm uses the priori information represented in the spatiotemporal probability of congestion matrix to decide at a particular time t_0, which AdaBoost classifiers in the bank of classifiers is used. The other unselected classifiers are assumed to give free-flow output based on the values of spatiotemporal probability of the congestion matrix within the interval $[t_0 + 1, t_0 + \Delta t]$. The proposed algorithm provides a very promising TPR and FPR when applied to 24 days' worth of archived data from the northbound Interstate 5 (I-5) corridor in Portland, OR. Specifically, the proposed algorithm produces a TPR slightly above0.99 and at the same time an FPR less than 0.001when predicting the traffic state 100 minutes into the future along the 22-mile test section using20 weak learners for each AdaBoost classifier.

As with the case with any research effort, further improvements and testing of the algorithm require further investigation. These enhancements include: (1) testing the algorithm on different freeway corridors; (2) developing an on-line calibration procedure to train the AdaBoost classifier to adapt to changes in the weather and roadway surface conditions; and (3) extending the algorithm to major arterial roadways.

Acknowledgements

This research effort was funded by the Mid-Atlantic University Transportation Center (MAUTC). The authors thank Mr. Wieczorek and Dr. Robert Bertini for providing and sharing the field data and that was used in this study.

References

[1] S. Ezell, "Explaining International IT Application Leaderhip:Intelligent Transportation Systems," January 2010.

[2] U. S. DOT. (March 2011). An Agency Guide on How to Establish Localized Congestion Mitigation Programs Congestion Mitigation Programs.

[3] B. Hamner, "Predicting Future Traffic Congestion from Automated Traffic Recorder Readings with an Ensemble of Random Forests," in Data Mining Workshops (ICDMW), 2010 IEEE International Conference on, 2010, pp. 1360-1362.

[4] L. Breiman, "Random forests," Machine learning, vol. 45, pp. 5-32, 2001.

[5] Y. Ando, O. Masutani, Y. Fukazawa, H. Iwasaki, and S. Honiden, "Performance of Pheromone Model for Predicting Traffic Congestion," in International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-06), 2006, pp. 73-80.

[6] R. Yasdi, "Prediction of Road Traffic using a Neural Network Approach," Neural Computing & Applications, vol. 8, pp. 135-142, 1999/05/01 1999.

[7] Y. Freund and R. Schapire, "A short introduction to boosting," Japonese Society for Artificial Intelligence, vol. 14, pp. 771-780, // 1999.

[8] R. Clarke, H. W. Ressom, A. Wang, J. Xuan, M. C. Liu, E. A. Gehan, et al., "The properties of high-dimensional data spaces: implications for exploring gene and protein expression data," Nat Rev Cancer, vol. 8, pp. 37-49, 01//print 2008.

[9] L. Smith. (2002, A Tutorial on Principal Components Analysis.

[10] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," Neural Networks, vol. 13, pp. 411-430, 6// 2000.

[11] R. T. Trevor Hastie, Jerome Friedman The Elements of Statistical Learning Data Mining, Inference, and Prediction, 2009.

[12] C. W. Hsu, C. C. Chang, and C. J. Lin, A practical guide to support vector classification, 2003.

[13] F. Guo, H. Rakha, and S. Park, "Multistate Model for Travel Time Reliability," Transportation Research Record: Journal of the Transportation Research Board 2010.

[14] F. Guo, Q. Li, and H. Rakha, "Multistate Travel Time Reliability Models with Skewed Component Distributions," Transportation Research Record: Journal of the Transportation Research Board, vol. 2315, pp. 47-53, 12/01/ 2012.

[15] M. Elhenawy and H. Rakha, "Title," unpublished|.

[16] C. Chen, A. Skabardonis, and P. Varaiya, "Systematic identification of freeway bottlenecks," Freeway Operations and Traffic Signal Systems 2004, pp. 46-52, 2004.

[17] J. Wieczorek, R. J. Fernández-Moctezuma, and R. L. Bertini, "Techniques for Validating an Automatic Bottleneck Detection Tool Using Archived Freeway Sensor Data," Transportation Research Record: Journal of the Transportation Research Board, vol. 2160, pp. 87-95, 2010.

[18] V. Bewick, L. Cheek, and J. Ball, "Statistics review 13: receiver operating characteristic curves," Crit Care, vol. 8, pp. 508-12, Dec 2004.

[19] T. Evgeniou, M. Pontil, and A. Elisseeff, "Leave One Out Error, Stability, and Generalization of Voting Combinations of Classifiers," Machine Learning, vol. 55, pp. 71-97, 2004/04/01 2004.

Part II

Chapter 7: Dynamic Travel Time Prediction Using Data Clustering and Genetic Programming

This chapter is based on
Mohammed Elhenawy, Hao Chen, and Hesham A. Rakha, "Dynamic travel time prediction using data clustering and genetic programming," Transportation Research Part C: Emerging Technologies, vol. 42, pp. 82-98, 2014.

Abstract

The current state-of-practice for predicting travel times assumes that the speeds along the various roadway segments remain constant over the duration of the trip. This approach produces large prediction errors, especially when the segment speeds vary temporally. In this paper, we develop a data clustering and genetic programming approach for modeling and predicting the expected, lower, and upper bounds of dynamic travel times along freeways. The models obtained from the genetic programming approach are algebraic expressions that provide insights into the spatiotemporal interactions. The use of an algebraic equation also means that the approach is computationally efficient and suitable for real-time applications. Our algorithm is tested on a 37-mile freeway section with several bottlenecks. The prediction error is demonstrated to be significantly lower than that produced by the instantaneous algorithm and the historical average averaged over seven weekdays (p-value<0.0001). Specifically, the proposed algorithm achieves more than a 25 and 76 percent reduction in the prediction error over the instantaneous and historical average, respectively on congested days. When we use bagging in addition to the genetic programming, the results show that the mean width of the travel time interval is less than 5 minutes for the 60-80 minute 37-mile trip.

Introduction

Congestion has proven to be a serious problem in most urban areas in the United States. In 2011, it caused urban Americans to spend 5.5 billion hours more in traveling and cost them an extra 2.9 billion gallons of fuel, for a congestion cost of $121billion. Congestion has also environmental side effects because of the CO2 produced during congested periods, which was estimated to be as much as 56 billion pounds in 2011 [1]. Adding capacity has been the traditional solution to the congestion problem, but this has become impractical given the financial, environmental, and social constraints. Consequently, highway agencies are seeking new solutions to overcome recurrent and non-recurrent congestion.

Thanks to advanced new technologies that enable continuous monitoring and dissemination of traffic information, it is possible to manage the transportation system more efficiently. The minimum that can be accomplished is to inform the potential users of a road what they can expect during their trip. Such information helps travelers compare alternative routes and make better routing and departure time decisions. This is the essence of Advanced Traveler Information Systems (ATISs), such as the 511 systems that have been implemented nationwide. In many states relevant traffic information is also posted on Variable Message Signs (VMSs) that are strategically positioned along highways. Because the infrastructure is already available, we can assist travelers in making better decisions by providing accurate travel time predictions. In case of congestion, many road users may change their routes of travel based on displayed travel time information.

Recently, various traffic-sensing technologies, such as point-to-point travel time measurement systems (e.g., license plate recognition systems, automatic vehicle identification systems, mobile devices, Bluetooth tracking systems, and probe vehicles, etc.) and station-based traffic-state-measuring devices (e.g., loop detectors, video cameras, remote traffic microwave sensors, etc.), have been used to collect traffic data. The data collected using these technologies are used in several applications, including computing travel times. Private companies such as INRIX integrate different sources of measured data to provide section-based traffic state data (speed, average travel time), which are used in our study to develop algorithms for predicting travel times. The benefit of using section-based traffic state data is that travel times can be easily calculated. More importantly, the section-based data provide the flexibility for scalable applications on traffic networks.

Travel time prediction algorithms that use section-based traffic state data can be categorized into two broad categories depending on the trip experience: dynamic and instantaneous travel time[2, 3]. Dynamic travel time reflects the actual, realized travel time that a vehicle experiences during a trip. Dynamic travel time algorithms account for speed changes over both space and time. Consequently, some algorithms predict future speed patterns and use them to predict travel times. Instantaneous travel time usually computes travel time using the current speed along the entire roadway; in other words, the speed distribution is assumed to remain constant for the duration of the trip. As long as the change of speed with time is not significant, both approaches provide comparable travel time estimates. However, instantaneous approaches may deviate substantially from the actual, experienced travel time under transient states during which congestion is forming or dissipating during a trip [4].

Some attempts have been conducted using macroscopic traffic modeling to predict short-term traffic states; however, this approach is computationally expensive and the accuracy degrades rapidly with the increase in the prediction temporal horizon[4, 5]. For long trips, traffic states may change significantly, and the traffic state in the near future usually cannot provide enough information to cover the entire trip. For example, in the case of a 100-mile trip, if the driver departs at the timet_d, and the trip would take one hour or more depending on traffic conditions, then the traffic state for the following one hour or more would need to be predicted in order to compute dynamic travel times.

An alternative approach to solving this problem is to assume that these states are hidden variables and are function of the current and previous traffic states. This function can be derived from historical data. The historical dataset provides a pool of past experienced traffic conditions and shows how traffic status changes over time and space. The key issue is how to develop this function (model) and its parameters and then use it to predict travel times. The purpose of this study is to develop a simple and fast algorithm to predict dynamic travel times. The proposed method searches the model space guided by the historical dataset to construct a model that best describes the relationship between traffic states along time. A freeway stretch from Newport News to Virginia Beach is selected to test the proposed algorithm using five-minute aggregated traffic data for 2010 provided by INRIX. The travel time prediction results from April to August demonstrate that the proposed method produces higher prediction accuracies compared to the state-of-practice instantaneous algorithm.

The remainder of this paper is organized as follows. A literature review of previous travel time prediction methods is provided. Subsequently, the proposed genetic programming–based approach is presented. This is followed by a description of the test data used for the case study and the results of a comparison of the proposed approach to traditional instantaneous algorithms.

The last section provides the conclusions of the research and some recommendations for future research.

Literature Review

During the past decades, many studies have been conducted to predict travel times. Some of the reviews of different methods can be found in earlier publications [6-9]. According to the manner of modeling, these methods can be classified into time series models or data-driven methods. Time series models include the Kalman filter [10, 11] and Auto-Regressive Integrated Moving Average (ARIMA) models[10, 12, 13].Data-driven methods include neural networks [7, 9-11, 13-15], support vector regression (SVR) [16, 17], and k-nearest-neighbor (k-NN) [8, 18, 19] models. These techniques are implemented through direct and indirect procedures to predict travel times using different types of state variables. Travel time is directly used as the state variable in model-based or data-driven methods to predict travel times. Indirect procedures are performed using other variables (such as traffic speed, density, flow, occupancy, etc.) as the state variable to predict the traffic status from which future travel times can be calculated based on some transition function.

Time series models construct the time series relationship of travel time or traffic state, and then current and/or past traffic data are used in the constructed models to predict travel times in the next time interval [20]. Kalman filters were proposed to predict travel times using Global Positioning System (GPS) information and probe vehicle data [10, 21]. A Kalman filter (KF) is a popular method for data estimation and tracking, in which the time update and measurement update processes are included. A time series equation is used to predict state variables, and then state values are corrected according to the new measurement data. The main advantage of a KF is that the recursive framework ensures traffic data is efficiently updated using only data from previous states and not the entire history [4]. The state transient parameter in the time series equation is defined from average historical data to calculate future travel times.

A similar idea was used in the Bayesian dynamic linear model for real-time, short-term travel time prediction [11]. The system noise can be adjusted for unforeseen events (e.g., incidents, accidents, or bad weather) and integrated into the recursive Bayesian filter framework to quantify random variations on travel times. Experimental results based on loop detector data from a segment of I-66 demonstrated that this method produced higher prediction accuracy under both recurrent and non-recurrent traffic conditions. However, a problem existed with these methods in that the travel time in the previous time interval was needed to calculate the future travel time. For real-time applications, the travel time is usually greater than the time interval step size. Hence, the actual travel time from the previous time interval is not available to apply in the algorithms used to predict travel times for the next time interval.

A seasonal ARIMA model was proposed to quantify the seasonal recurrent pattern of traffic conditions (occupancy) [13, 15]. Moreover, an embedded adaptive Kalman filter was developed in order to update the occupancy estimate in real-time using new traffic volume measurements. Consequently, multi-step, look-ahead occupancy information was estimated to obtain a data matrix representing the spatiotemporal traffic condition for the future trip. Since travel time cannot be directly computed through traffic conditions (occupancy), future traffic speed can be calculated using occupancy data by assuming an average vehicle length and using a constant conversion factor, known as the g-factor in the literature. Consequently, dynamic freeway corridor travel times are predicted with the consideration of traffic state evolution along

the corridor. However, this approach may be difficult to implement since the described recurrent pattern of traffic conditions may not be found everywhere.

Data-driven methods usually predict travel times using a large amount of historical traffic data. Time series models are not specified in the data-driven methods, considering the complexity and randomness in the system. Neural networks can be trained using historical data to identify hidden dependencies that can be used for predicting future states. A state space neural network (SSNN) method was proposed to predict freeway travel times for missing data[7]. The missing data problem was tackled by simple imputation schemes, such as exponential forecasts and spatial interpolation. Travel time was the direct state variable used for the training process, and the experimental results demonstrated that the SSNN methods produced accurate travel time predictions on inductive loop detector data. Supported vector machine (SVM) is a successor to artificial neural networks (ANNs). SVM has greater generalization ability than the ANNs and a superior empirical risk minimization principle [17]. The application of SVM to time series forecasting is called SVR. The SVR predictor was demonstrated to perform well for travel time prediction. The point-to-point travel time is usually used as the input to ANNs and SVRs. However, both methods require long training processes and are nontransferable to other sites [8].

The k-NN method can be used to find several candidate sequences from historical data by matching current to short, past data sequences. Travel time and occupancy sequences were used to predict dynamic travel times using the k-NN method with data combined from vehicle detectors and automatic toll collection systems in an earlier study [8]. The occupancy was used since the travel time sequence was collected for the recent past time intervals. The results from the case study demonstrated an improvement in the prediction accuracy by combining two types of sequences in the matching process. Moreover, a k-NN method was proposed by selecting candidates through the Euclidean distance and data trend measures to predict freeway travel times under different weather conditions [19]. Unlike ANNs and SVRs, k-NN methods are easy to implement and transfer to different sites without data training.

Genetic programming (GP) has two major advantages. The first advantage is the ability of GP to find a model to solve a problem without any pre-specified structure of the model. Based on the training data, the GP selects the best model to capture the underlying behavior. The second advantage is that the GP solution is that it is interpretable, which means it defines a logical relationship between the explanatory variables and the response variable. GP has been used successfully for regression and clustering [22, 23]. GP is used in different applications, including curve fitting, data modeling, image and signal processing, financial trading, time series prediction, and economic modeling [24]. In the field of transportation, GP is used to build models for real-time crash prediction[25]. It is used to perform new and complex tasks needed for vehicle guidance and control[26]. GP is applied to evaluate the performance of the pavement, where it is used to develop models to predict pavement rutting [27].To the best of our knowledge, we are the first to propose GP for dynamic travel time prediction.

In summary, existing methods are either insufficient or have limitations for predicting dynamic travel times for departures at the current time or future times. The proposed approach used in this study is a data-driven method, yet outperforms the previous methods by fully utilizing the relationship between traffic states and travel times. Moreover, other than previous studies using travel time sequences as input, the proposed method uses spatiotemporal traffic data to build a model using genetic programming and then uses it predict future travel times.

Background

The problem can be stated as follows: given the current traffic states (at time of departure), we need to accurately predict the traveler's travel time. In our case, the speed measurements along the road segments are available. Our task is, given the speeds at the time of departure, to predict the expected travel time. The hypothesis we assume is that the speed along each segment changes gradually, displaying a relationship between the current and the future speeds. This relationship can be described using a mathematical function, which we will attempt to derive. Another important point is that we know the relationship between speed, distance, and time so that given the speed and distance measurements we can accurately calculate the travel time. The proposed approach does not entail estimating the speed to compute the travel time; instead, we develop a model that links both the speeds before and at the time of departure to the actual experienced travel time. In this model, the future speeds are hidden and the output of the model is the travel time. This problem can be viewed as developing a model that relates the travel time to particular speed inputs using a functional form similar to Equation (1).

$$\hat{y} = \mathcal{F}(x, \beta); \quad (1)$$

where the predicted travel time $\hat{y}$ is a function of the speed vector x and the coefficient β. One possible model could have linear terms, interaction terms, and polynomial terms as shown in Equation (2).

$$\hat{y} = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n + \beta_{n+1} x_i \ldots x_j + \cdots + \beta_h x_l^2; \quad (2)$$

When we look at our problem this way, we discover that it becomes equivalent to searching for the model within the program space.

The proposed algorithm in this paper is developed using GP[28]. GP is an effective paradigm that provides a way to search a solution space for the optimum computer program. In a GP paradigm, the models are represented in a tree structure, and populations of these trees are genetically bred using the principle of survival of the fittest to select and generate new generations from current generations.

There are three main reasons to use the GP to solve this regression problem. First, we cannot satisfy the required assumption for the regression such as the equal variance assumption. Second, conventional regression seeks to estimate the regression coefficient for a pre-specified model. Alternatively, by using GP, there is no pre-specified model structure; instead the GP determines the model structure and regression coefficients simultaneously. The third advantage is the interpretability of the model obtained from the GP. Unlike other machine learning techniques like neural networks, GP generates analytical models that can be interpreted. In the context of travel time prediction, GP models give information about the interaction between the different road segments and which segments are more important for travel time prediction. In the following sections, we will briefly describe the GP approach and how we use it in our problem domain.

When the idea of automatic programming was first considered in the late 1940s and early 1950s, scientists attempted to develop programs without specifying how to carry out the tasks. In GP, the individuals of a generation are a tree-like structure built from genes. There are two types of genes: functional genes and terminal genes. Functional genes are tree nodes with children arguments. These functions depend on the problem domain and have to be defined and described. The other gene types are terminal genes, which are nodes of the tree without branches. In general, GP searches the space of computer programs to find the best fit program by executing three simple steps: (1) generate an initial random population; (2) conduct a fitness test; and (3) execute genetic operations. These three steps are briefly discussed in this section.

1. Generate Initial Random Population

Based on the problem domain, we determine our function set $F = \{f_1, f_2, \dots, f_g\}$,then we use the set F and the terminal set $T = \{t_1, t_2, \dots, t_s\}$, which includes the input attributes, constants, and random numbers to create individuals of the first generation. If the node is a function, we randomly select one of the functions from the set F using Equation (3).

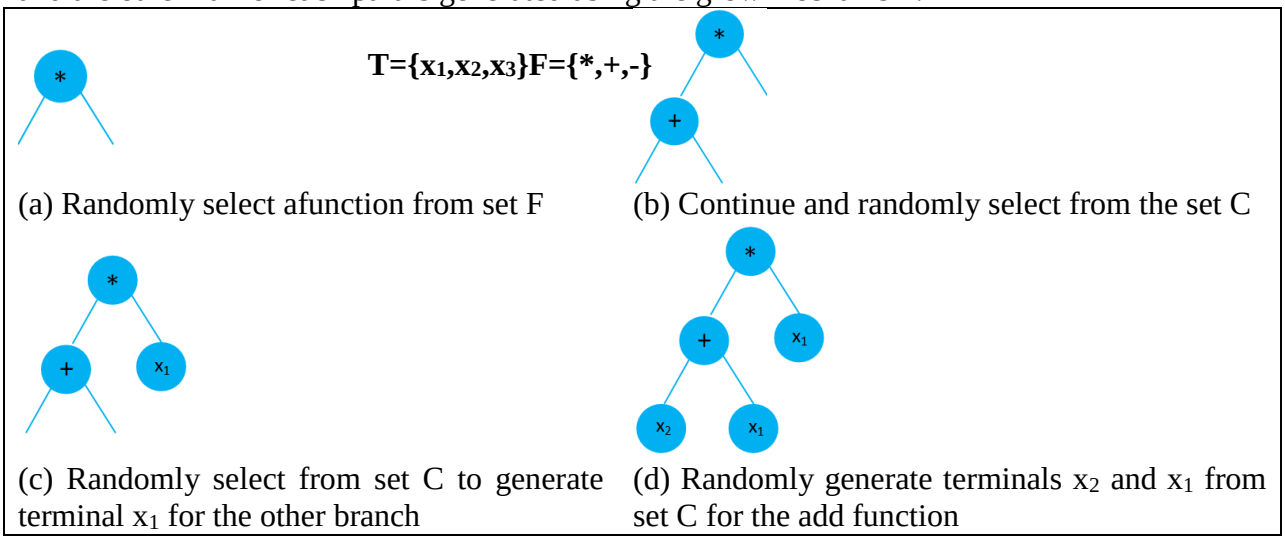
$$\text{index} = \text{INT}(g * \eta) + 1 \quad (3)$$

where g is the number of functions in the set F and ηis a random number drawn from a standard uniform distribution unif(0,1). If a tree node is labeled with a function f from the set F, then ϕ(f) branches are created to radiate out from that tree node, where ϕ(f)is the number of arguments taken by the function f. For each such radiating branch, a random number is drawn from a standard uniform probability distribution and an element from the combined set $C = F \cup T$ is chosen using Equation (4).

$$\text{index} = \text{INT}((g + s) * \zeta) + 1 \quad (4)$$

Where ζ is another random number.If the branch tree node is chosen from T, this node becomes terminal and does not have branches.

There are several other methods that can be used to randomly create the first generation of programs, as illustrated in Figure 33. Grow, full, and ramped-half-and-half are the most common techniques to create individuals in the first generation. In the grow method, starting from the root of the tree, the type of node is chosen randomly. If the new node is functional, the function is chosen randomly from a function set F and children nodes are created for this node. If the node is terminal, a random terminal is chosen from the terminal set T. This process continues until the depth of the tree reaches a maximum depth. In the full method, every node starting from the root node that has a depth less than a certain threshold is considered a functional node. If the node reaches the depth threshold, then the node type is selected randomly. In ramped-half-and-half, the population is divided into parts. Half of each part is generated using the full mechanism and the other half of each part is generated using the grow mechanism.



(a) Randomly select afunction from set F

(b) Continue and randomly select from the set C

(c) Randomly select from set C to generate terminal x_1 for the other branch

(d) Randomly generate terminals x_2 and x_1 from set C for the add function

Figure 33: Generating Initial Random Population

2. Fitness Test

Once we have the initial population, we test the fitness of each individual by executing each program in the population and assigning it a fitness value using the fitness measure. The fitness function could be a problem-specific issue. In many practical problems, the fitness of an

individual is measured by the mean of absolute difference between the predicted value using that individual and the true value. The closer this mean is to zero, the better the individual.

3. Genetic Operations

Once the fitness of each individual is evaluated, the evolutionary process starts and individuals in the next generation are created using the following techniques: reproduction, crossover, and mutation. Reproduction is simply copying individuals from the old generation to the next generation. Usually 10% of the old generation is copied to the next generation. Crossover is used to provide new and hopefully better individuals. It chooses two existing computer programs and randomly chooses a crossover point within each program to switch the chosen parts of the two programs.

For example, consider two individuals J1 and J2 shown in Equation (5) where ⟦…⟧ shows the crossover points in both individuals.

$$
\begin{aligned}
j_1 &= [G_{11}⟦G_{12}G_{13}\,G_{14}⟧G_{15}] \\
j_2 &= [G_{21}⟦G_{22}G_{23}⟧G_{24}] \\
o_1 &= [G_{11}⟦G_{22}G_{23}⟧G_{15}] \\
o_2 &= [G_{21}⟦G_{12}G_{13}\,G_{14}⟧G_{24}]
\end{aligned}
\tag{5}
$$

In genetic programming, there are many mutation operators in use. Sub-tree mutation is the simplest mutation operator, which replaces a randomly selected sub-tree with another randomly created sub-tree. The process of creating generations continues until we reach a stop criterion (e.g., the limit on the generation number). The final result of genetic programming is the best-so-far individual.

Methodology

1. Travel Time Prediction

Our problem entails predicting travel times given the current spatiotemporal variation in segment speeds. We hypothesize that speeds vary gradually in time and space unless a shockwave propagates through the system. The approach looks back L minutes into the past in predicting future travel times.

Travel Time Ground Truth Calculation

The calculation of the travel time ground truth is based on trajectory construction and the known speed through the trajectory's cells. A simple example of travel time ground truth calculation based on trajectory construction is demonstrated in Figure 34. In this example the roadway is divided into four sections using segments of length Δx and a time interval of Δt. We assume that the speed is homogenous within each cell. The average speed of the red-dotted cell (i=2, n=3) in the figure is u(x2,t3). Consequently, the trajectory slope represents the speed in each cell. Once the vehicle enters a new cell, the trajectory within this cell can be drawn as the straight blue line in Figure 34 using the cell speed as the slope. Finally, the ground truth travel time can be calculated when the trip reaches the downstream boundary of the last freeway section. It should be noted that the ground truth travel times were computed using the same INRIX dataset.

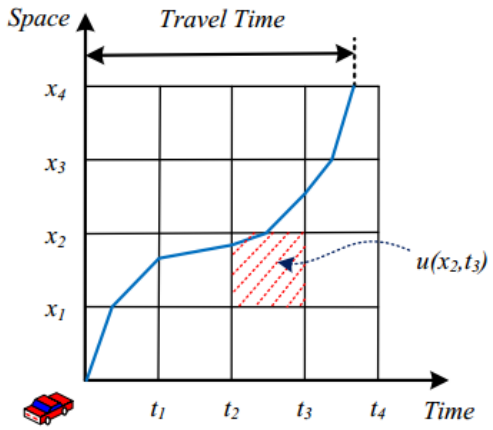


Figure 34: Illustration of Travel Time Ground Truth Calculation [4]

Instantaneous Travel Time

The instantaneous method is very simple where it assumes the segment speed does not change during the entire trip time. The travel time using the instantaneous approach is shown in equation (6)

$$\text{instantaneoustraveltime} = \sum_{i=1}^{h} \frac{L_i}{\upsilon_i^{t0}} \qquad (6)$$

Where L_i is the length of segment i , υ_i^{to} is the speed at segment i at the departure time t0 and h is the total number of segments.

Historical Average Method

If the spatiotemporal speed matrices are known for several previous days, we can calculate the ground truth travel time at each time interval for each day. The historical average at any time t0 is calculated using equation (7)

$$\text{historical averagetraveltime} = \sum_{i=1}^{Z} \frac{GTTT_i^{t0}}{Z} \qquad (7)$$

Where$GTTT_i^{t0}$ is the ground truth travel time at departure time t0 at historical day i and Z is number of days included in the average. In other words the historical average travel time at departure time t0 and current day is the average of the ground truth travel times at t0 for the previous Z days.

The historical average was calculated considering different Z values ranging from 5 to 30 days, where Z is number of days included in the average shown in Equation (7). As shown in Figure 35 there is no significance impact of z on the algorithm performance.

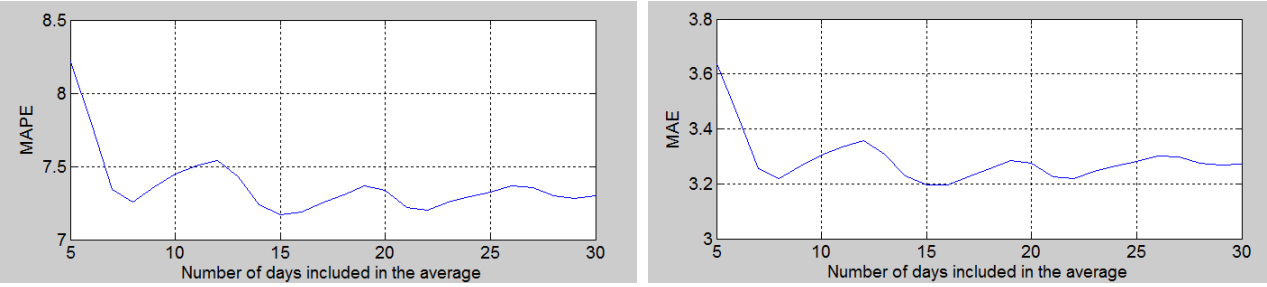


Figure 35: MAPE and MAE for the Historical Average at Different Number of Days Included in the Average.

Data Reshaping and Clustering

In general, data clustering is used to group objects of similar behavior into respective categories. These discovered categories have meaningful structures, within which the data are grouped in a way that the degree of association between two data vectors is maximal if they belong to the same group and minimal otherwise. By visually inspecting the spatiotemporal speed matrix we found that the speed patterns at different time intervals are different because of the activations of different bottlenecks. During these time periods the variance of the speed are different and we use clustering to group similar data vectors into clusters. The discovered clusters have similar structure and the GP can find better models for each cluster compared to only using a single model to predict the travel time. In this proposed algorithm to achieve more accurate results, we cluster the dataset into several partitions and build a model for each cluster. Building a model, also known as a program in genetic programming, is done through a training process. The training process is done using historical data that are arranged as shown in Figure 36. For each day (D_{58x288}speed matrix) from the historical dataset, we shift a window of size 58xL by one step, where 58 is number of road segments and 288 is the number of time intervals. For each sub-matrix D58xLwithin the (solid black rectangular), we reshape it into a vector x as in Equation (8).

$$D_{58xL} \xrightarrow{\text{yields}} [\grave{D}_{.1}\grave{D}_{.2} \dots \grave{D}_{.L}], \quad (8)$$

Where $\grave{D}_{.n}$ is the transpose of the nth column . The resulting vectors for each day are stacked together to construct the explanatory variables matrix X. The response vector Y is the ground truth travel time.

Before using matrix X and vector Y as inputs to the GP, we use k-means to cluster (partition) the X matrix. Given the rows of $X=(x_1; x_2; \dots; x_m)$ as our set of observations,where m is the number of rows in the matrix X and each row x is the reshaped sub matrix D58xL, we use the k-means to partition the m observations into k partitions ($k \ll m$) $S = \{S_1, S_2, \dots, S_k\}$ so that the objective function in Equation(9) is minimized:

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{x_{j.} \in S_i} d(x_{j.}, C_i). \quad (9)$$

Where d is the distance measure and C_i is the mean of the partition S_i. To minimize the objective function, k-mean sets are randomly selected as an initial set of k means $\left(C_1^{(0)}, C_2^{(0)}, \dots, C_k^{(0)}\right)$, then k-means alternate between the assignment step shown in Equation (10) and the update step shown in Equation (11) until the k-means converge.

$$S_i^{(t)} = \left\{x_{q.} : d(x_{q.}, C_i^{(t)}) \le d(x_{q.} - C_j^{(t)})\right\} \forall\, 1 \le j \le k \text{ and } 1 \le q \le m \quad (10)$$

$$C_i^{(t+1)} = \frac{1}{\left|S_i^{(t)}\right|} \sum_{x_j \in S_i^{(t)}} x_j \quad (11)$$

Where $\underline{\hat{Y}}$ is the matrix form of the predicted travel time from individuals in S_i.

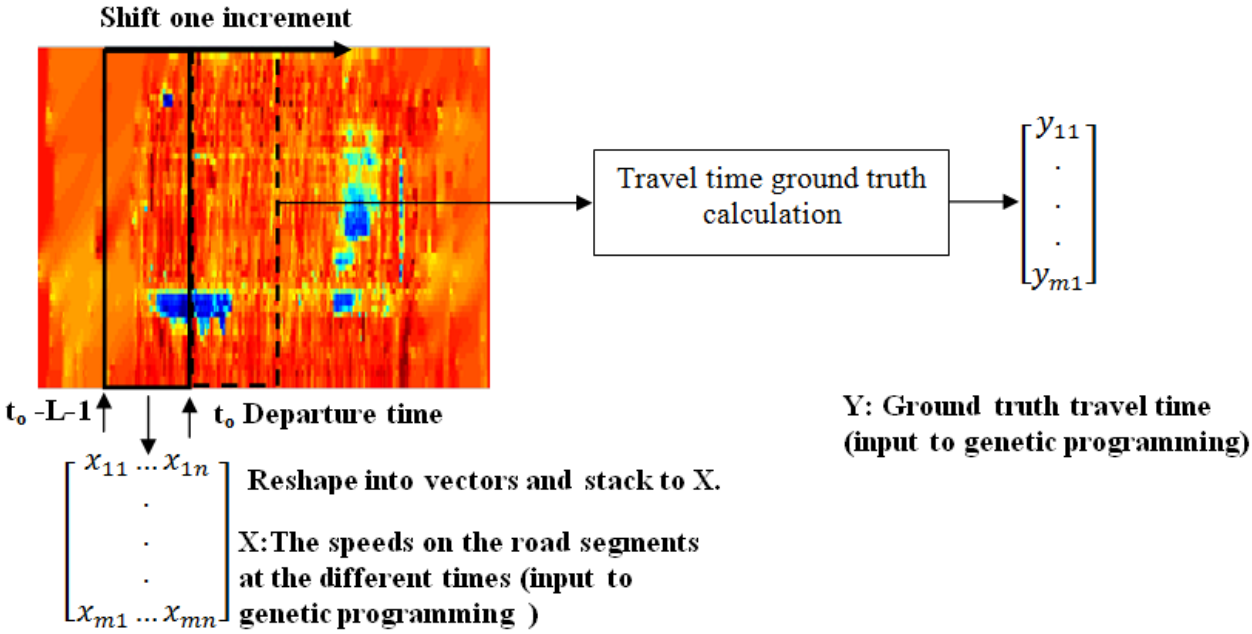


Figure 36: Illustration of the Preparation of the (X, Y) Inputs to Genetic Programming.

Model Building

The output of the k-means is the partitions set$S = \{S_1, S_2, \dots, S_k\}$ and the codebook set$C = \{C_1, C_2, \dots, C_k\}$. The next step entails building a model for each partition in S. We use GP to search the model space and find the best possible model. The inputs to the GP algorithm for partitionS_i are $x_j \in S_i$ for $\forall$ j and its corresponding ground truth travel time y_j.

In our approach we use a multi-individual GP. Multi-individual GP finds several models for each partition in S. The final model for a partition S_i is the linear combination of the models found by the GP for partition S_i. To find the linear combination coefficient, we regress the output of these models against the response, as demonstrated in Equation (12).

$$y_i = a_{i0} + a_{i1}\hat{y}_{i1} + \cdots + a_{ir}\hat{y}_{ir}; \quad (12)$$

Where $\hat{y}_{ir}$ is the predicted travel time from individual r in partition S_i. The values of the coefficients aij are found using the ordinary least squares estimation shown in its matrix form in Equation (13).

$$\underline{a} = (\underline{\hat{Y}}'\underline{\hat{Y}})^{-1}\underline{\hat{Y}}'\underline{Y} \quad (13)$$

After building all the models for all partitions, we are ready to estimate the travel time for any incoming new data. In estimating the travel time at a specific departure time, we use all the speeds across all segments at the current time and back in time L minutes. This matrix is reshaped into a vector using Equation (6), then the Euclidian distance between the incoming data and each code vector of the codebook is computed using Equation (14).

$$\arg\min_{C_i} \|x_{test.} - C_i\|^2, \forall\ 1 \le i \le k. \quad (14)$$

Based on the distance, the new data is assigned a partition, and the model for estimating the travel time of this cluster is used to estimate travel time as shown in Figure 37. The estimation process is very fast because we only substitute the values of the input variables into the model, which is very simple.

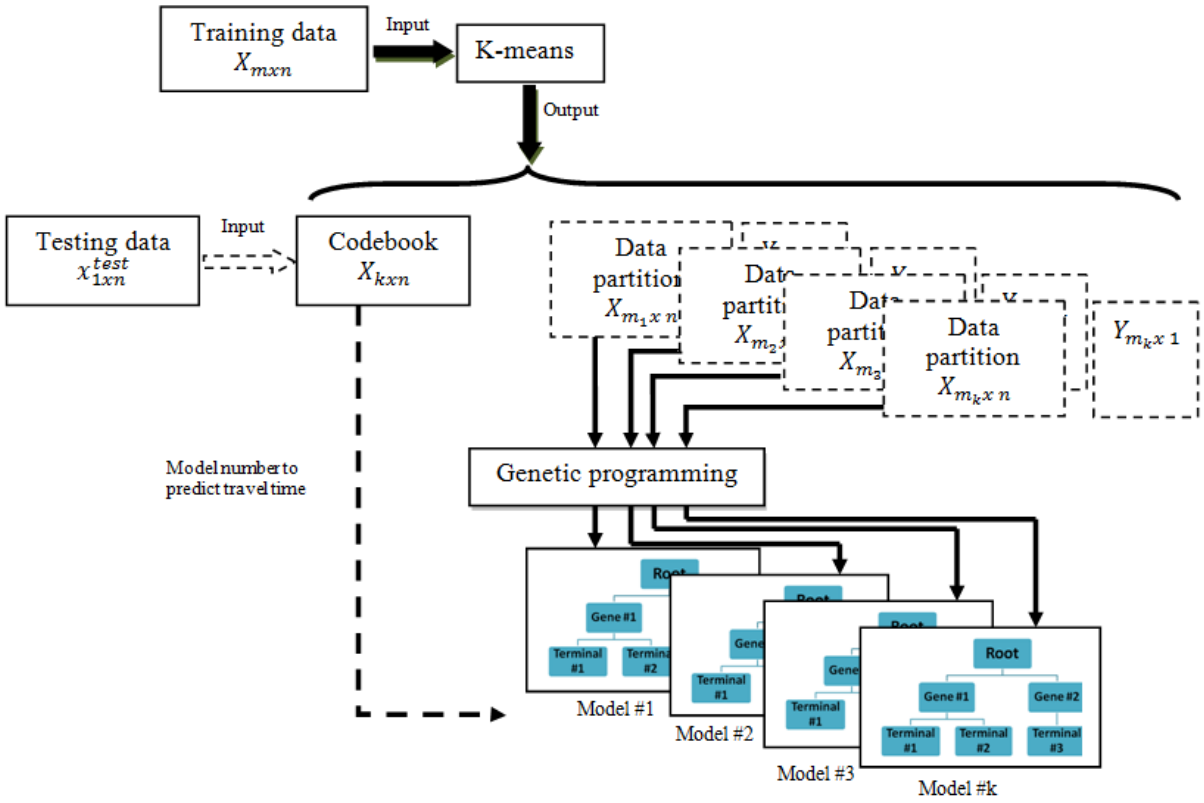


Figure 37: Block Diagram of the Proposed Algorithm. The Black Solid Arrows Show the Training Phase and the Dotted Arrows Show the Testing Phase (Predicting).

2. Estimation of Travel Time Lower and Upper Bounds

In the context of our problem, the training dataset π consists of$\{(x_i, y_i), i = 1, \dots, m\}$, where y_i is the ground truth travel time for speed pattern x_i. In the above sections, we developed a predictor$\varphi(x, \pi)$, such that when new input is received the model predicts y as $\varphi(x, \pi)$. Let us assume that we have multiple training datasets$\{\pi_v\}$ each coming from the same underlying distribution. Using multiple training datasets is typically expensive and time consuming. Consequently, we propose the use of bagging to generate multiple datasets. We use sampling from π with replacement to generate B new datasets $\{\pi_B\}$ each of the same size m. Each observation pair (x_i, y_i) may exist in the bootstrapped dataset several times, once or not at all. Each dataset in$\{\pi_B\}$ is a unique sample from the original dataset. Using the B datasets, K×B models are built using the GP algorithm, where K is the number of partitions. The use of bagging allows for the estimation of a travel time distribution as opposed to estimating a single travel time.

Case Study

The performance of the proposed GP algorithm was tested on a study 37-mile freeway section. A description of the test data is first introduced and then followed by a comparison of the proposed approach to the instantaneous method.

1. Data Description

The case study is conducted using privately developed INRIX traffic data, which are mainly collected using GPS-equipped probe vehicles. The collected probe data are supplemented with traditional road sensor data, as well as mobile devices and other sources [29]. As a result, the traffic data are the average speed of a roadway segment and aggregated at 5-minute intervals.

INRIX data are subject to continuous quality monitoring and improvement process. This process consists of five step process, which are described in detail in the literature [30]. Moreover, the quality of INRIX data was investigated and shown to be good for travel time prediction [31]. Finally, it should be noted that the ground truth travel times are computed using trajectory construction using the same INRIX data.

The INRIX data on the main segments along I-64 and I-264 in 2010 are used to construct the travel database. Since heavy traffic volumes are usually observed along I-64 and I-264 heading to Virginia Beach during the summer season and especially during the weekends, efficient and accurate travel time prediction can be helpful to travelers in planning their trips and reducing traffic congestion around the area. A 37-mile freeway stretch is selected to test the prediction algorithm, which includes most of the congested areas heading towards Virginia Beach from Richmond. The selected freeway stretch travels from Newport News to Virginia Beach along I-64 and I-264 and includes 58 sections as shown in Figure 38. The average length of all the sections is 0.65 miles and the longest section is the 3.7-mile segment located at the Hampton Roads Bridge-Tunnel (HRBT). Typically major congestion forms upstream of the HRBT and thus the freeway section include several congested locations with various backward forming shockwaves upstream of these bottlenecks.

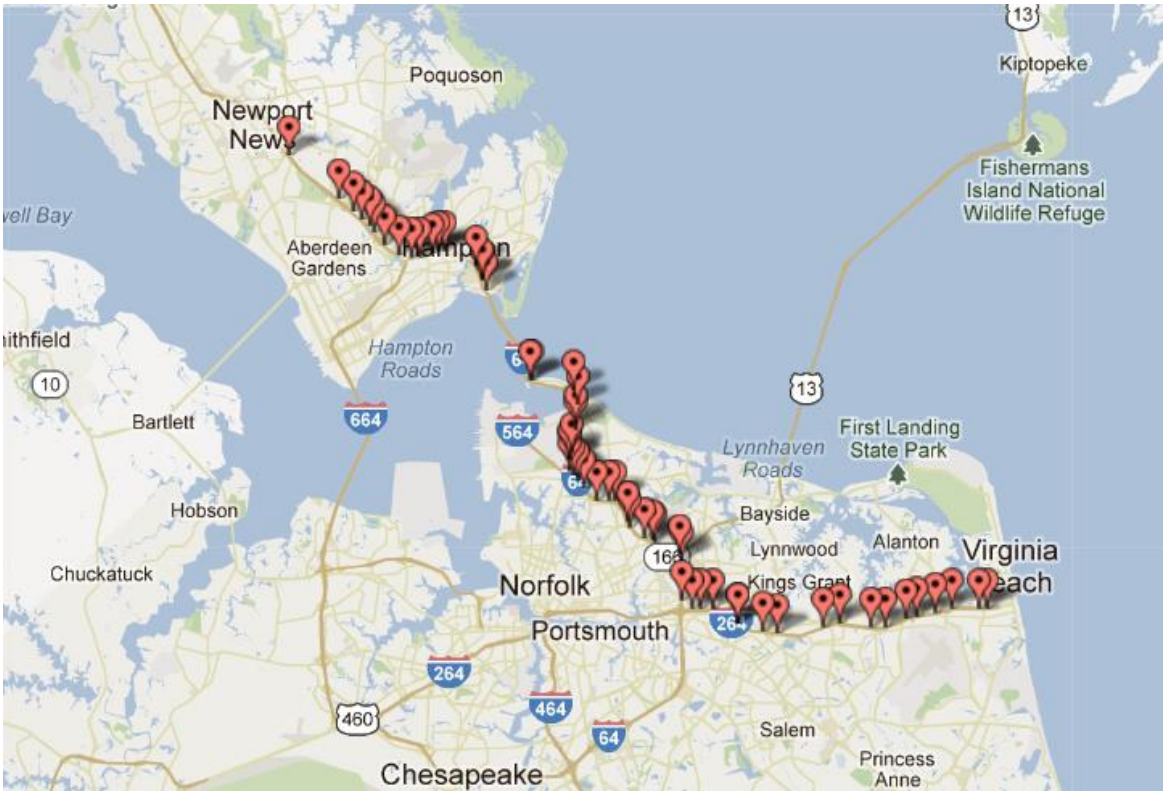


Figure 38: 37-mile Test Site.

A procedure of data reduction is conducted on the raw data to obtain daily traffic data, which is a spatiotemporal speed matrix. The data samples for typical weekday and weekend traffic occurring in June 2010 are presented in Figure 39. It should be noted that the traffic flow moves upwards. The figure illustrates a significant amount of missing data, especially for June5 and 6, 2010 (Saturday and Sunday). It appears from inspection of the data that the weekends involve more missing data than weekdays, which may pose a problem especially when making travel time predictions for weekends. According to the speed map of Figure 39 (a), most missing data (white areas) for a typical weekday occur between 21:00 p.m. and 5:00 a.m. (i.e., during the night and early morning hours). Normally there are few traffic volumes during this time period and free-flow speed could be assumed. However, sometimes the missing data also occur around a congested area (e.g., Figure(c) and (e)). Consequently, free-flow speed cannot be simply assumed for all missing data. A detailed description of the data imputation is provided in [4] and briefly described here.

As demonstrated earlier, various traffic data estimation algorithms have been developed for different data sources. Since ramp traffic data are not available, large errors will be introduced if macroscopic traffic models are used to estimate missing data. Alternatively, a statistical approach of data imputation is employed here that utilizes neighboring speed data over temporal and spatial conditions to estimate missing data. Here, the average value of eight neighboring cells is used to estimate the missing speed data in our dataset. Advanced approaches such as using kernel regression over temporal and spatial coordinates can be considered in the future. The samples of estimated speed maps for typical weekday and weekend traffic in June 2010 are presented in the right-hand column of Figure 39. Consequently, the full coverage daily temporal-spatial traffic data on the selected freeway stretch is estimated and can be used in the

proposed travel time prediction algorithm. Noteworthy is the fact that the full matrix of data is used for training and testing purposes, as will be described later.

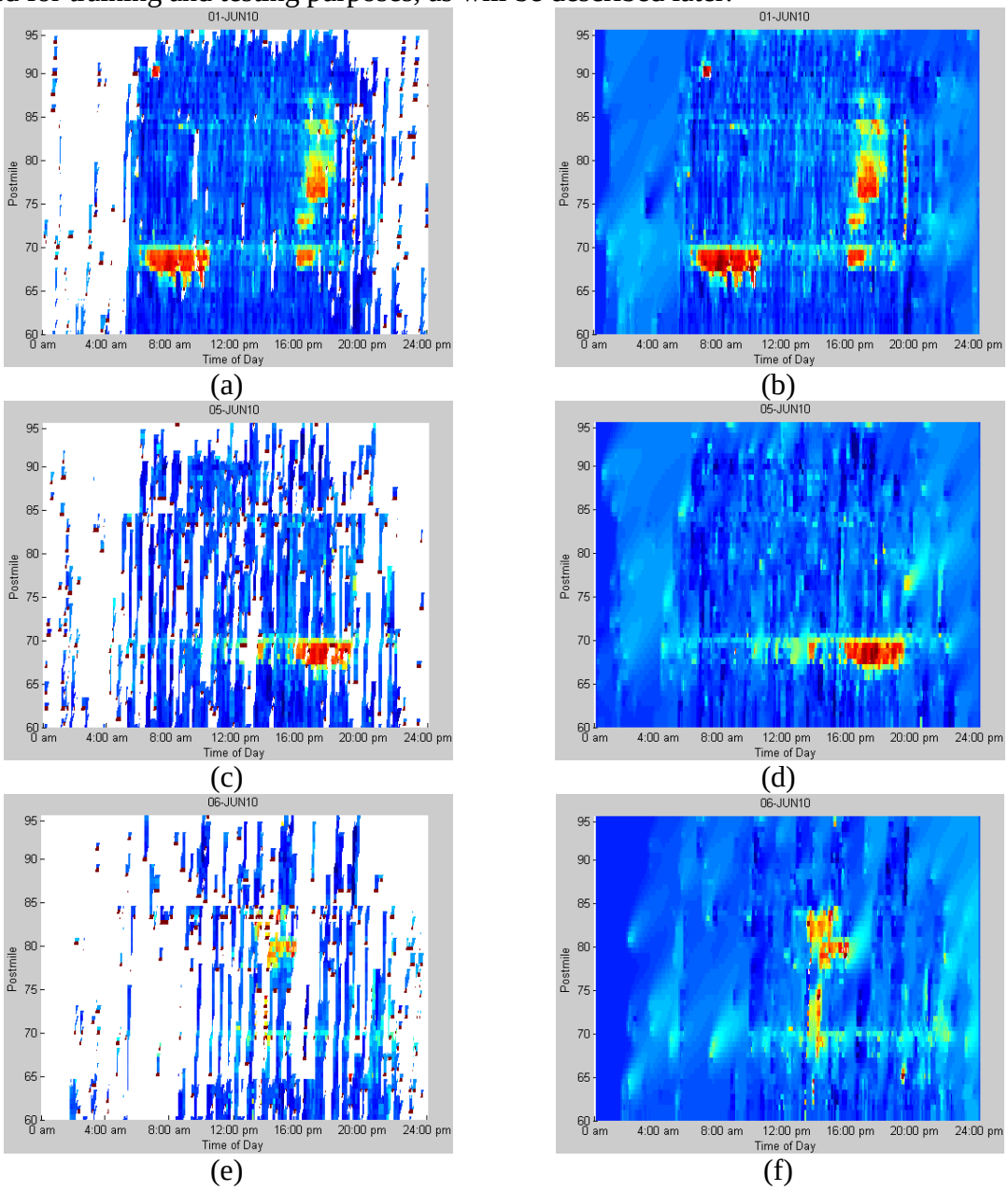


Figure 39: Samples of Daily Temporal-spatial Traffic State Variation.

2. Travel Time Prediction Results

Because congestion on the freeway stretch usually occurs during the summer months between 5:00 a.m. and 10:00 p.m., the evaluation of the prediction algorithm focuses on travel times from April to August of 2010 during this period (5:00 a.m. to 10:00 p.m.). Since the length of each section and the corresponding average speed for every time interval are known, the instantaneous travel time is calculated for each departure time. Here, we used the five-fold cross validation technique to test the proposed algorithm as shown in Figure 40 [32]. In the five-fold cross validation, the entire training and testing process is repeated five times (folds).In each fold, four

months (clear cells) are used as the training dataset to build the models, and the remaining months(yellow cells) are used to test the built models. The overall performance of the proposed algorithm is the average of the five folds' tests.

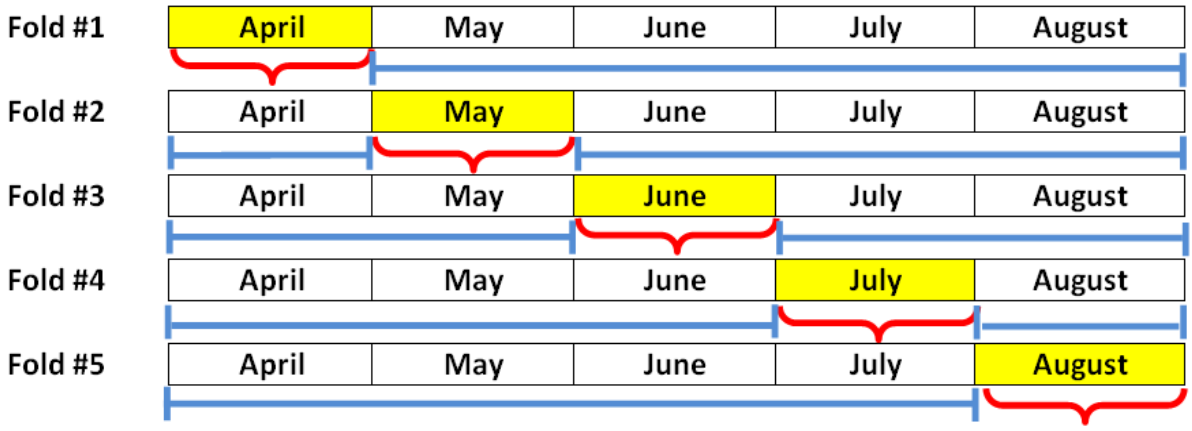


Figure 40: Five-fold Cross Validation Process.

The code for the proposed algorithm is written in Matlab using the Genetic Programming toolbox, which is available for free online [33]. The toolbox returns several models for each training cluster, and the estimated travel time is the weighted sum of individual models. We used the default settings of the toolbox, which are shown in Table 12. It is recommended that further experimentation with the various input parameters be conducted.

Table 12: Parameter Values Used to Setup the Genetic Programming Algorithm

Size of individual in the generation=300 Number of generations=100 Multi-gene option= true Maximum number of genes=4 Maximum depth of tree=5 Node functions = {'times', 'minus', 'plus'} (because the input for the model is the speed and the output is the travel time)

3. Selecting the Model Parameters

The proposed algorithm uses two parameters, namely: L, which is the temporal look back duration that we use to construct the model, and K, which is the number of clusters used in the algorithm. Optimizing these two parameters is a challenging task. In doing so, we ran the algorithm several times at different values of K and L. Both relative and absolute prediction errors are calculated for each set of parameters. The relative error is computed as the Mean Absolute Percentage Error (MAPE) using Equation (15). This error is the average absolute percentage change between the predicted and the true values. The corresponding absolute error is presented by the Mean Absolute Error (MAE) using Equation (16). This error is the absolute difference between the predicted and the true values.

$$\text{MAPE} = \frac{100}{I \text{x} J} \sum_{j=1}^{J} \sum_{i=1}^{I} \frac{\left| y_i^j - \widehat{y_i^j} \right|}{y_i^j} \qquad (15)$$

$$\mathrm{MAE} = \frac{1}{\mathrm{IxJ}} \sum_{j=1}^{J} \sum_{i=1}^{I} \left| y_i^j - \widehat{y_i^j} \right| \quad (16)$$

wher J is the total number of days in the testing dataset in each fold (i.e., 30 days or 31 days); I is the total number of time intervals in a single day; and y and $\hat{y}$ denote the ground truth and the predicted value, respectively, of the dynamic travel time for the i^{th}time interval on the j^{th}day. The relative and absolute errors calculated by the proposed method across various values of L and K are shown in Table 13 and Table 14. The L parameter was varied from 4 to 10at increments of 2, while the K parameter was varied from 2 to 8 at increments of 1.

Table 13: Calculated MAPE and MAE for Different Values of L and K

L / K	4		6		8		10	
	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE
2	4.32	1.93	4.40	1.97	4.37	1.95	4.40	1.96
3	4.28	1.92	4.35	1.94	4.36	1.94	4.43	1.97
4	4.25	1.90	4.28	1.92	4.35	1.94	4.37	1.95
5	4.24	1.90	4.26	1.90	4.32	1.93	4.32	1.93
6	4.24	1.90	4.25	1.90	4.30	1.92	4.32	1.93
7	4.24	1.90	4.29	1.92	4.25	1.90	4.31	1.93
8	4.24	1.90	4.27	1.91	4.27	1.91	4.34	1.94

The results demonstrate that the algorithm is not impacted significantly by changes in the L and K values and that the optimum performance occurs when K ranges from 3 to 7 and L equals 4.

4. Testing the Significance of the Proposed Algorithm

In comparing the proposed algorithm (using K=5 and L=4) with the instantaneous algorithm and the historical average of seven days, we compute the MAPE and MAE for each day in the dataset using the proposed algorithm and the pair methods, as presented in Table 14. Then, we apply the Wilcoxon Signed Rank Test to the MAE measures for the algorithms, as summarized in Table 15. We also applied the same test to the MAPE.

Wilcoxon Signed Rank Test is a non-parametric statistical hypothesis test. This test is equivalent to the paired t-test. It assumes that magnitudes of the differences between paired observations and the signs of differences carry information about the population. The test takes the paired observations and calculates the differences, where one pair is the error measure from the compared algorithm for the same day. The test then ranks the differences from smallest to largest by absolute value. The test statistic $W_{stat}$is calculated by adding all the ranks associated with positive differences. Finally, we use special tables of the Wilcoxon Signed Rank test to find the p-value associated with W_{stat}.

In both tests, our null hypothesis is that the distribution of differences is symmetrical around zero. The alternative hypothesis is that the differences tend to be smaller than zero.

Table 14: Summary Statistics for MAE and MAPE for GP, Instantaneous Method (INS), and Historical Average (HA)

Measure	MAPE(GP)	MAPE(INS)	MAPE(HA)	MAE(GP)	MAE(INS)	MAE(HA)
Mean	4.24	4.70	7.34	1.90	2.10	3.26
Std Dev	1.041	1.263	2.314	0.611	0.694	1.18
Median	4.184	4.631	7.06	1.858	2.041	3.15

Table 15: JMP Software's Output of the Wilcoxon Signed Rank

	MAE(GP)-MAE(INS)	MAPE(GP)-MAPE(INS)	MAE(GP)-MAE(HA)	MAPE(GP)-MAPE(HA)
Hypothesized Value	0	0	0	0
Actual Estimate	-0.1967	-0.4524	-1.3577	-3.0929
Std Dev	0.2079	0.46605	0.97598	2.29149
Signed-Rank test statistic	-5014.50	-5041.50	-5346.50	-5348.50
p-value	<0.0001	<0.0001	<0.0001	<0.0001

The experimental results show that our algorithm has a statistically lower MAPE and MAP compared to the instantaneous algorithm and the historical average method for 7 days. The p-value for the Wilcoxon Signed Rank Test is less than 0.0001. Consequently, the null hypothesis is rejected. We conclude that there is significant evidence that the MAE and MAPE for the proposed GP algorithm is less than the MAE and MAPE of both the instantaneous algorithm and the historical average method.

For individual days that have significant congestion and rapid temporal speed changes, the GP algorithm is significantly superior to the instantaneous algorithm. For these days, the MAPE and MAE are 25 percent less than those for the instantaneous algorithm and 76 percent less than those for historical average. As shown in Figure 41: Comparison of Proposed Algorithm and the Instantaneous Algorithm for Two Days. Figure 41, the instantaneous algorithm underestimates the travel time as congestion builds up. Another drawback of the instantaneous algorithm is that it overestimates the travel time as the peak period recedes. Alternatively, the GP algorithm responds quickly to changes in speeds at the shoulders of the peak period.

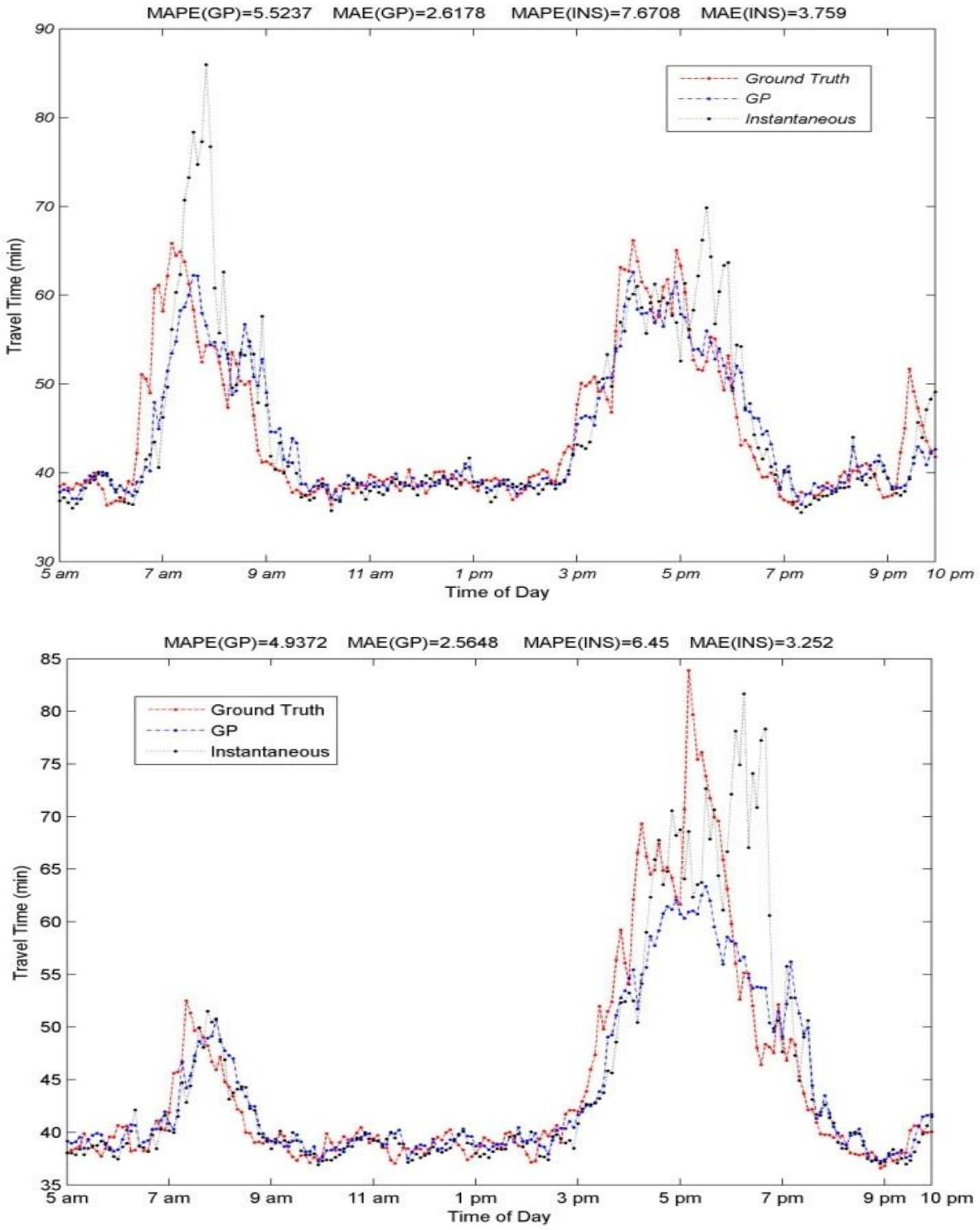


Figure 41: Comparison of Proposed Algorithm and the Instantaneous Algorithm for Two Days.

5. Model Interpretability

One of the advantages of using GP in predicting travel times is that the model is simple and interpretable. By analyzing the model variables, critical segments that affect the predicted travel time along a roadway can be identified. To illustrate this concept, Figure 42 visualizes the

codebook at K=4 and L=4. This codebook illustrates the four clusters that the model identified. In the figure, the blue color represents low speeds. The first two clusters show a red color at the tunnel location where congestion occurs. The third code vector shows light congestion at two different locations, one of them at the tunnel. The fourth code vector shows free-flow conditions. These clusters clearly demonstrate that for congested conditions the tunnel is the most critical location in predicting travel times given that it serves as a recurring bottleneck.

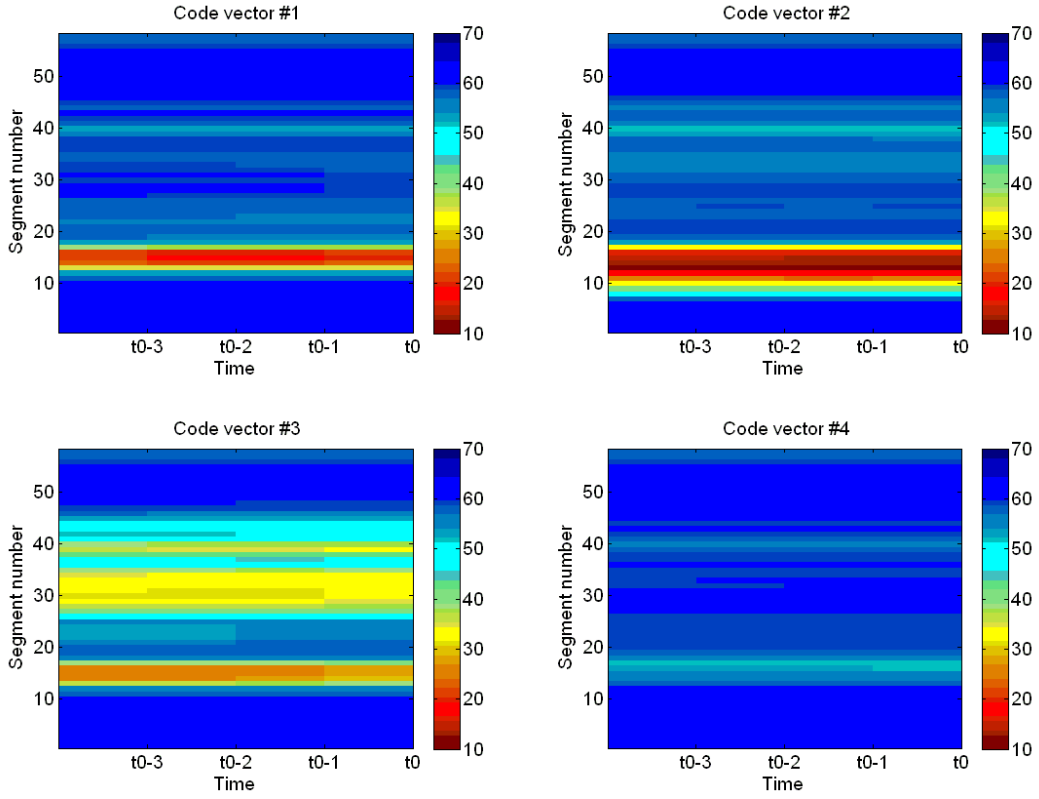


Figure 42: Visualization of Codebook at K=4 and L=4.

In order to interpret the GP models for the dataset partitions obtained using the above codebook, we visualize the coefficients of the first order terms in the models. As shown in Figure 43, the most important coefficients are those at the tunnel or near the tunnel area. Also, most of the important coefficients are those related to the speeds immediately prior to departure (i.e., at t0). The algebraic equations (models) built using the GP are presented in Appendix A.

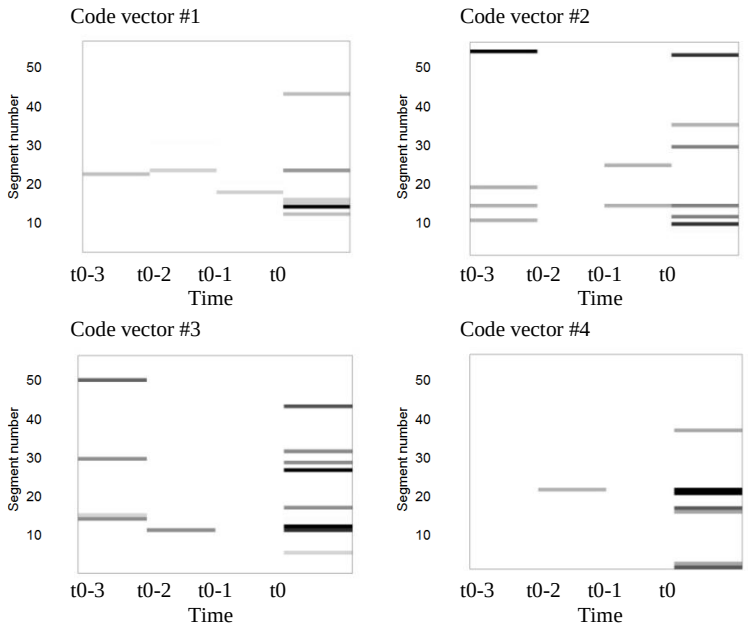


Figure 43: Coefficients of linear Term Models.

Bagging and Genetic Programming Results

The other important information are the upper and lower limits on the travel time. The travel time interval estimation is based on bootstrapping the original dataset. Given the dataset (X, Y) where each row in the X matrix is the explanatory variables and the corresponding element in Y is the ground truth travel time, we build 100 datasets $S = \{(\tilde{X}_1, \tilde{Y}_1), (\tilde{X}_2, \tilde{Y}_2), \dots, (\tilde{X}_{100}, \tilde{Y}_{100})\}$,where $(\tilde{X}_n, \tilde{Y}_n)$ is the dataset number *n*. The dimensions of $(\tilde{X}_n, \tilde{Y}_n)$ are the same as (X, Y). The $(\tilde{\mathrm{X}}_\mathrm{n}, \tilde{\mathrm{Y}}_\mathrm{n})$should be about 66 percent of the original training (X, Y) and the other cases are repeated cases. For each dataset $(\tilde{X}_n, \tilde{Y}_n)$, we apply our proposed algorithm using the same configuration shown in Table 12except the population size is set to 100to derive the models for this dataset.

The travel time interval for an unseen new data case (instance) can be defined by the maximum and minimum travel times estimated from the models from all 100 datasets. The histograms of the predicted travel time width for the five-month analysis period are shown in Figure 44 and Table 16 demonstrate that the width of the travel time interval is less than five minutes in duration. Furthermore, as illustrated in Figure 45, most of the ground truth travel time experiences are within the estimated travel time interval. Furthermore, points that are outside the interval are close to the interval borderlines. This accurate information, when provided to travelers, gives them a better idea about the expected conditions on the road. We also note that using the travel time interval is more robust than using a single value for the predicted travel time. When the speed pattern along the road varies, the difference between the predicted travel time and the ground truth is larger than the difference between the upper bound of the travel interval and ground truth. This makes travelers more confident when using the travel time interval.

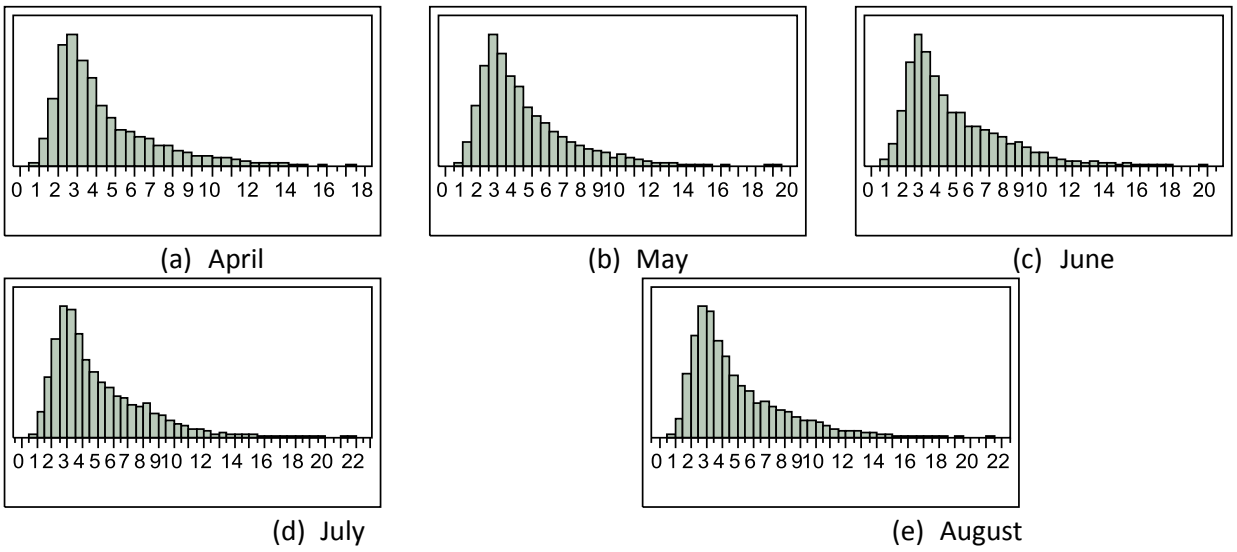


Figure 44: Width of Predicted Travel Time Interval.

Table 16: Monthly Variation in Predicted Travel Time Interval

	April	May	June	July	August
Mean	4.19	4.38	4.64	4.83	4.73
Std Dev	2.368	2.370	2.587	2.7534	2.704
Std Err Mean	0.03027	0.0298	0.0331	0.0346	0.0340

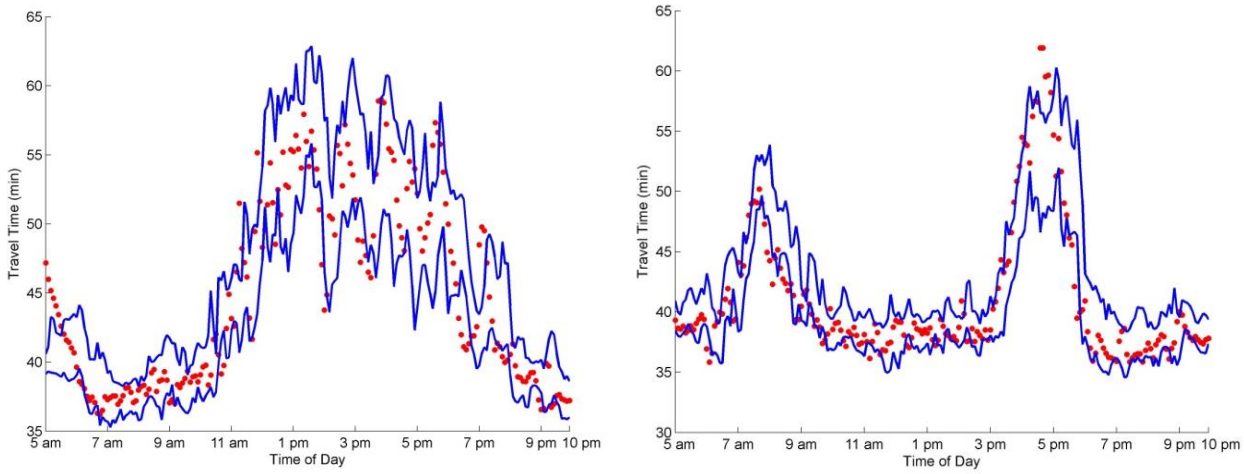


Figure 45: Temporal Variation in Travel Time Estimates Relative to Ground Truth.

Conclusions and Future Work

The research presented in this paper develops a genetic programming algorithm for predicting dynamic travel times. The proposed algorithm uses the k-means approach to partition the data into similar clusters in the training phase and to act as a simple classifier during the testing phase. The genetic programming approach is then used to build a model for each data partition. The proposed algorithm has two main advantages. The first advantage is the simplicity of the model and its computational efficiency. The second advantage of the model is that it is interpretable and

provides insight into critical segments that significantly impact future travel times. The results show the superior performance of the proposed algorithm when compared to the state-of-the-practice instantaneous algorithm. The proposed algorithm is tested on a 37-mile freeway section. The prediction error is demonstrated to be significantly lower than that produced by the instantaneous algorithm or the historical average (p-value<0.0001). Specifically, the proposed algorithm achieves more than a 25 percent and 76 percent reduction in the prediction error over the instantaneous and historical average, respectively on congested days. When we use bagging and genetic programming, the results show that the mean width of the travel time interval is less than 5 minutes for the 37-mile trip.

References

[1] D. Schrank, B. Eisele, and T. Lomax, "2012 Urban Mobility Report," Texas Transportation Institute2012.

[2] P.-E. Mazare, O.-P. Tossavainen, A. Bayen, and D. Work, "Trade-offs between Inductive Loops and GPS Vehicles for Travel Time Estimation: A Mobile Century Case Study," presented at the Transportation Research Board 91st Annual Meeting, Washington, D.C., 2012.

[3] H. Tu, "Monitoring Travel Time Reliability on Freeways," Ph.D., Department of Transport and Planning, Technische Universiteit Delft, 2008.

[4] H. Chen, H. A. Rakha, S. A. Sadek, and B. J. Katz, "A Particle Filter Approach for Real-time Freeway Traffic State Prediction," in *91st Transportation Research Board AnnualMeeting*, Washington D.C., 2012.

[5] H. Chen, H. A. Rakha, and S. A. Sadek, "Real-time Freeway Traffic State Prediction: A Particle Filter Approach," in *14th International IEEE Conference on Intelligent Transportation Systems*, Washington, DC, USA, 2011, pp. 626-631.

[6] L. Du, S. Peeta, and Y. H. Kim, "An Adaptive Information Fusion Model to Predict the Short-term Link Travel Time Distribution in Dynamic Traffic Networks,," *Transportation Research Part B: Methodological*, vol. 46, pp. 235-252, 2012.

[7] J. W. C. v. Lint, S. P. Hoogendoorn, and H. J. v. Zuylen, "Accurate Freeway 1 Travel Time Prediction with State-space Neural Networks Under Missing Data," *Transportation Research Part C: Emerging Technologies*, vol. 13, pp. 347-369, 2005.

[8] J. Myung, D.-K. Kim, Seung-Young Kho, and C.-H. Park, "Travel Time Prediction Using k Nearest Neighbor Method with Combined Data from Vehicle Detector System and Automatic Toll Collection System " *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2256, pp. 51-59, 2011 2011.

[9] E. I. Vlahogianni, J. C. Golias, and M. G. Karlaftis, "Short-term Traffic Forecasting: Overview of Objectives and Methods," *Transport Reviews*, vol. 24, pp. 533-557, 2004.

[10] J.-S. Yang, "Travel Time Prediction Using the GPS Test Vehicle and Kalman Filtering Techniques," in *Proceedings of the 2005 American Control Conference*, 2005, pp. 2128-2133.

[11] X. Fei, C.-C. Lu, and K. Liu, "A Bayesian Dynamic Linear Model Approach for Real time Short-term Freeway Travel Time Prediction," vol. 19, pp. 1306-1318, 2011.

[12] M. Chen and S. I. J. Chien, "Dynamic Freeway Travel-Time Prediction with Probe Vehicle Data: Link Based Versus Path Based," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1768, pp. 157-161, 2001.

[13] J. Xia, M. Chen, and W. Huang, "A Multistep Corridor Travel-Time Prediction Method Using Presence-Type Vehicle Detector Data," *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, vol. 15, pp. 104-113, 2011.

[14] C. P. I. J. v. Hinsbergen, A. Hegyi, J. W. C. v. Lint, and H. J. v. Zuylen, "Bayesian Neural Networks for the Prediction of Stochastic Travel Times in Urban Networks," *IET Intelligent Transport Systems*, vol. 5, pp. 259-265, 2011.

[15] J. Xia and M. Chen, "Freeway Corridor Travel Time Prediction Using Single Inductive Loop Detector Data," presented at the Transportation Research Board 88th Annual Meeting, Washington D.C., 2009.

[16] L. Vanajakshi and L. R. Rilett, "Support Vector Machine Technique for the Short Term Prediction of Travel Time," presented at the IEEE Intelligent Vehicles Symposium, Turkey, 2007.

[17] C.-H. Wu, J.-M. Ho, and D. T. Lee, "Travel-time Prediction with Support Vector Regression," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, pp. 276-281, 2004.

[18] B. I. Bustillos and Y.-C. Chiu, "Real-Time Freeway-Experienced Travel Time Prediction Using N-Curve and k Nearest Neighbor Methods," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2243, pp. 127-137, 2011.

[19] W. Qiao, A. Haghani, and M. Hamedi, "Short Term Travel Time Prediction Considering the Weather Impact," presented at the the Transportation Research Board 91st Annual Meeting, 2012.

[20] M. Yang, Y. Liu, and Z. You, "The Reliability of Travel Time Forecasting," *IEEE Trans. Intell. Transport. Syst*, vol. 11, pp. 162-171, 2010.

[21] C. Nanthawichit, T. Nakatsuji, and H. Suzuki, "Application of probe-vehicle data for real-time traffic-state estimation and short-term travel-time prediction on a freeway," *Transportation Research Record*, pp. 49-59, 2003.

[22] M. Oltean and L. Dioşan, "An autonomous GP-based system for regression and classification problems," *Applied Soft Computing*, vol. 9, pp. 49-60, 1// 2009.

[23] J. C. Bezdek, S. Boggavarapu, L. O. Hall, and A. Bensaid, "Genetic algorithm guided clustering," in *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*, 1994, pp. 34-39 vol.1.

[24] W. Langdon, R. Poli, N. McPhee, and J. Koza, "Genetic Programming: An Introduction and Tutorial, with a Survey of Techniques and Applications," in *Computational Intelligence: A Compendium*. vol. 115, J. Fulcher and L. C. Jain, Eds., ed: Springer Berlin Heidelberg, 2008, pp. 927-1028.

[25] C. Xu, W. Wang, and P. Liu, "A Genetic Programming Model for Real-Time Crash Prediction on Freeways," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 14, pp. 574-586, 2013.

[26] K. A. Marko and R. J. Hampo, "Application of genetic programming to control of vehicle systems," in *Intelligent Vehicles '92 Symposium., Proceedings of the*, 1992, pp. 191-195.

[27] C. Jia-Ruey, C. Shun-Hsing, C. Dar-Hao, and L. Yao-Bin, "Rutting Prediction Model Developed by Genetic Programming Method Through Full Scale Accelerated Pavement Testing," in *Natural Computation, 2008. ICNC '08. Fourth International Conference on*, 2008, pp. 326-330.

[28] J. R. Koza, *Genetic programming : on the programming of computers by means of natural selection*. Cambridge, Mass.: MIT Press, 1992.
[29] INRIX. (2012). *Traffic Information*. Available: http://www.inrix.com/trafficinformation.asp
[30] T. Trepanier. INRIX Data Services - Arterial System Performance Assessment and Management [Online]. Available: http://www.mtc.ca.gov/services/arterial_operations/downloads/6-3-13/5_INRIX_Data_Service_for_Arterials-2013June03.pdf
[31] H. Rakha, H. Chen, A. Haghani, and K. F. Sadabadi, "Assessment of Data Quality Needs for use in Transportation Applications " 2013.
[32] B. Efron and R. Tibshirani, "Improvements on cross-validation: The .632+ bootstrap method," *Journal of the American Statistical Association*, vol. 92, pp. 548-560, Jun 1997.
[33] D. searson. (2009). *Genetic programming & symbolic regression for Matlab* Available: http://gptips.sourceforge.net

Appendix E

Table 17 below shows the algebraic equations that are derived using the genetic program for each cluster. As shown in the table, the equations have linear terms and interaction terms. From the equation, we can see that most of the dominant terms are at time t0 or near t0. The $v(s,t)$ in the equation represents the speed at segment number s and time t.

Table 17: The Algebraic Equations (Models)

Model #1	y_{pred}=78.74+0.002495×v(t0,46)^2 - 0.2911×v(t0,46)- 0.05495×v(t0-1,42) + 0.002495×v(t0-1,39) - 0.07583×v(t0,48) - 0.05495×v(t0,45) - 0.05495×v(t0,44) - 0.1185×v(t0,36) - 0.07583×v(t0,15) - 0.07334×v(t0-3,22) - 0.05246×v(t0-2,36) + 0.002495×v(t0,28) ;
Model #2	y_{pred}=92.82+0.05417×v(t0-3,49) + 0.000487×v(t0-2,2) - 0.05417×v(t0-1,45) - 0.05417×v(t0-3,45) + 0.05417×v(t0-1,34) - 0.1419×v(t0,50) - 0.08775×v(t0,48) - 0.08775×v(t0,45) + 0.05417×v(t0-3,40) - 0.08775×v(t0,29) - 0.05417×v(t0,23) + 0.000487×v(t0,15) - 0.1419×v(t0,4) - 0.1755×v(t0-3,3) + 0.000487×v(t0-2,57) - 0.002273×v(t0,46)×(v(t0,52) + v(t0,51)) - 2.297×10^{-6}×(v(t0,18) - 8.683)×(6.616×v(t0-1,29)×v(t0,15) + 6.616×v(t0,15)×v(t0-3,36) - 1.221);
Model #3	y_{pred}=75.92+0.01898×v(t0-3,16) - 0.05009×v(t0-3,44) - 0.000311×v(t0,54) - 0.01898×v(t0,53) - 0.09342×v(t0,47) - 0.1124×v(t0,46) + 0.05009×v(t0,41) - 0.1124×v(t0,31) - 0.05009×v(t0,29) - 0.05009×v(t0,26) - 0.07592×v(t0,14) + 0.05009×v(t0-3,28) - 0.06907×v(t0-3,7) - 0.05009×v(t0-2,47) - 0.001875×v(t0,44)×v(t0,26) + 0.000311×v(t0-3,25)×v(t0-2,40)
Model #4	y_{pred}=67.03+1.929×10^{-5}×v(t0,38)^2×v(t0,37) - 0.03861×v(t0,57) - 0.03861×v(t0,43) - 0.07254×v(t0,42) - 0.1111×v(t0,38) - 0.1111×v(t0,37) - 0.03861×v(t0,21) - 1.929×10^{-5}×v(t0-3,2) - 0.03393×v(t0-2,37) - 1.929×10^{-5}×v(t0,43)×v(t0-3,2) - 0.07254×v(t0,58) - 4.49×10^{-6}×(v(t0-1,10) + v(t0,43) + v(t0,38) - v(:,80))×(v(t0,43) + v(t0,41) - v(t0,19) + v(t0,54)×v(t0,19) + 7.378)

Chapter 8: Random Forest Travel Time Prediction Algorithm using Spatiotemporal Speed Measurements

This chapter is based on
Mohammed Elhenawy, Hao Chen, and Hesham Rakha, "Random Forest Travel Time Prediction Algorithm using Spatiotemporal Speed Measurements," presented at the 2014 World Congress on Intelligent Transport Systems, Detroit, Michigan, United States, 2014.

Abstract

Accurate prediction of dynamic travel times can assist commuters in making better travel decisions. In this paper, a new algorithm is proposed to accurately predict the expected and confidence levels of dynamic travel times. The algorithm pre-processes the available historical data to identify recurring bottlenecks along the road. Subsequently, the algorithm builds a spatiotemporal congestion probability distribution. This distribution provides the probability of a spatiotemporal section being congested. The proposed algorithm integrates congestion probability and spatiotemporal speed measurements to construct feature vectors that are used as the travel time predictors. A random forest is used to model the relationship between the predictors and the travel time. Consequently, the built random forest can be used to predict the travel time by propagating the new features vector through all trees. The experimental results show that the proposed algorithm achieves more than a 38 percent reduction in the prediction error on congested days compared to the state-of-practice instantaneous algorithm and 28 percent reduction when compared to a genetic programming travel time prediction algorithm. Moreover, the predicted travel time bounds encompass all field observations.

Introduction

Traffic congestion has become a serious problem in medium and large cities. In 2007, it cost highway users 4.2 billion extra hours of sitting in traffic and an extra 2.8 billion gallons of fuel. This all translated into an additional $87.2 billion in congestion costs for road users in 2007, which showed a 50% increase in the cost compared to data from the previous decade. In developed countries, expanding roadway infrastructure is becoming less of an option for transportation and government agencies due to environmental, financial and social constraints. Under these circumstances, monitoring and disseminating travel time information through Advanced Traveler Information Systems (ATISs) drivers can make better travel decisions. The predictive travel time provides important information for travelers and transportation operators to schedule their trips and make better travel decisions that can reduce traffic congestion.

Various traffic sensing technologies have been used to collect traffic data for use in computing travel times, including point-to-point travel time collection (e.g., license plate recognition systems, automatic vehicle identification systems, mobile devices, Bluetooth sensors, probe vehicles, etc.) and station-based traffic state measuring devices (loop detectors, video cameras, remote traffic microwave sensors, etc.). Private companies such as INRIX integrate different sources of measured data to provide section-based traffic state data (speed, average travel time), which is used in our study to develop algorithms for predicting travel times. The benefit of using section-based traffic state data is that travel time can be easily calculated from traffic state data. More importantly, section-based data provide the flexibility in developing scalable applications.

By providing section-based traffic state data, there are two approaches to compute travel times depending on the trip experience [1, 2]. The dynamic travel time is the actual, realized travel time that a vehicle experiences during a trip. If a vehicle leaves its origin at the current time, the roadway speed will not only change over space but also over time during the entire trip. Consequently, dynamic travel times can be obtained by using a prediction algorithm to compute the speed evolution downstream in future time steps. Instantaneous travel time is the other approach available to compute travel times without the consideration of speed evolution over time. It is usually computed using the current speed along the entire roadway; in other words, the speed field is assumed to remain constant in time. Instantaneous travel time is close to the dynamic travel time when the roadway speed does not change significantly along the time axis over the trip duration. However, this approach may deviate substantially from the actual, experienced travel time under transient states during which congestion is forming or dissipating during a trip [3].

Over the past decades, many research efforts have attempted to predict travel times. According to the manner of modeling, these methods can be classified into time series models including Kalman filter [4], Auto-Regressive Integrated Moving Average (ARIMA) models [5] and data-driven models, such as artificial neural networks (ANNs) [6], support vector regression (SVR) [7] and K-Nearest Neighbor (k-NN) [8, 9] models. These techniques are implemented through direct and indirect procedures to predict travel times using different types of state variables. Travel time is directly used as the state variable in model-based or data-driven methods to predict travel times. Indirect procedures are performed using other variables (such as traffic speed, density, flow, occupancy, etc.) as the state variable to predict the future traffic state, and then future travel times can be calculated using some transition matrix. However, existing methods are either insufficient or have limitations to predict dynamic travel times for departures at the current or future times. For real time applications, instantaneous travel times can be obtained by summing the section travel times at the current time interval. But experienced travel time can only be obtained on the completion of the trip, because the temporal and spatial evolution of speed should be considered. In this case, the experienced travel time from the previous time interval usually is not available for predicting travel times in the next interval, especially for long trips [10].

In order to solve the above problems, a prediction approach using random forests is proposed in this paper. A random forest (RF) is an ensemble learning method for classification or regression and is widely used in machine learning and statistics analysis [11]. RF methods have been applied in many applications and different fields in the past decades. In the biological studies, it is used for a large variety of tasks, including identification of DNA-binding proteins [12], bacterial species identification [13], classification of protein-localization patterns within florescent microscope images [14], discrimination between acidic and alkaline enzymes [15], and diagnosing Alzheimer's disease based on single photon emission computed tomography (SPECT) data. Computer vision engineers use RF successfully in many image and video processing tasks such as segmentation of video objects [16], and recognition of face images [17]. RF is also applied in diverse engineering applications such as aircraft engine fault diagnosis [18], automatic e-mail filing into folders (multi-class problem) and spam e-mail filtering (two-class problem) [19], hyper-spectral remote sensing and geographic data classification [20] and network intrusion detection [21]. Recently, RF was used in the transportation field in some problems, including: classifying and counting vehicles detected by multiple inductive loop

detectors [22], identifying motorway rear-end crash risks using disaggregate data [23], and automatic traffic incident detection [24].

Considering that a roadway bottleneck includes the dynamic propagation information of traffic status, a spatiotemporal probability distribution is constructed in this paper to provide additional feature information that is integrated with spatiotemporal speed data as regressor variables in the construction of a decision tree. The spatiotemporal probability distribution is calculated from the historical dataset using a bottleneck identification technique. A simple statistical approach called Automated Statistically-principled Bottleneck Identification Algorithm (ASBIA) is used in this paper to identify freeway bottlenecks [25]. Compared to other bottleneck identification methods, ASBIA is a simple statistical approach that requires only one pre-defined parameter, yet outperforms other state-of-the-art algorithms.

This paper develops a new algorithm to accurately predict expected freeway dynamic travel times and reliability measures using decision tree and bottleneck identification techniques. The algorithm pre-processes the available historical dataset to identify congested roadway segments and construct a spatiotemporal congestion probability matrix. The proposed algorithm integrates the congestion probability matrix with spatiotemporal speed measurements to construct feature vectors that used as predictors and the corresponding historical travel time as the response. We model the relationship between the predictors and the response using a random forest. The built random forest can be used to predict the travel time by sending the new features vector through all trees. The experimental results show significant improvement of the proposed algorithm over state-of-practice algorithms.

The remainder of the paper is organized as follows. First, the bottleneck identification and decision tree techniques are introduced. This is followed by the details of the proposed prediction algorithm. Subsequently, a case study using probe data is provided to validate the proposed algorithm and compare its performance to other prediction methods. Finally, conclusions and future research directions are presented.

Methodology

1. Bottleneck Identification

A bottleneck identification algorithm called Automated Statistically-principled Bottleneck Identification Algorithm (ASBIA) was developed by researchers at the Virginia Tech Transportation Institute (VTTI). This algorithm uses speed measurements at location x and its neighbors in the time-space domain to evaluate the status of speed at x. The output of the algorithm is the status of the roadway segment (free-flow or congested) and the confidence level of the test (p-value). Specifically, the ASBIA makes use of the correlation between a point in the spatiotemporal domain and its neighbors. This correlation exists along both the time and space domains (i.e. temporal and spatial correlations). The algorithm uses the fact that points in close proximity (both temporally and spatially) to any point x provide additional information about point x. ASBIA assumes a two-phase traffic flow state, where traffic states are either free-flow or congested. In addition, it assumes that the speed data are sampled from two component Gaussian distributions and modeled as a mixture model. The first component represents the congested regime and all speed measurements in the congested regime are drawn from this component. The second component represents the uncongested regime and all speed measurements within this regime are drawn from free-flow conditions. The two-component multi-state model based on the normal distribution is presented the equation below. The λ parameter is the mixture proportion,

i.e., the probability that the traffic stream speed is in the first state; and u is the traffic stream space-mean-speed.

$$f(u|\lambda, \mu_1, \mu_2, \sigma_1, \sigma_2) = \lambda \frac{1}{\sqrt{2\pi}\sigma_1} e^{\frac{(u-\mu_1)^2}{2{\sigma_1}^2}} + (1-\lambda) \frac{1}{\sqrt{2\pi}\sigma_2} e^{\frac{(u-\mu_2)^2}{2{\sigma_2}^2}} \quad (1)$$

where, (μ_1, σ_1) and (μ_2, σ_2) are the mean and standard deviation of the first and second component distributions, respectively.

ASBIA uses a δ1 by δ2 window to select a sample from the speed matrix and identify the status of the point at the center of the window. The algorithm simply consists of the following steps:

1. Move a window δ1 by δ2 over the time-space domain such that the window scans all the data points in the domain.
2. If the speed measurements within the spatiotemporal window are equal then the center point is identified as free-flow if the speed is greater than the speed-at-capacity (uc), otherwise it is identified as congested.
3. If the speed values within the spatiotemporal window are not constant, then we use the t-test to characterize the status of window center point. The null hypotheses H0: μobs≤uc and the alternative hypothesis is μobs>uc. The center point is congested if the test fails to reject the null hypothesis; otherwise it is considered free-flow if we reject the null hypothesis.

ASBIA is the first automatic congestion identification algorithm that is based on statistical concepts. This approach identifies the status of the current point using its speed reading and the readings from its neighbors in space and time. ASBIA demonstrated promising performance in terms of its high true positive and low false positive rates. ASBIA will be applied to identify typical recurrent bottlenecks to improve the travel time prediction algorithm.

2. Random Forests

The random forest [11] is an ensemble approach that is very efficient in dealing with data prediction problems. The biggest advantage of the random forest approach is that it does not suffer from over fitting because of the law of Large Numbers. Ensembles are techniques that use the divide-and-conquer approach to improve performance. The main idea behind ensemble methods is that building a large group of simple models will give an overall improved performance. In other words a group of weak models will give a resultant strong model. The random forest is a large group of un-pruned decision trees with randomized selection of features at each split. The well-known machine learning technique called Classification And Regression Tree (CART) [26] is one of the common decision trees used in random forests. A random forest starts with the CART which, in ensemble terms, corresponds to the weak model. CART is a greedy and recursive top-down binary partitioning that divides the feature space into sets of disjoint regions. These regions should be pure with respect to the response variable. The random forest algorithm for regression can be simply described in the following section.

Assume the training dataset has H cases, P predictors, and N represents the number of trees to build for each of the N iterations

1. Sample H cases randomly with replacement to create a bootstrap sample from the original dataset. The subset should be about 66% of the original training and the other cases are repeated cases.
2. At each node P/3 predictor variables are selected randomly from all the predictor variables.

3. The predictor variable out of the P/3 predictors that provides the best split (minimum squared error), is used to conduct a binary split on that node.
4. At the next node, randomly choose another P/3 predictors from all the P explanatory variables and conduct the same process.
5. Do not perform cost complexity pruning and save trees as is with the other built trees before this iteration.

At the testing phase, when a new case arrives (data vector) we propagate the vector down all of the trees to predict an output. Each tree will give a prediction and the result can either be an average or weighted average of all of individual predictions.

Proposed Algorithm

In order to integrate the prior bottleneck information into the travel time prediction, the proposed algorithm uses the available historical dataset during the training phase to build a stochastic congestion spatiotemporal map. This map is built once during the training phase and then is used during testing phase. We assume that the probability distribution of being congested at each segment follows a Bernoulli distribution with mean π (2).

$$p(Z/z) = (1-\pi)^{1-z}(\pi)^{z}, \quad (2)$$

Where z equals one if the segment is congested and zero otherwise. π is not constant, instead it is a function of location and time $\pi_{s,t} = \varphi(\text{segment}, t)$. It may change from a segment to another segment and from a time period to another time period, so that estimating $\pi_{s,t}$ is the very first step in our algorithm. The following sections describe the training and testing phases in detail.

1. Training Phase

In this phase we start by estimating $\pi_{s,t}$ and then we show how we fuse the spatiotemporal speed matrix with the congestion probability map to build a random forest model for travel time prediction. As shown in Figure 46, the core of estimating $\pi_{s,t}$ is the ASBIA algorithm, which statically processes each day's spatiotemporal speed data $V_{s,t}$. The output of ASBIA is a spatiotemporal binary matrix where one indicates a congested segment and zero indicates a free-flow segment. After processing all historical days, $\pi_{s,t}$ is simply the frequency of ones at each spatiotemporal element as shown in (3)

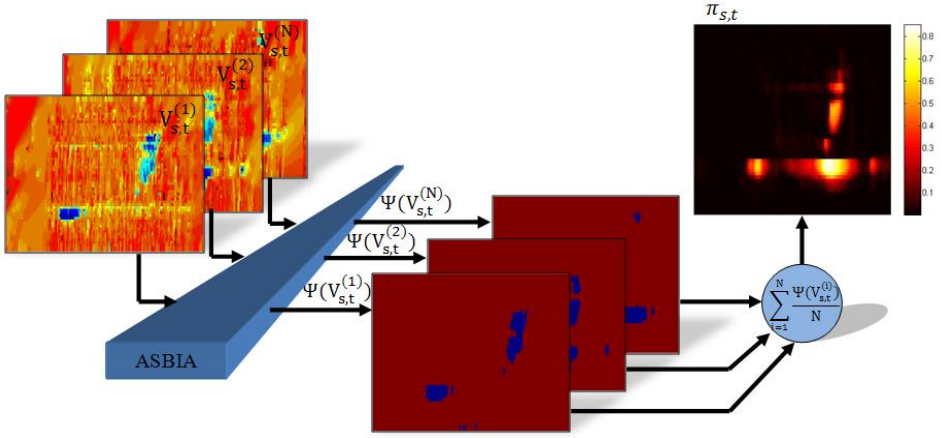


Figure 46: Illustration of $\pi_{s,t}$ Estimation

$$\pi_{s,t} = \sum_{i=1}^{N} \frac{\Psi(V_{s,t}^{(i)})}{N}, \quad \text{.....(3)}$$

where Ψ is the ASBIA mapping function from the spatiotemporal speed matrix $V_{s,t}$ to the spatiotemporal binary matrix and N is the number of days in the historical dataset.

After estimating $\pi_{s,t}$, the next step entails fusing both $\pi_{s,t}$ and the spatiotemporal speed matrix to construct the training dataset for the development of the random forest model. The random forest trees are regression trees so that the training set should include the explanatory variable vector X (regressors) and the corresponding response Y. In our problem context the response is the ground truth travel time corresponding to each assumed departure time. The regressors include the speed distribution along the road segments at the times $\{t_{-m+1}, t_{-m+2}, \ldots, t_{-1}, t_0\}$, where t_0 is the departure time. The regressors also should include information that shows what the expected future conditions of the road segments are expected to be. This is achieved by using some of the regressors as the means of the Bernoulli distribution at each road segment at the times $\{t_1, t_2, \ldots, t_m, \}$. The training regressors vector is the concatenation of all the above speeds and expected means as shown in equation (4)

$$X_{t_0} = [\grave{V}_{.,t}\big|_{t=t_{-m+1}} \grave{V}_{.,t}\big|_{t=t_{-m+2}} \ldots \grave{V}_{.,t}\big|_{t=t_0} \grave{\pi}_{.,t}\big|_{t=t_1} \grave{\pi}_{.,t}\big|_{t=t_2} \ldots \grave{\pi}_{.,t}\big|_{t=t_m}], \quad (4)$$

Where X_{t_0} is a row vector of the regressors at the departure time t_0 , and $\grave{V}_{.,t}\big|_{t=t_{-m+1}}$ is the transpose of the $V_{.,t}\big|_{t=t_{-m+1}}$. The training dataset is the collection of all X_{t_0} at different departure times and the corresponding ground truth travel time response Y_{t_0}. The training dataset is used to build the random forest and the final step after training is assigning weights to the trees. The weight of each tree is based on how much of the travel time variability is explained by this tree. These weights are used to give one predicted travel time using the T predicted times from the T trees

$$w_q = \frac{R^2{}_q}{\sum_{q=1}^{T} R^2{}_q},$$

$$R^2{}_q = 1 - \frac{SS_{RES_q}}{SS_T}, \quad (5)$$

$$SS_{RES} = \sum_{j=1}^{k} (y_j - \hat{y}_j)^2 ,$$

$$SS_T = \sum_{j=1}^{k} (y_j - \bar{y})^2$$

Where T is the number of trees in the forest. The outputs of the training phase are three components, namely the random forest, the $\pi_{s,t}$ matrix, and the weights of the trees.

2. Testing Phase

This phase is simpler than the training phase and is much faster from a computational standpoint. In this phase we predict the travel time given the traveler departure time t_0 . The first step entails constructing the predictor vector by concatenating the speed distribution along the road segments at the times $\{t_{-m+1}, t_{-m+2}, \ldots, t_{-1}, t_0\}$ of the current day and the means of the Bernoulli distribution at each road segment at the times $\{t_1, t_2, \ldots, t_m, \}$ as shown in equation (6).

$$X_{t_0}{}^{(test)} = [\grave{V}_{.,t}\big|_{t=t_{-m+1}} \grave{V}_{.,t}\big|_{t=t_{-m+2}} \ldots \grave{V}_{.,t}\big|_{t=t_0} \grave{\pi}_{.,t}\big|_{t=t_1} \grave{\pi}_{.,t}\big|_{t=t_2} \ldots \grave{\pi}_{.,t}\big|_{t=t_m}], \quad (6)$$

The Bernoulli distribution means in the above vector do not depend on the day, instead they only depend on departure time t_0. The speed portion of the vector depends on both the day and time of departure t_0. The second step entails running the regressor vector through each tree in the random forest to compute the predicted travel time from each tree. The final predicted travel time is the weighted average of the tree predictions as in equation (7)

$$\hat{y} = \frac{\sum_{q=1}^{T} w_q * \hat{y}^{(q)}}{\sum_{q=1}^{T} w_q} \quad , \tag{7}$$

where $\hat{y}^{(q)}$ is predicted travel time from tree q.

Case Study

This section describes the test site that was used to test the proposed algorithm. The description of the test site and the field data is first introduced followed by a comparison of the proposed approach to the instantaneous and GP method.

1. Data Description

The case study is conducted using privately developed INRIX traffic data that are mainly collected using GPS-equipped probe vehicles. The collected probe data are supplemented with traditional road sensor data, as well as mobile devices and other sources [27]. As a result, the traffic data represent the average space-mean speed of a roadway segment over a 5-minute interval. The 2010 INRIX data along I-64 and I-264 are used to construct the travel database. Since heavy traffic volumes are usually observed along I-64 and I-264 heading toward Virginia Beach during the summer and especially on the weekends, efficient and accurate travel time prediction can be of importance to travelers in planning their time of departure and route of travel. A 37-mile freeway stretch is used to test the prediction algorithm, which includes most of the congested areas heading towards Virginia Beach from Richmond. The selected freeway stretch is located from Newport News to Virginia Beach along I-64 and I-264 and includes 58 sections as shown in Figure 47. The average section length is 0.65 miles and the longest section is 3.7 miles in length and is located at Hampton Roads Bridge-Tunnel (HRBT). The speed matrix is a 58 by 720 matrix $V_{s,t}$. The ground truth dynamic travel times are computed using piecewise constant speed values and the trip trajectory is a combination of diagonal curves over time and space [28].

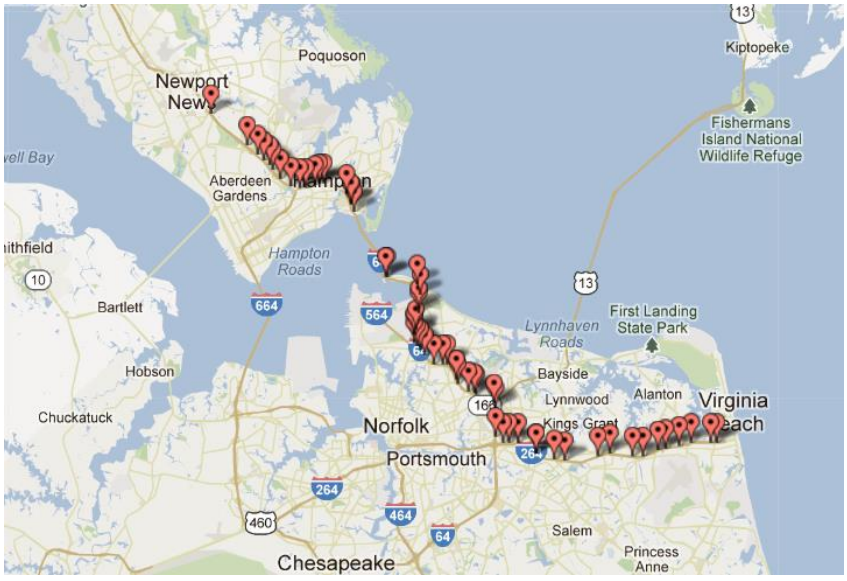


Figure 47: Selected I-66 and I-264 Freeway Stretch.

Considering the congestion on the selected freeway stretch usually occurs during the summer months between 5:00 am to 10 p.m., the evaluation of the prediction algorithm focuses

on travel times from April to August of 2010 from 5:00 a.m. to 10:00 p.m. Since the length of each section is known the corresponding average speed can be computed from the average travel time. The instantaneous travel time is calculated for each departure time. In this paper, we used the five-fold cross validation technique to test the proposed algorithm [29]. In the five-fold cross validation the training and testing process is repeated five times (folds) in each fold four months are used as the training dataset to estimate $\pi_{s,t}$ and build the random forest trees and remainder month is used to test the built random forest. The overall performance of the proposed algorithm is the average of the five-fold tests.

2. Number of Trees

The number of trees is an important parameter in RF methods. Usually the number of trees is application dependent. In order to find the appropriate number of trees, the RF was run several times for different number of trees and an objective function was computed for each forest. The number of trees that produced the best objective function was used. In the proposed algorithm, another important parameter is the number of regressors. For the sake of simplicity, we assume there is no interaction between the number of regressors and the number of trees. In other words the curves of the objective function at different number of trees and regressors are parallel. In the case study we fix (m) at four, and the algorithm was run several times for different number of trees. Afterwards, the relative error was computed as the Mean Absolute Percentage Error (MAPE) using Equation (8). This error was the average absolute percentage change between the predicted and the true values. The corresponding absolute error is presented by the Mean Absolute Error (MAE) using Equation (9). This error is the absolute difference between the predicted and the true values.

$$\mathrm{MAPE} = \frac{100}{I \times J} \sum_{j=1}^{J} \sum_{i=1}^{I} \frac{\left| y_i^j - \widehat{y_i^j} \right|}{y_i^j} \tag{8}$$

$$\mathrm{MAE} = \frac{1}{I \times J} \sum_{j=1}^{J} \sum_{i=1}^{I} \left| y_i^j - \widehat{y_i^j} \right| \tag{9}$$

Here J is the total number of days used in the testing dataset in each fold (i.e., 30 days or 31 days); I is the total number of time intervals in a single day; and y and $\hat{y}$ denote the ground truth and the predicted value, respectively, of the dynamic travel time for the i^{th}time interval on the j^{th}day.

The relative and absolute errors calculated by the proposed method considering different number of trees are shown in Figure 48. According to the results, an elbow of the curve appears to occur at 100 trees and thus was used in the experiments.

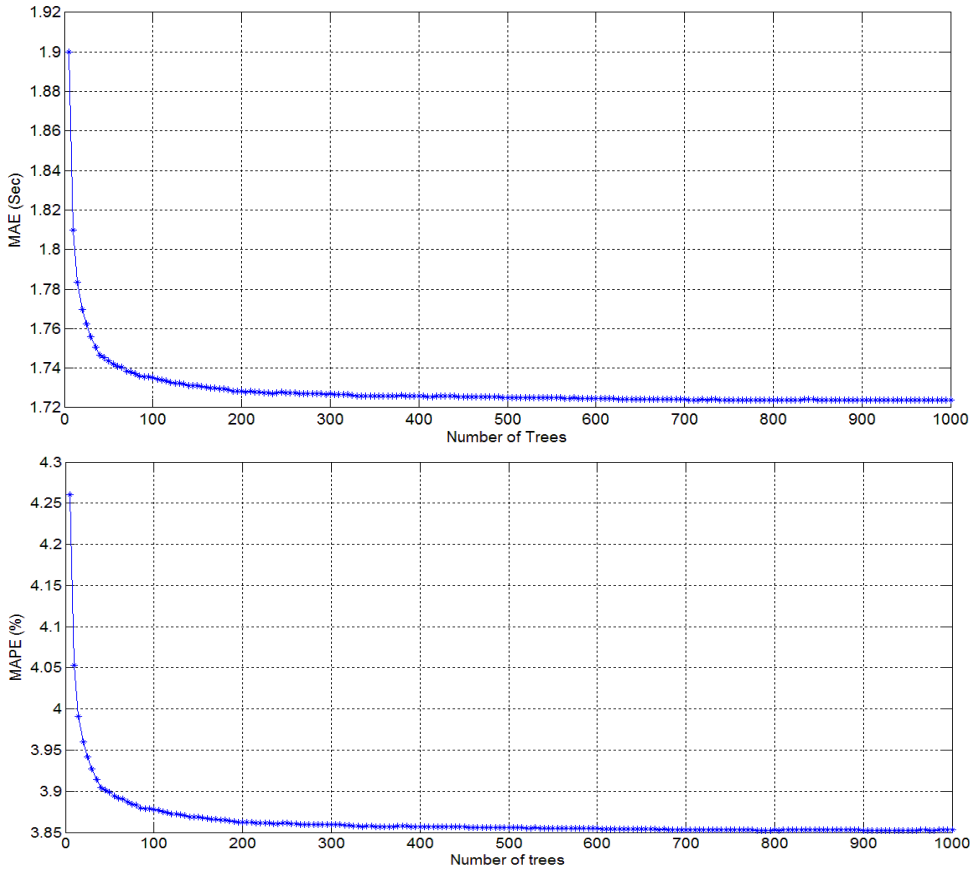


Figure 48: Variation in MAPE and MAE as a Function of the Number of Trees

3. Number of Predictors or Regressors

Model selection is an important step in any modeling exercise. The modeler typically looks for a model that accurately captures the data while at the same time is simple. An important parameter in the proposed algorithm is how far back in time from the departure time t_0 should the predictor include in the speed matrix $V_{s,t}$. Another important parameter is how far forward in time relative to the departure time should the algorithm consider in the $\pi_{s,t}$. For simplicity we set up both parameters equal to m. Optimizing this parameter (m) is a challenging task. In doing so the algorithm was run several times for different values of m and the MAPE and MAE were calculated. The relative and absolute errors calculated by the proposed method across various values of m are shown in Table 18. The m parameter is varied from 4 to 10 at a step size of 2. The results demonstrate that the algorithm is not impacted significantly by changes in the m parameter. Consequently, we chose the simplest model (lowest number of regressors at m=4). This corresponds to a look ahead and look back of 20 minutes.

Table 18: Calculated MAPE and MAE for Different Values of m

m	4	6	8	10
MAE	1.7240	1.7366	1.7390	1.7390
MAPE	3.8529	3.8800	3.8860	3.8850

4. Model Testing and Evaluation

In order to better evaluate the performance of the proposed predictor, the instantaneous travel time and a GP method were applied on the same dataset and compared to the proposed algorithm. The instantaneous travel time method is the easiest alternative to predict future travel times by assuming the current traffic speed along all the segments remains constant until the completion of the trip. This method is currently used by the Virginia Department of Transportation (VDOT) to display travel time information on variable message signs. Consequently, instantaneous travel times are considered the state-of-practice and used to quantify the tradeoff between simplicity and prediction accuracy.

A GP approach for modeling and predicting the expected dynamic travel times along freeways was developed in an earlier publication [30]. It is computationally efficient and suitable for real-time applications. This approach was tested on the same 37-mile freeway segment. The prediction errors were found to be significantly lower than the prediction errors associated with the instantaneous algorithm. Specifically, the algorithm achieved more than a 25 percent reduction in the prediction error for congested days compared to the instantaneous algorithm. Consequently, the GP algorithm was selected as a comparison method in addition to the instantaneous approach while testing the proposed algorithm.

To compare the proposed algorithm (at m=4) with the instantaneous algorithm and GP algorithm, the average MAPE and MAE were computed, as summarized in Table 19. Subsequently, the matched pair permutation test was applied to compare the proposed algorithm with each of the other algorithms [31]. The statistical tests are conducted using MAEs and MAPEs. The null hypotheses of the tests are shown in Table 20.

Table 19: Summary Statistics for MAE and MAPE by Three Approaches

	MAE(INS)	MAE(GP)	MAE(RF)	MAPE(INS)	MAPE(GP)	MAPE(RF)
Mean	2.100	1.900	1.724	4.700	4.240	3.853
Std Dev	0.694	0.611	0.591	1.263	1.041	0.963
Median	2.041	1.858	1.591	4.631	4.184	3.672

Table 20: The Null Hypotheses and Alternative Hypotheses for Statistical Tests

Comparing MAE between ntaneous and random insta forest	$H_0: MAE(INS)_{true} - MAE(RF)_{true} < MAE(INS)_{org} - MAE(RF)_{org}$ -vs- $H_a: MAE(INS)_{true} - MAE(RF)_{true} \geq MAE(INS)_{org} - MAE(RF)_{org}$
Comparing MAE between instantaneous and genetic programming	$H_0: MAE(INS)_{true} - MAE(GP)_{true} < MAE(INS)_{org} - MAE(GP)_{org}$ -vs- $H_a: MAE(INS)_{true} - MAE(GP)_{true} \geq MAE(INS)_{org} - MAE(GP)_{org}$
Comparing MAE between instantaneous and random forest	$H_0: MAPE(INS)_{true} - MAPE(RF)_{true} < MAPE(INS)_{org} - MAPE(RF)_{org}$ -vs- $H_a: MAPE(INS)_{true} - MAPE(RF)_{true} \geq MAPE(INS)_{org} - MAPE(RF)_{org}$
Comparing MAE between instantaneous and genetic programming	$H_0: MAPE(INS)_{true} - MAPE(GP)_{true} < MAPE(INS)_{org} - MAPE(GP)_{org}$ -vs- $H_a: MAPE(INS)_{true} - MAPE(GP)_{true} \geq MAPE(INS)_{org} - MAPE(GP)_{org}$

The experimental results show that the proposed algorithm has a statistically lower MAPE and MAE compared to the instantaneous algorithm. Specifically, the p-value for all matched pair permutation tests is less than 0.0001. Consequently, the null hypothesis is rejected concluding that there is significant evidence that the MAE and MAPE for the proposed RF algorithm is less than MAE and MAPE of both the instantaneous and the GP algorithms.

For individual days that experience significant congestion and rapid temporal speed changes, the GP algorithm is significantly superior to the instantaneous algorithm. For these days the MAPE and MAE is 38 percent lower than the instantaneous algorithm and 28 percent for the GP algorithm. As shown in Figure 49 the instantaneous algorithm underestimates the travel time as congestion builds up. Another drawback of the instantaneous algorithm is that it overestimates the travel time as the peak period recedes. Alternatively, the random forest algorithm responds quickly to changes in speeds at the shoulders of the peak period.

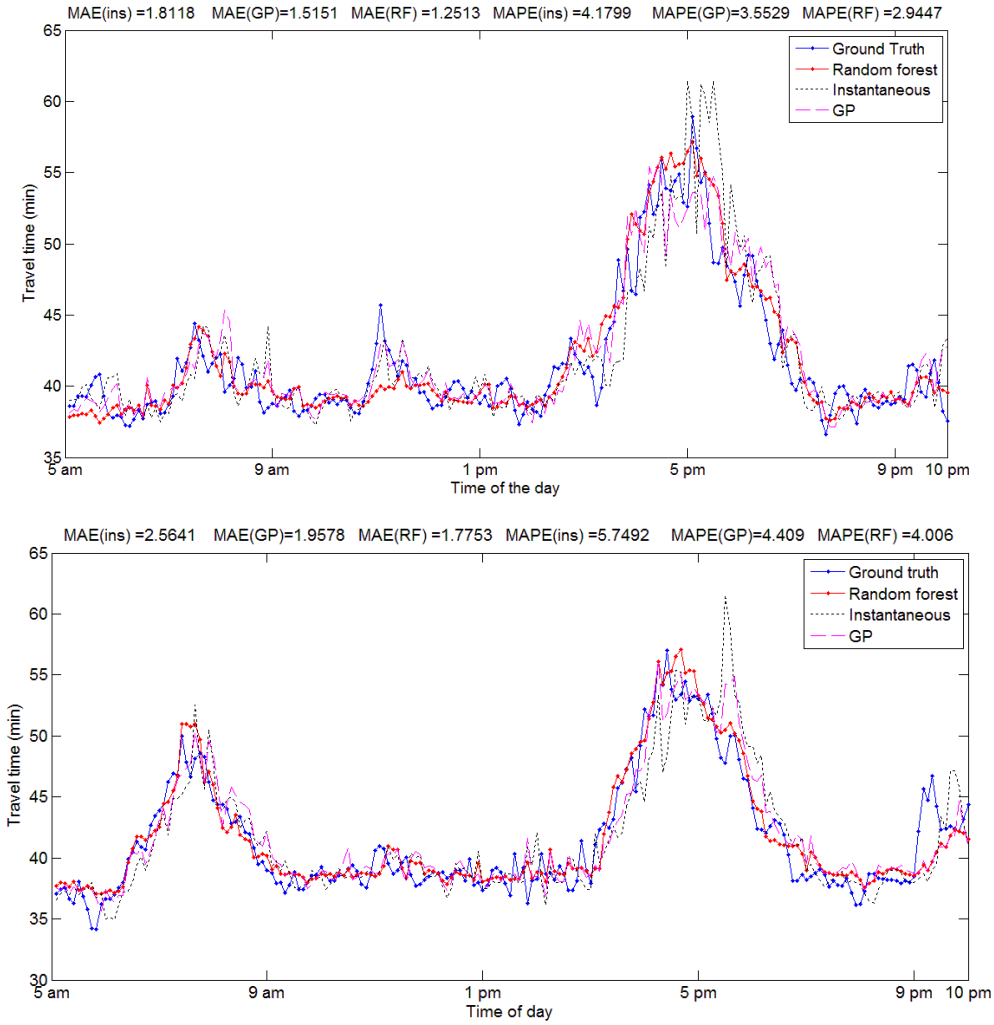


Figure 49: Samples of Prediction Results for Different Methods.

5. Random Forest Travel Time Confidence Limits

The other important piece of information is the travel time upper and lower bounds, which is a measure of the travel time reliability. One advantage of the random forest approach is that it computes the bounds with no additional effort. Since the random forest resamples the training

dataset several times with replacement to generate each tree. For each sample builds a regression tree. Each tree in the random forest provides an estimate of the travel time. The travel time reliability can be computed as the 5% to 95% quintile.

According to Table 21, the width of the travel time interval is less than ten minutes in duration. Furthermore, as illustrated in Figure 50, most of the ground truth travel time experiences are within the estimated confidence interval. Furthermore, points that are outside the interval are very close to the interval borderlines. This accurate information, when provided to travelers gives them a better idea about the expected conditions on the road. We also note that using the travel time interval is more robust than using a single value for the predicted travel time. When the speed pattern along the facility varies, the difference between the predicted travel time and the ground truth is larger than the difference between the upper bound of the travel interval and ground truth. This makes travelers more confident when using the travel time interval.

Table 21: Monthly Variation in Predicted Travel Time Interval

	April	May	June	July	August
Mean (min)	8.4894035	8.5106378	9.2514588	9.5293581	9.1267532
Std Dev (min)	5.0365181	4.815047	5.4910914	5.9737553	5.479364
Std Err Mean	0.0643806	0.0615495	0.0701912	0.076361	0.0700413

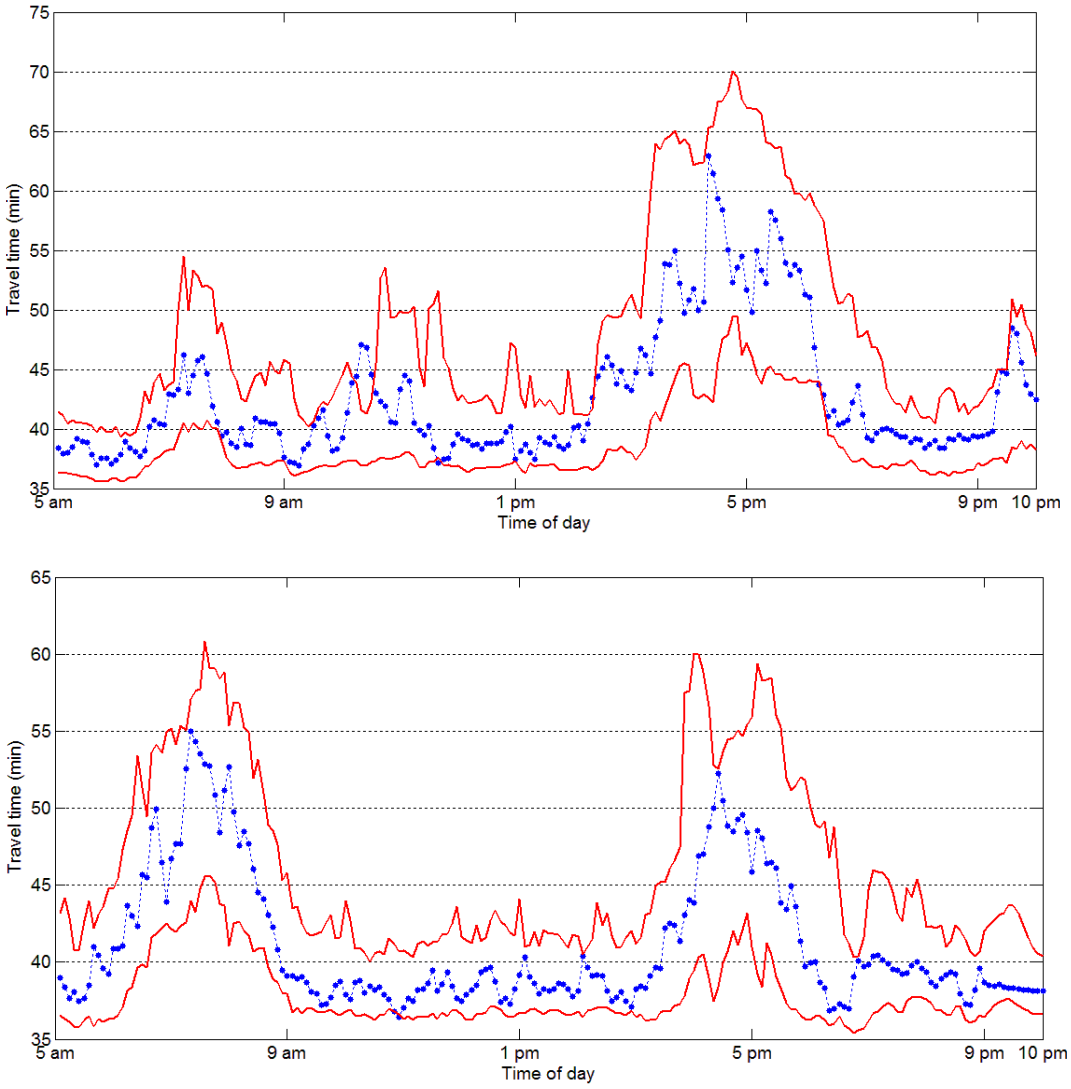


Figure 50: Temporal Variation in Travel Time Estimates Relative to Ground Truth.

Conclusion and Future Work

The research presented in this paper develops a random forest based algorithm for predicting dynamic travel times. The proposed algorithm uses the previously developed ASBIA algorithm to build a spatiotemporal probability of congestion matrix. Subsequently, the priori information derived from congestion matrix and the current spatiotemporal speed measurements are fused together. A random forest is used to build a model using this integrated input dataset. The proposed algorithm has the advantage of giving the travel time reliability without any extra processing. The results show the superior performance of the proposed algorithm when compared to the state-of-the-practice instantaneous algorithm and the GP algorithm. The proposed algorithm was tested on a 37-mile freeway section in Virginia. The prediction error was demonstrated to be significantly lower than that produced by both the instantaneous algorithm and the GP algorithm ($p < 0.0001$). Specifically, the proposed algorithm achieved more than a 38 percent reduction in the prediction error on congested days compared to the instantaneous algorithm and 28 percent reduction when compared to GP algorithm. Moreover,

the computed travel time confidence limits show that the mean width of the travel interval is less than 10 minutes for the 37-mile trip.

Acknowledgements

This research effort was jointly funded by the Mid-Atlantic University Transportation Center (MAUTC) and the Virginia Department of Transportation (VDOT).

References

[1] H. Tu, "Monitoring Travel Time Reliability on Freeways," Ph.D., Department of Transport and Planning, Technische Universiteit Delft, 2008.

[2] P.-E. Mazare, O.-P. Tossavainen, A. Bayen, and D. Work, "Trade-offs between Inductive Loops and GPS Vehicles for Travel Time Estimation: A Mobile Century Case Study," presented at the Transportation Research Board 91st Annual Meeting, Washington, D.C., 2012.

[3] H. Chen and H. A. Rakha, "Prediction of Dynamic Freeway Travel Times based on Vehicle Trajectory Construction," in *15th International IEEE Conference on Intelligent Transportation Systems*, 2012.

[4] X. Fei, C.-C. Lu, and K. Liu, "A Bayesian Dynamic Linear Model Approach for Real-time Short-term Freeway Travel Time Prediction," *Transportation Research Part C: Emerging Technologies*, vol. 19, pp. 1306-1318, 2011.

[5] J. Xia, M. Chen, and W. Huang, "A Multistep Corridor Travel-Time Prediction Method Using Presence-Type Vehicle Detector Data," *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, vol. 15, pp. 104-113, 2011.

[6] J. W. C. v. Lint, S. P. Hoogendoorn, and H. J. v. Zuylen, "Accurate Freeway Travel Time Prediction with State-space Neural Networks Under Missing Data," *Transportation Research Part C: Emerging Technologies*, vol. 13, pp. 347-369, 2005.

[7] C.-H. Wu, J.-M. Ho, and D. T. Lee, "Travel-time Prediction with Support Vector Regression," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, pp. 276-281, 2004.

[8] J. Myung, D.-K. Kim, S.-Y. Kho, and C.-H. Park, "Travel Time Prediction Using k Nearest Neighbor Method with Combined Data from Vehicle Detector System and Automatic Toll Collection System," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2256, pp. 51-59, 2011.

[9] W. Qiao, A. Haghani, and M. Hamedi, "Short Term Travel Time Prediction Considering the Weather Impact," presented at the Transportation Research Board 91st Annual Meeting, Washington D.C., 2012.

[10] M. Yildirimoglu and N. Geroliminis, "Experienced travel time prediction for congested freeways," *Transportation Research Part B: Methodological*, vol. 53, pp. 45-63, 2013.

[11] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5-32, 2001.

[12] G. Nimrod, A. Szilagyi, C. Leslie, and N. Ben-Tal, "Identification of DNA-binding Proteins Using Structural, Electrostatic and Evolutionary Features," *Journal of Molecular Biology*, vol. 387, pp. 1040-1053, Apr 10 2009.

[13] B. Slabbinck, B. De Baets, P. Dawyndt, and P. De Vos, "Towards large-scale FAME-based bacterial species identification using machine learning techniques," *Systematic and Applied Microbiology*, vol. 32, pp. 163-176, May 2009.

[14] A. Z. Kouzani, "Subcellular Localisation of Proteins in Fluorescent Microscope Images Using a Random Forest," *2008 Ieee International Joint Conference on Neural Networks, Vols 1-8*, pp. 3926-3932, 2008.

[15] G. Y. Zhang, H. C. Li, and B. S. Fang, "Discriminating acidic and alkaline enzymes using a random forest model with secondary structure amino acid composition," *Process Biochemistry*, vol. 44, pp. 654-660, Jun 2009.

[16] H. T. Chen, T. L. Liu, and C. S. Fuh, "Segmenting highly articulated video objects with weak-prior random forests," *Computer Vision - Eccv 2006, Pt 4, Proceedings*, vol. 3954, pp. 373-385, 2006.

[17] A. Z. Kouzani, "Faceparts for recognition," *Tencon 2006 - 2006 Ieee Region 10 Conference, Vols 1-4*, pp. 1232-1235, 2006.

[18] W. Z. Yan, "Application of random forest to aircraft engine fault diagnosis," *2006 IMACS: Multiconference on Computational Engineering in Systems Applications, Vols 1 and 2*, pp. 468-475, 2006.

[19] I. Koprinska, J. Poon, J. Clark, and J. Chan, "Learning to classify e-mail," *Information Sciences*, vol. 177, pp. 2167-2187, May 15 2007.

[20] J. A. Benediktsson, J. Chanussot, and M. Fauvel, "Multiple classifier systems in remote sensing: From basics to recent developments," *Multiple Classifier Systems, Proceedings*, vol. 4472, pp. 501-512, 2007.

[21] A. Zainal, M. A. Maarof, S. M. Shamsuddin, and A. Abraham, "Ensemble of One-class Classifiers for Network Intrusion Detection System," *Fourth International Symposium on Information Assurance and Security, Proceedings*, pp. 180-185, 2008.

[22] S. S. M. Ali, N. Joshi, B. George, and L. Vanajakshi, "Application of Random Forest Algorithm to Classify Vehicles Detected by a Multiple Inductive Loop System," *2012 15th International Ieee Conference on Intelligent Transportation Systems (Itsc)*, pp. 491-495, 2012.

[23] M.-H. Pham, A. Bhaskar, E. Chung, and A.-G. Dumont, "Random Forest Models for Identifying Motorway Rear-End Crash Risks Using Disaggregate Data," presented at the 13th International IEEE Annual Conference on Intelligent Transportation Systems, Madeira Island, Portugal, 2010.

[24] Qingchao Liu, Jian Lu, and S. Chen, "Traffic Incident D etection Using Random Forest," presented at the Transportation Research Board 92nd Annual Meeting, Washington DC.

[25] Mohammed Elhenawy, H. A. Rakha, and H. C. , "An Automated Statistically-principled Bottleneck Identification Algorithm (ASBIA)," presented at the IEEE Intelligent Transportation Systems Society Conference Management System, The Hague, The Netherlands, 2013.

[26] L. Breiman, *Classification and regression trees*. Belmont, Calif.: Wadsworth International Group, 1984.

[27] INRIX. (2012). *Traffic Information*. Available: http://www.inrix.com/trafficinformation.asp

[28] J. Van Lint, N. Van der Zijpp, and R. Rijweg, "An improved travel-time estimation algorithm using dual loop detectors," in *82nd Transportation Research Board Annual Meeting*, 2002.

[29] B. Efron and R. Tibshirani, "Improvements on cross-validation: The .632+ bootstrap method," *Journal of the American Statistical Association*, vol. 92, pp. 548-560, Jun 1997.

[30] M. Elhenawy, H. Chen, and H. Rakha, "Title," unpublished|.

[31] D. S. Moore, *The practice of business statistics : using data for decisions*, Comprehensive . ed. New York: W.H. Freeman, 2003.

Chapter 9: Speed and Travel Time Prediction Based on Partial Least Squares Regression

This chapter is based on
Mohammed Elhenawy and Hesham Rakha, "Speed and Travel Time Prediction Based on Partial Least Squares Regression," under review paper.

Abstract

Traffic congestion is a serious problem facing commuters every day. Traditional solutions are very expensive and have social and environmental impacts. However, advances in communications and computing can produce innovative solutions to the traffic congestion problem. One solution is to use a historical dataset of road segments to build models for predicting these segments' future status, such as speed. Predicting segments' speeds for two hours or more in the future helps predict congestion and travel time, helping commuters make better route choices. Broadcasting predicted congestion and travel time will help reduce the traffic demand on congested segments and relieve congestion quickly. This paper proposes using the partial least square regression to predict the speed and then predict the travel time of road segments by using historical datasets. The experimental results show a promising performance of the proposed algorithm.

Introduction

Most urban areas in the United States suffer from traffic congestion. Congestion carries a high economical cost. In 2011, 5.5 billion hours were wasted sitting in traffic in urban areas. An extra 2.9 billion gallons of fuel were consumed, which boosts the congestion cost to $121 billion. Congestion causes some environmental side effects because of the CO2 emissions produced during congested periods, which was estimated to be as much as 56 billion pounds in 2011 [1]. Addition of physical capacity to roadways has been the traditional solution to the congestion problem. This solution has become impractical because of limited budgets as well as environmental and social constraints. Consequently, highway agencies are seeking innovative solutions to overcome and ease the traffic congestion problem.

New, advanced sensing and wireless communication technologies enable continuous monitoring and dissemination of traffic information, which in turn enables more efficient management of the transportation system. The minimum that can be accomplished with this technology is to inform potential users of a road what the estimated travel time is or the travel time reliability for their trips. Using this information could help commuters choose alternative routes and departure times. This is the essence of Advanced Traveler Information Systems (ATISs), such as the 511 systems that have been implemented nationwide. In many states, Variable Message Signs (VMSs) are used to post relevant traffic information, which, with enough signs and strategic positioning, makes the dissemination of the travel time easy and cost effective. Providing accurate travel time predictions can assist travelers in making better decisions. In case of congestion, many road users may change their routes of travel based on displayed travel time information which release the congestion faster. In the near future with the wide use of connected cars, this kind of trip related information exchange will be routine procedure to relax and avoid congestion.

Traffic data have been collected using various traffic-sensing technologies, such as point-to-point travel time measurement systems and station-based, traffic-state-measuring devices.

License plate recognition systems, automatic vehicle identification systems, mobile devices, Bluetooth tracking systems, and probe vehicles are among the point-to-point systems, and loop detectors, video cameras, and remote traffic microwave sensors are station-based systems. The data collected using these technologies are used in several applications, including the computation of travel times. Private companies such as INRIX integrate different sources of measured data to provide section-based traffic state data (speed, average travel time), which were used in the present study to develop algorithms for predicting travel times. The benefit of using section-based traffic state data is that travel times can be easily calculated. More importantly, the section-based data provide the flexibility for scalable applications on traffic networks.

Dynamic and instantaneous travel time [2, 3] are the two classes of travel time prediction algorithms that use section-based traffic state data. Dynamic travel time reflects the actual, realized travel time that a vehicle experiences during a trip. Dynamic travel time algorithms consider the variation of speed over both space and time. Consequently, some algorithms indirectly predict travel time by first predicting the future speed patterns. Instantaneous travel time usually computes travel time using the current speed along the entire roadway; in other words, the speed distribution is assumed to remain unchanged for the whole time of the trip. As long as the road stretch is not congested, the speed variation with time is not remarkable, and both approaches provide comparable travel time estimates. However, this is not the case if the road stretch is congested. Instantaneous approaches may deviate substantially from the actual, experienced travel time under a trip's transient states, during which congestion is forming or dissipating [4].

Some research efforts have used macroscopic traffic modeling to predict short-term traffic states. The disadvantage of this approach is the expensive computations and the rapid degradation in the accuracy with the increase in the prediction temporal horizon [4, 5]. For long trips, traffic states may change remarkably, and the traffic state in the near future usually cannot provide enough information to cover the entire trip. For example, in the case of a 100-mile trip, if the driver departs at the time t_d, and the trip would take one hour or more depending on traffic conditions, then the traffic state for the following one hour or more would need to be predicted in order to compute dynamic travel times.

Partial least squares (PLS) models the relations between a set of observed variables using a set of latent variables. PLS assumes the observed data are generated by a process driven by a small number of latent variables [6]. PLS regression (PLSR) is an extension of PLS to the regression problem. It combines features from principal component analysis and multiple regression to predict the response matrix from the predictor matrix.

Recently, PLSR has been used to study many problems in the transportation engineering field. Wei present the development of freeway incident detection models based on PLSR [7]. A hybrid model that combines PLS and neural network (NN) was developed to detect traffic incidents automatically [8]. PLSR is used in the prediction of traffic incident duration, which is a very important issue to Advanced Traffic Incident Management (ATIM) [9]. The influences of roadways' attributes on vehicle speed is investigated by taking advantage of logarithmic regression and then applying PLSR for computing the modified coefficients of roadway attributes towards vehicle speed [10]. A PLSR-based fusion model has been proposed for modeling and predicting driver drowsiness [11].

The remainder of this paper is organized as follows. A description of the proposed algorithm is provided. This is followed by some methods used in this report. Then the paper presents a description of the test data used for the case study and the results of a comparison of

the proposed approach to PLSR. The last section provides the conclusions of the research and some recommendations for future research.

Partial Least Squares Regression

Multiple linear regression (MLR) is a good tool to model the relationship between predictors and responses. MLR is effective when the number of predictors is small, there is no significant multicollinearity, and there is a well-understood relation between predictors and responses [12]. In many scientific problems, the relation between the predictors and responses are poorly understood, and the main goal is to construct a good predictive model using large number of predictors. In this case, MLR is not a suitable tool. If the number of predictors gets too large, an MLR model will over-fit the sampled data perfectly but fail to predict new data well.

When the number of the observations is less than the number of predictors, the chance of multicollinearity increases and ordinary MLR fails. Several approaches have been proposed to overcome this problem. Principal component regression is used to remove the multicollinearity by projecting the X into orthogonal component and then regressing the X's scores on Y. These orthogonal components explain X but may not be relevant for Y.

PLSR is a recent technique that generalizes and combines features from principal component analysis and MLR. It is used to predict Y from X and to describe their common structure. PLSR assumes that there are few latent factors that account for most of the variation in the response. The general idea of PLSR is to try to extract those latent factors, accounting for as much of the predictors' X variation as possible and at the same time modeling the responses well.

PLSR finds two sets of weights, w and c, to create a linear combination of the columns of X and Y such that their covariance is maximized. Specifically, the goal is to obtain a first pair of vectors $t = Xw$ and $u = Yc$ with the constraints that $w^T w = 1$, $t^T t = 1$ and $t^T u$ is maximized. A number of variants of PLSR algorithms exist. Most of the algorithms estimate the coefficients of the linear regression between X and Y as $= X\tilde{\beta} + \widetilde{\beta_0}$. Some PLSR algorithms are designed for the case where Y is a column vector, while others are suitable for the general case, in which Y is a matrix. One of the PLSR algorithms is the simple NIPALS algorithm [6]:

1. Before starting the iteration process, the vector u is initialized at random, usually with a values from one of the Y columns.
2. Calculate the X-weights, $w = X^T u / u^T u$.
3. Calculate X-scores, $t = Xw$.
4. The Y-weights, $c = Y^T t / t^T t$.
5. Calculate the updated Y-scores, $u = Yc / c^T c$.

If $\frac{\|t_{old} - t_{new}\|}{\|t_{new}\|} > \varepsilon$, then t is not converged, and the user can proceed to Step 2. If t has converged, the user can compute the value of b, which is used to predict Y from t as $b = t^T u$ and compute the factor loadings for X as $p = X^T t$. Remove the present component from X and Y by partially removing (deflating) the effect of t from both X and Y as follows $X = X - tp^T$ and $Y = Y - btc^T$. The deflated matrices X and Y are used in the next component estimation. The vectors t, u, w, c, and p are then stored in the corresponding matrices, and the scalar b is stored as well as a diagonal element of matrix B. Continue with next component (back to step 1) until there is no more significant information in X about Y.

Methods

1. Travel-Time Ground Truth Calculation

The travel-time ground truth can be calculated based on trajectory construction and the known speed at each trajectory's cell. Figure 51 shows an example of the travel-time ground truth calculation using trajectory construction. The roadway is divided into four sections using segments of length Δx and a time interval of Δt. Within each cell, the speed is assumed to stay unchanged. Given the average speed of the red-dotted cell (i=2, n=3) in the figure is u(x2, t3) then the trajectory slope can be defined as the speed at this cell. Once the vehicle enters a new cell, the segment of the trajectory within this cell is the line that has slope equal to the cell speed and passes through the entrance point, as does the blue line in Figure 51. Finally, the ground truth travel time can be calculated when the trip reaches the downstream boundary of the last freeway section. It should be noted that the ground-truth travel times were computed using the same INRIX dataset.

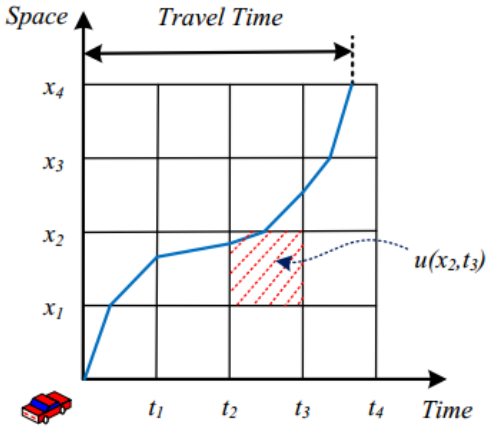


Figure 51: Illustration of Travel-Time Ground Truth Calculation [4].

2. Historical Average Method

The INRIX dataset can be used to calculate the historical average travel time by calculating the ground truth travel time at each time interval for each day. The historical average at any time t_0 is calculated using equation (1).

$$\text{Historical average travel time} = \sum_{i=1}^{Z} \frac{GTTT_i^{t_0}}{Z} \quad (1)$$

Where $GTTT_i^{t_0}$ is the ground truth travel time at departure time t_0 for historical day i and Z is the number of historical days included in the average. In other words, the historical average travel time at departure time t_0 and at the current day is the average of the ground-truth travel times at t_0 for the previous Z days.

The historical average was calculated considering different Z values ranging from 5 to 30 days, where Z is number of days included in the average shown in Equation (1). As shown in Figure 58, there is no significance impact of Z on algorithm performance.

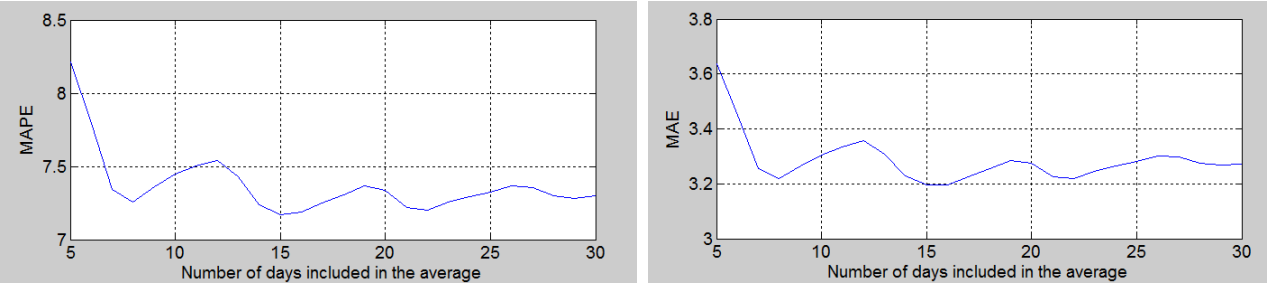


Figure 52: Mean Absolute Percentage Error (MAPE) and Mean Absolute Error (MAE) for the Historical Average at Different Number of Days Included in the Average.

Applying the PLSR algorithm to predict speed and travel time

The travel time prediction problem could be stated as given the road segments' speeds at the current time t_0 and back $(L-1)$ time interval, then the goal is to predict the travel time for a commuter (traveler) who is leaving at time t_1. To predict the travel time, first the speeds of the future H time intervals are predicted starting from t_1, then the travel trajectory is constructed and the travel time is estimated. To estimate the speeds, PLSR was applied in two different ways.

The first was to build a model at each interval of time, in this case every 5 minutes. This produced different model matrices at every time interval, and the total number of the model matrices was 24x12. This approach assumes the model matrices are time dependent. At time interval t_i, to find the model matrices it is necessary to define the predictors matrix X^{t_i} and the response matrix Y^{t_i} as follows:

1. Each row m in the predictors' matrix X^{t_i} is a reshaped window of the spatiotemporal speed matrix between time intervals $t_i - L + 1$ and t_i of day n in the training set.
2. Each row m in the response matrix Y^{t_i} is a reshaped window of the spatiotemporal speed matrix between time intervals $t_i + 1$ and $t_i + H$ of day n in the training set.

The second approach assumes that each day of the week has its own model matrices. Seven models, one model for each day, had to be built. This means the matrices are independent of the time of the day and dependent on the day of the week. To build the predictors matrix X^S and the response matrix Y^S, the training dataset was divided into seven partitions S={1,2,3,4,5,6,7}. Each partition consists of the spatiotemporal speed matrices for one day of the week. Assume the number of days within each partition is m^S. In this case, the predictors matrix X^S and the response matrix Y^S are defined as follows:

3. Each row in the predictors matrix X^S is a reshaped window of the spatiotemporal speed matrix between time intervals $t - L + 1$ and t of each single day ϵ S.
4. Each row in the response matrix Y^S is a reshaping window of the spatiotemporal speed matrix between time intervals $t + 1$ and $t + H$ of each single day ϵ S, where t is the time interval that is varied to cover the time of the day between 5:00 a.m. to 10:00 p.m.

In both approaches, after constructing the predictors' matrix X and the response's matrix Y, the PLSR was applied to get the model matrices. The model matrices can be used to predict the response for unseen (new) predictors as shown in Equation (2):

$$\hat{Y} = X^{unseen} B_{PLS} \text{ where } B_{PLS} = (P^{T+})BC^T \quad (2)$$

In Equation (2), P^{T+} is the Moore-Penrose inverse of P^{T} [13], and X and Y are assumed to be standardized.

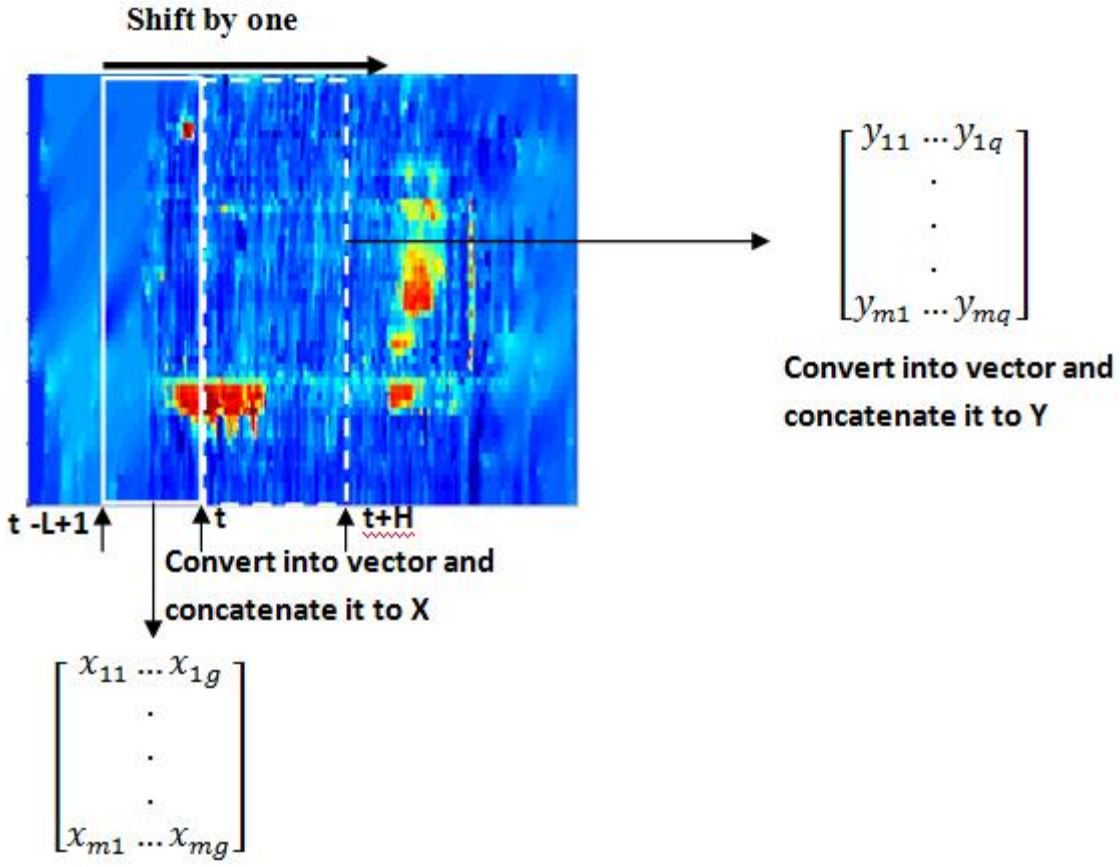


Figure 53: Illustration of the Preparation of the (X, Y) Inputs to PLSR for the Second Approach Using Day Number m.

Case Study

The performance of the proposed algorithm was tested on a 37-mile freeway section. This section first describes the test data and then shows the experimental result of the proposed speed and travel-time prediction based on PLSR.

1. Data Description

The case study is conducted using privately developed INRIX traffic data. INRIX is mainly probe data collected using GPS-equipped vehicles. For the sake of data quality, the collected probe data are supplemented with data collected using different technologies such as road sensors and mobile devices [14]. The data are organized as a spatiotemporal speed matrix for each day, and each matrix cell is the average speed of a roadway segment aggregated at 5-minute intervals.

INRIX assures high-quality data by conducting a continuous quality monitoring and improvement process. The details of the five steps' quality procedure are described in detail in the literature [15]. Moreover, the quality of INRIX data was investigated and shown to be good for travel time prediction [16]. Finally, it should be noted that the ground-truth travel times are computed using trajectory construction using the same INRIX data.

The main segments of I-64 and I-264 from June to August 2010 were used to construct the travel database [14, 17, 18]. The 37-mi stretch goes from Newport News to Virginia Beach and includes most of the congested areas heading toward Virginia Beach from Richmond. It is divided into 58 sections as shown in Figure 54. The 58 sections have different lengths averaging 0.65 mi. The longest section is a 3.7-mile length located at the Hampton Roads Bridge-Tunnel (HRBT). Major congestion usually forms upstream of the HRBT, so this freeway section included several congested locations with various backward-forming shockwaves upstream of these bottlenecks.

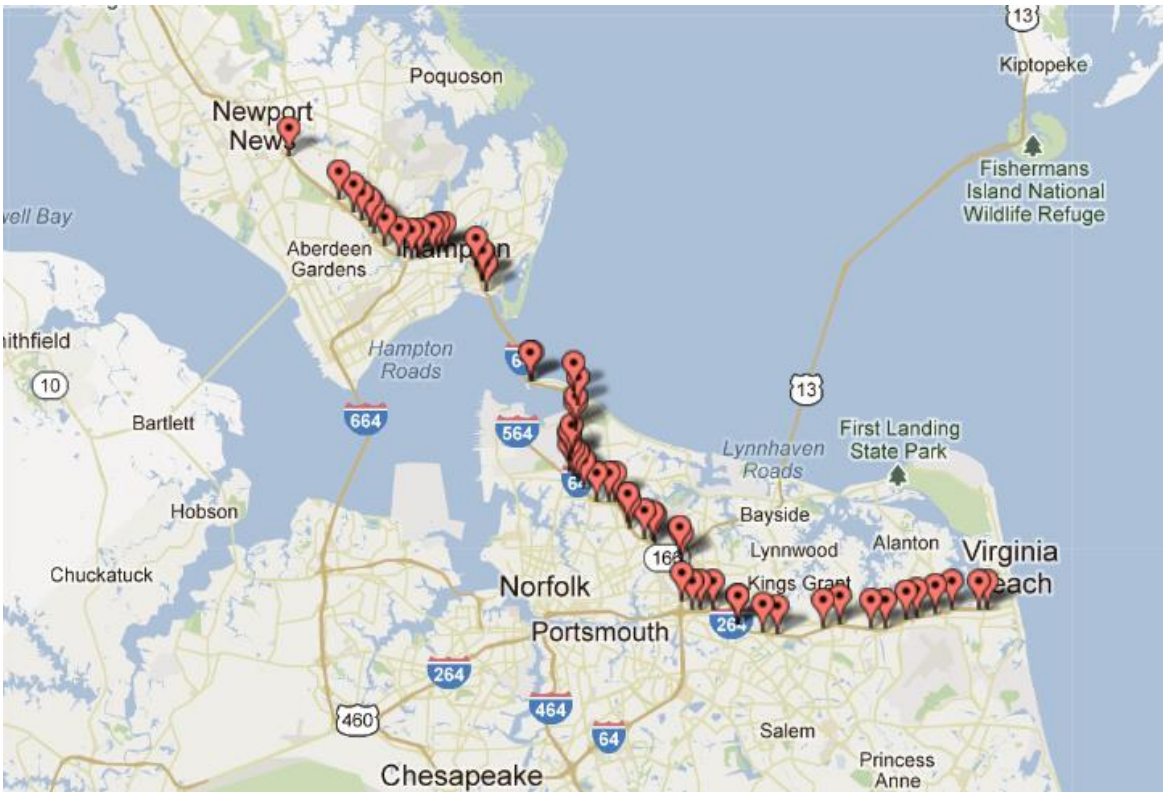


Figure 54: The 37-mile Test Site.

It should be noted that the traffic flow moves north. Pre-processing of the data showed that weekends have more missing data than weekdays. Missing data affects the prediction of both speed and travel time. Consequently, the missing data needs be estimated because it cannot simply be imputed with the free-flow speed. There are many available imputation algorithms [19]. In the present research, any missing cell's data were statistically imputed by using the average temporal and spatial values of neighboring cells.

2. Speed Prediction Experimental Results

To evaluate the two proposed approaches to predicting speed, the dataset described above was used. It was divided into a training set consisting of two months, June and July, and a testing set consisting of one month, August. The proposed algorithm has two parameters: L, which is the duration in the past that was used to construct the model, and H, which is the duration in the future (prediction horizon). Optimizing L and H is a challenging task that required running the algorithm several times at different values of L and H. The values of L and H were changed from 10 to 30 with step size 2. The mean absolute error (MAE) of the speed, which is shown in

Equation (3), was calculated for each combination of (L,H) parameters. This error is the mean of the absolute difference between the predicted and the true values of speeds.

$$\text{MAE of the speed} = \frac{1}{58xIxHxJ}\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{h=1}^{H}\left|v_{ih}^{j} - \widehat{v_{ih}^{J}}\right| \quad (3)$$

Where

J = total number of days in the testing dataset (i.e., 31 days),

I = total number of time intervals in a single day,

v = ground truth,

$\hat{v}$ = predicted value of speed, and

58 = number of road segments.

The MAE of the speed across various values of L and H for both approaches are shown in Figure 55 and Figure 56.

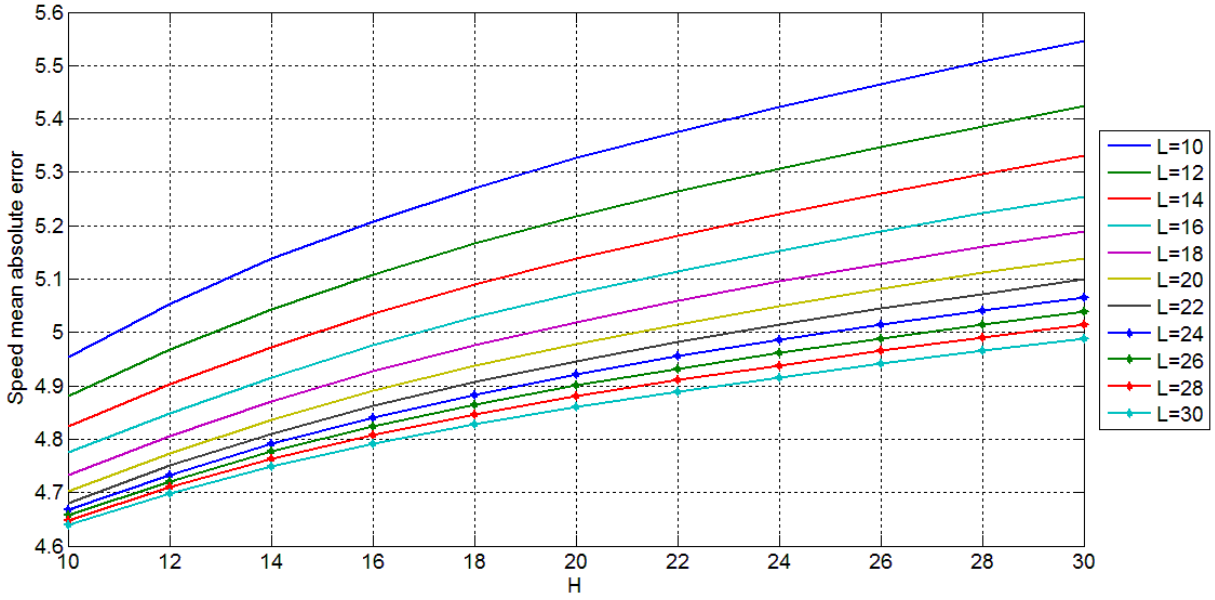


Figure 55: The Speed MAE at Different Values of L and H Using the First Approach.

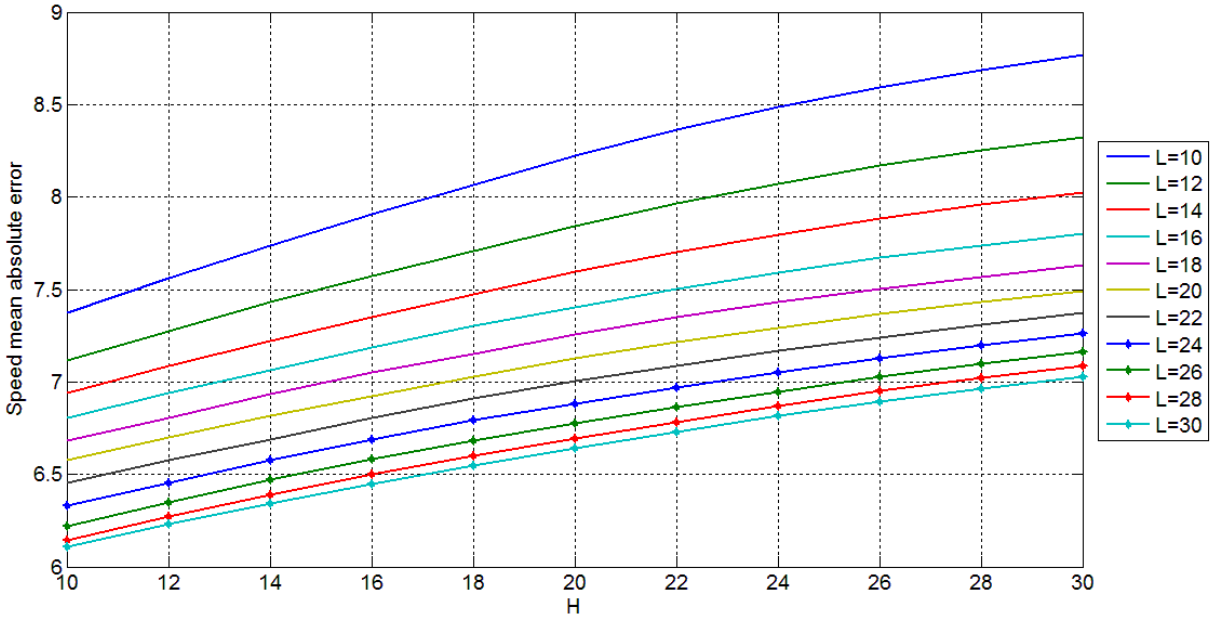


Figure 56: The speed MAE at Different Values of L and H Using the Second Approach.

As shown in Figure 55 and Figure 56, the first approach of applying PLSR, which assumes the model matrices are time dependent, gives better results in terms of speed MAE. In both approaches, error in speed prediction is increased as the prediction horizon (H) increases and decreases as the duration in the past (L) increases.

3. Travel Time Prediction Experimental Result

Travel time is the second quantity of interest for evaluating the performance of the proposed algorithm. The superior speed predictions from the first approach were used. The errors of estimated travel time of the proposed algorithm at different values of L from 10 to 40 and at constant H = 20 are shown in Table 22. The errors of estimated travel time using the state of practice (historical average) ,the pattern-matching recognition [20], and the traditional K nearest neighbors (K-NN) [21, 22] are shown in the same table as well. Relative and absolute prediction errors were calculated for the proposed algorithm. The relative error was computed as the Mean Absolute Percentage Error (MAPE) using Equation (4). This error is the average absolute percentage change between the predicted and the true values. The corresponding absolute error is presented by the MAE using Equation (5). This error is the absolute difference between the predicted and the true values.

$$\text{MAPE} = \frac{100}{I \times J} \sum_{j=1}^{J} \sum_{i=1}^{I} \frac{\left| y_i^j - \hat{y}_i^j \right|}{y_i^j} \quad (4)$$

$$\text{MAE} = \frac{1}{I \times J} \sum_{j=1}^{J} \sum_{i=1}^{I} \left| y_i^j - \hat{y}_i^j \right| \quad (5)$$

Where

J = total number of days in the testing dataset in each fold (i.e., 31 days),

I = total number of time intervals in a single day,

y = ground truth, and

$\hat{y}$ = predicted value of the dynamic travel time for the i^{th} time interval on the j^{th} day.

The MAE of the speed across various values of L and constant H are shown in Table 22.

Table 22: MAE and MAPE at Different Values of L (Varied from 10 to 40 at Increments of 2) and H = 20.

L	MAE	MAPE
10	2.4087	5.2029
12	2.3871	5.1596
14	2.3761	5.1323
16	2.3617	5.0989
18	2.3574	5.0866
20	2.3546	5.0766
22	2.3650	5.0898
24	2.3703	5.0963
26	2.3725	5.1013
28	2.3639	5.0851
30	2.3677	5.0937
32	2.3978	5.1532
34	2.4237	5.2047
36	2.4444	5.2480
38	2.4553	5.2684
40	2.4590	5.2753
Pattern recognition [20]	2.96	5.96
K-NN method [21, 22]	3.47	6.59
Historical average	3.26	7.34

Table 22 shows that the PLSR algorithm gives the best travel time estimation at any L when compared with the other prediction algorithms. Moreover, the PLSR at L = 20 gives the best travel time prediction in terms of MAE and MAPE.

Conclusion

This paper proposes a new algorithm for predicting the speed across road segments and then calculating the travel time using the predicted speeds. The algorithm is based on PLSR. Two approaches to applying PLSR to speed and travel time prediction were proposed. The first approach assumes the model matrices are time dependent, and the second approach assumes the model matrices are day dependent and time independent. The experimental results show that the first approach gives better result in terms of speed prediction error. The predicted travel time of the proposed algorithm was compared with predicted travel time using pattern recognition, K-NN, and historical average methods. The proposed algorithm gives better predicted travel time in terms of smaller MAE and MAPE. Although the PLSR gives a better result, building the models takes time because PLSR is computationally expensive. Future work should develop a new algorithm that is computationally efficient. This algorithm would transform the multivariate predictors and the corresponding multivariate responses into two different spaces in the model training phase so that the projections of predictors (scores) and responses are equal and the prediction is a simple matrix multiplication.

References

[1] D. Schrank, B. Eisele, and T. Lomax, "2012 Urban Mobility Report," Texas Transportation Institute2012.

[2] P.-E. Mazare, O.-P. Tossavainen, A. Bayen, and D. Work, "Trade-offs between Inductive Loops and GPS Vehicles for Travel Time Estimation: A Mobile Century Case Study,"

presented at the Transportation Research Board 91st Annual Meeting, Washington, D.C., 2012.

[3] H. Tu, "Monitoring Travel Time Reliability on Freeways,," Ph.D., Department of Transport and Planning, Technische Universiteit Delft, 2008.

[4] H. Chen, H. A. Rakha, S. A. Sadek, and B. J. Katz, "A Particle Filter Approach for Real-time Freeway Traffic State Prediction," in *91st Transportation Research Board AnnualMeeting*, Washington D.C., 2012.

[5] H. Chen, H. A. Rakha, and S. A. Sadek, "Real-time Freeway Traffic State Prediction: A Particle Filter Approach," in *14th International IEEE Conference on Intelligent Transportation Systems*, Washington, DC, USA, 2011, pp. 626-631.

[6] S. Wold, A. Ruhe, H. Wold, and I. Dunn, W., "The Collinearity Problem in Linear Regression. The Partial Least Squares (PLS) Approach to Generalized Inverses," *SIAM Journal on Scientific and Statistical Computing*, vol. 5, pp. 735-743, 1984.

[7] W. Wang, S. Chen, and G. Qu, "Incident detection algorithm based on partial least squares regression," *Transportation Research Part C: Emerging Technologies*, vol. 16, pp. 54-70, 2// 2008.

[8] J. Lu, S. Chen, W. Wang, and H. van Zuylen, "A hybrid model of partial least squares and neural network for traffic incident detection," *Expert Systems with Applications*, vol. 39, pp. 4775-4784, 4// 2012.

[9] X. Wang, S. Chen, and W. Zheng, "Traffic Incident Duration Prediction based on Partial Least Squares Regression," *Procedia - Social and Behavioral Sciences*, vol. 96, pp. 425-432, 11/6/ 2013.

[10] L. Weiguo, Z. Hanjie, D. Xiaoping, Q. Kun, and L. Cuiying, "Data analysis of roadway attributes through Partial Least Squares regression," in *Information and Financial Engineering (ICIFE), 2010 2nd IEEE International Conference on*, 2010, pp. 466-468.

[11] S. Hong and Z. Gangtie, "A Partial Least Squares Regression-Based Fusion Model for Predicting the Trend in Drowsiness," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 38, pp. 1085-1092, 2008.

[12] H. Abdi, "Partial least square regression (PLS regression)," ed.

[13] K. Schmidt, "Computing the Moore-Penrose Inverse of a Matrix with a Computer Algebra System," *International Journal of Mathematical Education in Science and Technology*, vol. 39, pp. 557-562, 2008.

[14] INRIX. (2012). *Traffic Information*. Available: http://www.inrix.com/trafficinformation.asp

[15] T. Trepanier. INRIX Data Services - Arterial System Performance Assessment and Management [Online]. Available: http://www.mtc.ca.gov/services/arterial_operations/downloads/6-3-13/5_INRIX_Data_Service_for_Arterials-2013June03.pdf

[16] H. Rakha, H. Chen, A. Haghani, and K. F. Sadabadi, "Assessment of Data Quality Needs for use in Transportation Applications " 2013.

[17] M. Elhenawy, H. Chen, and H. A. Rakha, "Dynamic travel time prediction using data clustering and genetic programming," *Transportation Research Part C: Emerging Technologies*, vol. 42, pp. 82-98, 5// 2014.

[18] H. Chen and H. A. Rakha, "Real-time travel time prediction using particle filtering with a non-explicit state-transition model," *Transportation Research Part C: Emerging Technologies*, vol. 43, Part 1, pp. 112-126, 6// 2014.

[19] G. B. Durrant, "Imputation methods for handling item nonresponse in the social sciences: A methodological review," ESRC National Centre for Research Methods, University of Southampton2005.
[20] Hao Chen, Hesham A. Rakha, and C. C. McGhee, "Dynamic Travel Time Prediction using Pattern Recognition," presented at the 20th ITS World Congress 2013.
[21] B. Bustillos and Y.-C. Chiu, "Real-Time Freeway-Experienced Travel Time Prediction Using N -Curve and k Nearest Neighbor Methods," *Transportation Research Record: Journal of the Transportation Research Board,* vol. 2243, pp. 127-137, 12/01/ 2011.
[22] W. Qiao, A. Haghani, and M. Hamedi, "Short-Term Travel Time Prediction Considering the Effects of Weather," *Transportation Research Record: Journal of the Transportation Research Board,* vol. 2308, pp. 61-72, 12/01/ 2012.

Chapter 10: A Matrix Projection Approach for Predicting Freeway State Evolution and Dynamic Travel Times

This chapter is based on
Mohammed Elhenawy and Hesham Rakha, "A Matrix Projection Approach for Predicting Freeway State Evolution and Dynamic Travel Times," presented at the Transportation Research Board 94th Annual Meeting, Washington DC, United States, 2015.

Abstract

An important measure of the transportation systems' performance is travel time. Dissemination of predicted travel times may help roadway travelers make better travel decisions. Moreover, travel time prediction is critical to other intelligent transportation system applications, such as dynamic route guidance, congestion management, and incident detection systems. In this paper, a novel algorithm that predicts dynamic travel times is developed. This algorithm transforms the multivariate predictors and the corresponding multivariate responses into two different spaces in the model training phase, such that, the projections of predictors (scores) and responses are equal. The proposed algorithm is tested using archived data along a congested 37-mile freeway stretch from Newport News to Virginia Beach along I-64 and I-264. The prediction results demonstrate that the proposed method produces predictions that are more accurate than the state-of-the-art K-Nearest Neighbor and a pattern recognition algorithm that was developed in an earlier publication with significantly reduced computational times.

Introduction

Congestion has proven to be a serious problem in most urban areas in the United States. For example, in 2011, congestion resulted in urban Americans spending 5.5 billion hours more in traveling and cost them an extra 2.9 billion gallons of fuel, resulting in a congestion cost of $121billion. Congestion also results in environmental side effects because of the CO2 produced during congested periods, which was estimated to be as much as 56 billion pounds in 2011 [1]. Adding capacity has been the traditional solution in addressing the congestion problem, but this has become impractical given the financial, environmental, and societal constraints. Consequently, highway agencies are seeking new solutions to overcome recurrent and non-recurrent congestion problems.

Thanks to advanced new technologies that enable continuous monitoring and dissemination of traffic information, it is possible to manage the transportation system more efficiently. The minimum that can be accomplished is to inform the potential users of a road what they can expect during their trip. Such information helps travelers compare alternative routes and make better routing and departure time decisions. This is the essence of Advanced Traveler Information Systems (ATISs), such as the 511 systems that has been implemented nationwide. In many states relevant traffic information is also posted on Variable Message Signs (VMSs) that are strategically positioned along highways. Because the infrastructure is already available, we can assist travelers in making better decisions by providing accurate travel time predictions. In case of congestion, many road users may change their routes of travel based on displayed travel time information.

Traffic data have been collected using various traffic-sensing technologies, such as point-to-point travel time measurement systems and station-based traffic-state-measuring devices. License plate recognition systems, automatic vehicle identification systems, mobile devices,

Bluetooth tracking systems, and probe vehicles are among the point-to-point systems while loop detectors, video cameras, and remote traffic microwave sensors are station-based systems. The data collected using these technologies are used in several applications, including computing travel times. Private companies such as INRIX integrate different sources of measured data to provide section-based traffic state data (speed, average travel time), which are used in our study to develop algorithms for predicting travel times. The benefit of using section-based traffic state data is that travel times can be easily calculated. More importantly, the section-based data provides the flexibility for scalable applications on traffic networks.

Dynamic and instantaneous travel time [2, 3] are the two classes of travel time prediction algorithms that use section-based traffic state data. Dynamic travel time reflects the actual, realized travel time that a vehicle experiences during a trip. Dynamic travel time algorithms consider the variation of speed over both space and time. Consequently, some algorithms indirectly predict travel time by first predicting the future speed patterns. Instantaneous travel time usually computes travel time using the current speed along the entire roadway; in other words, the speed spatiotemporal evolution is assumed to remain unchanged for the entire trip duration. As long as the road stretch is not congested the spatiotemporal speed variation is typically minimal and both approaches provide comparable travel time estimates. However, this is not the case if the road stretch is congested. instantaneous approaches may deviate substantially from the actual, experienced travel time under transient states during which congestion is forming or dissipating during a trip [4].

All the above sensing technologies collect traffic data in real-time which reflects past or current conditions of the different road segments. Because travel time prediction requires the future conditions of the road segments, it is complex and difficult to apply. Consequently, to provide travel time information to travelers, we need to predict the future conditions on the roadway. We can avoid the required road condition prediction, if the instantaneous travel times are used. We have to highlight that the instantaneous approach results in considerable errors when traffic conditions are varying in time and space.

During the last decade, travel time prediction algorithms have emerged as an active research area, and thus have attracted the attention of many researchers. For example, artificial neural networks are used as a non-linear predictor for short-term travel time forecasting in [5, 6]. State-space neural networks (SSNN) are used for freeway travel time prediction demonstrating accurate and robust travel time predictions especially with missing or corrupt data [7].In addition, many linear models have been developed. Linear regression using the stepwise-variable-selection method and advanced tree-based methods have been used to predict travel times on freeways using flow and occupancy data from single-loop detectors and historical travel-time information [8]. A straightforward and computationally efficient algorithm that uses a linear model with coefficients that vary as smooth functions of the departure time was proposed in [9, 10]. This algorithm models travel time as a function of the instantaneous travel time and the average historical travel time. Because of the data used to predict travel times were non-stationary, an Autoregressive Integrated Moving Average (ARIMA) model was developed and showed good accuracy when used to predict arterial short-term travel times [11]. The rapid development of Machine learning algorithms has encouraged researchers to actively apply these algorithms to travel time prediction. For example, the support vector machine (SVM) method was applied to time series forecasting to develop the support vector regression (SVR) approach. SVR is used in[12] to predict travel times using a set of SVR parameters. Context-dependent Random Forests (RFs) were used to predict future travel times on certain road segments given

simulated GPS transmissions of cars across a city in [13]. In [14] a data clustering and genetic programming approach was used to model and predict the expected, lower, and upper bounds of dynamic travel times along freeways.

The remainder of this paper is organized as follows. A description of the methods used in this paper is provided. This is followed by a description of the proposed algorithm. Subsequently, a description of the test data used for the case study and the results of the proposed algorithm are presented. The last section provides the conclusions of the research and some recommendations for future research.

Methods

This section describes the methods used in conducting the research presented in this paper.

1. Travel Time Ground Truth Calculation

The travel time ground truth can be calculated by reconstructing vehicle trajectories using the known speed in each cell. Figure 57 shows an example of the travel time ground truth calculation using trajectory construction. As shown in the illustrative example, the roadway is divided into four segments of length Δx and discretizing over a time interval of Δt. Within each cell, the speed is assumed to remain constant. The average speed of the red-dotted cell (i=2, n=3) is denoted as u(x2, t3). Consequently, the speed in each cell can be defined as the trajectory slope. Once the vehicle enters a cell, the segment of the trajectory within this cell is a straight line given that the speed is assumed to be constant in each cell, as illustrated in Figure 57. Finally, the ground truth travel time can be calculated when the trip reaches the downstream boundary of the last freeway section. It should be noted that the ground truth travel times were computed using the same INRIX dataset.

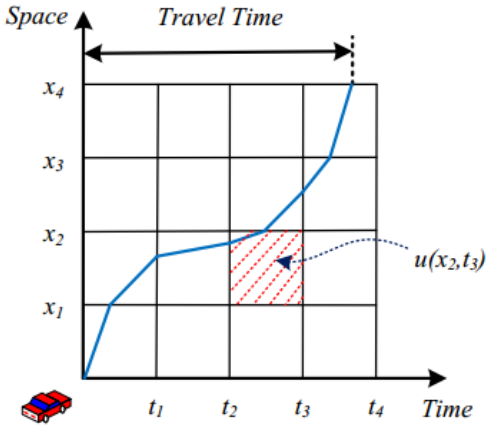


Figure 57: Illustration of Travel Time Ground Truth Calculation [4]

2. Historical Average Method

We can use the INRIX dataset to compute the historical average travel time by calculating the ground truth travel time at each time interval for each day. The historical average at any time t0 is calculated using Equation (1).

$$\text{historical average travel time} = \sum_{i=1}^{Z} \frac{GTTT_i^{t0}}{Z} \quad (1)$$

Where $GTTT_i^{t0}$ is the ground truth travel time at departure time t0 for historical day i and Z is the number of historical days included in the average. In other words, the historical average travel

time at departure time t0 and current day is the average of the ground truth travel times at t0 for the previous Z days.

The historical average was calculated considering different Z values ranging from 5 to 30 days, where Z is number of days included in the average. As shown in Figure 58, there is no significant impact of z on the algorithm performance after a Z value of 8 days. It, should be noted that the mean absolute error (MAE) and mean absolute percent error (MAPE) will be presented later in the paper.

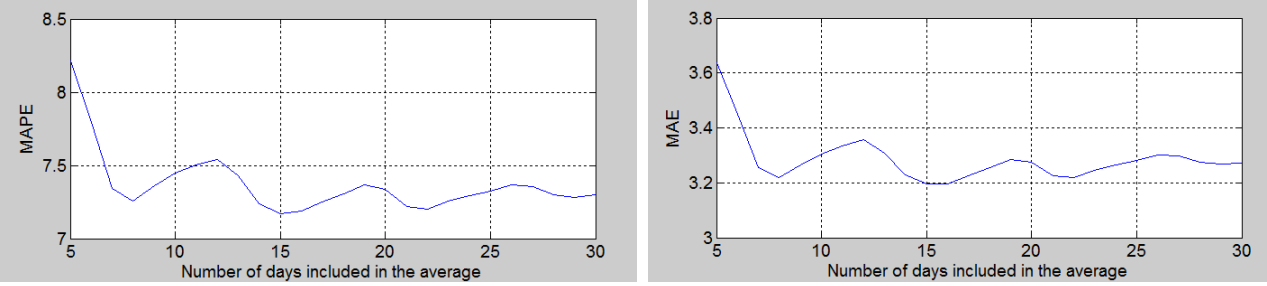


Figure 58: MAPE and MAE for the Historical Average at Different Number of Days Included in The Average.

Proposed Algorithm

Assuming that we have the matrix of predictors $X_{m\times n}$ and the matrix of responses $Y_{m\times p}$where m is the number of cases, n is the number of predictors and p is the number of responses. The proposed algorithm finds two spaces for the predictors and responses, respectively. The main idea of the proposed algorithm is finding the two spaces such that the scores of predictors and the scores of the responses are equal. The algorithm starts by factorizing the predictor and response matrices, as shown in Equations (2) and (3).

$$X_{m\times n} = A_{m\times k}B_{k\times n} \quad (2)$$

$$Y_{m\times p} = C_{m\times k}D_{k\times p} \quad (3)$$

Where $A_{m\times k}$and $C_{m\times k}$ are the score matrices and $B_{k\times n}$and $D_{k\times n}$ are the loading matrices. The factorization should be done such that $A_{m\times k} = C_{m\times k}$. In order to satisfy this condition, the proposed algorithm uses any of the well-known factorization techniques to factorize the predictor matrix. Then based on the above condition, we can write the response matrix as shown in Equation (4).

$$Y_{m\times p} = A_{m\times k}D_{k\times p} \quad (4)$$

In Equations (2) and (3), the $Y_{m\times p}$ and $A_{m\times k}$matrices are known and can be obtained from the loading of the second space, as shown in Equation (5).

$$D_{k\times p} = (A^{T}_{m\times k}A_{m\times k})^{-1}A^{T}_{m\times k}Y_{m\times p} \quad (5)$$

The $B_{k\times n}$ and $D_{k\times p}$ matrices represent the loading matrices of the required two spaces $\mathbb{R}$ and $\mathbb{N}$. $B_{k\times n}$ and $D_{k\times p}$ are the models that will be used to predict responses for an unseen predictor matrix $X^{(new)}_{l\times n}$. The prediction of the response for $X^{(new)}_{lxn}$is done in two steps. The first step, is solving Equation (6) to find the scores of the $X^{(new)}_{l\times n}$, which is shown in Equation (7).

$$X^{(new)}_{l\times n} = A_{l\times k}B_{k\times n} \quad (6)$$

$$A^{(new)}_{l\times k} = X^{(new)}_{l\times n}B^{T}_{n\times k}(B_{k\times n}B^{T}_{n\times k})^{-1} \quad (7)$$

Because we choose the loading matrices such that the scores of both the predictors and responses are equal, the second step is done by multiplying the $A_{l\times k}^{(new)}$ by the $D_{k\times p}$ to obtain $Y_{l\times p}^{(predicted)}$, as shown in Equation (8).

$$Y_{l\times p}^{(predicted)} = A_{l\times k}^{(new)} D_{k\times p} \quad (8)$$

In the following section we describe the application of the above algorithm to predict the speeds of a road segment and the travel time.

1. Estimating Travel Times

The sensing technologies collect speed data in real-time which reflects past or current conditions of the different road segments. In other words, the segment speeds along the roadway at the current time t_0 and back (L-1) time intervals are available. Given the speed data, our objective is to predict the travel time. Numerous algorithms that predict the travel time directly without predicting the segment speeds by training the model using the true travel time as a response variable. The approach entails first predicting the speeds of the future H time intervals, then using these predicted speeds to construct the travel trajectories to compute the travel times. In order to predict the speeds we build a model for each time interval. Since, the day is divided into 5-minute intervals, a set of B and D matrices are needed for each time interval. At time interval t_i, to find the two matrices $B^{t_i}$and D^{t_i}we need to define the predictor matrix X^{t_i} and the response matrix Y^{t_i}, as follows:

1. Each row m in the predictor matrix X^{t_i} is a reshaped window of the spatiotemporal speed matrix between time intervals $(t_i - L + 1)$ and (t_i) for day m in the training set, where $m \in \{1,2,\dots,M\}$ and M is the total number of days in the training dataset.
2. Each row m in the response matrix Y^{t_i} is a reshaped window of the spatiotemporal speed matrix between time intervals $(t_i + 1)$ and $(t_i + H)$ on day m in the training set.
3. The final step after defining the predictor and response matrices at time t_i entails calculating $B^{t_i}$and D^{t_i} for the pair (X^{t_i}, Y^{t_i}) as described in the above section. B^{t_i} and D^{t_i} are the models used to predict speed and travel time at time t_i.

Case Study

The performance of the proposed algorithm was tested on a 37-mile freeway section. A description of the test data is first presented and then followed by a comparison of the proposed approach to two state-of-the-practice algorithms.

1. Data Description

The case study is conducted using privately developed INRIX traffic data. INRIX data is mainly probe data collected using GPS-equipped vehicles. For the sake of data quality, the collected probe data are supplemented with data collected using different technologies such as road sensor and mobile devices [15]. The data are summarized in a spatiotemporal speed matrix for each day. The value in each matrix cell reflects the average speed of a roadway segment aggregated at 5-minute intervals. INRIX assures high quality data by conducting continuous data quality monitoring. The details of the five-step data quality procedure are provided in the literature [16]. Moreover, the quality of the INRIX data was evaluated in an earlier study and shown to be sufficient for travel time prediction [17]. Finally, it should be noted that the ground truth travel

times were computed using the trajectory construction procedure that was presented earlier on the same INRIX data.

The road stretch on the main segments along I-64 and I-264 from June to August 2010 were used to construct the travel database [15]. The selected freeway stretch has 37 miles in length and included most of the congested sections from travel from Richmond towards Virginia Beach. Specifically, the selected freeway stretch ran from Newport News to Virginia Beach including at total of 58 sections, as shown in Figure 59. The 58 sections have different lengths with and average length equal to 0.65 miles in length. The longest section was 3.7 miles in length, which is located at the Hampton Roads Bridge-Tunnel (HRBT). Typically major congestion forms upstream of the HRBT and thus this freeway section includes several congested locations with various backward forming shockwaves upstream of these bottlenecks.

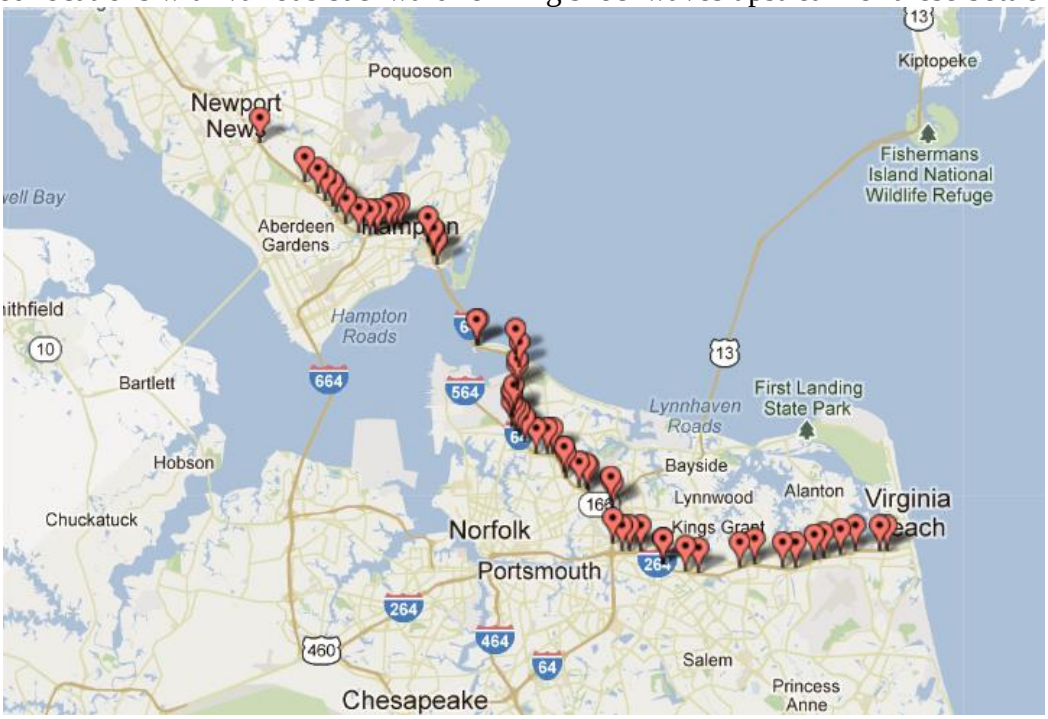


Figure 59: 37-mile Test Site

While pre-processing the data, the weekends were identified to have more missing data than weekdays. Missing data affects the prediction of both speed and travel time. Consequently, the missing data were imputed and cannot simply impute it with free-flow speeds. There are many available imputation algorithms and the interested reader can find details in [18]. In this paper we used a statistical approach for data imputation that used the temporal and spatial contiguous cells of the missing cell to estimate the speed of the missing cell. The average value of eight neighboring cell speeds was used to impute the missing cell data in this study.

2. Experimental Results

The evaluation of the prediction algorithm focuses on travel times from June to August of 2010 [15] during the time period between 5:00 a.m. to 10:00 p.m. In this experimental work, we divided the dataset into two parts one for training and the other for testing. The training set consisted of two months (June and July) and the testing set was only one month in duration (August). The singular value decomposition was used to factorize the predictor matrix, where the

predictor matrix is decomposed using Equation (9) such that the product of S and V^T was the loading matrix and U was the scoring matrix.

$$X = USV^T \quad (9)$$

The proposed algorithm uses two parameters, namely: L, which is the temporal look back duration that is used to construct the model, and H, which is the prediction horizon. Optimizing these two parameters is a challenging task. In doing so, the algorithm was several times using different values for the L and H parameters. The mean absolute error (MAE) of the speed, computed as shown in Equation (10), was calculated for each set of parameters. This error is computed as the mean of absolute difference between the predicted and the true values of speeds.

$$MAE = \frac{1}{58xIxHxJ}\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{h=1}^{H}\left|v_{ih}^{j} - \widehat{v_{ih}^{J}}\right| \quad \ldots(10)$$

Here J is the total number of days in the testing dataset in each fold (i.e., 31 days); I is the total number of time intervals in a single day; v and $\hat{v}$ denote the ground truth and the predicted value of speed, respectively, and 58 is the number of road segments. The speed MAE for various values of L and H are shown in Table 13 The L parameter was varied from 10 to 30 at increments of 2, and the H parameter was varied from 10 to 30 at increments of 2.

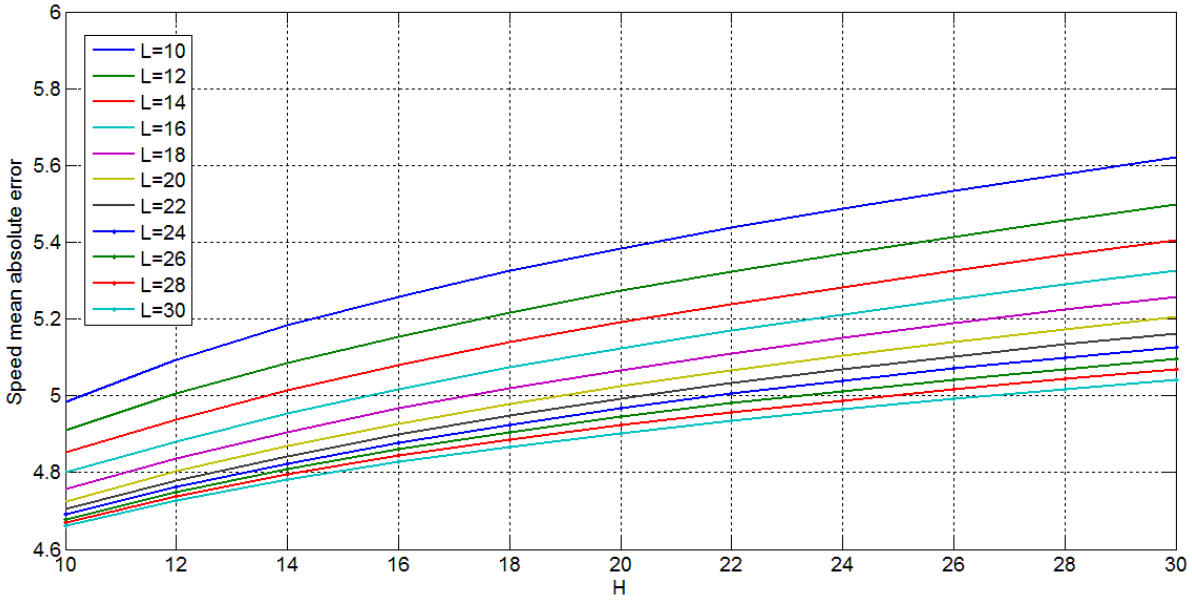


Figure 60: The Proposed Algorithm MAE for Different Values of L and H

As shown in Figure 60 the speed MAE increases as H increases and decreases as L increases. In order to obtain better insights into the performance of the proposed algorithm we use the predicted speed for different L values ranging from 10 to 30 at increments of 2 and H equal to 20 to predict the travel time. The relative and absolute travel time prediction errors are calculated for each set of parameters. The relative error is computed as the Mean Absolute Percentage Error (MAPE) using Equation (10). This error is the average absolute percentage change between the predicted and the true values. The corresponding absolute error is presented using the Mean Absolute Error (MAE) using Equation (11). The definitions of I and J are the same as stated earlier and y and $\hat{y}$ denote the ground truth and the predicted value of the dynamic travel time for the i^{th} time interval on the j^{th}day.

$$\text{MAPE} = \frac{100}{IxJ}\sum_{j=1}^{J}\sum_{i=1}^{I}\frac{\left|y_i^j - \widehat{y_i^j}\right|}{y_i^j} \quad (10)$$

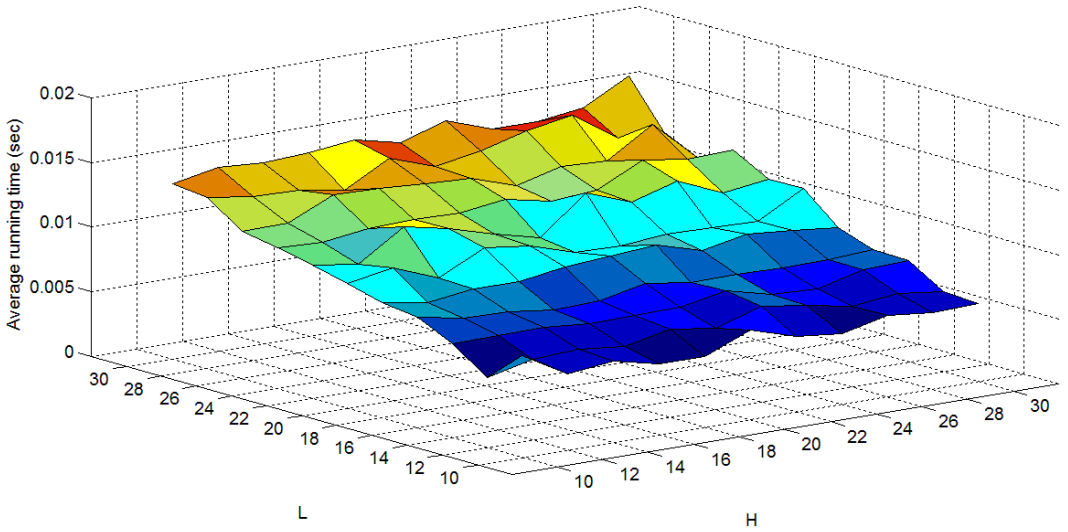
$$\text{MAE} = \frac{1}{IxJ}\sum_{j=1}^{J}\sum_{i=1}^{I}\left|y_i^j - \widehat{y_i^j}\right| \quad (11)$$

Table 23 shows the MAE and MAPE when using the proposed algorithm and compares it to the lowest MAE and MAPE of the K-Nearest Neighbor method [19] , a pattern recognition method that was developed earlier by the authors [20] and PLSR. The data in Table 23 shows that the proposed algorithm is comparable with the PLSR and produces lower error than the other two methods.

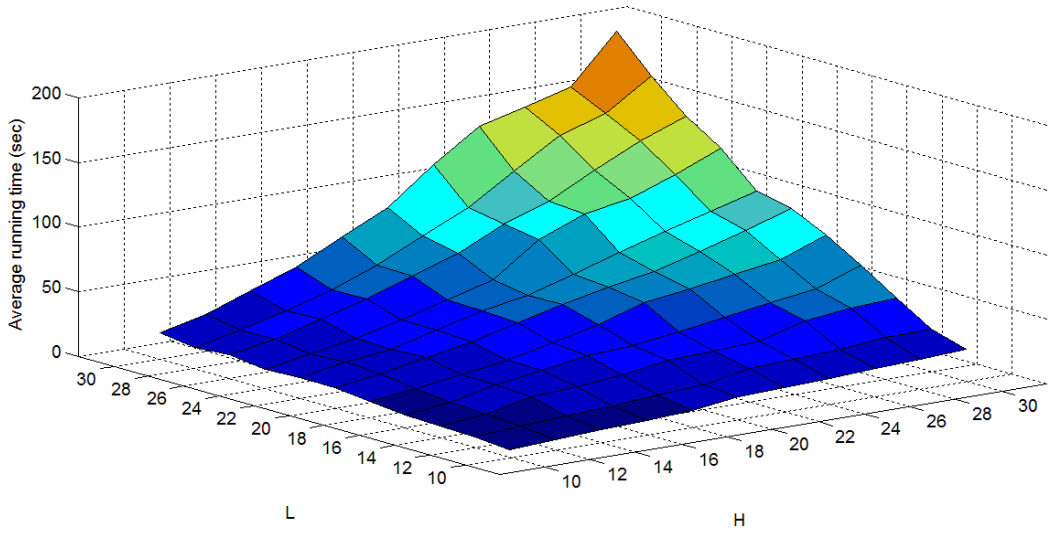
Table 23: MAE and MAPE for Different Values of L for H=20 for both PLSR and the Proposed Algorithm.

	The proposed algorithm		PLSR	
L	MAE	MAPE	MAE	MAPE
10	2.44	5.30	2.41	5.20
12	2.40	5.22	2.39	5.16
14	2.37	5.14	2.38	5.13
16	2.36	5.11	2.36	5.10
18	2.35	5.10	2.36	5.09
20	2.36	5.10	2.35	5.08
22	2.37	5.11	2.37	5.09
24	2.37	5.12	2.37	5.10
26	2.37	5.12	2.37	5.10
28	2.37	5.12	2.36	5.09
30	2.37	5.13	2.37	5.09
32	2.41	5.19	2.40	5.15
34	2.43	5.24	2.42	5.20
36	2.46	5.30	2.44	5.25
38	2.47	5.33	2.46	5.27
40	2.48	5.33	2.46	5.28
K-Nearest Neighbor	3.47	6.59	3.47	6.59
Pattern Recognition [20]	2.96	5.96	2.96	5.96

Because our focus is to propose a computationally efficient algorithm which is suitable for real-time applications, we report here the running time needed to compute the model matrices for both PLSR and the proposed algorithm. We ran the algorithms using a laptop equipped with an Intel (R) core (TM) i7-2670 QM CPU @ 2.20 Hz 2.20 Hz and 8.00 GB RAM. The size of X and Y matrices varied as a function of the L and H parameters and calculate the model matrices. The above experiment was repeated 20 times and the average for each L and H parameter combination is shown in the top panel of Figure 61. The figure shows that the required time to compute the model matrices of the proposed algorithm is very small even for large L and H values. In the case of the large parameters values, the computation time is almost 0.015 sec. The bottom panel of Figure 61 shows the running time for the PLSR. We found our algorithm is 4509 faster than PLSR in the case of the large parameters values. The results demonstrate that the proposed algorithm is very efficient computationally because it is a non-iterative algorithm and the model is obtained through matrix multiplications only.



(a)



(b)

Figure 61: Average Run Time of (a) Proposed Algorithm and (b) PLSR

Conclusions

The research presented in this paper developed a novel algorithm to predict the spatiotemporal evolution of roadway speeds and then computes dynamic travel times using the predicted speeds. The algorithm finds two spaces with different loadings that produce identical scores for the predictor and response matrices. The proposed algorithm was compared to the K-Nearest Neighbor and a pattern recognition algorithm developed earlier. The results demonstrate that the proposed algorithm outperforms both approaches. The proposed algorithm is efficient computationally because it is a not iterative algorithm. Moreover, the model building and the prediction involve simple matrix multiplications and thus does not require any time consuming pattern matching as done in other pattern matching techniques.

Acknowledgements

This research effort was co-funded by the Mid-Atlantic University Transportation Center (MAUTC) and the Connected Vehicle Initiative University Transportation Center (CVI-UTC).

References

[1] D. Schrank, B. Eisele, and T. Lomax, "2012 Urban Mobility Report," Texas Transportation Institute2012.

[2] P.-E. Mazare, O.-P. Tossavainen, A. Bayen, and D. Work, "Trade-offs between Inductive Loops and GPS Vehicles for Travel Time Estimation: A Mobile Century Case Study," presented at the Transportation Research Board 91st Annual Meeting, Washington, D.C., 2012.

[3] H. Tu, "Monitoring Travel Time Reliability on Freeways,," Ph.D., Department of Transport and Planning, Technische Universiteit Delft, 2008.

[4] H. Chen, H. A. Rakha, S. A. Sadek, and B. J. Katz, "A Particle Filter Approach for Real-time Freeway Traffic State Prediction," in *91st Transportation Research Board AnnualMeeting*, Washington D.C., 2012.

[5] D. Park and L. Rilett, "Forecasting Multiple-Period Freeway Link Travel Times Using Modular Neural Networks," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1617, pp. 163-170, 01/01/ 1998.

[6] D. Park, L. Rilett, and G. Han, "Spectral Basis Neural Networks for Real-Time Travel Time Forecasting," *Journal of Transportation Engineering*, vol. 125, pp. 515-523, 1999.

[7] J. W. C. van Lint, S. P. Hoogendoorn, and H. J. van Zuylen, "Accurate freeway travel time prediction with state-space neural networks under missing data," *Transportation Research Part C: Emerging Technologies*, vol. 13, pp. 347-369, 10// 2005.

[8] J. Kwon, B. Coifman, and P. Bickel, "Day-to-Day Travel-Time Trends and Travel-Time Prediction from Loop-Detector Data," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1717, pp. 120-129, 01/01/ 2000.

[9] X. Zhang and J. A. Rice, "Short-term travel time prediction," *Transportation Research Part C: Emerging Technologies*, vol. 11, pp. 187-210, 6// 2003.

[10] J. Rice and E. van Zwet, "A simple and effective method for predicting travel times on freeways," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 5, pp. 200-207, 2004.

[11] D. Billings and Y. Jiann-Shiou, "Application of the ARIMA Models to Urban Roadway Travel Time Prediction - A Case Study," in *Systems, Man and Cybernetics, 2006. SMC '06. IEEE International Conference on*, 2006, pp. 2529-2534.

[12] W. Chun-Hsin, H. Jan-Ming, and D. T. Lee, "Travel-time prediction with support vector regression," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 5, pp. 276-281, 2004.

[13] B. Hamner, "Predicting Travel Times with Context-Dependent Random Forests by Modeling Local and Aggregate Traffic Flow," in *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*, 2010, pp. 1357-1359.

[14] M. Elhenawy, H. Chen, and H. A. Rakha, "Dynamic travel time prediction using data clustering and genetic programming," *Transportation Research Part C: Emerging Technologies*, vol. 42, pp. 82-98, 5// 2014.

[15] INRIX. (2012). *Traffic Information*. Available: http://www.inrix.com/trafficinformation.asp

[16] T. Trepanier. INRIX Data Services - Arterial System Performance Assessment and Management [Online]. Available: http://www.mtc.ca.gov/services/arterial_operations/downloads/6-3-13/5_INRIX_Data_Service_for_Arterials-2013June03.pdf

[17] H. Rakha, H. Chen, A. Haghani, and K. F. Sadabadi, "Assessment of Data Quality Needs for use in Transportation Applications " 2013.

[18] G. B. Durrant, "Imputation methods for handling item nonresponse in the social sciences: A methodological review," ESRC National Centre for Research Methods, University of Southampton2005.

[19] J. Myung, D.-K. Kim, S.-Y. Kho, and C.-H. Park, "Travel Time Prediction Using k Nearest Neighbor Method with Combined Data from Vehicle Detector System and Automatic Toll Collection System," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2256, pp. 51-59, 12/01/ 2011.

[20] Hao Chen, Hesham A. Rakha, and C. C. McGhee, "Dynamic Travel Time Prediction using Pattern Recognition," presented at the 20th ITS World Congress 2013.

Chapter 11: Travel Time Modeling Using Spatiotemporal Speed Variation and Mixture of Linear Regression

This chapter is based on
Mohammed Elhenawy, Abdallah Hassan, and Hesham Rakha, "Travel Time Modeling Using Spatiotemporal Speed Variation and Mixture of Linear Regression," under review paper.

Abstract

Real-time and accurate travel time prediction algorithms are needed for individual travelers, business sectors, and government agencies. It helps commuters make better travel decisions, avoid traffic congestion, save the environment by reducing carbon emissions, and improve traffic operation efficiency. Travel time prediction attracted more attention with the rapid development and deployment of intelligent transportation systems (ITSs). It is considered one of the important elements required for the successful deployment of ITS subsystems, However it is not an easy task because of the stochastic nature of the travel time. The ITS application can use these travel time model to predict travel time and travel time reliability. In this paper, we propose modeling the travel time using a mixture of linear regression. The proposed model consists of two normal components. One component is used to model congested regime while the other is used to model free-flow regime. The means of the two components are modeled by two linear regression equations. The predictors used in the linear regression equation are selected out of the spatiotemporal Speed matrix using the random forest machine learning algorithm. The proposed model is tested using archived data along a 74.4-mile freeway stretch of I-66 eastbound to connect I-81 and Washington D.C. The experimental results show the ability of the model to capture the stochastic nature of the travel time and gives good travel time predictions.

Introduction

One of the main objectives of the Intelligent Transportation Systems ITS is to minimize travel times for vehicles from their origins to their destinations. This goal is extremely challenging due to the dynamic nature of the traffic flow that is in most cases highly unpredictable. A straightforward strategy would be to direct vehicles or guide drivers to follow the routes that avoid congested paths. A critical step for route planning or guidance to be effective is to be able to accurately predict the travel time between different alternative routes that can be followed from the source point to the destination point.

Besides that, the travel time is in itself an important performance measure for traffic system evaluation. It is easily understood by drivers and operators of traffic management systems. It can be viewed as a simple summary of the complex behavior of a traffic system. In order for the travel time to be successfully and accurately predicted, the ITS is required to be capable of:

- Sensing and acquiring the current state of the transportation network of interest, where several data values need to be sensed and collected such as the traffic conditions and parameters at different parts of the network and if some roads are currently congested, the current weather condition, the time of the day, whether there are any incident on any road in the network, etc. It may be fairly expensive for the ITS to gather these data values on all roads and intersection with the quality and quantity that can allow the accurate forecasting of travel time between two points in the network.

- Storing a long history of traffic parameters for the transportation network of interest that can support the future prediction of travel times, this historical dataset may end up into a very large dataset that is difficult to use and manage.
- Feeding the current state of the network beside its traffic history to some model that predicts the travel time if a trip is to start from some point to another in the network at some specific time. It is very challenging to design such a model, and to select a set of current or historical parameters of prediction power. The most useful model may be road dependent, and even for a single road, it has been shown that two or more models may describe the traffic behavior accurately at different traffic conditions, for instance a model is more useful when the road is congested, another model is accurate when vehicles are flowing freely, etc.

In short, the accurate traffic time forecasting or prediction is challenging due to the high cost of sensing and collecting enough useful current and historical traffic data. Even when such data is available, it is still challenging and tricky to figure out which type of model best describes the traffic behavior, and which traffic data should be fed to the model for best predictions. Moreover, we may decide to use two or more models and switch between them depending on the current traffic conditions, this adds a new challenge as we need to decide which model from the set of models we shall use for some specific input data, or we may use different models for prediction and apply some weight to each output prediction to reach a final travel time prediction.

In this paper, a new method for travel time prediction is proposed. The proposed method uses a mixture of linear regressions. This is motivated by the fact that the travel time distribution is not unimodal, since two modes or regimes of traffic can exist, one at congestion state, and the other at free-flow state. The proposed model is built and tested using probe data provided by INRIX and supplemented with traditional road sensor data, as well as mobile devices and other sources. The dataset is collected from the freeway stretch of I-66 eastbound to connect I-81 and Washington D.C. The traffic on this stretch is often extremely heavy. This makes the problem of travel time prediction more challenging, but also means that the data is valuable and can help build a realistic model.

The remainder of this paper is organized as follows. A review of the related work is provided. Subsequently, the proposed model based on a mixture of linear regression is presented. This is followed by a description of the test data used for the case study and the results of the travel time prediction and reliability. The last section provides the conclusions of the research and some recommendations for future research.

Related Work

Various methods and algorithms have been proposed in the literature for the travel time prediction problem. These methods can roughly be classified into two main categories: statistical-based data-driven methods and simulation-based methods. This section will focus on the statistical-based methods since the proposed solution in this paper fall under this class of methods, and since more research effort can be found in the literature using statistical methods.

Several researchers fit different regression models to predict travel time. A typical approach is to fit an MLR (Multiple Linear Regression) model using explanatory variables representing instantaneous traffic state and historical traffic data, for example [1-3]. Notice that the model proposed in [1] is even an SLR but was successful in providing acceptable travel time predictions. Some researchers developed hybrid methods where a regression model is used in

conjunction with other advanced statistical methods, for example [4] used regression with statistical tree methods. Another approach was followed in [5] where an SLR model used bus travel time to predict automobile travel time.

Generally speaking, regression models are powerful in predicting travel time especially for short-term prediction, long-term predictions are less accurate. It is also claimed that regression models are more suitable for use free-flow traffic than congested traffic and fails to predict with incidents [6].

The idea of using a mixture of linear regressions for different traffic regimes has also previously been explored[7]. This paper tries to overcome the drawbacks of previous work that used mixture models of two or three components to model travel time reliability. The previously proposed mixture models suffer from the following limitations:

1. The mean of each component is not modeled as a function of the available predictors.
2. Proportion variable is fixed at each time slot, which limits the model flexibility.
3. Provided information given the time slot of the day is the probability of each component (fixed) and the 90% percentile.

Another class of statistical-based methods in literature uses time series models for travel time prediction. For example the use of auto-regressive prediction models [8-11], the use of multivariate time series models [12], and the use of auto-regressive integrated moving average (ARIMA) technique [13]. Similar to regression models, the use time series model is more suitable for free-flow traffic than for congested traffic, may fail with unusual incidents, and is more accurate for short-term predictions [6].

Another common technique used for travel time modeling is the use of artificial neural networks. For example, a feed-forward neural network is used in [14] to predict journey time. Later more researchers used advanced neural network technique to model travel time and to predict it, for example [15-21]. The accuracy of predictions are high for most proposed models, for example in [22] the prediction error was about 4% only.

Methods

In this section, a brief introduction to the modeling techniques used in this paper is presented to familiarize readers with these powerful techniques. The random forest (RF) is used to select a subset of important predictor for travel time modeling. The expectation-maximization (EM) is used to fit the mixture of linear regression model to the historical data. The used techniques are among a variety of machine learning and statistical learning techniques that demonstrates the wide variety of algorithms that can be used by transportation practitioners.

1. Variables (Predictors) Selection[23]

The I66 stretch consists of 64 segments. The dataset consist of the spatiotemporal speed matrices for every day in 2013. The default approach for modeling and predicting travel time is taking all the speeds within a window. This window started right before the departure time t_0 and cover L time back until $t_0 - L$. If we used this approach and set L=30 minutes, the number of predictors will be 64*6 at 5 minutes time aggregation. In order to reduce the dimension of the predictors' vector, we use RF to select the most important predictors to include in the travel time model. The steps to select the most important predictors are as follows:

1. For each month, we build an RF consists of 100 trees and find the out of bag samples that are not used in the training for each tree.
2. Find the mean square error of the RF using out of bag samples and denote it as $\mathrm{MSE}_{\text{out of bag}}$.

3. For each predictor x_i we randomly permutes its values between the out of bag samples and calculate the mean square error of the RF and denote it $\mathrm{MSE}_{\text{out of bag}}^{\text{permuted } x_i}$.
4. Then we rank the predictors in descending order based on the $\frac{1}{12}\sum_{\text{month}=1}^{12}\left(\mathrm{MSE}_{\text{out of bag}}^{\text{permuted } x_i} - \mathrm{MSE}_{\text{out of bag}}\right)$ and choose the top m ranked predictors.

Based on the above approach the higher the value of the predictor importance the more important is this predictor. As shown in Figure 62 almost most of the important predictors are segments' speeds at time $t_0 - 5$. We also noted these segments have a high probability of getting congested during peak hours.

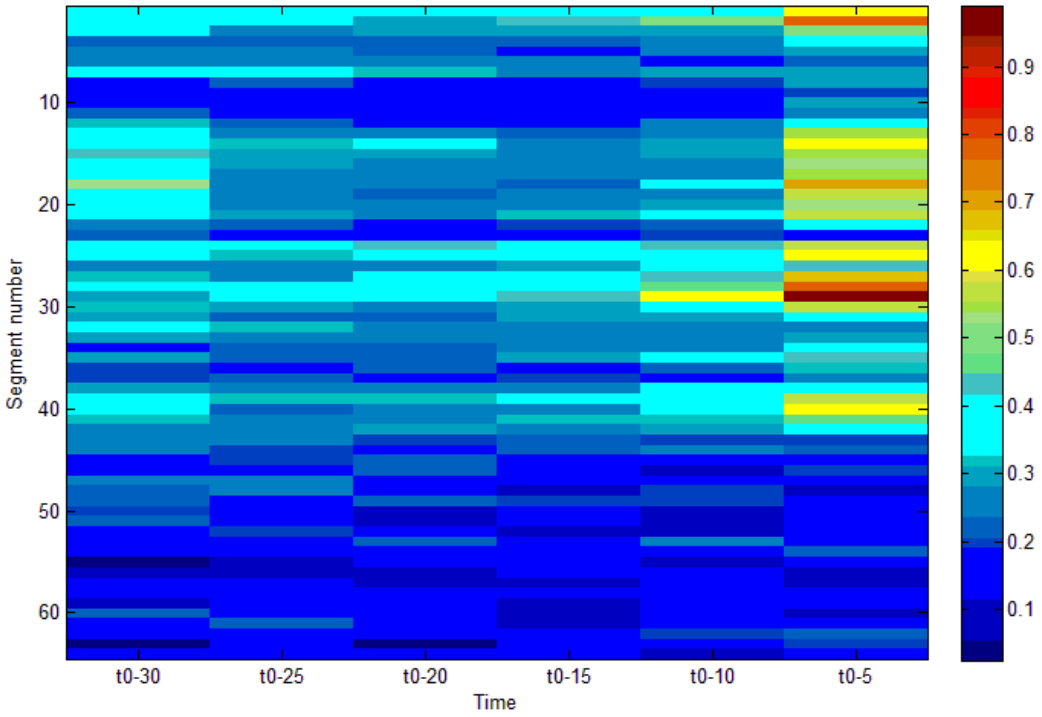


Figure 62: The Variable Importance Based on RF

In addition to the speed predictors chosen by the RF, we add another predictor. This predictor is historical average travel time at t_0 given the day of the week. As shown in Figure 63 we calculated the historical average for each day of the week. The historical average figure shows two travel time peaks at the weekdays and only one evening peak at Saturdays and almost no peaks on Sundays.

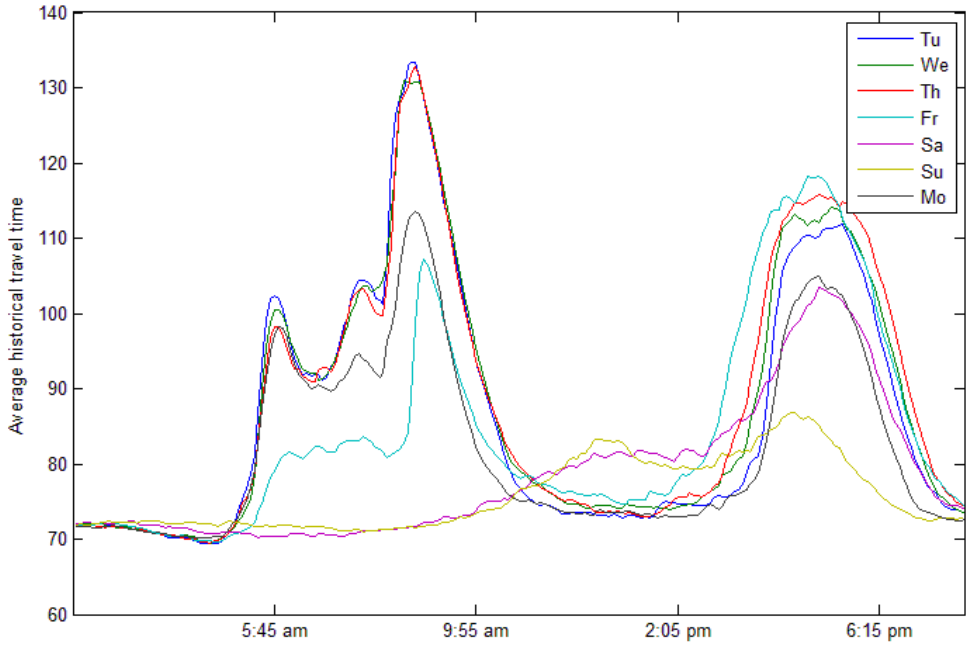


Figure 63: Historical Travel Time Average for the Different Days of the Week

2. Mixture of Linear Regressions[24, 25]

Finite mixture models are powerful tools for modeling a wide variety of random phenomena. It is used to model random phenomena in many fields of statistical applications such as agriculture, biology, economics, medicine, and genetics. A Mixture of linear regressions is one of the mixture family that is studied carefully and can be used to model the travel time under different traffic regimes. The mixture of linear regression can written as

$$f(y|X) = \sum_{j=1}^{m} \frac{\lambda_j}{\sigma_j\sqrt{2\pi}} e^{-\frac{\left(y-x^T\beta_j\right)^2}{2\sigma_j^2}} \quad (1)$$

or as

$$y_i = \begin{cases} x_i^T\beta_1 + \epsilon_{i1} & \text{with probability } \lambda_1 \\ x_i^T\beta_2 + \epsilon_{i2} & \text{with probability } \lambda_2 \\ \vdots & \\ x_i^T\beta_m + \epsilon_{im} & \text{with probability } 1 - \sum_{q=1}^{m-1}\lambda_q \end{cases} \quad (2)$$

where y_i is the response corresponding to a vector p of predictors x_i^T ,β_j is the vector of regression coefficients for the j^{th} component, λ_j is mixing probability of the j^{th} component and ϵ_{ij} are normal random errors. The model parameters $\psi=\{\beta_1, \beta_2, \ldots, \beta_m, \sigma_1^2, \sigma_2^2, \ldots, \sigma_m^2, \lambda_1, \lambda_2, \ldots, \lambda_m\}$ can be estimated by maximizing the log-likelihood of equation (1) given a set of response predictors pairs $(y_1, x_1), (y_2, x_2), \ldots, (y_n, x_n)$ using Expectation-Maximization algorithm (EM).

The EM algorithm iteratively finds the maximum likelihood estimates by alternating the E-step and M-step. let $\psi^{(k)}$ the parameters estimates after the k^{th} iteration. In the E-step, the

posterior probability of the i^{th} observation comes from component j is computed using equation (3).

$$w_{ij}^{(k+1)} = \frac{\lambda_j^{(k)} \phi_j\left(y_i \middle| x_i, \psi^{(k)}\right)}{\sum_{j=1}^{m} \lambda_j^{(k)} \phi_j\left(y_i \middle| x_i, \psi^{(k)}\right)} \quad (3)$$

where $\phi_j\left(y_i \middle| x_i, \psi^{(k)}\right)$ is the probability density function of the j^{th} component.

In the M-step, the new parameters' estimates $\psi^{(k+1)}$ that maximizes log-likelihood function in equation (1) is calculated using equations (4-6)

$$\lambda_j^{(k+1)} = \frac{\sum_{i=1}^{n} w_{ij}^{(k+1)}}{n} \quad (4)$$

$$\hat{\beta}_j^{(k+1)} = (X^T W_j X)^{-1} X^T W_j Y \quad (5)$$

where X is the predictors matrix which has n rows and $(p + 1)$ columns, Y is the corresponding nx1 response vector , and W is nxn diagonal matrix which has $w_{ij}^{(k+1)}$ on its diagonal.

$$\hat{\sigma}_j^{2(k+1)} = \frac{\sum_{i=1}^{n} w_{ij}^{(k+1)} (y_i - x_i^T \hat{\beta}_j^{(k+1)})^2}{\sum_{i=1}^{n} w_{ij}^{(k+1)}} \quad (6)$$

the E-step and M-step are alternated repeatedly until the change in the incomplete log-likelihood is arbitrary small as shown in equation (7)

$$\left| \prod_{i=1}^{n} \sum_{j=1}^{m} \lambda_j^{(k+1)} \phi_j\left(y_i \middle| x_i, \psi^{(k+1)}\right) - \prod_{i=1}^{n} \sum_{j=1}^{m} \lambda_j^{(k)} \phi_j\left(y_i \middle| x_i, \psi^{(k)}\right) \right| < \xi \quad (7)$$

where ξ is a small number.

3. Travel Time Ground Truth Calculation

The calculation of the travel time ground truth is based on trajectory construction and the known speed through the trajectory's cells. A simple example of travel time ground truth calculation based on trajectory construction is demonstrated in Figure 64. In this example, the roadway is divided into four sections using segments of length Δx and a time interval of Δt. We assumed that the speed is homogenous within each cell. The average speed of the red-dotted cell (i=2, n=3) in the figure is $u(x2, t3)$. Consequently, the trajectory slope represents the speed in each cell. Once the vehicle enters a new cell, the trajectory within this cell can be drawn as the straight blue line in Figure 64 using the cell speed as the slope. Finally, the ground truth travel time can be calculated when the trip reaches the downstream boundary of the last freeway section. It should be noted that the ground truth travel times were computed using the same dataset and used as the response (y).

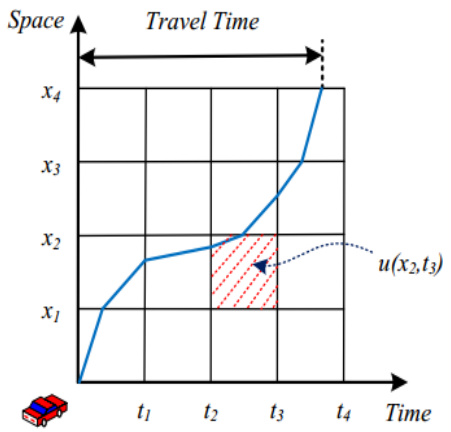


Figure 64: Illustration of Travel Time Ground Truth Calculation [26]

Data Description

The freeway stretch of I-66 eastbound to connect I-81 and Washington D.C. is selected as the test site for this study. High traffic volumes are usually observed during morning and afternoon peak hours on I-66 heading towards Washington D.C., therefore the study site provides a great environment to test the proposed travel time model.

In this study, the traffic data are provided by INRIX, which mainly collects probe data by GPS-equipped vehicles and supplemented with traditional road sensor data, as well as mobile devices and other sources [27]. The probe data on the test site covers 64 freeway segments with a total length of 74.4 miles. The average segment length is 1.16 miles long, and the length of each segment is unevenly divided in the raw data from 0.1 to 8.22 miles. The location of the study site and deployment of roadway segments are presented in Figure 65. The raw data provides the average speed for each roadway segment and is collected at one-minute intervals.

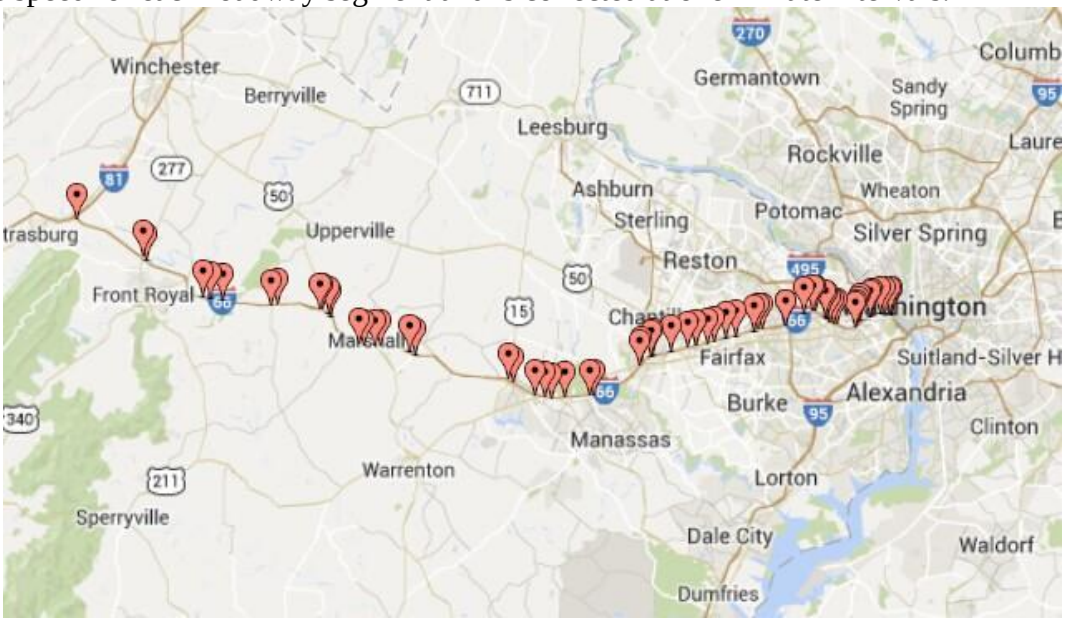


Figure 65: The Study Site on I-66 Eastbound (Source: Google Maps).

In order to use the collected traffic data in the proposed algorithm, data reduction is needed to transfer the raw measurement data into the required format of input data. In general, spatiotemporal traffic state matrix is the main component of input data. The INRIX data is collected by each roadway segment at different time interval. Each roadway segment represents a Traffic Management Center (TMC) station, and the geographic information of TMC station is also provided. The average speed for each TMC station can be used to derive spatiotemporal traffic state matrix. However, the raw INRIX data includes several problems, such as geographically inconsistent sections, irregular time intervals of data collection, and missing data. Considering these problems, the data reduction process is illustrated in Figure 66. Note that the data reduction is not constrained to the INRIX probe data, other types of traffic measurement data from various sensing techniques (e.g. loop detector, GPS and etc.) can also be used as the input to generate spatiotemporal traffic state matrix in the data reduction process.

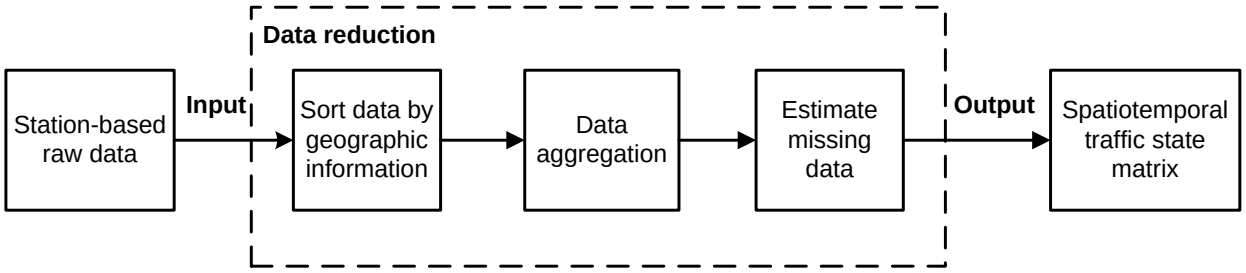


Figure 66: Data Reduction of INRIX Probe Data.

According to the geographic information of each TMC station, the raw data is sorted along the roadway direction (e.g. towards eastbound of I-66). An examination should be adopted to check any overlapping or inconsistent stations along the direction. Afterward, the speed data are aggregated by time intervals (e.g. 5 minutes in this study) to reduce the noise and smooth measurement error in this study. In this way, the raw data is aggregated to the form of daily data matrix along spatial and temporal intervals. It should be noted that missing data usually exist in the developed data matrix, therefore data imputation methods should be conducted to estimate the missing data by values of neighboring cells. Consequently, the daily spatiotemporal traffic state matrix can be generated for travel time modeling.

Experimental Work

The experimental work is divided into three parts the first part is modeling the travel time using a mixture of two linear regression with fixed proportions (λ_1, λ_2) and comparing the proposed model with the linear regression model. The second part is modeling the travel time using mixture of two linear regression with a variable proportions function of the same predictors used in the linear regression equations. The last part explains how the proposed model can be used to convey travel time reliability to user

1. Modeling Travel Time Using Mixture of Regression with Fixed Proportions

The purpose of this section is to experimentally proof that the mixture of two linear regression model is better than the linear regression model (one component). In order to show that, we fitted the proposed model to four months of the data then we compared the proposed model and the linear regression model. We used three measures to compare the two models. We used the Mean Absolute Percentage Error (MAPE) and Mean Absolute Error (MAE) to quantify the errors of both models with respect to the ground truth. MAPE is the average absolute percentage change

between the predicted $\widehat{y_i^j}$ and the true values y_i^j MAE is the absolute difference between the predicted and the true values.

$$\text{MAPE} = \frac{100}{IxJ}\sum_{j=1}^{J}\sum_{i=1}^{I}\frac{\left|y_i^j - \widehat{y_i^j}\right|}{y_i^j} \quad (8)$$

$$\text{MAE} = \frac{1}{IxJ}\sum_{j=1}^{J}\sum_{i=1}^{I}\left|y_i^j - \widehat{y_i^j}\right| \quad (9)$$

Here J is the total number of days in the testing dataset; I is the total number of time intervals in a single day; and y and $\hat{y}$ denote the ground truth and the predicted value, respectively, of the travel time for the time interval on the day. The lower the value of these errors measures the better the model. The other measure used for comparison is the histogram intersection. It measures how much the histogram of the predicted travel time, using certain model, is similar to the histogram of ground truth travel time. The higher the value of the histogram intersection the better the model.

$$H(y) \cap H(\hat{y}) = \frac{1}{Q}\sum_{q=1}^{Q}\min(H_q(y), H_q(\hat{y})) \quad (10)$$

where $H(y)$ and $H(\hat{y})$ are the histogram of ground truth travel time and the predicted travel time respectively. Table 24 shows values for the MAE, MAPE and the histogram intersection for models using different number of top ranked predictors. As shown in Table 24, for all models that are built using different number of predictors, the models that are built using proposed mixture of regression are better than the linear regression models because it has less MAE and MAPE and greater histogram intersection.

Table 24: Comparison Between One and Two Components Models

	MAPE		MAE		similarity	
Number of predictors	one component	two component	one component	two component	one component	two component
6	7.1877	5.6878	6.5664	5.2201	189.5400	217.6000
11	6.9909	5.6299	6.3871	5.1032	192.0600	223.7400
16	6.9559	5.5737	6.3584	5.0543	196.6500	224.4900
21	6.8948	5.5587	6.3159	5.0397	200.4000	225.4000
26	6.8983	5.5869	6.3116	5.0578	199.2500	227.1500
31	6.8998	5.6392	6.3153	5.0907	198.6900	228.3300
36	6.8768	5.6215	6.2959	5.0765	199.0800	228.7800
41	6.8820	5.6944	6.2973	5.1303	200.0600	231.8800
46	6.8733	5.6776	6.2902	5.1172	200.3100	232.0800
51	6.7996	5.6997	6.2258	5.1313	208.6100	232.5100
56	6.8195	5.6926	6.2374	5.1239	207.7700	232.5000
61	6.7687	5.7443	6.1843	5.1631	215.6500	233.9900
66	6.7621	5.7395	6.1798	5.1590	215.6600	234.3400
71	6.7882	5.7256	6.2024	5.1454	216.2500	234.2600
76	6.7779	5.7336	6.1920	5.1536	215.8000	233.9600
81	6.7886	5.7406	6.1990	5.1574	215.7400	233.8900
86	6.7758	5.7817	6.1849	5.1930	215.9800	233.9300
91	6.7912	5.7969	6.1930	5.2080	214.8400	234.1100
96	6.7865	5.7985	6.1890	5.2081	214.9900	234.0600
101	6.7830	5.8242	6.1868	5.2289	215.3400	233.8300

2. Travel Time Prediction

One goal of modeling travel time is using this model to predict travel time. Conveying this information to travelers helps them make better decisions. If we are interested in providing travel time information, we usually convey the expected travel time as one value and sometimes we also provide upper and lower pounds for the travel time. This travel time interval makes more sense because it reflects the uncertainty of the travel time. Moreover, it gives the travelers a better idea about the travel time variance for his planned trip.

In this work, for a given unseen new vector of predictors we get the mean of each component and then calculate travel time prediction as a weighted average of the travel time means. The weights used are the $\lambda_j's$. The travel time interval for unseen predictors' vector is calculated as the weighted average of the 95% confidence interval for each component. To evaluate the proposed model in travel time prediction, we test the two regression mixture models by four unseen months. MAPE and MAE are used to measure how much accurate is the expected travel time. To evaluate the travel time interval we define a hitting rate measure. It is the ratio of the number of ground truth travel times lay within the calculate interval to the total number of ground truth travel time. Table 25 shows the MAPE, MAE, hitting rate, and travel time width at different number of predictors. As shown in the table the models which are built using number of predictors larger than or equal 16 predictors almost have the same accuracy. The parameters' estimates for the model which uses a predictor vector of 16 dimension is shown in Appendix A. Figure 67 gives a better idea about how well the predicted travel time and the interval are.

Table 25: Travel Time Accuracy in Terms of MAPE and MAE, Travel Time Interval's Width and Hitting Rate

# of predictors	MAPE	MAE	% Hitting rate	interval width in minutes
6	7.9682	7.6353	81.9078	26.3633
11	7.7443	7.4271	81.0470	24.6697
16	7.7018	7.3915	80.9587	24.5197
21	7.6990	7.3861	80.8483	24.4428
26	7.6873	7.3636	80.7637	24.3143
31	7.6790	7.3520	80.6423	24.0701
36	7.6689	7.3418	80.6313	24.0629
41	7.6733	7.3357	80.4363	23.8458
46	7.6764	7.3366	80.4841	23.8178
51	7.6760	7.3286	80.4106	23.7539
56	7.6981	7.3418	80.5356	23.7617
61	7.6886	7.3228	80.4289	23.6568
66	7.6929	7.3200	80.4179	23.6497
71	7.6932	7.3190	80.4289	23.6205
76	7.7037	7.3228	80.3407	23.5557
81	7.7142	7.3263	80.3627	23.5332
86	7.7398	7.3414	80.1788	23.4517
91	7.7311	7.3329	80.1273	23.4112
96	7.7337	7.3349	80.0868	23.4040
101	7.7451	7.3392	79.9507	23.3756

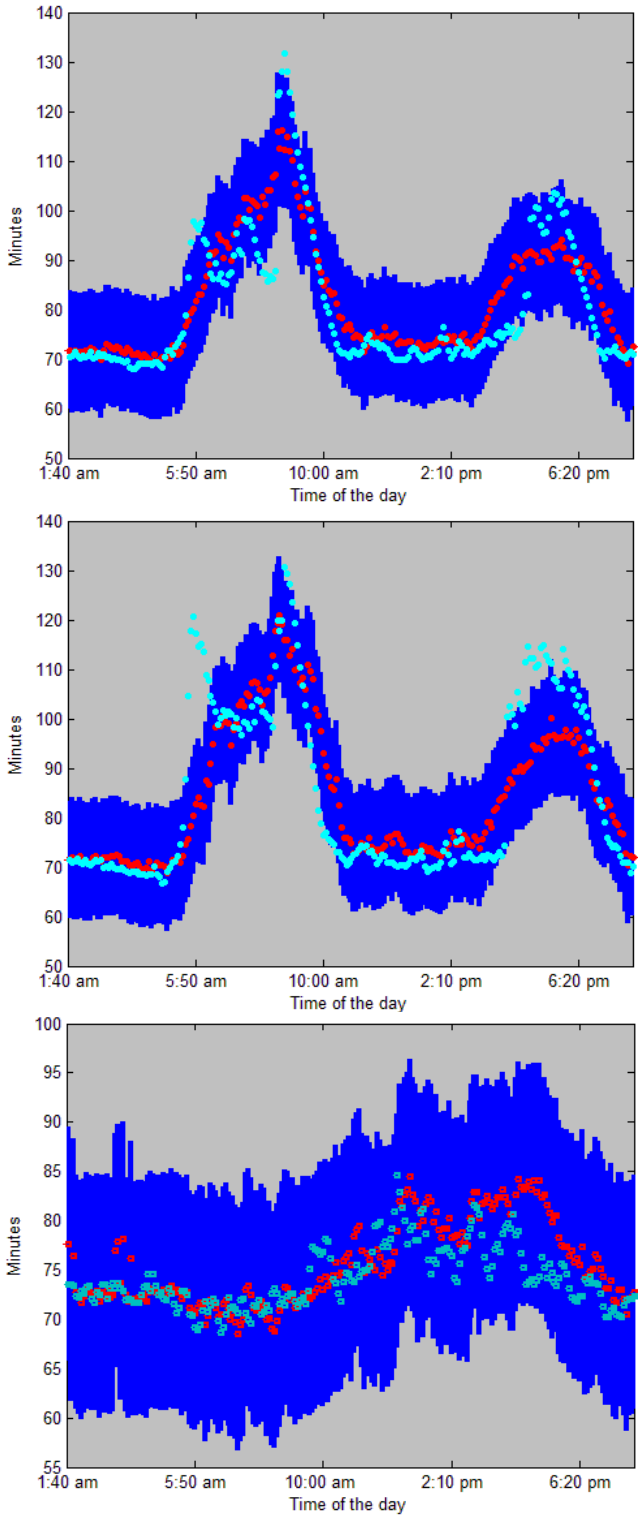


Figure 67: Shows the Travel Time Ground Truth (Red), the Predicted Travel Time (Cyan), And the Travel Time Interval (Blue)

3. Travel Time Reliability

Travel time reliability is the other form of information that we can convey to the user using the travel time model. Using the proposed model we can provide the traveler what are the probabilities of congestion and free-flow. Moreover, the expected and 90% percentile travel time for each regime can be provided. In order to get good estimates for the above quantities, the proportions should be a function of the predictor which means it varies depending on the values of the predictors. Revisiting the EM algorithm, it estimates the posterior probabilities w_{ij} and The model parameters ψ and returns only ψ at convergence and does not use w_{ij}. As shown in equation (4), the returned λ_j is an average of the posterior probabilities w_{ij}. In the two components model, if we modeled w_{ij} using logistic regression at the convergence of the EM, this means that λ_j becomes a function of the predictors as well as the components' means. We used the w_{ij} we get when we fit the model described in Appendix A to build a logistic regression. This logistic regression model the probability of predictor vector being drawn from component number two. Then using simple algebra manipulation we got the equation (11) for λ_2 . Now the new model is exactly the model in Appendix A but with variable λ_2 and λ_1

$$\lambda_2 = 1/1 - \exp([1\ x_{29,t0-1}\ x_{18,t0-1}\ \ x_{40,t0-1}\ \ x_{25,t0-1}\ \ x_{14,t0-1}\ \ x_{1,t0-1}\ \ x_{21,t0-1}\ x_{30,t0-1}\ x_{13,t0-1}] \begin{bmatrix} -1.8828 \\ -0.0249 \\ -0.0062 \\ -0.0114 \\ -0.0305 \\ -0.0155 \\ -0.0141 \\ -0.0131 \\ 0.0042 \\ 0.0896 \end{bmatrix}) \quad (11)$$

We test this model by calculating the mean, 90% percentile, and probabilities of congestion and free-flow for each predictor vector in each day of May 2013. Then we divided the day into four time interval and calculate the mean of the above quantities within each time interval given the day. The result shown in Table 26 is consistent with the travel time pattern we observe in Figure 63 where at the congestion time of the day the probability of the congestion component becomes higher. Also, the model shows that the probability of the morning congestion during weekends is lower than its values at weekdays.

Table 26: Testing The Model For Travel Time Reliability Using May 2013

		1:40 am - 4:55 am		5:00 am - 10:00 am		10:05 am - 3:00 pm		3:05 am - 7:00 pm	
		congested	free-flow	congested	free-flow	congested	free-flow	congested	free-flow
Tuesday	Mean (min)	87.07	73.07	127.66	85.51	94.88	75.53	120.96	81.44
	90% percentile (min)	71.94	70.80	112.53	83.23	79.75	73.25	105.83	79.17
	probability	0.0046	0.9954	0.8241	0.1759	0.1334	0.8666	0.8516	0.1484
Wednesday	Mean (min)	87.09	73.09	127.71	85.65	95.45	75.91	121.44	81.85
	90% percentile (min)	71.96	70.82	112.58	83.37	80.32	73.63	106.31	79.57
	probability	0.0051	0.9949	0.8114	0.1886	0.1488	0.8512	0.8684	0.1316
Thursday	Mean (min)	87.41	73.28	127.01	85.02	96.26	76.23	122.50	82.55
	90% percentile (min)	72.28	71.00	111.87	82.75	81.13	73.96	107.37	80.28
	probability	0.0050	0.9950	0.8035	0.1965	0.1581	0.8419	0.9057	0.0943
Friday	Mean (min)	87.24	73.12	119.99	81.62	95.53	75.95	122.95	82.96
	90% percentile (min)	72.11	70.84	104.86	79.34	80.40	73.67	107.82	80.69
	probability	0.0045	0.9955	0.7499	0.2501	0.1432	0.8568	0.9146	0.0854
Saturday	Mean (min)	87.47	73.30	109.75	75.64	98.78	78.32	123.52	83.33
	90% percentile (min)	72.34	71.03	94.62	73.36	83.65	76.05	108.39	81.06
	probability	0.0048	0.9952	0.5760	0.4240	0.3129	0.6871	0.9588	0.0412
Sunday	Mean (min)	86.84	73.07	110.00	76.00	99.38	78.38	120.64	81.81
	90% percentile (min)	71.71	70.80	94.87	73.73	84.25	76.11	105.51	79.54
	Probability	0.0038	0.9962	0.5908	0.4092	0.3237	0.6763	0.9145	0.0855
Monday	Mean (min)	87.19	73.18	122.06	82.46	93.21	74.46	117.66	79.51
	90% percentile (min)	72.06	70.90	106.93	80.18	78.08	72.19	102.53	77.24
	Probability	0.0046	0.9954	0.7524	0.2476	0.0738	0.9262	0.8304	0.1696

Conclusions and Future Work

In this paper, we proposed a travel time model based on two component mixture of linear regression. The proposed model can capture the stochastic nature of the travel time. The model assigns one component for the free-flow regime and the other component for the congested regime. The means of the components are a function in the input predictors which are chosen using random forest algorithm. The proposed model can be used to predict the travel time and the upper and lower pounds for the travel time as well. Moreover, the proposed model can be used to give the travel time reliability information and any time at any day. The experimental results show a promising performance of the proposed algorithm.

The current model does not consider the weather condition, incident, or work zones; however this model has the ability to easily integrate these factors if the historical dataset which includes them is available. So that our future work will focus on extending the proposed model to include these factors and study their effect on the travel time distribution.

References

[1] J. Rice and E. van Zwet, "A simple and effective method for predicting travel times on freeways," *Intelligent Transportation Systems, IEEE Transactions on,* vol. 5, pp. 200-207, 2004.

[2] X. Zhang and J. A. Rice, "Short-term travel time prediction," *Transportation Research Part C: Emerging Technologies,* vol. 11, pp. 187-210, 6// 2003.

[3] E. Sullivan, "New Model for Predicting Freeway Incidents and Incident Delays," *Journal of Transportation Engineering,* vol. 123, pp. 267-275, 1997.

[4] J. Kwon, B. Coifman, and P. Bickel, "Day-to-Day Travel-Time Trends and Travel-Time Prediction from Loop-Detector Data," *Transportation Research Record: Journal of the Transportation Research Board,* vol. 1717, pp. 120-129, 01/01/ 2000.

[5] P. Chakroborty and S. Kikuchi, "Using Bus Travel Time Data to Estimate Travel Times on Urban Corridors," *Transportation Research Record: Journal of the Transportation Research Board,* vol. 1870, pp. 18-25, 01/01/ 2004.

[6] A. Guin, J. Laval, and B. R. Chilukuri, "Freeway Travel-time Estimation and Forecasting," 2013.

[7] F. Guo, Q. Li, and H. Rakha, "Multistate Travel Time Reliability Models with Skewed Component Distributions," *Transportation Research Record: Journal of the Transportation Research Board,* vol. 2315, pp. 47-53, 12/01/ 2012.

[8] H. J. M. v. Grol, C. D. R. Lindveld, M. Papageorgiou, H. Haj Salem, S. Manfredi, G. Tognoni, *et al.*, "D'ACCORD Development and Application of Co-ordinated Control of Corridors," 1996.

[9] T. Oda, "An algorithm for prediction of travel time using vehicle sensor data," in *Road Traffic Control, 1990., Third International Conference on*, 1990, pp. 40-44.

[10] M. Iwasaki and K. Shirao, "A short term prediction of traffic fluctuations using pseudo-traffic patterns," presented at the the Third World Congress on Intelligent Transport Systems, Orlando, Florida 1996.

[11] M. D'Angelo, H. Al-Deek, and M. Wang, "Travel-Time Prediction for Freeway Corridors," *Transportation Research Record: Journal of the Transportation Research Board,* vol. 1676, pp. 184-191, 01/01/ 1999.

[12] H. M. Al-Deek, M. P. D'Angelo, and M. C. WANG, "TRAVEL TIME PREDICTION WITH NON-LINEAR TIME SERIES," presented at the Fifth International Conference

on Applications of Advanced Technologies in Transportation Engineering, Newport Beach, California, 1998.
[13] B. Williams and L. Hoel, "Modeling and Forecasting Vehicular Traffic Flow as a Seasonal ARIMA Process: Theoretical Basis and Empirical Results," *Journal of Transportation Engineering*, vol. 129, pp. 664-672, 2003.
[14] T. J. A Cherrett, H. A Bell, and M. A McDonald, "The use of SCOOT type single loop detectors to measure speed, journey time and queue status on non SCOOT controlled links," presented at the Proceedings of the Eighth International Conference on Road Traffic Monitoring and Control, 1996.
[15] L. Rilett and D. Park, "Direct Forecasting of Freeway Corridor Travel Times Using Spectral Basis Neural Networks," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1752, pp. 140-147, 01/01/ 2001.
[16] H. F. M. Matsui, "Travel time prediction for freeway traffic information by neural network driven fuzzy reasoning," *Neural networks in transport applications.*, 1998.
[17] J. You and T. J. Kim, "Development and evaluation of a hybrid travel time forecasting model," *Transportation Research Part C: Emerging Technologies*, vol. 8, pp. 231-256, 2// 2000.
[18] J. Guiyan and Z. Ruoqi, "Travel time prediction for urban arterial road," in *Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE*, 2003, pp. 1459-1462 vol.2.
[19] J. Guiyan and Z. Ruoqi, "Travel-time prediction for urban arterial road: a case on China," in *Vehicle Electronics Conference, 2001. IVEC 2001. Proceedings of the IEEE International*, 2001, pp. 255-260.
[20] C.-H. Wei, S.-C. Lin, and Y. Lee, "Empirical Validation of Freeway Bus Travel Time Forecasting," *Transportation Planning Journal*, vol. 32, 2003.
[21] L. . Kisgyorgy and L. R. Rilett, "Travel Time Prediction by Advanced Neural Network," *Periodica Polytechnica Civil Engineering*, vol. 46, 2002.
[22] L. Kisgyörgy and L. R. Rilett, "Travel time prediction by advanced neural network," *Civil Engineering*, vol. 46, pp. 15-32, 2002.
[23] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, pp. 5-32, 2001/10/01 2001.
[24] R. D. De Veaux, "Mixtures of linear regressions," *Computational Statistics & Data Analysis*, vol. 8, pp. 227-245, 11// 1989.
[25] S. Faria and G. Soromenho, "Fitting mixtures of linear regressions," *Journal of Statistical Computation and Simulation*, vol. 80, pp. 201-225, 2010/02/01 2009.
[26] C. Hao, H. A. Rakha, and S. Sadek, "Real-time freeway traffic state prediction: A particle filter approach," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, 2011, pp. 626-631.
[27] INRIX. (2012). *http://www.inrix.com/trafficinformation.asp*. Available: http://www.inrix.com/trafficinformation.asp

Appendix F

Table 27 shows the parameter estimates of the proposed model. In this table $x_{seg\#,time}$ is the speed at certain segment and time.

Table 27: Parameters' Estimates for Mixture of Two Regressions

		1st component	2nd component
β	$intercept$	79.4354	96.5943
	$x_{29,t0-1}$	-0.0153	-0.0148
	$x_{2,t0-1}$	-0.0903	-0.0250
	$x_{28,t0-1}$	-0.0668	0.0061
	$x_{18,t0-1}$	-0.0912	-0.0519
	$x_{27,t0-1}$	0.0187	-0.0449
	$x_{40,t0-1}$	-0.2107	-0.1107
	$x_{25,t0-1}$	-0.0652	-0.0603
	$x_{14,t0-1}$	-0.0245	-0.0136
	$x_{29,t0-2}$	-0.0106	-0.0224
	$x_{1,t0-1}$	-0.0745	-0.0150
	$x_{39,t0-1}$	-0.0174	-0.0331
	$x_{21,t0-1}$	-0.0203	-0.0252
	$x_{24,t0-1}$	-0.0742	-0.0239
	$x_{19,t0-1}$	0.0075	-0.0078
	$x_{30,t0-1}$	-0.1269	-0.0558
	$x_{13,t0-1}$	0.6767	0.0834
σ^2		11.8066	1.7746
λ		0.4466	0.5534

Chapter 12: Travel Time Reliability Modeling using a Mixture of Linear Regressions

This chapter is based on
Mohammed Elhenawy, Mohammed Almannaa, and Hesham Rakha, "Travel Time Reliability Modeling using a Mixture of Linear Regressions," under review paper.

Abstract

The rapid development and deployment of intelligent transportation systems (ITSs) attracts attention to developing data-driven algorithms for many of ITS functions. Travel time modeling can be considered a part of the travel and transportation management and travel demand management functions. Travel time has a huge impact on driver route choice behavior and assessment of transportation system performance. In this paper, a mixture of linear regression is proposed for travel time modeling. The mixture of linear regression model has three advantages. First, it provides better model fitting as compared to the linear regression. Second, the proposed can capture the bi-modal nature of the travel time distributions and link it to the free-flow and congested traffic regimes. Third, the means of the bi-modal distribution are modeled as functions of the input predictors. This last advantage allows for the quantitative evaluation of the probability of each travel time state as well as the uncertainty associated with each state at any time of the day given the values of predictors at that time. The proposed model is applied to archived data along a 74.4-mile freeway stretch of I-66 eastbound to connect I-81 and Washington D.C. The experimental results show the ability of the model to capture the stochastic nature of the travel time and gives good travel time predictions.

Introduction

The repaid deployment of the ITS during the last two decades to improve the performance of transportation systems, enhance safety, and provide more choices to travelers motivates the development of new algorithms satisfy its needs. Travel time modeling is one of the critical needs for ITS which is used to get accurate travel time predictions to guide the travelers route choice. The travel time models need data collection which is done by loop detectors, video detection, and/or probe vehicles. The state of practice algorithm used for travel time prediction is the instantaneous method due to its simplicity and ease of implementation. This algorithm assumes that speed on all road segments remain constant until the trip is completed, then trip time is the summation of the travel times of all the road stretch's segments[1, 2]. This algorithm works well if the change in the speed is small. In case of congestion the change in speed is large and the travel time prediction using instantaneous method is bad. The travel time models should be able to explain the variability in travel time. Travel time prediction has been the focus of many research efforts and, in general, the proposed algorithms can be categorized into two broad classes: statistical-based data-driven methods and simulation-based methods. Since the proposed model in this work is a statistical model we give a brief summary of the effort done in this class of algorithms.

Several researchers fit different regression models to predict travel time. A typical approach is to fit an MLR (Multiple Linear Regression) model using explanatory variables representing instantaneous traffic state and historical traffic data, for example [3-5]. Notice that the model proposed in [3] is even an SLR but was successful in providing acceptable travel time predictions. Some researchers developed hybrid methods where a regression model is used in

conjunction with other advanced statistical methods, for example [6] used regression with statistical tree methods. Another approach was followed in [7] where an SLR model used bus travel time to predict automobile travel time. Generally speaking, regression models are powerful in predicting travel time especially for short-term prediction, long-term predictions are less accurate. It is also claimed that regression models are more suitable for use free-flow traffic than congested traffic and fails to predict with incidents [8].

Travel time reliability is coined to use the travel time model to show the uncertainty associated with travel time. Much of the work done in travel time modeling assume travel time is generated from a single stochastic process and ignore the fact that travel time distribution is bi-model. So that, single-mode distributions are used to model travel time. Log-normal, gamma, Weibull, and exponential single mode distributions are used to model travel time data [9]. The resultant models are compared and log-normal has the best fit. However, single mode distribution cannot explain the variation in travel time because of the complex traffic conditions [10].

The next advance was proposing two normal component mixture to fit travel time data[11]. This model fits well the data with a small proportion of the congested state. However, this model underestimates the true proportion and overestimates the variance of the travel time in the free-flow state for data with 75% of time units in congested state [11]. This model underestimates the true proportion and overestimates the variance of the travel time in the free-flow state. To overcome this bias Guo et, al proposed adding a third component or using alternative component distributions such as log-normal or Gamma. Skewed component distributions are introduced to better fit nonsymmetrically distributed travel times, which is the case in congested states [12].This study found that the multistate lognormal model is the best model for modeling travel time under moderate to heavy traffic conditions along freeways based on the Akaike's information criterion (AIC). The previously proposed mixture models suffer from the following limitations:

1. The mean of each component is not modeled as a function of the available predictors.
2. Proportion variable is fixed at each time slot, which limits the model flexibility.
3. Provided information given the time slot of the day is the probability of each component (fixed) and the 90% percentile.

In this paper, we propose a model addressing these limitations by using a mixture of linear regression. It is based on a set of predictors which are the instantaneous travel time and average of historical travel time. This model generalizes the travel time reliability and gives the probability and travel time of congested and uncongested condition at any time given the predictors vector at $t0$. Also, we compare three algorithms of parameters estimation in the mixture of linear regression. Because previous research showed that the log-normal distribution gives the best fit for travel data, so we set assume each component proposed model is log-normal. The remainder of this paper is organized as follows. A review of the three algorithms of parameters estimation is provided. Subsequently, the proposed model based on a mixture of linear regression is presented. This is followed by a description of the test data used for the case study and the results of the travel time prediction and reliability. The last section provides the conclusions of the research and some recommendations for future research.

Methods

1. Mixture of Linear Regressions[13, 14]

Finite mixture models are powerful tools for modeling a wide variety of random phenomena. It is used to model random phenomena in many fields of statistical applications such as agriculture, biology, economics, medicine, and genetics. The mixture of linear regressions is one of the mixture families that is studied carefully and can be used to model the travel time under different traffic regimes.

The mixture of linear regression can be written as

$$f(y|X) = \sum_{j=1}^{m} \frac{\lambda_j}{\sigma_j\sqrt{2\pi}} e^{-\frac{\left(y-x^T\beta_j\right)^2}{2\sigma_j^2}} \quad (1)$$

Or as

$$y_i = \begin{cases} x_i^T\beta_1 + \epsilon_{i1} & \text{with probability } \lambda_1 \\ x_i^T\beta_2 + \epsilon_{i2} & \text{with probability } \lambda_2 \\ \vdots & \\ x_i^T\beta_m + \epsilon_{im} & \text{with probability } 1 - \sum_{q=1}^{m-1}\lambda_q \end{cases} \quad (2)$$

Where

y_i is the response corresponding to a vector of p predictors x_i^T

β_j is the vector of regression coefficients for the j^{th} component

λ_j is mixing probability of the j^{th} component

ϵ_{ij} is a normal random error.

The model parameters $\psi = \{\beta_1, \beta_2, \ldots, \beta_m, \sigma_1^2, \sigma_2^2, \ldots, \sigma_m^2, \lambda_1, \lambda_2, \ldots, \lambda_m\}$ can be estimated by maximizing the log-likelihood of equation (1) given a set of response predictors pairs $(y_1, x_1), (y_2, x_2), \ldots, (y_n, x_n)$ using Expectation-Maximization algorithm (EM).

2. EM Algorithm

The EM algorithm iteratively finds the maximum likelihood estimates by alternating the E-step and M-step. let $\psi^{(k)}$ the parameters estimates after the k^{th} iteration. on the E-step, the posterior probability of the i^{th} observation comes from component j is computed using equation (3).

$$w_{ij}^{(k+1)} = \frac{\lambda_j^{(k)}\phi_j\left(y_i|x_i, \psi^{(k)}\right)}{\sum_{j=1}^{m}\lambda_j^{(k)}\phi_j\left(y_i|x_i, \psi^{(k)}\right)} \quad (3)$$

where $\phi_j\left(y_i|x_i, \psi^{(k)}\right)$ is the probability density function of the j^{th} component.

on the M-step the new parameters' estimates $\psi^{(k+1)}$ that maximizes log-likelihood function in equation (1) is calculated as shown in equations (4-6)

$$\lambda_j^{(k+1)} = \frac{\sum_{i=1}^{n} w_{ij}^{(k+1)}}{n} \quad (4)$$

$$\hat{\beta}_j^{(k+1)} = (X^T W_j X)^{-1} X^T W_j Y \quad (5)$$

where X is $nx(p+1)$ predictor matrix, Y is the corresponding nx1 response vector, and W is nxn diagonal matrix which has $w_{ij}^{(k+1)}$ on its diagonal.

$$\hat{\sigma}_j^{2(k+1)} = \frac{\sum_{i=1}^{n} w_{ij}^{(k+1)} (y_i - x_i^T \hat{\beta}_j^{(k+1)})^2}{\sum_{i=1}^{n} w_{ij}^{(k+1)}} \quad (6)$$

the E-step and M-step are alternated repeatedly until the change in the incomplete log-likelihood is arbitrary small as shown in equation (7)

$$\left| \prod_{i=1}^{n} \sum_{j=1}^{m} \lambda_j^{(k+1)} \phi_j\left(y_i | x_i, \psi^{(k+1)}\right) - \prod_{i=1}^{n} \sum_{j=1}^{m} \lambda_j^{(k)} \phi_j\left(y_i | x_i, \psi^{(k)}\right) \right| < \xi \quad (7)$$

where ξ is a small number.

3. Classification EM (CEM) Algorithm

The CEM algorithm adds a step called C-step between the E- and M-steps of EM. In the C-step, the data is partitioned by assigning each observation to be one of the m components that has the largest posterior probability. The E-step of the CEM algorithm is identical to the E-step of the EM algorithm.

On the C-step of the $(k+1)^{th}$ iteration, the data is partitioned into m partitions $\left\{ p_1^{(k+1)} \; p_2^{(k+1)} \ldots p_m^{(k+1)} \middle| w_{ij}^{(k+1)}, \psi^{(k)} \right\}$.

The M-step updates the estimate $\psi^{(k+1)}$ using the partitions $p_j^{(k+1)}$ instead of the whole data. Thus on the M-step of the $(k+1)^{th}$ iteration, the $\lambda_j^{(k+1)}$ and $\hat{\beta}_j^{(k+1)}$ estimates are as shown in equations (8-9) while the estimates $\hat{\sigma}_j^{2(k+1)}$ are exactly like the EM.

$$\lambda_j^{(k+1)} = \frac{n_j}{n} \qquad j = (1,2,\ldots,m) \quad (8)$$

where n_j is the number of observations in partition j

$$\hat{\beta}_j^{(k+1)} = (X_j^T W_j X_j)^{-1} X_j^T W_j Y_j \qquad j = (1,2,\ldots,m) \quad (9)$$

Where

X_j is $n_j x (p+1)$ predictor matrix for the j partition

Y_j is the corresponding $n_j x 1$ response vector

W_j is $n_j x n_j$ diagonal martrix which has $w_{ij}^{(k+1)}$ on its diagonal.

CEM steps are repeated until a convergence criterion is met.

4. Stochastic EM (SEM) Algorithm

SEM hopes to escape local maximum and finds a better solution than EM by incorporating a stochastic step (S-step) between the E- and M-steps of EM. The E-step is exactly like the EM. On $(k+1)^{th}$ iteration, during the S-step, SEM randomly partition the data into m partitions based $w_{ij}^{(k+1)}$ where observation i can be assigned to partition j with probability $w_{ij}^{(k+1)}$. The greater the $w_{ij}^{(k+1)}$ the greater the chance observation i is assigned to partition j. The M-step is exactly like the M-step of CEM where the $\lambda_j^{(k+1)}$ and $\hat{\beta}_j^{(k+1)}$ estimates are based on the partitioned data.

5. Predictors

The proposed model is based on a set of predictors which are the instantaneous travel time and the average of historical travel time. For example, if we are interested in the travel time reliability at $t0$, the predictor vector will be the instantaneous travel time at the times $\{t0-60, t0-55, \ldots, t0-5\}$ and the average of the historical travel time at times $\{t0, t0+5, \ldots, t0+60\}$. In this paper, we have the average of the historical travel time for each day of the week. As

shown in Figure 68, there are two peaks of the travel time during morning and evening hours. The height of the peaks is different from one day to another especially on weekdays and weekends.

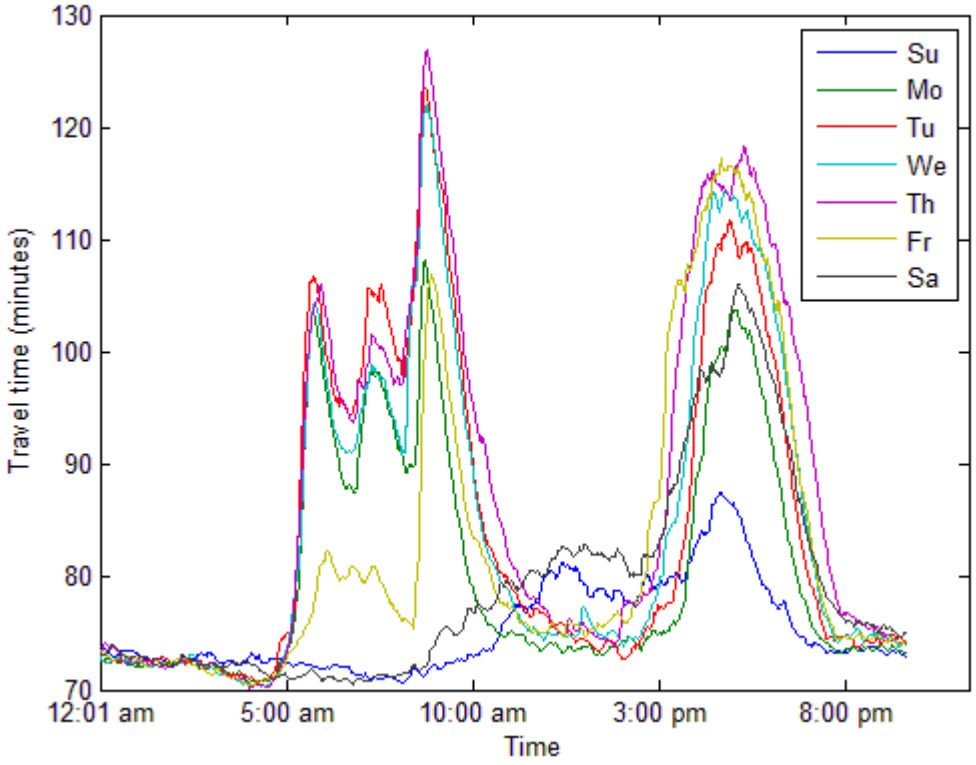


Figure 68: The Average of Historical Travel Time for Each Day of The Week

6. Travel Time Ground Truth Calculation

The calculation of the travel time ground truth is based on trajectory construction and the known speed through the trajectory's cells. A simple example of travel time ground truth calculation based on trajectory construction is demonstrated in Figure 64: Illustration of Travel Time Ground Truth Calculation . In this example, the roadway is divided into four sections using segments of length Δx and a time interval of Δt. We assumed that the speed is homogenous within each cell. The average speed of the red-dotted cell (i=2, n=3) in the figure is $u(x2, t3)$. Consequently, the trajectory slope represents the speed in each cell. Once the vehicle enters a new cell, the trajectory within this cell can be drawn as the straight blue line in Figure 69 using the cell speed as the slope. Finally, the ground truth travel time can be calculated when the trip reaches the downstream boundary of the last freeway section. It should be noted that the ground truth travel times were computed using the same dataset and used as the response (y).

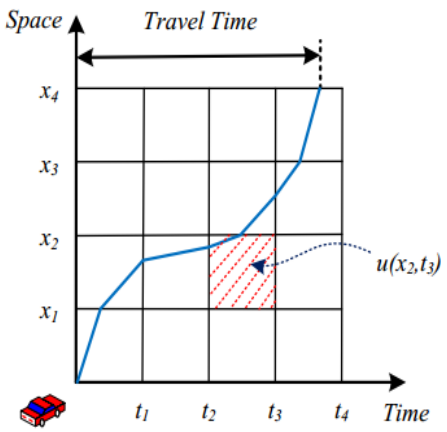


Figure 69: Illustration of Travel Time Ground Truth Calculation [2]

7. Instantaneous Travel Time

The instantaneous method is very simple where it assumes the segment speed does not change during the entire trip time. The travel time using the instantaneous approach is shown in equation (10)

$$\text{instantaneous travel time} = \sum_{i=1}^{h} \frac{L_i}{\upsilon_i^{t0}} \quad (10)$$

Where

L_i is the length of segment i

υ_i^{t0} is the speed at segment i at the departure time t0

h is the total number of segments.

8. Historical Average Method

If the spatiotemporal speed matrices are known for several previous months, then the ground truth travel time at each time interval for each day can be calculated. The historical average at any time of day D is calculated using equation (11)

$$\text{average historical travel time} = \sum_{i=1}^{Z_{D_i}} \frac{GTTT_{D_i}^{t0}}{Z_{D_i}} \quad \forall D_i = \text{Saturday}, \ldots, \text{Friday} \quad (11)$$

Where $GTTT_{D_i}^{t0}$ is the ground truth travel time at departure time t0 at historical day D_i and Z_{D_i} is number of days included in the average.

Data Description

The freeway stretch of I-66 eastbound to connect I-81 and Washington D.C. is selected as the test site for this study. High traffic volumes are usually observed during morning and afternoon peak hours on I-66 heading towards Washington D.C., therefore the study site provides a great environment to test the proposed travel time model.

In this study, the traffic data are provided by INRIX, which mainly collects probe data by GPS-equipped vehicles and supplemented with traditional road sensor data, as well as mobile devices and other sources [15]. The probe data on the test site covers 64 freeway segments with a total length of 74.4 miles. The average segment length is 1.16 miles long, and the length of each segment is unevenly divided in the raw data from 0.1 to 8.22 miles. The location of the study site and deployment of roadway segments are presented in Figure 18. The raw data provides the average speed for each roadway segment and is collected at one-minute intervals.

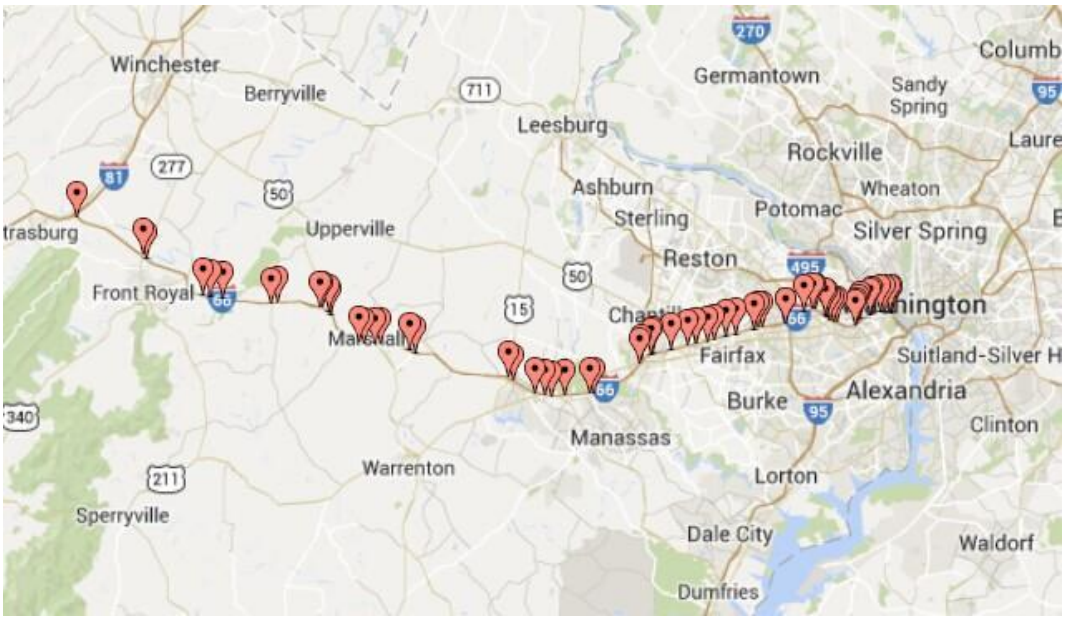


Figure 70: The Study Site on I-66 Eastbound (Source: Google Maps).

In order to use the collected traffic data in the proposed algorithm, data reduction is needed to transfer the raw measurement data into the required format of input data. In general, spatiotemporal traffic state matrix is the main component of input data. The INRIX data is collected by each roadway segment at different time interval. Each roadway segment represents a Traffic Management Center (TMC) station, and the geographic information of TMC station is also provided. The average speed for each TMC station can be used to derive spatiotemporal traffic state matrix. However, the raw INRIX data includes several problems, such as geographically inconsistent sections, irregular time intervals of data collection, and missing data. Considering these problems, the data reduction process is illustrated Figure 19. Note that the data reduction is not constrained to the INRIX probe data, other types of traffic measurement data from various sensing techniques (e.g. loop detector, GPS and etc.) can also be used as the input to generate spatiotemporal traffic state matrix in the data reduction process.

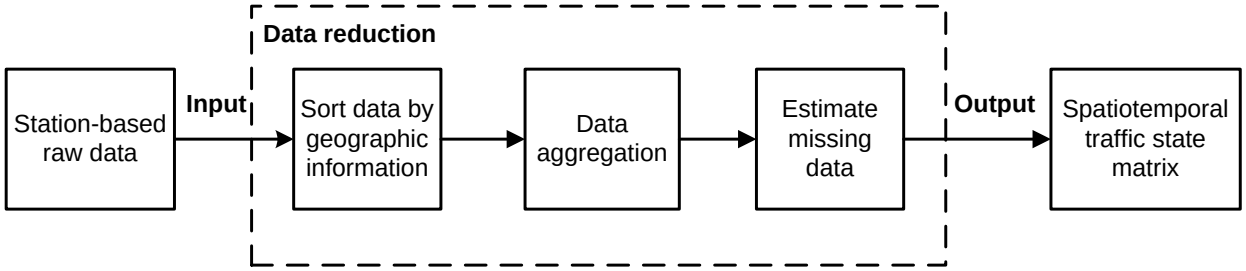


Figure 71: Data reduction of INRIX Probe Data.

According to the geographic information of each TMC station, the raw data is sorted along the roadway direction (e.g. towards eastbound of I-66). An examination should be adopted to check any overlapping or inconsistent stations along the direction. Afterward, the speed data are aggregated by time intervals (e.g. 5 minutes in this study) to reduce the noise and smooth measurement error in this study. In this way, the raw data is aggregated to the form of daily data

matrix along spatial and temporal intervals. It should be noted that missing data usually exist in the developed data matrix, therefore data imputation methods should be conducted to estimate the missing data by values of neighboring cells. Consequently, the daily spatiotemporal traffic state matrix can be generated for travel time modeling.

Experimental Work

The experimental work is divided into two parts. The first part is focusing on investigating which model is the best in terms of number of components and normal or log-normal. Moreover, the first comparison between the EM, CEM, and SEM, in order to choose the best one, is to use for travel time reliability. The second part shows that how we can use the built model and the posteriors probabilities to get travel time reliability at any time of the day.

1. Modeling Travel Time Using Mixture of Regression

The goal in this section is to choose the best model and best estimation algorithm. For the sake of completeness, each model has been compared with the multiple linear regression model which assumes the travel time distribution is unimodal distribution. In order to show that, we used three months of the data to estimate the parameters for mixture of two and three linear regression using the response vector log(Y) and the corresponding features matrix X. Then we compared the different models using the Mean Absolute Percentage Error (MAPE) and Mean Absolute Error (MAE) to quantify the errors of each model with respect to the ground truth. MAPE is the average absolute percentage change between the predicted $\widehat{y_i^j}$ and the true values y_i^j . MAE is the absolute difference between the predicted and the true values.

$$\text{MAPE} = \frac{100}{IxJ}\sum_{j=1}^{J}\sum_{i=1}^{I}\frac{\left|y_i^j - \widehat{y_i^j}\right|}{y_i^j} \tag{12}$$

$$\text{MAE} = \frac{1}{IxJ}\sum_{j=1}^{J}\sum_{i=1}^{I}\left|y_i^j - \widehat{y_i^j}\right| \tag{13}$$

Here J is the total number of days in the testing dataset; I is the total number of time intervals in a single day; and y and $\hat{y}$ denote the ground truth and the predicted value, respectively, of the travel time for the time interval on the day. The lower the value of these errors measures the better the model. The MAE and MAPE of all models fitted using the three estimation algorithms and different number of predictors are shown in Table 28. We tried different number of predictors in order to find the simplest model. As shown in this table, the mixture of two linear regression is the best regardless the estimation algorithm used. Another important result is that the accuracy of the model in terms of MAE and MAPE improved as we add predictors. In real time running, we prefer simple models so that we can choose the mixture of two linear regression with 11 predictors and use it for the next set of experiments. The estimates of the model parameters using the EM algorithm are shown in Table 29.

Table 28: Comparison Between the Different Log-normal Models Using Different Estimation Algorithms

	EM					
	MAE			MAPE		
Component # / Predictors #	1	2	3	1	2	3
19	5.0276	4.2488	4.3321	5.4581	4.5565	4.6760
17	5.0260	4.2584	4.3658	5.4521	4.5623	4.7099
15	5.0251	4.2619	4.3652	5.4472	4.5621	4.7062
13	5.0268	4.2766	4.3613	5.4461	4.5784	4.7011
11	5.0287	4.2733	4.3725	5.4461	4.5736	4.7128
9	5.0297	4.2757	4.3858	5.4467	4.5764	4.7270
7	5.0333	4.2757	4.3883	5.4507	4.5771	4.7302
5	5.0360	4.2784	4.3851	5.4544	4.5798	4.7241
3	5.0403	4.2815	4.3783	5.4587	4.5802	4.7145
	CEM					
	MAE			MAPE		
Component # / Predictors #	1	2	3	1	2	3
19	5.0276	4.3662	4.3687	5.4581	4.7520	4.7573
17	5.0260	4.3765	4.3829	5.4521	4.7600	4.7705
15	5.0251	4.3965	4.4018	5.4472	4.7822	4.7903
13	5.0268	4.4076	4.4098	5.4461	4.7955	4.7982
11	5.0287	4.4114	4.4161	5.4461	4.7992	4.8075
9	5.0297	4.4118	4.4206	5.4467	4.7998	4.8123
7	5.0333	4.4191	4.4223	5.4507	4.8087	4.8143
5	5.0360	4.4246	4.4248	5.4544	4.8167	4.8177
3	5.0403	4.4258	4.4225	5.4587	4.8183	4.8156
	SEM					
	MAE			MAPE		
Component # / Predictors #	1	2	3	1	2	3
19	5.0276	4.4470	8.1002	5.4581	4.8329	8.1654
17	5.0260	4.4577	8.1140	5.4521	4.8408	8.1745
15	5.0251	4.4628	8.1779	5.4472	4.8435	8.2287
13	5.0268	4.4704	8.2594	5.4461	4.8492	8.2950
11	5.0287	4.4748	8.2892	5.4461	4.8543	8.3120
9	5.0297	4.4796	8.2274	5.4467	4.8630	8.2526
7	5.0333	4.4815	8.0576	5.4507	4.8641	8.1044
5	5.0360	4.4856	7.9124	5.4544	4.8703	7.9793
3	5.0403	4.4936	7.8538	5.4587	4.8794	7.9172

Table 29: The EM Parameters' Estimates for Mixture of Linear Regression Assuming Log-normal Distribution (Log(y)):

		First component	second component
β_j	intercept	3.4875	3.9663
	$x_{(to-25)}^{ins}$	-0.0003	-0.0001
	$x_{(to-20)}^{ins}$	-0.0002	0.0001
	$x_{(to-15)}^{ins}$	0.0001	-0.0001
	$x_{(to-10)}^{ins}$	0.0000	0.0000
	$x_{(to-5)}^{ins}$	0.0020	0.0026
	$x_{(to)}^{his}$	0.0017	-0.0002
	$x_{(to+5)}^{his}$	0.0036	0.0010
	$x_{(to+10)}^{his}$	0.0028	0.0012
	$x_{(to+15)}^{his}$	0.0013	-0.0008
	$x_{(to+20)}^{his}$	-0.0022	0.0011
	$x_{(to+25)}^{his}$	0.0028	-0.0003
σ_j^2		0.1008	0.0222
λ_j		0.4655	0.5345

2. Travel Time Reliability

Travel time reliability is the form of information that we can convey to traveler using the travel time model. Using the proposed model, we can provide the traveler what are the probabilities of congestion and free-flow. Moreover, the expected and 90% percentile travel time for each regime can be provided. In order to get good estimates for the above quantities, the proportions should be a function of the predictor which means it varies depending on the values of the predictors. Revisiting the EM algorithm, it estimates the posterior probabilities w_{ij} and the model parameters ψ and returns only ψ at convergence and does not use w_{ij}. As shown in equation (4), the returned λ_j is an average of the posterior probabilities w_{ij}.In the two component models, if we modeled w_{ij} using logistic regression at the convergence of the EM, this means that λ_j becomes a function of the predictors as well as the components' means. We used the obtained w_{ij} when we fit the model described in Table 29 to build a logistic regression. This logistic regression models the probability of predictor vector being drawn from component number two. Then using simple algebra manipulation, we got the coefficient of the logistic model for λ_2 which are shown in Table 30. Now, the new model is exactly the model in Table 29 but with variable λ_2 and λ_1.

Table 30: The Estimated Coefficient for the Logic Model For λ_2

predictors	coefficient
intercept	-12.1702
$x_{(to-10)}^{ins}$	-0.0046
$x_{(to-5)}^{ins}$	0.0354
$x_{(to)}^{his}$	0.0890
$x_{(to+15)}^{his}$	0.0476
$x_{(to+20)}^{his}$	-0.1015
$x_{(to+25)}^{his}$	0.0718

We tested the proposed model by visually inspect the ground truth travel time for each day and the mean of each component as well as the λ_1 , which is the probability of congestion in the fitted model. We visually check if the value of λ_1 is large at the time when the ground truth becomes large. As shown in Figure 72 for weekday (top panel), there are two peaks at morning and evening and at the same time the values of λ_1 approach one which means the probability of congestion is high. The bottom panel shows a weekend where this no morning congestion but there is an evening congestion and λ_1 has only high values at the evening peak.

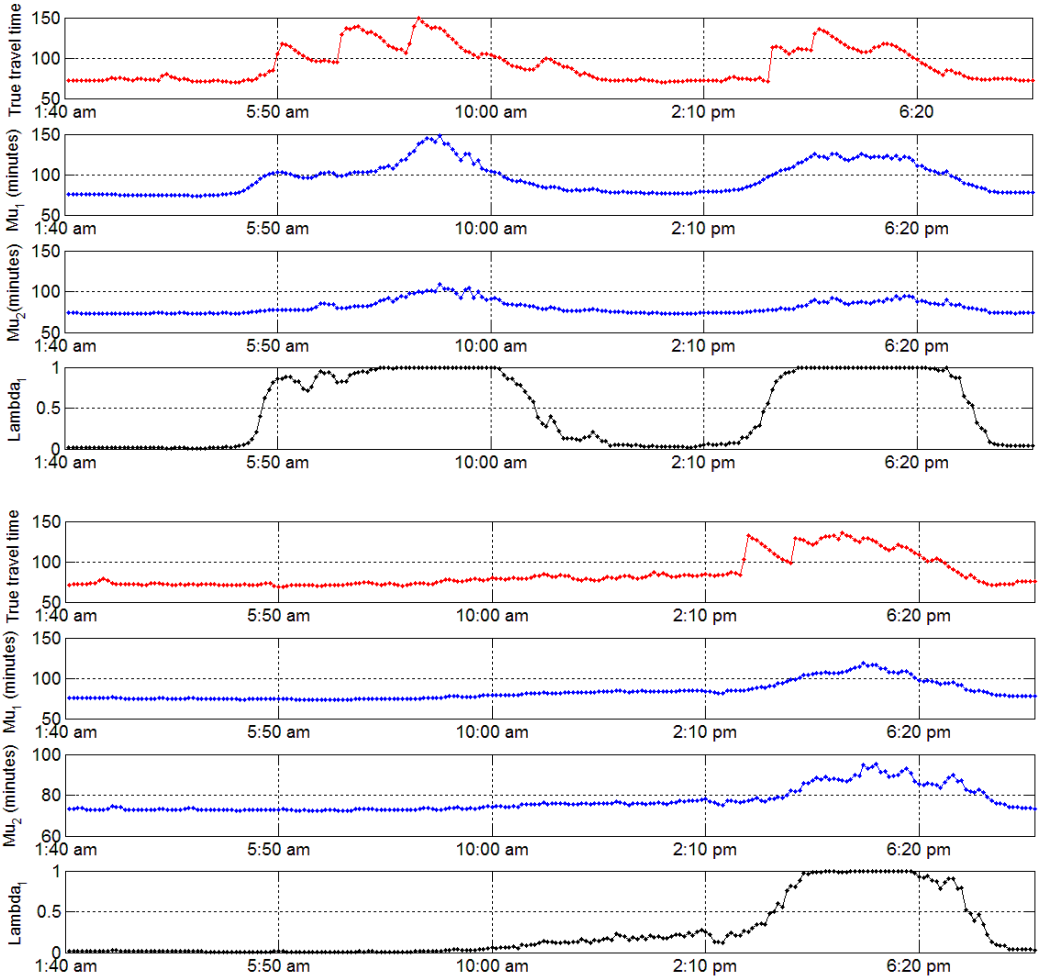


Figure 72: The Ground Truth Travel Time (Red Curve), the Mean of Each Component of the Proposed Model (Blue Curves), and the λ_1 Which Is the Congestion Probability for Two Different Days. The Upper Panel Is Weekday and the Bottom Panel is a Weekend Day.

In order to better test the proposed model, we calculate the mean, 90% percentile, and probabilities of congestion and free-flow for each predictor vector in each day of May 2013. Then based on the curves in Figure 68, we divided the day into four time interval and calculated the mean of the above quantities within each time interval given for each day of the week. The result shown in Table 31 is consistent with the travel time pattern that we observe in Figure 68 where at the congestion time of the day the probability of the congestion component becomes

higher. Also, the model shows that the probability of the morning congestion during weekends is lower than its values at weekdays.

Table 31: Testing the Model for Travel Time Reliability Using May 2013.

		1:40 am - 4:55 am		5:00 am - 10:00 am		10:05 am - 3:00 pm		3:05 am - 7:00 pm	
		free-flow	congested	free-flow	congested	free-flow	congested	free-flow	congested
Sunday	mean	73.0488	75.1705	73.1512	74.7273	75.8167	80.7624	76.2411	82.8952
	90% percentile	75.1582	85.5391	75.2660	85.0381	78.0486	91.9409	78.4755	94.4364
	probability	0.9853	0.0147	0.9864	0.0136	0.8769	0.1231	0.7993	0.2007
Monday	mean	72.9965	74.6206	82.0748	95.9163	75.7127	77.9892	78.8482	91.4993
	90% percentile	75.1046	84.9159	84.7607	109.6074	78.0015	88.8062	81.2451	104.6935
	probability	0.9876	0.0124	0.3014	0.6986	0.9014	0.0986	0.4528	0.5472
Tuesday	mean	73.0480	74.8349	86.1908	107.3710	76.1483	79.6850	81.1346	98.8168
	90% percentile	75.1577	85.1629	89.0161	123.0241	78.4567	90.8143	83.6036	113.2747
	probability	0.9859	0.0141	0.1140	0.8860	0.8604	0.1396	0.2819	0.7181
Wednesday	mean	73.0569	74.5786	83.5223	102.1581	76.8479	80.2636	84.2647	105.4945
	90% percentile	75.1674	84.8699	86.1598	117.0035	79.1299	91.4155	86.9091	120.8847
	probability	0.9874	0.0126	0.1823	0.8177	0.8525	0.1475	0.1676	0.8324
Thursday	mean	73.0237	74.5813	84.5361	106.4077	77.6081	82.3683	85.9053	112.9694
	90% percentile	75.1332	84.8728	87.2393	121.9683	79.9461	93.9328	88.5890	129.1630
	probability	0.9874	0.0126	0.1352	0.8648	0.7630	0.2370	0.0709	0.9291
Friday	mean	72.8766	74.3745	76.9303	85.4773	75.7830	80.4316	85.9168	111.0949
	90% percentile	74.9815	84.6359	79.2372	97.6921	78.0162	91.6230	88.5674	127.1507
	probability	0.9888	0.0112	0.6883	0.3117	0.8689	0.1311	0.0733	0.9267
Saturday	mean	73.0011	74.7768	73.0327	74.9951	75.8940	82.7500	80.5937	97.4901
	90% percentile	75.1093	85.0923	75.1442	85.3594	78.1081	94.1929	83.0425	111.2402
	probability	0.9870	0.0130	0.9837	0.0163	0.8343	0.1657	0.2593	0.7407

Conclusion and Future Work

In this paper, we proposed a travel time model based on a mixture of linear regression. The distribution of each component in this mixture is assumed to follow log-normal. Three parameter estimation algorithms are used and we found EM is the best out of the three algorithms. We compare three component and two component mixture with linear regression and we found two components fits data the best whatever the estimation algorithm used. The proposed model can capture the stochastic nature of the travel time. The two component model assigns one component for free-flow regime and the other component for the congested regime. The means of the components are a function of the input predictors. The proposed model can be used to give the travel time reliability information and anytime at any day if the predictors' vector is available. The experimental result shows a promising performance of the proposed algorithm.

The current model does not consider the weather condition, incident, or work zones; however this model has the ability to easily integrate these factors if the historical dataset which includes them is available. So that our future work will focus on extending the proposed model to include these factors and study their effect on the travel time distribution.

References

[1] X. J. Ban, R. Herring, J. Margulici, and A. M. Bayen, "Optimal sensor placement for freeway travel time estimation," in *Transportation and Traffic Theory 2009: Golden Jubilee*, ed: Springer, 2009, pp. 697-721.

[2] X. Ban, Y. Li, A. Skabardonis, and J. Margulici, "Performance evaluation of travel time methods for real time traffic applications," in *11th World Conference on Transport Research*, 2007.

[3] J. Rice and E. van Zwet, "A simple and effective method for predicting travel times on freeways," *Intelligent Transportation Systems, IEEE Transactions on,* vol. 5, pp. 200-207, 2004.

[4] X. Zhang and J. A. Rice, "Short-term travel time prediction," *Transportation Research Part C: Emerging Technologies,* vol. 11, pp. 187-210, 6// 2003.

[5] E. Sullivan, "New Model for Predicting Freeway Incidents and Incident Delays," *Journal of Transportation Engineering,* vol. 123, pp. 267-275, 1997.

[6] J. Kwon, B. Coifman, and P. Bickel, "Day-to-Day Travel-Time Trends and Travel-Time Prediction from Loop-Detector Data," *Transportation Research Record: Journal of the Transportation Research Board,* vol. 1717, pp. 120-129, 01/01/ 2000.

[7] P. Chakroborty and S. Kikuchi, "Using Bus Travel Time Data to Estimate Travel Times on Urban Corridors," *Transportation Research Record: Journal of the Transportation Research Board,* vol. 1870, pp. 18-25, 01/01/ 2004.

[8] A. Guin, J. Laval, and B. R. Chilukuri, "Freeway Travel-time Estimation and Forecasting," 2013.

[9] E. B. Emam and H. Ai-Deek, "Using real-life dual-loop detector data to develop new methodology for estimating freeway travel time reliability," *Transportation Research Record: Journal of the Transportation Research Board,* vol. 1959, pp. 140-150, 2006.

[10] P. Sangjun, H. Rakha, and G. Feng, "Multi-state travel time reliability model: Impact of incidents on travel time reliability," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, 2011, pp. 2106-2111.

[11] F. Guo, H. Rakha, and S. Park, "Multistate Model for Travel Time Reliability," *Transportation Research Record: Journal of the Transportation Research Board,* vol. 2188, pp. 46-54, 12/01/ 2010.

[12] F. Guo, Q. Li, and H. Rakha, "Multistate Travel Time Reliability Models with Skewed Component Distributions," *Transportation Research Record: Journal of the Transportation Research Board,* vol. 2315, pp. 47-53, 12/01/ 2012.

[13] R. D. De Veaux, "Mixtures of linear regressions," *Computational Statistics & Data Analysis,* vol. 8, pp. 227-245, 11// 1989.

[14] S. Faria and G. Soromenho, "Fitting mixtures of linear regressions," *Journal of Statistical Computation and Simulation,* vol. 80, pp. 201-225, 2010/02/01 2009.

[15] INRIX. (2012). http://www.inrix.com/trafficinformation.asp. Available: http://www.inrix.com/trafficinformation.asp

Appendix G

The model used for travel time reliability is

$$p(\log(y)|X) = \frac{\lambda_1}{\sqrt{2*0.1008*\pi}} e^{-\frac{\left(y-\left[1\ x^{ins}_{(to-25)}\ x^{ins}_{(to-20)}\cdots x^{ins}_{(to-5)}\ x^{his}_{(to+5)}\ x^{his}_{(to+10)}\cdots x^{his}_{(to+30)}\right]\begin{bmatrix}3.4875\\-0.0003\\-0.0002\\0.0001\\0.0000\\0.0020\\0.0017\\0.0036\\0.0028\\0.0013\\-0.0022\\0.0028\end{bmatrix}\right)^2}{2*0.1008}} + \frac{\lambda_2}{\sqrt{2*0.0222*\pi}} e^{-\frac{\left(y-\left[1\ x^{ins}_{(to-25)}\ x^{ins}_{(to-20)}\cdots x^{ins}_{(to-5)}\ x^{his}_{(to+5)}\ x^{his}_{(to+10)}\cdots x^{his}_{(to+30)}\right]\begin{bmatrix}3.9663\\-0.0001\\0.0001\\-0.0001\\0.0000\\0.0026\\-0.0002\\0.0010\\0.0012\\-0.0008\\0.0011\\-0.0003\end{bmatrix}\right)^2}{2*0.0222}} \quad (14)$$

where

$$\lambda_2 = \frac{1}{1-\exp\left(\left[1\ x^{ins}_{(to-10)}\ x^{ins}_{(to-5)}\ x^{his}_{(to)}\ x^{his}_{(to+15)}\ x^{his}_{(to+20)}\ x^{his}_{(to+25)}\right]\begin{bmatrix}-12.1702\\-0.0046\\0.0354\\0.0890\\0.0476\\-0.1015\\0.0718\end{bmatrix}\right)}$$

and

$\lambda_1 = 1 - \lambda_2$

Part III

Chapter 13: Enhanced Modeling of Driver Stop-or-Run Actions at a Yellow Indication Use of Historical Behavior and Machine Learning Methods

This chapter is based on
Mohammed Elhenawy, Hesham Rakha, and Ihab El-Shawarby, "Enhanced Modeling of Driver Stop-or-Run Actions at a Yellow Indication," Transportation Research Record: Journal of the Transportation Research Board, vol. 2423, pp. 24-34, 12/01/ 2014.

Abstract

The ability to model driver stop/run behavior at signalized intersections is critical in the design of advanced driver assistance systems. Such systems can reduce intersection crashes and fatalities by predicting driver stop/run behavior. The research presented in this paper uses data collected from a controlled field experiment on the Smart Road at the Virginia Tech Transportation Institute (VTTI) to model driver stop/run behavior at the onset of a yellow indication. The paper offers three contributions. First, it evaluates the importance of various model predictors in the modeling of driver stop/run behavior in the vicinity of signalized intersections. Second, it introduces a new variable related to the driver aggressiveness and demonstrates that this measure enhances the modeling of driver stop/run behavior. Third it applies well-known machine learning techniques, including: K-nearest neighbors (K-NN), random forests, and Adaptive Boosting (AdaBoost) techniques on the data and compares their performance to standard logistic models in an attempt to identify the optimum modeling framework. The experimental work shows that by adding the driver aggressiveness predictor to the model, the model accuracy increases by approximately 10% for the logistic, random forest and K-NN models and by 7% for the AdaBoost model. The paper also demonstrates that all modeling frameworks produce similar prediction accuracies.

Introduction

With advances in sensing, communications and computational technologies, research in the area of safety is increasing significantly. Most new cars have active safety features including anti-lock braking and adaptive cruise control systems to reduce road accidents [1]. In the US, the Department of Transportation (DOT) reported 32,367 fatalities caused by road accidents in 2011 [2]. A significant percentage of these road accidents occurred at signalized intersections as a result of the dilemma zone problem. At the onset of a yellow indication vehicle crashes may result from dilemma zone and/or red light running problems. A driver is trapped in the dilemma zone, if at the onset of yellow indication, the driver's distance to the intersection is less than the distance required for safe stopping and greater than the distance required to proceed and clear the intersection before the yellow interval ends. If the driver decides to stop when he/she should have proceeded, a rear-end crash could take place. While if the driver proceeds when he/she should have stopped, he/she would run the red light and a right-angle crash with side-street traffic could occur. There are many parameters that can be used to model or classify the driver stop/run behavior while approaching high-speed signalized intersections at the onset of yellow. These parameters can be categorized as driver related parameters and intersection related parameters. Age, gender, and perception-reaction time (PRT) are examples of driver related

parameters while approaching speed, distance to intersection (DTI) and time to intersection (TTI), and safe acceptable acceleration and deceleration rates are intersection related.

An aggressive driver who sometimes drives recklessly can cause accidents at signalized intersections. Aggressive driving could be critical in modeling driver stop/run behavior at signalized intersections; however measuring driver aggressiveness may not be plausible. Previous research uses five driver actions to measure aggressive driving. These five measures include: short or long honk of the horn, cutting in front of other vehicles in a passing lane maneuver, cutting in front of other vehicles in a multi-lane passing maneuver, and passing one or more vehicles by driving on the shoulder and then cutting in [3]. In this paper we introduce a new parameter related to the driver aggressiveness. This new predictor can be observed directly from stop/run historical behavior. Using this new predictor we demonstrate that the modeling of driver stop/run behavior can be enhanced. The use of such models can then be integrated with in-vehicle safety systems to predict the action of a driver and thus warn other drivers or take action to ensure that no collisions occur.

The past two decades have seen numerous research efforts and advances in both machine learning and computers. Many machine learning techniques require a large number of computations and are infeasible without computers. The available machine learning algorithms and computation power encourages researchers to transfer this knowledge into their fields. Transportation engineers are among people who are interested in applying these algorithms to address transportation problems. This interest increases with the availability of datasets from fixed detectors or data probes and intelligent transportation systems (ITSs). Recently, some machine learning algorithms were used in the transportation field, including: classifying and counting vehicles detected by multiple inductive loop detectors [4], identifying motorway rear-end crash risks using disaggregate data [5], automatic traffic incident detection [6], and real-time detection of driver distraction [7, 8]. Modeling driver stop/run behavior at signalized intersections is very important and is ideal for applying machine learning techniques. At first glance, driver stop/run behavior modeling seems to be a good candidate for straightforward application of machine learning algorithms. Observations of the driver stop/run behavior from naturalized datasets or from controlled field experiment datasets can be used to train machine learning algorithms. The trained models can then be used to predict future driver decisions for implementation in in-vehicle safety systems. However, machine learning modeling of driver stop/run behavior faces some challenges, including: the need for large labeled datasets, driver stop/run behavior drift and computational complexity.

Large datasets is the first challenge for any data analysis techniques such as machine learning. Machine learning algorithms cannot build a reasonably accurate model unless the training data is a good sample from the population describing the driver stop/run behavior. In order to ensure that the training dataset is a good sample representing the population it should be large. The label (in this case the decision to stop or to run) is not a problem in driver modeling because it is observed easily in experimental and/or in naturalistic data.

Driver stop/run behavior can change over time, also known as drift. This means that the driver modeling is very dynamic and learning algorithms and the predictors should be capable of adjusting to these changes quickly. Learning algorithms are always compared in terms of prediction accuracy. In many applications, an algorithm that has a predictive accuracy of 88% is better than algorithm that achieves 90% if the first algorithm is computationally simpler and more suitable for real-time applications because computation complexity is very important in

real-time applications. However, the driver model is built offline in the training phase in our application in an attempt to overcome this challenge.

Recently accumulating large data instances and features is easy and storing it is inexpensive. In machine learning the terms feature selection, predictor selection, and variable selection are used interchangeably. This large amount of machine readable information needs the machine learning tools' ability to understand and make use of it. Fundamental to machine learning is a feature selection that maintains the dimensionality of the pattern representation (i.e., the number of features) as small as possible to save measurement cost and improve classification accuracy. Feature selection is done before applying the learning algorithm to choose the best subset of features. The best subset should consist of the least number of dimensions that are important for increasing accuracy; we discard the remaining, unimportant dimensions. Feature selection algorithms should not discard features that lead to a loss in the discrimination power and thereby lower the accuracy of the resulting recognition system. Feature selection algorithms are classified into two broad model categories, filter models and the wrapper models [9]. Filter models select some features independently of any learning algorithm based on the general characteristics of the training data. Alternatively, wrapper models require the determination of certain learning algorithms and use their performance to evaluate and determine which features should be selected.

In terms of the paper layout, after the introduction, a brief background of the machine learning methods used in this study are presented, followed by our insights about the proposed predictor variables for consideration. Thereafter, a description of the data collection procedure, the experimental work, results, and discussion are presented. Finally, the conclusions of the study are presented along with recommendations for further work.

Methods

In this section a brief introduction to the modeling techniques used in this paper are presented to familiarize readers with these emerging techniques. The strengths of each modeling technique are presented given that the models will be compared later on the same dataset. The modeling techniques represent a variety of machine learning techniques. They range from very simple algorithms to complicated and high computational demand algorithms. The used algorithms demonstrate the wide variety of algorithms that can be used by transportation practitioners.

1. K-Nearest Neighbors Algorithm

K-nearest neighbors (K-NN) is a naive technique that is good for classification [10]. The advantage of K-NN is that it does not require training to build the model because the training data represents the model. The simplest version of a K-NN model is when the training portion of the algorithm simply stores the data points of the raw training data $T = \{x_1, \dots, x_n\}$ without any processing. To classify the unseen instance x_t , the algorithm calculates the distances from x_t to all instances in the dataset, then it finds the K smallest distances (nearest neighbors). Finally, the algorithm assigns x_t to the class that has the majority in K. This algorithm has two important parameters, the distance metric and the number K. A typical distance metric $d(x_t, x_i)$ is the Euclidean distance, where the value of k is typically between 5 and 20 and is set up experimentally as,

$$d(x_t, x_i) = \sqrt{(x_t - x_i)^T (x_t - x_i)} \quad (1)$$

When the training dataset is very large, the K-NN algorithm has a memory problem. There is another version of K-NN called Agglomerative Nearest Neighbor (A-NN) that is used to

overcome memory problems in K-NN algorithms. During the training phase, the A-NN algorithm clusters the training instances that have the same label using any of the well-known clustering algorithms and saves the cluster centroids for each class $C_j = \{c_{j1}, \dots, c_{jm}\}$. To classify unseen instances x_t, the A-NN algorithm calculates the distances from x_t to all instances in C_j $\forall j$, and then finds the K smallest distances (nearest neighbors). Finally, the algorithm assigns x_t to the class that has the majority centroids in K. A-NN is not required in our case because we only have 24 drivers and thus there is no memory problem.

2. Generalized Linear Models

Generalized linear models (GLMs) are statistical models that model responses as linear combinations of regressors (predictors). It can be used as regression models for continuous dependent variables, models for rates and proportions, binary, ordinal and multinomial variables, and counts. The GLM approach has two big advantages: (a) its theoretical framework is valid for many commonly encountered statistical models; (b) the software implementation is simple because the same algorithm can be used for estimation, inference and assessing model adequacy for all GLMs.

Logistic regression is a commonly used GLM when the response variable is a binomial response. It is used to model the driver stop/run behavior and gives a good accuracy [11]. Equation (2) shows the fitted model of the driver stop/run behavior.

$$\ln\frac{P_s}{P_r} = \ln\frac{P_s}{1-P_s} = logit(P_s) = \beta_0 + \beta_1 g + \beta_2 a + \beta_3 \frac{TTI}{y} + \beta_4 \frac{v}{v_f} \qquad (2)$$

Where P_s is the probability of stopping, P_r is the probability of running, β_i's are model constants, g is the gender (0 = female, 1 = male), a is the age (years), TTI is the time-to-intersection (s), y is the yellow time (s), v is the approach speed (km/h) and v_f is the speed limit (km/h). These variables were introduced into the model using a stepwise regression technique and including only the statistically significant variables as described in an earlier study [11, 12]. This testing tried various combinations of variables and finally identified this set of explanatory variables. These will be used for the remainder of the paper.

3. Random Forests

The random forest is an ensemble approach that is an effective tool in prediction [13]. Random forests do not suffer from over fitting because of the Law of Large Numbers. Ensembles are techniques that use the divide-and-conquer approach to improve performance. The main idea behind ensemble methods is that building a large group of simple models will give an overall improved performance. The group of weak models will give a resultant strong model. The random forest is a large group of un-pruned decision trees with randomized selection of features at each split. The well-known machine learning technique called Classification And Regression Tree (CART) is one of the common decision trees used in random forests [14]. Random forests start with the CART that, in ensemble terms, corresponds to the weak model. CART is a greedy and recursive top-down binary partitioning that divides feature space into sets of disjointed regions. These regions should be pure with respect to the response variable. The random forest algorithm for classification can be simply described, assuming the training dataset has *H* cases, *P* predictors, and *N* be the number of trees to build for each of the *N* iterations, in the following steps:

1. Sample *H* cases at random with replacement to create a bootstrap sample from the original dataset. The subset should be approximately 66% of the original training set and the other cases are repeated cases.
2. For some number $\sqrt{p}$ at each node, $\sqrt{p}$ predictor variables are selected at random from all the predictor variables.
3. The predictor variable out of the $\sqrt{p}$ predictors that provides the best split (minimum squared error), is used to make a binary split on that node.
4. At the next node, choose randomly another $\sqrt{p}$ predictors from all regressor variables and do the same.
5. Do not perform cost complexity pruning and save the tree as is with the other built trees before this iteration.

At the testing phase, the new arrived case is pushed down all the trees. Each tree will give a class label and the result is majority voting. This modeling technique could improve the modeling of driver stop/run behavior and thus will be tested as part of this research effort.

4. Adaptive Boosting Algorithm

The Adaptive Boosting (AdaBoost) is a machine learning algorithm that is based on the idea of incremental contribution [15]. AdaBoost was introduced as an answer to the question of whether a group of "weak" learner algorithms that each has low accuracy can be grouped together and boosted into an arbitrarily accurate "strong" learning algorithm. Before introducing the idea of AdaBoost, the traditional way of thinking in machine learning was based on choosing the most possible class discriminating features. In other words algorithms are required to be as class discriminatory as possible. Then, using these features to find the most discriminating learning algorithm to predict the class label for an unseen data instance, labeled data are collected and used as a training dataset. After that, feature selection is done. If the number of features is larger than the number of instances, then we suffer from the curse of dimensionality [16]. At this point, Principal Components Analysis [17] or Independent Component Analysis [18] can be used to reduce the space dimensions and address of curse of dimensionality problem . The last step after defining the feature space is choosing an algorithm such as K-NN [2], Support Vector Machine [19], or parametric models that give the highest accuracy.

AdaBoost does not use one classifier; instead it uses a set of weak classifiers, each is trained using the same training dataset but with a different weight distribution. Each of the weak learners focuses on the instances that are misclassified by the previous learner. The output of AdaBoost is the weighted average of all weak learner outputs. AdaBoost is proved to have smaller misclassification error compared to the summation of weak learners; also it has a bound on the generalization error. To describe the AdaBoost algorithm, let us assume the training set consists of *n* instances $\mathrm{T} = \{(\mathrm{x}_1, \mathrm{y}_1), \ldots, (\mathrm{x}_\mathrm{n}, \mathrm{y}_\mathrm{n})\}$ where, x_i is the vector of predictors that can be represented by a point in the multidimensional feature (predictor) space and y_i is the corresponding label. Because the algorithm predicts whether a segment will be congested or not, the focus will be on the binary classification problems with $\mathrm{y}_\mathrm{i} = \{-1, +1\}$. The pseudo-code of the classic AdaBoost is described in Table 32.

After training T weak learners the model is ready to predict the label for test instance (unseen) x_{test}. The label of the test instance is defined using Equation (3)

$$sign\left(\sum_{t=1}^{T} \propto_\mathrm{t} \mathrm{L}_\mathrm{t}(\mathrm{x}_\mathrm{test})\right) \qquad (3)$$

The label is set equal to 1 if the output of Equation (3) is positive and -1 if the output is negative. Again this algorithm will be tested on the dataset to develop driver stop/run models.

Table 32: Proposed AdaBoost Algorithm Pseudo Code

Set a probability distribution $P_t(x_i)$ over all the training samples. Initially, $P_t(x_i)$ is set to be uniform then it is modified iteratively with each selection of a weak classifier.
for iteration t do

1. Train weak learner L_t with weighted sample.
2. Test weak learner L_t on all data and get the predicted label $L_t(x_i)$ for each x_i.
3. Compare the predicted labels $L_t(x_i)$ with y_i for i = 1, …, n and calculate the classification error ϵ_t.
4. Calculate the trustiness level $\propto_t$ of the L_t the following equation
$$\propto_t = \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t}$$
5. Update F_t such that misclassified instances weights are increased using,

$P_{t+1}(x_i) = \frac{P_t(x_i) e^{-\propto_t y_i L_t(x_i)}}{Z_t}$, where Z_t serves as a normalize such that $\sum_{i=1}^{n} P_{t+1}(x_i) = 1$
end for

Predictor Selection

1. Overview

Previous studies that modeled driver stop/run behavior, found that the factors that affect the models are the driver's gender, age, the distance to the intersection, the travel time to the stop bar, and the approach speed [11, 20]. In this work we used gender, age, time-to-intersection divided by the yellow time in seconds (*TTI*/*y*), and the approaching speed divided by the speed limit (v/v_f). These variables were selected after testing many forms of the model as explained earlier. The normalization of *TTI* and *v* ensures that we have a non-dimensional model and thus can easily be generalized. Furthermore, the magnitude of the various model coefficients provides insight into the importance of specific variables in a non-dimensional model. The main idea of predictor selection is reducing the measuring cost without decreasing the prediction accuracy, so that the main focus is on investigating the usefulness of the last two predictors. This investigation goes through three steps. First, the frequency of each predictor is drawn and labeled by its class. Second, the learning algorithms are run using the original training dataset to determine the classification accuracy, then randomly permute the predictor and use the permuted dataset to train and test the new models and compute the classification accuracy. Third, train and test the models with the dataset after removing the predictor from the dataset. In the second step, the predictor under investigation is deleted from the dataset and a model is built using this version of dataset. Then the difference in classification accuracy between the full model and the model built using the dataset without the investigated predictor is calculated. In the third step, the investigated predictor is randomly permuted instead of deleted. Another model is built using a permuted version of the dataset. Subsequently, the difference in classification accuracy between the full model and the model built using the dataset with the permuted investigated predictor is calculated. In each step, as the difference increases, the predictor becomes more important.

A statistical test was conducted to quantify the independence of the proposed driver aggressiveness predictor with the gender and age variables. The statistical tests confirmed that

the stop propensity variable was independent of the age and gender variables at the 0.05 significance level.

2. Proposed Model Predictors

The predictors used in the classification or modeling of driver stop/run behavior are driver and intersection related. The used driver related parameters such as age and gender are not sufficient to measure the level of aggressiveness of the driver. This measure is computed by counting the number of stops the driver makes divided by the total number of observations (stops plus runs) during a yellow indication. In real situations, the vehicle would monitor the driver stop/run behavior and compute the frequency of stops at the onset of a yellow interval. If the value of the new predictor is very close to zero that means the driver rarely stops and thus is more aggressive than the driver who has large value of the new predictor. This new predictor is important because it gives prior information about the stop/run tendencies of that specific driver. It is envisioned that this parameter can be computed through some form of infrastructure-to-vehicle (I2V) communication in which the vehicle receives Signal Phasing and Timing (SPaT) information to identify the indication of the traffic signal. Vehicle-to-vehicle (V2V) communication would be required to identify if the lead vehicle was not forced to stop because the vehicle ahead of it stopped. Using the SPaT and surrounding vehicle information the vehicle would count the number of times the driver stopped and ran the yellow indication when s/he at the freedom to proceed.

With advances in telecommunication and computation power, connected cars have become a reality in which cars can exchange information with each other and with the traffic signal controller. We can use this technology advantage to allow vehicles to maintain a small record describing many behavioral related parameters. These parameters can describe the level of aggressiveness of the driver. One simple parameter we propose is the probability of stopping at the onset of a yellow indication at signalized intersections. This parameter goes from zero to one, with the value closer to one meaning that the driver is more conservative and more likely to stop. In order to motivate the usage of the new predictor, the predictor (TTI/y) is scattered versus the predictor (v/vf). The scatter plot of the instances is shown in Figure 29. Each scattered instance is labeled red if it is a stop instance and blue if it is a run instance. From the figure it is very clear that it is difficult to find a classifier that separates the two classes (run, stop). Specifically, the instances of the two classes overlap reducing discrimination power of the joined two predictors.

This severely overlapped classification will result in low true positive rates, high false positives, and low classification accuracies when we use any classifier to divide the predictor space. Alternatively, the (v/v_f) predictor is replaced with the new driver aggressiveness predictor, as demonstrated in Figure 30. The figure clearly demonstrates that the driver aggressiveness predictor increases the discrimination power when added to (TTI/y). Consequently, adding the new predictor decreases the overlap between the two classes and makes it easier to use a classifier to model the dataset.

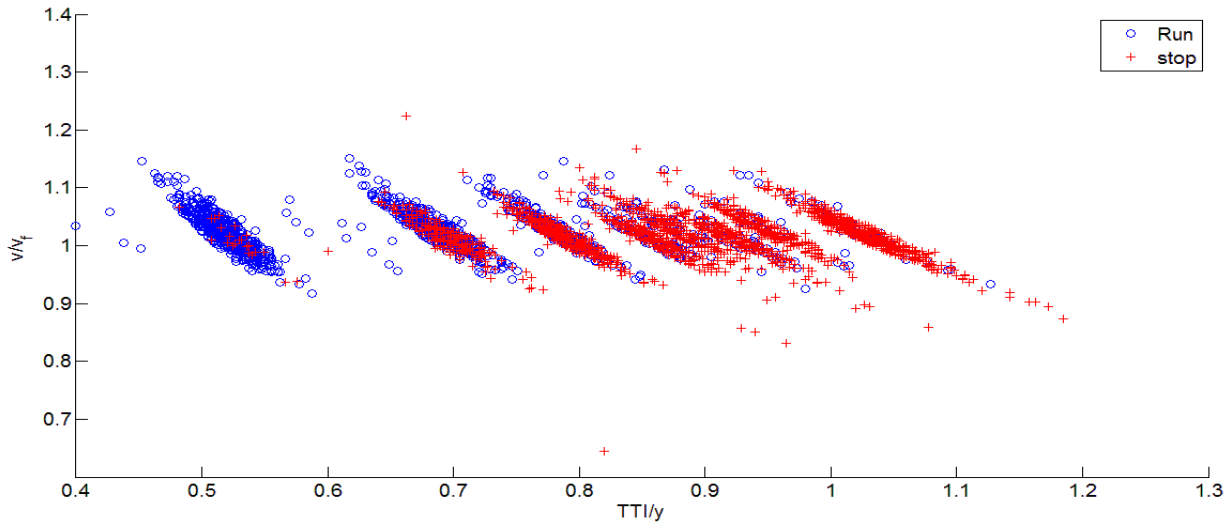


Figure 73: Scatter Plot of the Real Data Collected In the Field Experiment Using (TTI/y) Versus (v/vf) Predictors

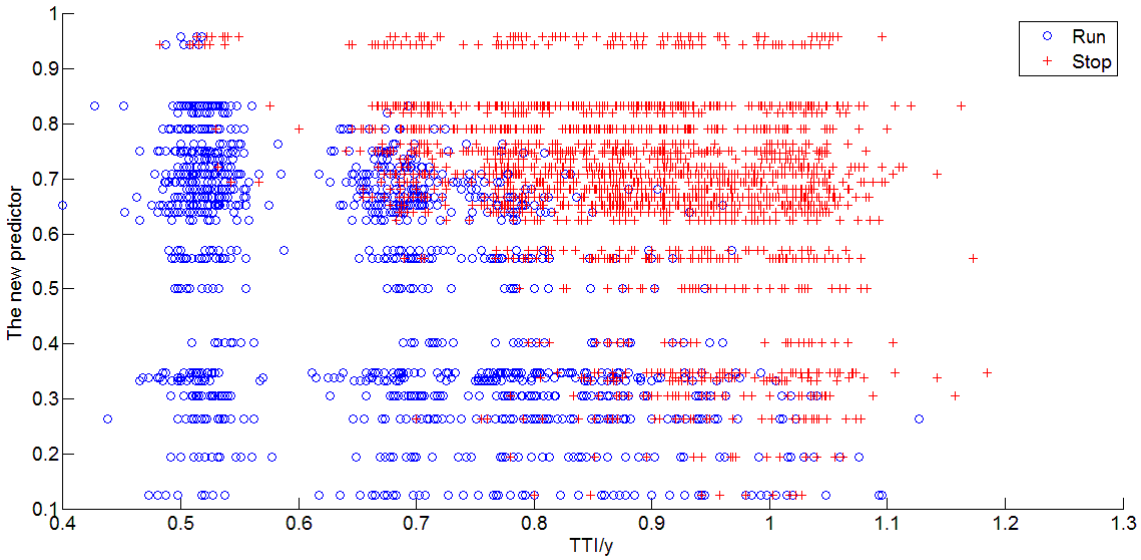


Figure 74: Scatter Plot of the Real Data Collected In the Field Experiment of (TTI/y) Versus the New Predictor

Data Description

The field experimental data used in this paper was conducted in the summer of 2008 at the Virginia Department of Transportation's (VDOT) Smart Road facility, located at the Virginia Tech Transportation Institute (VTTI). The Smart Road is a 3.5 km (2.2 mile) two-lane road with one four-way signalized intersection, and access to the road is controlled by electronic gateways, making the test facility a safe location to conduct field tests. The horizontal layout of the test section is fairly straight, and the vertical layout has a substantial grade of 3% [21]. Because participants turned around at the end of each run, half of the trials run by each participant were on a 3% upgrade and the other half were on a 3% downgrade.

Three vehicles were used in the study, one was driven by test participants (accompanied by the in-vehicle experimenter) and the other two vehicles were driven by two research assistants. A real-time data acquisition system (DAS) was installed inside the trunk of a 2004 Chevrolet Impala. The vehicle was also equipped with a differential Global Positioning System (GPS) unit, a longitudinal accelerometer, sensors for accelerator position and brake application, and a computer to run the different experimental scenarios. The data recording equipment had a communications link to the intersection signal control box that synchronized the vehicle data stream with changes in the traffic signal controller. Phase changes were controlled from the instrumented car using the GPS unit to determine the distance from the intersection and a wireless communications link to trigger the phase changes. Twenty-four licensed drivers were recruited in three equal age groups (under 40-years-old, 40 to 59-years-old, and 60-years-old or older); equal numbers of males and females were assigned to each group. The experiment involved test-track driving, for six sessions, once per day, where each participant was assigned to six different test conditions. The different test conditions were based on two instructed vehicle speeds of 72.4 km/h (45 mi/h) and 88.5 km/h (55 mi/h) and three platoon conditions (leading, following, and no other vehicle).

Participants drove loops on the Smart Road, crossing the four-way signalized intersection where the data were collected, 24 times for a total of 48 trials, where a trial consists of one approach to the intersection. Among the 48 trials, there were 24 trials in which each yellow trigger time to stop-line occurred four times. On the remaining 24 trials the signal indication remained green. This scheme would result in yellow/red signals being presented on 50 percent of the 48 trials; conversely, 50 percent of intersection approaches would be green indication. To examine whether willingness to stop varies with speed, the onset of yellow was based on the time-to-stop line (between 2.0 s and 4.6 s) at the instructed speed rather than on distance from the stop line. Radar was used to determine vehicle distance from the intersection. Outputs from the radar triggered the phase change events. Each participant drove to the Smart Road with the in-vehicle experimenter. The participant was asked to follow all normal traffic rules and to obey all traffic laws, and was also told that maintenance vehicles will occasionally be entering and leaving the road via a standard signalized intersection. These maintenance vehicles were the two confederate vehicles driven by trained experimenters who were involved in the study. The first vehicle was either leading or following the test vehicle, whereas the second vehicle was crossing the intersection from the conflicting approach when the traffic light was green.

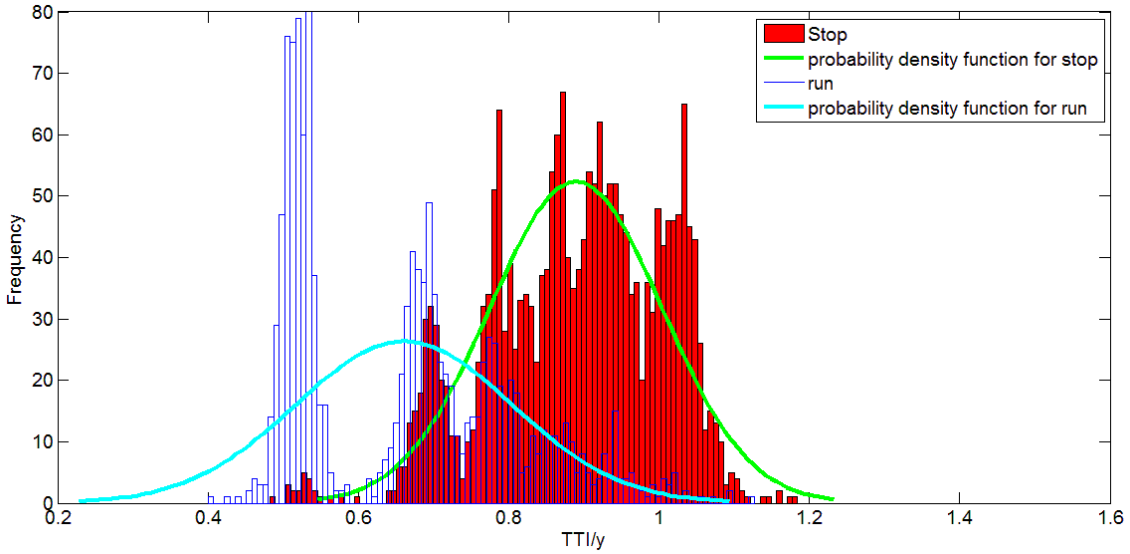
A 4-second yellow indication at the 72 km/h (45 mi/h) instructed speed and a 4.5-second yellow indication at the 88 km/h (55 mi/h) instructed speed were triggered for a total of 24 times (four repetitions at six distances). The yellow indications were triggered when the front of the test vehicle was 40.2, 54.3, 62.5, 70.4, 76.5, and 82.6 m (132, 178, 205, 231, 251, and 271 ft) from the intersection for the 72 km/h (45 mi/h) instructed speed and 56.7, 76.2, 86, 93.6, 101, and 113 m (186, 250, 282, 307, 331, and 371 ft) for the 88 km/h (55 mi/h) instructed speed to ensure that the entire dilemma zone was within the range.

Two major datasets are available from this experiment for the two instructed approach speeds; 72 km/h (45 mi/h) and 88 km/h (55 mi/h). Each dataset includes a complete tracking data every deci-second of the subject vehicle within about 150 m (500 ft) before and after the intersection. The 72 km/h dataset includes 1658 valid record (687 running records and 971 stopping records), whereas the 88 km/h dataset includes 1670 valid record (625 running records and 1045 stopping records). This yields a total of 3328 stop-run records (1312 running records and 2016 stopping records).

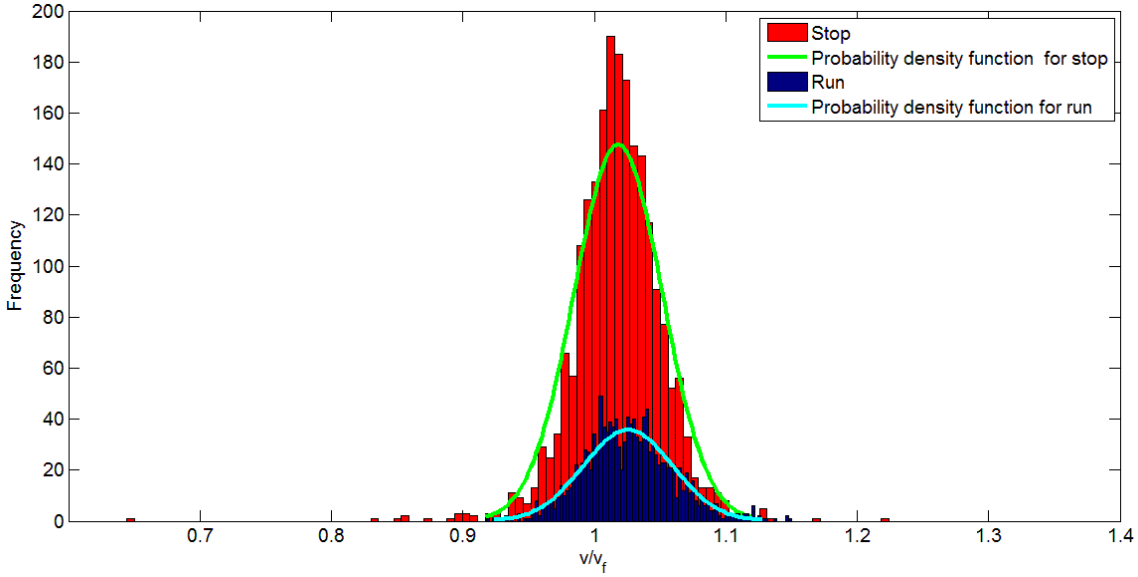
It should be noted that earlier publications demonstrated that the driver perception reaction time, deceleration levels, and stop/run behavior are consistent with other empirical observations in North America [22-27].

Data Analysis

Experimental work starts by drawing the histogram and probability density function using two components one for run and the other for stop as a function of (TTI/y) and (v/vf), as illustrated in Figure 31. Figure 75(a) demonstrates that (TTI/y) has a separation power (i.e. discrimination) in the driver stop/run decision, which means that the (TTI/y) predictor is useful in building the driver model. Alternatively, there is a complete overlap in the run and stop distribution in the case of the (v/vf) predictor, as illustrated in Figure 75(b), which means that the (v/vf) predictor may add limited information to the driver decision model. The next step after visually inspecting both predictors is to check the predictors again in presence of other predictors. Sometimes, a predictor seems not useful when checked alone but when other predictors are added the predictors together give better discrimination power and boost the performance of the machine learning algorithm as shown in Figure 4. Specifically, the figure demonstrates that when looking at a single dimension the histograms could appear to overlap, however when the data are analyzed in a multi-dimensional space the data clusters are distinct.

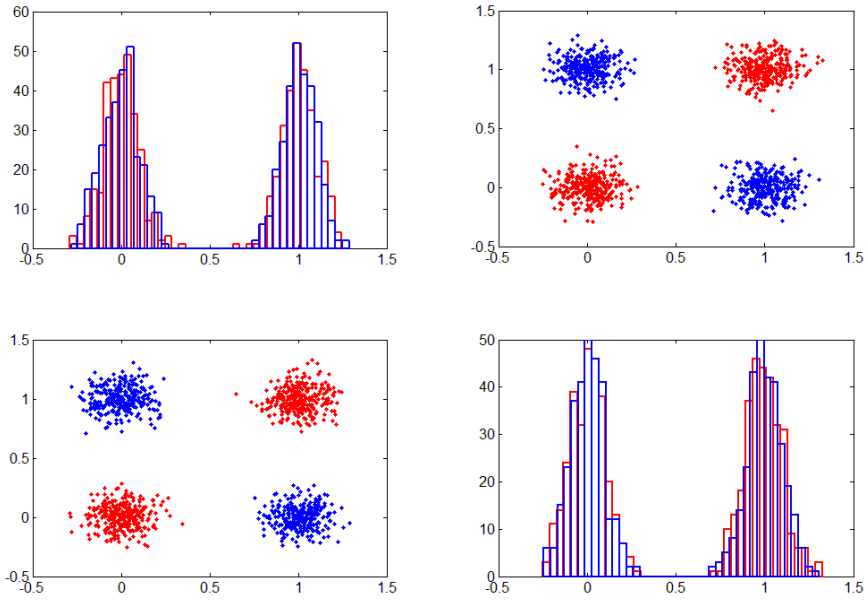


(a)

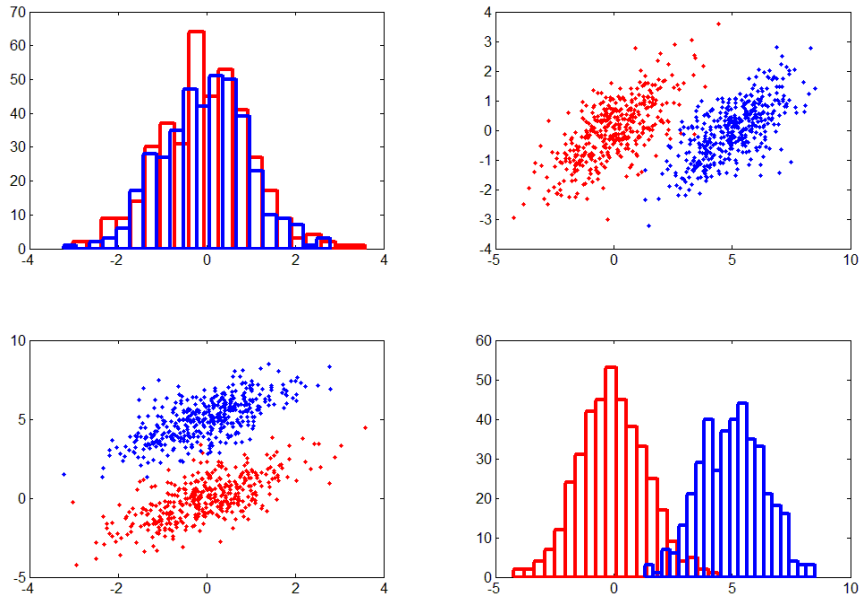


(b)

Figure 75: Histogram and Probability Density Function of (TTI/y) And (v/vf)



(a)



(b)

Figure 76: Demonstration of How Weak Predictors by Themselves Can Become Strong Predictors When Combined With Other Predictors [9]

Consequently, the next step after visually inspecting both predictors entails building the driver decision model using the training dataset. The model is first built using all predictors that were identified earlier in the literature and tested in this paper. Then, the (*TTI*/*y*) is removed and the model is rebuilt, tested, and compared to the full model. The classification accuracy with and without the (*TTI*/*y*) can then be quantified. The same procedure is repeated using the (v/v_f) predictor. The classification accuracy can be computed using Equation (4).

$$\text{classification accuracy} = 100\text{x}\frac{\text{True classifed run} + \text{True classifed stop}}{\text{Actual run} + \text{Actual stop}} \quad (4)$$

The next step is to randomly permute the predictor and using this version of the dataset build and test the models. The classification accuracy obtained from this step is then compared with the full model (all predictors included). At each run, the performance of the used machine learning algorithm is evaluated. The classification accuracy is calculated as an average of classification accuracy of each set of trials using the leave-one-out (LOO) cross-validation method [28]. In the LOO approach, the classifier is built using labeled data from 23 drivers only. The remaining driver is used as the unseen test driver and the classification accuracy is calculated for that day. The entire process is repeated where each driver is used once as a test driver and the average classification accuracy is calculated across all drivers. The same experiment was run considering different number of trees, different number of weak learners, and different number of nearest neighbors using the random forest, AdaBoost, and KNN algorithms, respectively. Figure 77 and Table 33 show the results of the experimental sets. Figure 77 is divided into two columns of figures, where the left column is for the (v/vf) predictor and the right column is for the (TTI/y) predictor. At each column, the top figure shows the classification accuracy for the full model, deleted predictor model, and permuted predictor model for the speed and TTI predictors.

Regarding the speed predictor, the difference between the model without this predictor and the full model is small. Also, the difference between the full model and the model with the permuted predictor is small for all learning algorithms. For models built using the AdaBoost classifier, the three curves are exactly the same. When the curves for the (TTI/y) predictor is checked, a big difference between the full model and the other models is observed (i.e. permuted and without the predictor models). The difference between the models is greater than 20%. From the above, it can be concluded that the (v/v_f) predictor is not important and does not explain a significant portion of the variability in the response and thus is recommended that it not be considered as a predictor. At the same time the (TTI/y) predictor is very important and removing this predictor will decrease the classification accuracy significantly. It should be noted that the *TTI* variable implicitly includes the *v* variable given that it is computed as the distance to the intersection divided by the speed of the vehicle at the onset of yellow.

The second set of the experiment is done to demonstrate the effectiveness of the new proposed driver aggressiveness measure is in the model development. Removing the speed predictor and adding the driver aggressiveness predictor and running the same machine learning algorithms that were run in the first experiments, produces considerable enhancements in the prediction accuracy. In the case of the logit model, the classification accuracy is increased from 80.05% to 90.56% after adding the new predictor. Figure 77 shows the improvement in the ACC after adding the new predictor.

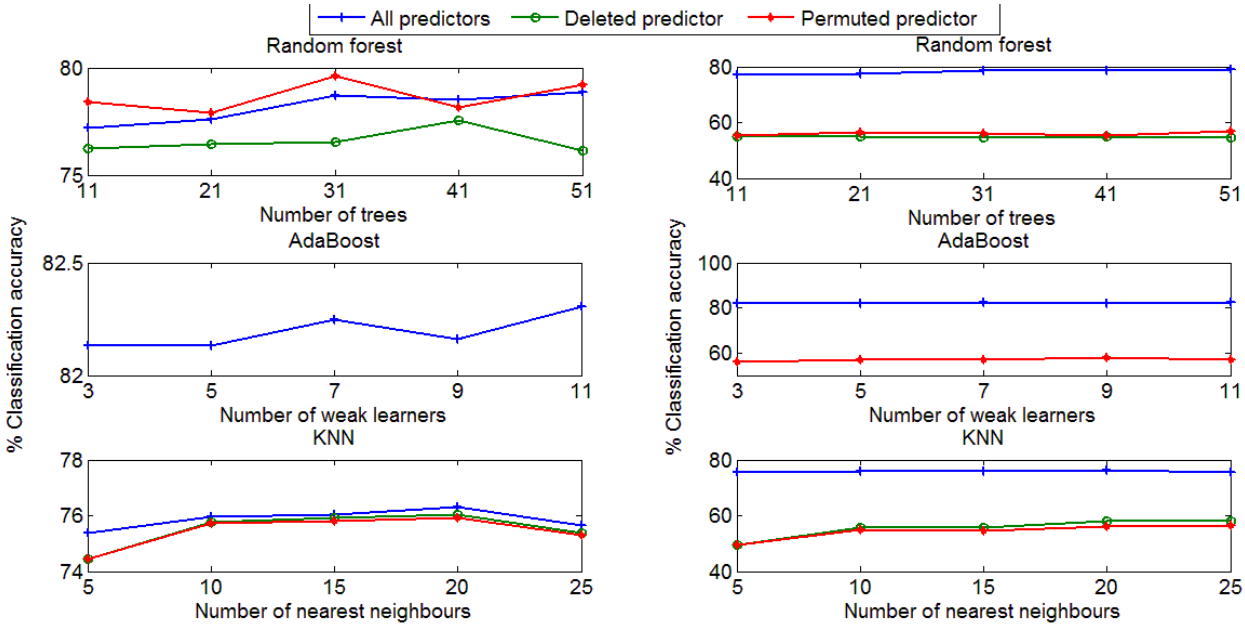


Figure 77: Classification Accuracy of Full, Deleted Predictor, and Permuted Predictor Models Using Different Learning Algorithms

Table 33: Classification Accuracy Using the Logit Model

	The (v/v_f) predictor.	**The (TTI/y) predictor**
All predictors (full model)	80.05%	
Deleted predictor	80.20%	59.75%
Permuted predictor	80.05%	59.67%

From Figure 78 it is evident that the new predictor boosts the classification accuracy, irrespective of the learning algorithm used to build the model. Before adding the proposed predictor, the AdaBoost model had an 82.3 percent classification accuracy, which was the highest classification accuracy. We tested the significance of the improvement of AdaBoost over the logit model using the matched pair permutation test. The null hypothesis for the test is that both the AdaBoost and logit models have similar classification accuracy. The p-value for the test is 0.0277 indicating that we can reject the null hypothesis at significance level of 0.05 and conclude that AdaBoost has better classification accuracy After adding the new predictor, the logit model became the best model with a 90.56% classification accuracy. We tested the significance of this improvement after adding the new predictor by comparing the classification accuracy of the logit model with and without this predictor. The null hypothesis for the test is the new predictor does not improve the classification accuracy of the logit model. The matched pair permutation test gives a p-value of 0.0001 and thus rejecting the null hypothesis. From the experiments, the addition of the driver aggressiveness predictor adds explanatory power to the model and thus is recommended for consideration in the modeling of driver stop/run behavior. Finally, all the learning algorithms used to model the driver stop/run behavior are all good except for the K-NN model. The availability of multiple modeling techniques to solve the same problem is beneficial given that it enables the use of a specific model depending on the available hardware platform.

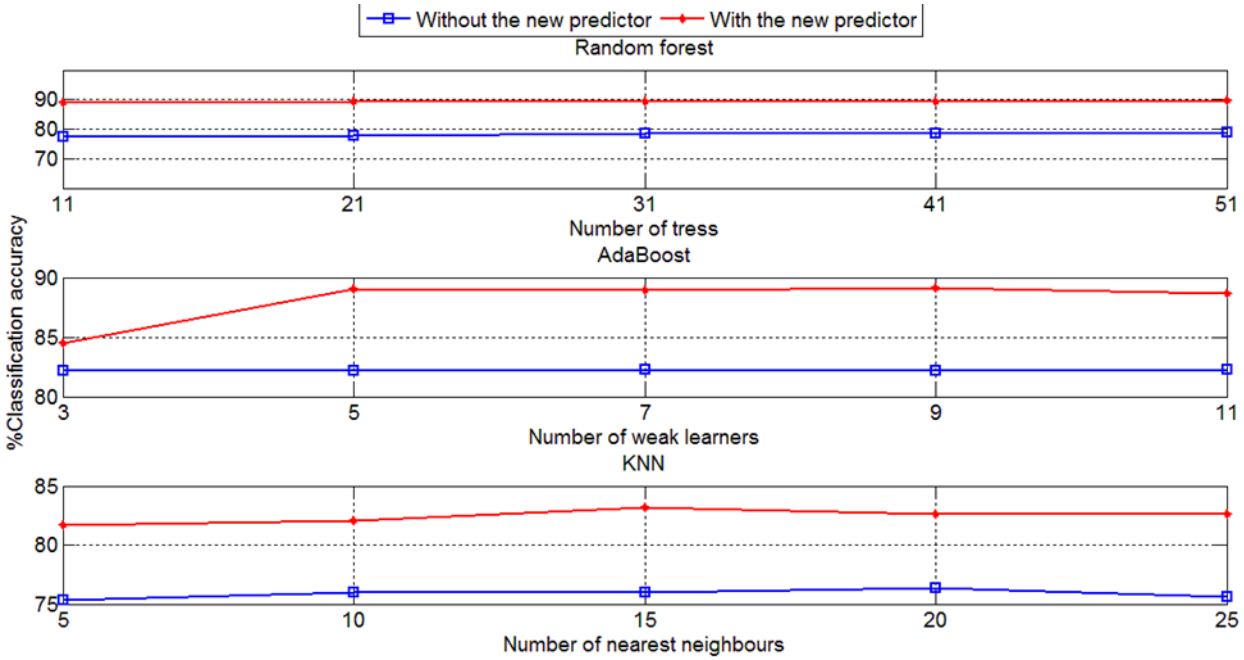


Figure 78: Comparison Between Classification Accuracy With/Without the New Predictor

Study Conclusions and Future Work

In this paper, we introduce a measure of driver aggressiveness into the modeling of driver stop/run behavior at the onset of a yellow indication. The driver aggressiveness parameter can be computed by monitoring the driver response to the onset of a yellow indication and counting the probability the driver would stop based on these observations. The parameter can then be added to the model after some period of monitoring. The experimental results show that the ability of the new predictor to explain part of the variability in the driver stop/run decision. Specifically, the addition of this predictor significantly increases the prediction accuracy by up to 10 percent. The study also demonstrates that machine learning algorithms can enhance the modeling of driver stop/run behavior and offer a modeling technique that in some cases is better than traditional statistical modeling techniques. All modeling algorithms were found to be comparable in terms of classification accuracy but machine learning techniques have the luxury of adapting to changes in driver behavior if a non-supervised learning technique were applied.

As is the case with any research effort further enhancements to the model are required to model driver stop/run behavior under inclement weather, considering the impact of the vehicle type (bus or truck) on the driver behavior, and developing real-time machine learning techniques that can adapt to changes in driver behavior.

Acknowledgments

The authors acknowledge the support of the Center for Technology Development at VTTI, Yu Gao, Sangjun Park, Aly Tawfik, Sashikanth Gurram, Stephanie Shupe, and Molataf Al-Aromah for running the tests. This work was supported in part by grants from the Virginia Department of Transportation, SAFETEA-LU funding, and the Mid-Atlantic Universities Transportation Center (MAUTC) funding.

References

[1] W. D. Jones, "Keeping cars from crashing," *IEEE Spectrum,* vol. 38, pp. 40–45, 2001.

[2] R. T. Trevor Hastie, Jerome Friedman *The Elements of Statistical Learning Data Mining, Inference, and Prediction*, 2009.

[3] D. Shinar and R. Compton, "Aggressive driving: An observational study of driver, vehicle, and situational variables," *Accident Analysis & Prevention,* vol. 36, pp. 429-437, 2004.

[4] S. S. M. Ali, N. Joshi, B. George, and L. Vanajakshi, "Application of Random Forest Algorithm to Classify Vehicles Detected by a Multiple Inductive Loop System," *2012 15th International Ieee Conference on Intelligent Transportation Systems (Itsc),* pp. 491-495, 2012.

[5] M.-H. Pham, A. Bhaskar, E. Chung, and A.-G. Dumont, "Random Forest Models for Identifying Motorway Rear-End Crash Risks Using Disaggregate Data," presented at the 13th International IEEE Annual Conference on Intelligent Transportation Systems, Madeira Island, Portugal, 2010.

[6] Qingchao Liu, Jian Lu, and S. Chen, "Traffic Incident D etection Using Random Forest," presented at the Transportation Research Board 92nd Annual Meeting, Washington DC.

[7] L. Yulan, M. L. Reyes, and J. D. Lee, "Real-Time Detection of Driver Cognitive Distraction Using Support Vector Machines," *Intelligent Transportation Systems, IEEE Transactions on,* vol. 8, pp. 340-350, 2007.

[8] F. Tango and M. Botta, "Real-Time Detection System of Driver Distraction Using Machine Learning," *Intelligent Transportation Systems, IEEE Transactions on,* vol. 14, pp. 894-905, 2013.

[9] I. Guyon, Andr, #233, and Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157-1182, 2003.

[10] J. H. Friedman, F. Baskett, and L. J. Shustek, "An Algorithm for Finding Nearest Neighbors," *Computers, IEEE Transactions on,* vol. C-24, pp. 1000-1006, 1975.

[11] A. M. M. Amer, H. A. Rakha, and I. El-Shawarby, "A Behavioral Modeling Framework of Driver Behavior at Onset of Yellow a Indication at Signalized Intersections," presented at the Transportation Research Board 89th Annual Meeting, Washington DC, 2010.

[12] H. Rakha, A. Amer, and I. El-Shawarby, "Modeling driver behavior within a signalized intersection approach decision-dilemma zone," *Transportation Research Record: Journal of the Transportation Research Board,* vol. 2069, pp. 16-25, 2008.

[13] L. Breiman, "Random Forests," *Machine Learning,* vol. 45, pp. 5-32, 2001.

[14] L. Breiman, *Classification and regression trees*. Belmont, Calif.: Wadsworth International Group, 1984.

[15] Y. Freund and R. Schapire, "A short introduction to boosting," *Japonese Society for Artificial Intelligence,* vol. 14, pp. 771-780, // 1999.

[16] R. Clarke, H. W. Ressom, A. Wang, J. Xuan, M. C. Liu, E. A. Gehan, *et al.*, "The properties of high-dimensional data spaces: implications for exploring gene and protein expression data," *Nat Rev Cancer,* vol. 8, pp. 37-49, 01//print 2008.

[17] L. Smith. (2002, A Tutorial on Principal Components Analysis.

[18] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural Networks,* vol. 13, pp. 411-430, 6// 2000.

[19] C. W. Hsu, C. C. Chang, and C. J. Lin, *A practical guide to support vector classification*, 2003.

[20] T. Hicks, R. Tao, and E. Tabacek, "Observations of Driver Behavior in Response to Yellow at Nine Intersections in Maryland," presented at the Transportation Research Board 84th Annual Meeting, Washington D.C., 2005.
[21] H. Rakha, I. Lucic, S. H. Demarchi, J. R. Setti, and M. Van Aerde, "Vehicle dynamics model for predicting maximum truck acceleration levels," *Journal of Transportation Engineering,* vol. 127, pp. 418-425, 2001.
[22] I. El-Shawarby, H. Rakha, V. Inman, and G. Davis, "Effect of yellow-phase trigger on driver behavior at high-speed signalized intersections," in *Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE*, 2006, pp. 683-688.
[23] H. Rakha, I. El-Shawarby, and J. R. Setti, "Characterizing driver behavior on signalized intersection approaches at the onset of a yellow-phase trigger," *IEEE Transactions on Intelligent Transportation Systems,* vol. 8, pp. 630-640, 2007.
[24] H. Rakha, A. Amer, and I. El-Shawarby, "Modeling Driver Behavior within Signalized Intersection Approach Decision-Dilemma Zone," in *Transportation Research Board 87th Annual Meeting [Paper 08-0475]*, Washington D.C., 2008.
[25] I. El-Shawarby, A. Amer, and H. Rakha, "Driver Stopping Behavior an High-Speed Signalized Intersection Approaches," *Transportation Research Record,* pp. 60-69, 2008.
[26] I. El-Shawarby, H. A. Rakha, A. M. M. Amer, and C. McGhee, "Characterization of Driver Perception Reaction Time at Onset of Yellow Indication," 2010.
[27] A. Amer, H. Rakha, and I. El-Shawarby, "Novel Stochastic Procedure for Designing Yellow Intervals at Signalized Intersections," *Journal of transportation engineering,* vol. 138, pp. 751-759, 2011.
[28] T. Evgeniou, M. Pontil, and A. Elisseeff, "Leave One Out Error, Stability, and Generalization of Voting Combinations of Classifiers," *Machine Learning,* vol. 55, pp. 71-97, 2004/04/01 2004.

Chapter 14: Modeling Driver Stop/Run Behavior at the Onset of a Yellow Indication Considering Driver Run Tendency and Roadway Surface Conditions

This chapter is based on
Mohammed Elhenawy, Arash Jahangiri, Hesham Rakha, and Ihab El-Shawarby, "Modeling Driver Stop/Run Behavior at the Onset of a Yellow Indication Considering Driver Run Tendency and Roadway Surface Conditions," under review journal paper.

Abstract

The ability to model driver stop/run behavior at signalized intersections considering the roadway surface condition is critical in the design of advanced driver assistance systems. Such systems can reduce intersection crashes and fatalities by predicting driver stop/run behavior. The research presented in this paper uses data collected from two controlled field experiments on the Smart Road at the Virginia Tech Transportation Institute (VTTI) to model driver stop/run behavior at the onset of a yellow indication for different roadway surface conditions. The paper offers two contributions. First, it introduces a new predictor related to driver aggressiveness and demonstrates that this measure enhances the modeling of driver stop/run behavior. Second, it applies well-known Artificial Intelligence techniques including: Adaptive Boosting (AdaBoost), random forest, and Support Vector Machine (SVM) algorithms as well as traditional logistic regression techniques on the data in order to develop a model that can be used by traffic signal controllers to predict driver stop/run decisions in a connected vehicle environment. The research demonstrates that by adding the driver aggressiveness predictor to the model, there is a statistically significant increase in the model accuracy. Moreover, the false alarm rate is remarkably reduced but not statistically significant. The study demonstrates that, for the subject data, the SVM machine learning algorithm performs the best in terms of optimum classification accuracy and false positive rates. However, the SVM model produces the best performance in terms of the classification accuracy only.

Introduction

With advances in sensing, communications, and computational technologies, research in the area of vehicle safety is increasing. Most new cars have active safety features including anti-lock braking and adaptive cruise control systems to reduce road accidents [1]. In the US, the Department of Transportation (DOT) reported 32,367 fatalities caused by road accidents in 2011 [2]. A significant percentage of these road accidents occurred at signalized intersections as a result of driver behavior in the decision/dilemma zone while approaching signalized intersections [3].

When drivers approaching a signalized intersection encounter a yellow light, they need to make a decision whether to stop or proceed. When they are too far from the intersection they normally tend to stop as they know when they reach the stop bar the traffic light would be red. However, as their distance to the intersection at the yellow onset decreases the drivers may make a wrong decision as to stopping or proceeding, which may lead to crashes with the side-street traffic or rear-end crashes. The area in which the drivers need to make the decision to stop or go is known as the Dilemma Zone (DZ) [4]. When the DZ exists, it is a small strip of road located between the minimum stopping distance and the maximum clearing distance. Where clearing

distance is defined as the distance required for an approaching vehicle to pass the stop bar before the end of yellow indication and the stopping distance is defined as the distance required for an approaching vehicle to come to a complete stop before the stop bar. Hence, the DZ is not the whole distance to the stop bar. DZ was first introduced by Gazis, et al. [5] and then have been studied in many other studies such as [4, 6-16].

Several factors influence the driver behavior when they are in the DZ area. The factors can be divided into three categories; driver-related, intersection related and vehicle-related. These factors that have been studied throughout the literature [4, 8, 12-14, 17-20] include the driver perception-reaction time; the driver's acceptable deceleration rate; the driver's age; the driver's gender; the time-to-intersection (TTI) at the onset of yellow; distance-to-intersection DTI at the onset of yellow; approach speed; vehicle type; presence of side-street vehicles, pedestrians, bicycles, or opposing vehicles waiting to turn left; flow rate; length of yellow interval; cycle length; presence of police. Moreover, El-Shawarby et al. [21] compared the driver stopping/running probabilities in clear weather and in rainy weather and found a slight shift between the two probabilities. El-Shawarby et al. [21], correlated this shift to the decrease in the probability of stopping in case of wet pavement surface and rainy weather conditions. Consequently, we added the roadway surface condition as an input variable to the proposed classifiers. It appears that the driver aggressiveness as a factor has not been considered in the past studies, and thus this paper attempts to develop a factor that can capture the driver aggressiveness and then use it in driver stop/run models.

The intersection safety needs identification report published by federal highway administration in July 2009 showed that in 2007, 22% of the total fatal crashes were intersection-related with an estimated cost of 27.8 US billion dollars while 44.8% of the total injury crashes were also intersection-related with an estimated cost of 51.3 US billion dollars [22]. Based on The National Highway Traffic Safety Administration, two-thirds of all fatal crashes are caused by aggressive driving [23]. Therefore, aggressive driving is critical in modeling driver stop/run behavior at signalized intersections; however measuring driver aggressiveness may not be plausible. In a previous research study, five driver actions were used to measure aggressive driving behavior. These five measures include: short or long honk of the horn, cutting in front of other vehicles in a passing lane maneuver, cutting in front of other vehicles in a multi-lane passing maneuver, and passing one or more vehicles by driving on the shoulder and then cutting in [24]. Other studies classified drivers into three categories aggressive, conservative, and normal groups based on their decision (stop /run) and the distances to the stop line when the signal turns yellow[25, 26] . In our previous work, we proposed the use of the frequency of running a yellow indication as a measure of driver aggressiveness [27]. In the current study, a better sold and formal definition and formulation of the driver aggressiveness is proposed. The measure proposed here is a continuous measure of aggressiveness varies from zero to one and not categories as in the current practice.

Consider a vehicle approaching a signalized intersection, our goal is to build a model that predicts driver stop or run behavior at the onset of the yellow indication. This model uses many predictors such as TTI and driver's age to predict the driver decision. Because in real-life, different drivers behave differently, we added the proposed predictor to explain some of the variation between drivers based on their history. Such a model should be one of the main building blocks in more advanced driver assistance systems. These systems should be able to predict the driver behavior and warn them if their decision is incorrect. Moreover, it would warn the driver if there is any potential violation from other drivers/vehicles approaching the

intersection. The system should ensure the algorithm produces minimum false positives in order to encourage drivers trust their output.

The past two decades have seen numerous research efforts and advances in both machine learning techniques and computer computational power. Many machine learning techniques require a large number of computations and are infeasible without computers. The available machine learning algorithms and computational power have made such techniques feasible for real-time implementation. Transportation engineers are among people who are interested in applying these algorithms to address transportation problems. This interest increases with the availability of datasets from fixed detectors, data probes and intelligent transportation systems (ITSs). Recently, some machine learning algorithms were used in the transportation field, including: classifying and counting vehicles detected by multiple inductive loop detectors [28], identifying motorway rear-end crash risks using disaggregate data [29], automatic traffic incident detection [30], real-time detection of driver distraction [31, 32], transportation mode recognition using smartphone sensor data [33, 34], and video-based highway asset segmentation and recognition [35]. Modeling driver stop/run behavior at signalized intersections is very important and is ideal for applying machine learning techniques [36]. At first glance, driver stop/run behavior modeling seems to be a good candidate for a straightforward application of machine learning algorithms. Observations of driver stop/run behavior from naturalistic driver datasets or from controlled field experiment datasets can be used to train machine learning algorithms. The trained models can then be used to predict future driver decisions for implementation in in-vehicle safety systems. However, machine learning modeling of driver stop/run behavior faces some challenges including the need for large labeled datasets, driver stop/run behavior drift, and computational complexity.

In this paper, we introduce a new parameter related to the driver aggressiveness. This new predictor can be observed directly from historical driver stop/run behavior. Using this new predictor, we demonstrate that the modeling of driver stop/run behavior can be enhanced. The use of such models can then be integrated with in-vehicle safety systems to predict the action of a driver and thus warn other drivers or take action to ensure that no vehicle collisions occur.

Methods

We define our problem by defining the input variables (predictors) and the output (response). There are six predictors used as inputs to the model:

G is the gender (1 = female, 0 = male),
A is the age (years),
TTI is the time-to-intersection (s),
V is the approach speed (km/h),
S is the roadway surface condition (0= rainy/wet surface, 1=dry),
D is the new proposed driver aggressiveness predictor which will be discussed in the next section.
G is the gender (1 = female, 0 = male),

The output of the model is a label shows if the driver will stop or run. This kind of modeling problems is a binary classification problem. Classification is one of the main areas of machine learning. In order to build such models, labeled data is needed. Once the model is built, it can be used to classify (predict) the behavior of the driver at the onset of yellow light based on the values of the input predictors. There are many machine learning classification algorithms that are suitable for the driver run/stop modeling. A brief introduction to the modeling algorithms

used in this paper is presented to familiarize readers with these emerging techniques. The strengths of each modeling technique are presented given that the models will be compared later on the same dataset. They range from simple algorithms to complicated and high computational demand algorithms.

1. Generalized Linear Models

Generalized linear models (GLMs) are statistical models that model responses as linear combinations of regressors (predictors). It can be used as regression models for continuous dependent variables, models for rates and proportions, binary, ordinal and multinomial variables, and counts. The GLM approach has two significant advantages: (a) its theoretical framework is valid for many commonly encountered datasets; (b) the software implementation is simple because the same algorithm can be used for estimation, inference and assessing model adequacy for all GLMs. Logistic regression is a commonly used GLM when the response variable is a binomial response and is thus used to model driver stop/run behavior, as was demonstrated in an earlier study [37].

2. Random Forests

The random forest is an ensemble approach that is an effective tool in prediction [38]. Breiman used the Strong Law of Large Numbers and proofed that Random forests do not suffer from over fitting as more trees are added[39]. The main idea behind ensemble methods is that building a large group of simple models will give an overall improved performance. The random forest is a large group of un-pruned decision trees with a randomized selection of features at each split. The well-known machine learning technique, Classification And Regression Tree (CART), is one of the common decision trees used in random forests [40]. Random forests start with the CART that, in ensemble terms, corresponds to the weak model. CART starts by splitting the feature space into two partitions (children) such that its objective function is locally optimized. For each child, CART repeats this splitting process until the stopping criteria is reached. The cases in each region have (almost) the same outcome. The random forest algorithm for classification can be simply described, assuming the training dataset has H cases, P predictors, and M be the number of trees to build for each of the M iterations, in the following steps:

1. Sample H cases at random with replacement to create a bootstrap sample from the original dataset. The subset should be approximately 66% of the original training set and the other cases are repeated cases.
2. For some number, $\sqrt{p}$ at each node, $\sqrt{p}$ predictor variables are selected randomly from all the predictor variables.
3. The predictor variable out of the $\sqrt{p}$ predictors that provides the best split is used to produce a binary split on that node.
4. At the next node, choose randomly another $\sqrt{p}$ predictors from all regressor variables and do the same.
5. Do not perform cost complexity pruning and save the tree as is with the other built trees produced from earlier iterations.

At the testing phase, the newly arrived case is pushed down all the trees. Each tree votes for one class by providing a class label. The output of the random forest is the class which has most votes. This modeling technique could improve the modeling of driver stop/run behavior and thus will be tested as part of this research effort.

3. Adaptive Boosting Algorithm

The Adaptive Boosting (AdaBoost) is a machine learning algorithm that is based on the idea of incremental contribution[41]. AdaBoost was introduced as an answer to the question of whether a group of "weak" learner algorithms that each has low accuracy can be grouped into learning algorithm with high accuracy. Before introducing the idea of AdaBoost, the traditional way of thinking in machine learning was based on choosing the most possible class discriminating features. In other words, algorithms are required to be as class discriminatory as possible. Then, using these features to find the most discriminating learning algorithm to predict the class label of an unseen data instance, labeled data are collected and used as a training dataset. After that, feature selection is done. If the number of features is larger than the number of instances, then we suffer from the curse of dimensionality [42]. At this point, Principal Components Analysis [43] or Independent Component Analysis [44] can be used to reduce the space dimensions and address the dimensionality problem. The last step after defining the feature space is choosing an algorithm such as K-NN [2], Support Vector Machine [45], or parametric models that give the highest accuracy.

AdaBoost does not use one classifier; instead it uses a set of weak classifiers, each is trained using the same training dataset but with a different weight distribution. Each of the weak learners focuses on the instances that are misclassified by the previous learner. The output of AdaBoost is the weighted average of all weak learner outputs. AdaBoost is likely to have smaller misclassification error compared to the summation of weak learners; also it has a bound on the generalization error[41, 46].To describe the AdaBoost algorithm, let us assume the training set consists of K instances $T = \{(x_1, y_1), \dots, (x_K, y_K)\}$ where, x_i is the vector of input predictors that can be represented by a point in the multidimensional feature (predictor) space and y_i is the corresponding label (response). Because the algorithm predicts whether the driver will run or stop, the focus will be on the binary classification problems with the label y_i is either $+1$ or-1. The pseudo-code of the classic Adaboost is described in Table 11.

After training T weak learners the model is ready to predict the label for test instance (unseen) x_{test}. The label of the test instance is defined using Equation (1).

$$\text{sign}\left(\sum_{t=1}^{T} \propto_t L_t(x_{test})\right) \qquad (1)$$

The label is set equal to 1 if the output of Equation (1) is positive and -1 if the output is negative. Again this algorithm will be tested on the dataset to develop driver stop/run models.

Table 34: Proposed AdaBoost Algorithm Pseudo Code

Set a probability distribution $P_t(x_i)$ over all the training samples. Initially, $P_t(x_i)$ is set to be uniform then it is modified iteratively with each selection of a weak classifier.
for iteration t do

1. Train weak learner L_t with weighted sample.
2. Test weak learner L_t on all data and get the predicted label $L_t(x_i)$ for each x_i.
3. Compare the predicted labels $L_t(x_i)$ with y_i for i = 1, …, n and calculate the classification error ϵ_t.
4. Calculate the trustiness level $\propto_t$ of the L_t the following equation

$$\propto_t = \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t}$$

5. Update F_t such that misclassified instances weights are increased using,

$P_{t+1}(x_i) = \frac{P_t(x_i) e^{-\propto_t y_i L_t(x_i)}}{Z_t}$, where Z_t serves as a normalizer such that $\sum_{i=1}^{n} P_{t+1}(x_i) = 1$

end for

4. Support Vector Machine

Support Vector Machine (SVM) is a rather complex machine learning technique that can be employed in classification problems. SVM is known as a large margin classifier, which means that while this method attempts to find decision boundaries between different classes, it tries to maximize the gap or margin between classes.

The objective function of the SVM formulation and the associated constraints are presented below in Equation (2) through Equation (4) [47]. The sum of two terms are minimized in the objective function; minimizing the first term is basically equivalent to maximizing the margin between classes, and the second term consists of an error term multiplied by the regularization (penalty) parameter denoted by C. The regularization is designed to deal with the problem of over-fitting a model. The value of the C parameter should be adjusted to obtain the best possible performance.

$$\min_{w,b,\xi} \left(\frac{1}{2} w^T w + C \sum_{n=1}^{K} \xi_n \right) \quad (2)$$

Subject to:

$$y_n (w^T \phi(x_n) + b) \geq 1 - \xi_n \, , n = 1, \ldots, K \quad (3)$$

$$\xi_n \geq 0 \, , n = 1, \ldots, K \quad (4)$$

where,

- w Parameters to define decision boundary between classes
- C Regularization (or penalty) parameter
- ξ_n Error parameter to denote margin violation
- b Intercept associated with decision boundaries
- $\phi(x_n)$ Function to transform data from X space into some Z space
- K the number of observations in the dataset

When using SVM, the data are transformed from the X space to the Z space using some function $\phi(x_n)$. The reason for conducting the transformation is to obtain a space in which identifying decision boundaries between classes becomes easier. However, in solving the problem, there is no need to actually do the transformation. Instead, some other functions, known as kernels, are adopted. The kernels, which appear in the dual formulation of the problem, correspond to the vector inner product in the Z space. To construct the model, the kernel type

should be selected (e.g. linear, polynomial, Gaussian). Depending on the problem, one kernel may perform better than the other. Some practical considerations suggest using certain kernels for specific problems based on the data size [45].

Proposed Driver Aggressiveness Predictor

Models used to classify the driver stop/run behavior, use driver- and intersection-related predictors. Driver's age and gender are used as the driver-related predictors. But these predictors are not sufficient to measure the level of aggressiveness of the driver. Thanks to advances in telecommunication and computation power, connected vehicles have become a reality in which vehicles can exchange information with each other (V2V) and with the traffic signal controllers (V2I). We can use this technology advantage to allow vehicles to learn from the behavior of its driver by maintaining a small record describing many behavioral-related parameters. These parameters can describe the level of aggressiveness of the driver. One intuitive parameter we propose is the probability of running at the onset of a yellow indication at signalized intersections when stopping is a better decision.

We propose a new predictor that can be used as a measure of the driver's aggressiveness. The new measure is based on a count of the number of runs the driver makes when the time-to-intersection at the onset of the yellow indication is greater than the yellow time and his/her speed is equal or greater than the posted speed limit. The value of the new predictor θ_j for driver j, can be estimated based on the Bayesian approach by finding the posterior density distribution, as shown in Equation (5).

$$f\left(\theta_j \middle| y_{j1}, y_{j2}, \dots, y_{jN_j}\right) \propto f\left(y_{j1}, y_{j2}, \dots, y_{jN_j} \middle| \theta_j\right) f(\theta_j) \quad (5)$$

where,

$f\left(y_{j1}, y_{j2}, \dots, y_{jN_j} \middle| \theta_j\right)$ is the sampling distribution for driver j

$f(\theta_j)$ is the prior distribution for driver j

N_j is the number of cases for driver j when the time-to-intersection at the onset of the yellow indication is greater than the yellow time and his/her speed is equal or greater than the posted speed limit

The distribution of the $f(y_{ji}|\theta_j)$ is Bernoulli because the random variable is either one or zero. The new predictor can take any value between zero and one. This means the domain of the $f(\theta_j)$ is from zero to one [0, 1]. So that reasonable choice of $f(\theta_j)$ is a Beta distribution and the problem can be viewed as a Beta-Bernoulli model, as shown in Equation (6).

$$f(y_{ji}|\theta_j) \sim \text{Bernoulli}\ (\theta_j) \text{ where } y_{ji} \text{ is the case i for driver j}$$
$$\theta_j \sim \text{Beta}\ (a, b) \quad (6)$$
$$a = 1 \text{ and } b = 1000$$
$$f\left(\theta_j \middle| y_{j1}, y_{j2}, \dots, y_{jN_j}\right) \propto \left\{\prod_{i=1}^{N_j} \theta_j^{y_{ji}} (1-\theta_j)^{1-y_{ji}}\right\} \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \theta_j^{a-1} (1-\theta_j)^{b-1}$$

The above equation can be simplified as shown in Equation (7)

$$f\left(\theta_j \middle| y_{j1}, y_{j2}, \dots, y_{jN_j}\right) \propto \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \left\{\theta_j^{\sum_{i=1}^{N_j} y_i + a - 1} (1-\theta_j)^{N_j - \sum_{i=1}^{N_j} y_i + b - 1}\right\} \quad (7)$$

By removing the constants $\frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}$ from the above equation, the kernel of the posterior is Beta distribution with the parameters shown in Equation (8).

$$f\left(\theta_j \middle| y_{j1}, y_{j2}, \dots, y_{jN_j}\right) \sim \text{Beta}\ \left(\sum_{i=1}^{N_j} y_i + a\ , N_j - \sum_{i=1}^{N_j} y_i + b\right) \quad (8)$$

From Equation (8) we can estimate the expectation $E\left[f\left(\theta_j \middle| y_{j1}, y_{j2}, \dots, y_{jN_j}\right)\right]$ and use it as the aggressiveness measure for driver j, as shown in Equation (9).

$$E\left[f\left(\theta_j \middle| y_{j1}, y_{j2}, \dots, y_{jN_j}\right)\right] = \frac{\sum_{i=1}^{N_j} y_i + a}{a+b+N_j} \quad (9)$$

To calculate the new predictor for each driver, we find the number of cases N_j for which both the driver's time-to-intersection at the onset of the yellow indication is greater than the yellow time and the speed of the vehicle is greater than or equal to the posted speed limit. Let $y_{ji} = 1$ if the driver's decision was run and $y_{ji} = 0$ otherwise. We adopt equation (9) to estimate the aggressiveness of each driver. If the value of the new predictor is close to one that means the driver rarely stops and thus is more aggressive than the driver who has a smaller value of the new predictor. This new predictor is important because it captures the stop/run tendencies of that specific driver. It is envisioned that the computation of this new predictor can be done through some form of infrastructure-to-vehicle (I2V) communication in which the vehicle receives Signal Phasing and Timing (SPaT) information to identify the indication of the traffic signal. Moreover, Vehicle-to-vehicle (V2V) communication would be required to exchange information with surrounding vehicles to ensure that the driver was not forced to stop because the vehicle ahead of it stopped. Using the SPaT and surrounding vehicle information the vehicle would count the number of times the driver stopped and ran the yellow indication when she or he has the freedom to proceed.

Data Description

The data used in this paper were collected from two different field experiments which were done at two different roadway surface conditions. Both experiments were conducted on the Virginia Department of Transportation's (VDOT) Smart Road facility, located at the Virginia Tech Transportation Institute (VTTI). The length of the Smart Road is a 3.5 km (2.2 mile). It is a two-lane road with one four-way signalized intersection. The entrance to this test facility is through an electronic gate to make it safe location to conduct field tests. The horizontal layout of the test section is fairly straight, and the vertical layout has a substantial grade of 3% [48].

1. Dry Roadway Surface Field Experiment

This experiment was only run in clear weather and dry pavement surface condition. in this experiment, Three vehicles were used, one was driven by test participants and accompanied with in-vehicle experimenter [37]. The other two vehicles were driven by trained experimenters who were involved in the study. One of them following the test vehicle, whereas the other vehicle was crossing the intersection from the conflicting approach when the traffic light was green. The test vehicle is equipped with a real-time data acquisition system (DAS), differential Global Positioning System (GPS) unit, a longitudinal accelerometer, sensors for accelerator position and brake application, and a computer to run the different experimental scenarios.

The vehicle data stream was synchronized with changes in the traffic signal controller by the communication channel that linked the data recording equipment to the intersection signal controller. The phase changes were triggered by the test vehicle using a GPS unit that determined the distance from the intersection. Twenty-four licensed drivers were recruited in three equal age groups (under 40-years-old, 40 to 59-years-old, and 60-years-old or older); each group was male-female balanced.

Participants were asked to follow all normal traffic rules and to obey all traffic laws while driving. They drove loops on the Smart Road at a 72.4 km/h (45 mi/h) instructed speed, crossing

the four-way signalized intersection 24 times for a total of 48 trials, where a trial consisted of one approach to the intersection. Among the 48 trials, a 4-second yellow indication at the 72 km/h (45 mi/h) instructed speed were triggered for a total of 24 times (four repetitions at six distances). The yellow indications were triggered when the front of the test vehicle was 40.2, 54.3, 62.5, 70.4, 76.5, and 82.6 m (132, 178, 205, 231, 251, and 271 ft.) from the intersection for the 72 km/h (45 mi/h) instructed speed to ensure that the entire dilemma zone was within the range. On the remaining 24 randomized trials, the signal indication remained green. The Dry dataset included a complete tracking data every deci-second of the subject vehicle within about 150 m (500 ft.) before and after the intersection.

2. Rainy/Wet Roadway Surface Field Experiment

This experiment was only run in rainy weather and wet pavement surface condition. in this experiment, two vehicles were used because the platooning factor was not found to be a significant factor in the stop/go model [37]. One vehicle was driven by test participants (accompanied by the in-vehicle experimenter) and the other vehicle was driven by a trained research assistant to simulate real-world conditions[19]. The confederate vehicle crossed the intersection from the side street when the signal was red for the test vehicle. The participant was asked to follow all normal traffic rules and to obey all traffic laws.

As was the case in the first study, the test vehicle was equipped with a differential Global Positioning System (GPS), a real-time Data Acquisition System (DAS), and a computer to run the different experimental scenarios. A communications link to the intersection signal control box was used by the data recording equipment to synchronize the vehicle data stream with changes in the traffic. The two vehicles were equipped with a communications system between vehicles, operated by the research assistants.

Twenty-six drivers were recruited in three age groups (under 40-years-old, 40 to 59-years-old, and 60 years of age or older), equal number of male and female participants were assigned to each group.

During the test runs the participants drove a distance of 1.6 km (1 mile) going downhill to approach the intersection followed by a 0.5 km (0.3 mile) leg to a high-speed turnaround, and another 0.5 km (0.3 mile) approach going back to the intersection. A 4-second yellow interval at the 72.4 km/h (45 mi/h) instructed speed was triggered for a total of 24 times (four repetitions at six distances). The yellow indications were triggered when the front of the test vehicle was 54.3, 62.5, 70.4, 76.5, 82.6, and 92.7 m (178, 205, 231, 251, 271, and 304 ft.) from the intersection to ensure that the entire dilemma zone was within the range. An additional 24 green trials were randomly introduced into the 24 yellow trials to introduce an element of surprise into the experiment. The run sequence was generated randomly and was different from one trial to another.

Results

This section presents the classification results of the logistic regression and the machine learning algorithms (i.e. random forest, AdaBoost, and SVM). Using the two datasets collected in the two previous studies, as described in the above section. The logistic model and the machine learning models are evaluated using both the classification accuracy and the false positive rate, computed using Equation (9) and Equation (10), respectively. In our context stop is positive and run is negative. So, false positive (error type I) is predicting the driver stops while he actually runs which is very dangerous and may cause crashes.

Our goal is to identify the model that has;

1. The highest classification accuracy to increase user's acceptance of the advanced driver assistance system.
2. The lowest false positive to provide the driver with the highest degree of safety.

$$\text{Classification accuracy} = 100\text{x}\frac{\sum \text{True classified run}+\sum \text{True classified stop}}{\sum \text{Actual run}+\sum \text{Actual stop}} \quad (9)$$

$$\text{False positive rate (FPR)} = \frac{\sum \text{misclassifed run}}{\sum \text{Actual stop}} \quad (10)$$

For each classifier, the average of classification accuracy and FPR of each set of the trials are calculated using the Leave-One-Out (LOO) cross-validation method [49]. In the LOO approach, the classifier is built using labeled data from all drivers except one single driver. This driver is used as the unseen test driver and the classification accuracy and FPR are calculated for that driver. The entire process is repeated where each driver is used once as a test driver and the average classification accuracy is calculated across all drivers.

1. Logistic Regression

We built two logistic models, one without the new predictor, and the other with the new predictor. The Bayesian information criterion (BIC) of the model without the new predictor is 1018.1 and after adding the new predictor becomes 877.5. The Akaike information criterion (AIC) of the model without the new predictor is 987.6 and after adding the new predictor becomes 841.9. Both BIC and AIC shows that the model with the new predictor has lower values hence, is better. The parameter estimates are shown in Table 35 and Table 36.

Table 35: Parameter Estimates of the Logistic Model without the New Predictor.

Term	Estimate	Std Error	ChiSquare	Prob>ChiSq
Intercept	4.05225895	2.192154	3.42	0.0645
Time to intersection (TTI)	-2.8928114	0.1708661	286.63	<.0001
Age (A)	-0.0195942	0.0048147	16.56	<.0001
Gender (G)	0.25835955	0.162566	2.53	0.1120
Speed (V)	0.07641985	0.0286772	7.10	0.0077
Road condition (S)	-0.2666143	0.1621986	2.70	0.1002

As shown in Table 35 all factors are significant except the road condition and the driver gender. After adding the new predictor which is shown in Table 36 to be significant, the speed became insignificant in addition to the road condition and gender.

Table 36: Parameter Estimates of the Logistic Model with the New Predictor.

Term	Estimate	Std Error	ChiSquare	Prob>ChiSq
Intercept	5.85016118	2.3571039	6.16	0.0131
Time to intersection (TTI)	-3.4510272	0.2081167	274.97	<.0001
Age (A)	-0.0174145	0.0052297	11.09	0.0009
Gender (G)	-0.3409965	0.1854972	3.38	0.0660
Speed (V)	0.03591772	0.030821	1.36	0.2439
Road condition (S)	-0.1600429	0.1768756	0.82	0.3656
New predictor	2696.46047	255.80208	111.12	<.0001

Equation (11) and Equation (12) show the fitted model of the driver stop/run behavior.

$$\ln\frac{P_s}{P_r} = \ln\frac{P_s}{1-P_s} = 4.05 - 2.89\text{TTI} - 0.02\text{A} + 0.26\text{G} + 0.08\text{V} - 0.27\text{S} \quad (11)$$

$$\ln\frac{P_s}{P_r} = \ln\frac{P_s}{1-P_s} = 5.85 - 3.45TTI - 0.02A - 0.34G + 0.04V - 0.16S + 2696.5D \quad (12)$$

Where P_s is the probability of stopping and P_r is the probability of running. The classification accuracy and FPR of the logistic models without the new predictor are 80.76 and 0.2451, respectively. After adding the new predictor, the classification accuracy is raised to 84.19 and the FPR is reduced slightly to 0.2417.

2. Support Vector Machine

To implement SVM, the LibSVM library of SVMs was applied [50]. With regard to the size of the data, Gaussian kernel was selected to adopt for model development [45]. Furthermore, complete model selection was conducted by changing the regularization parameter and the Gaussian parameter to achieve the highest performance. Classification accuracy of about 90% and 86% were obtained with and without including the new predictor, respectively. Moreover, the SVM model resulted in FPR of about 24% and 17% with and without the new predictor, respectively. As mentioned earlier, LOO cross-validation technique was applied to assess the model. Figure 79 presents the complete model selection for two of the SVM models, with and without the new predictor, while using the LOO cross-validation technique. Figure 3 illustrates how varying the model parameters (i.e. regularization and gaussian parameters) changes the cross-validation accuracy. In fact, the best performance (i.e. accuracy of 90% and 86%) for the two SVM models were obtained by adjusting the model parameters as presented in the figure.

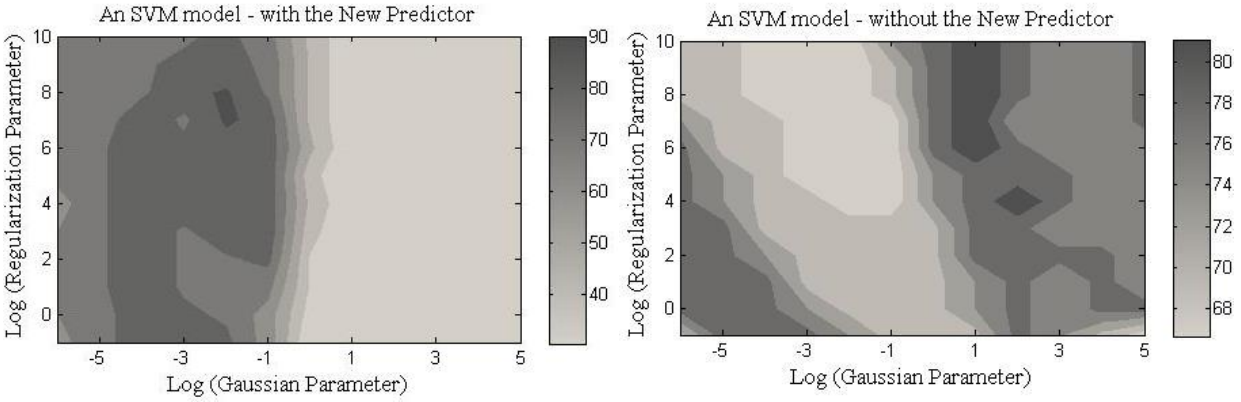


Figure 79: Complete Model Selection for SVM Models

3. AdaBoost and Random Forest

The sequence in the above subsection was repeated using the AdaBoost machine and random forest learning algorithms. Figure 80 and Figure 81 show the classification accuracy and FPR of the AdaBoost models with and without the new predictor for different number of weak learners. We can observe the reduction in the FPR and an increase in the classification accuracy compared to the model which does not use the new predictor.

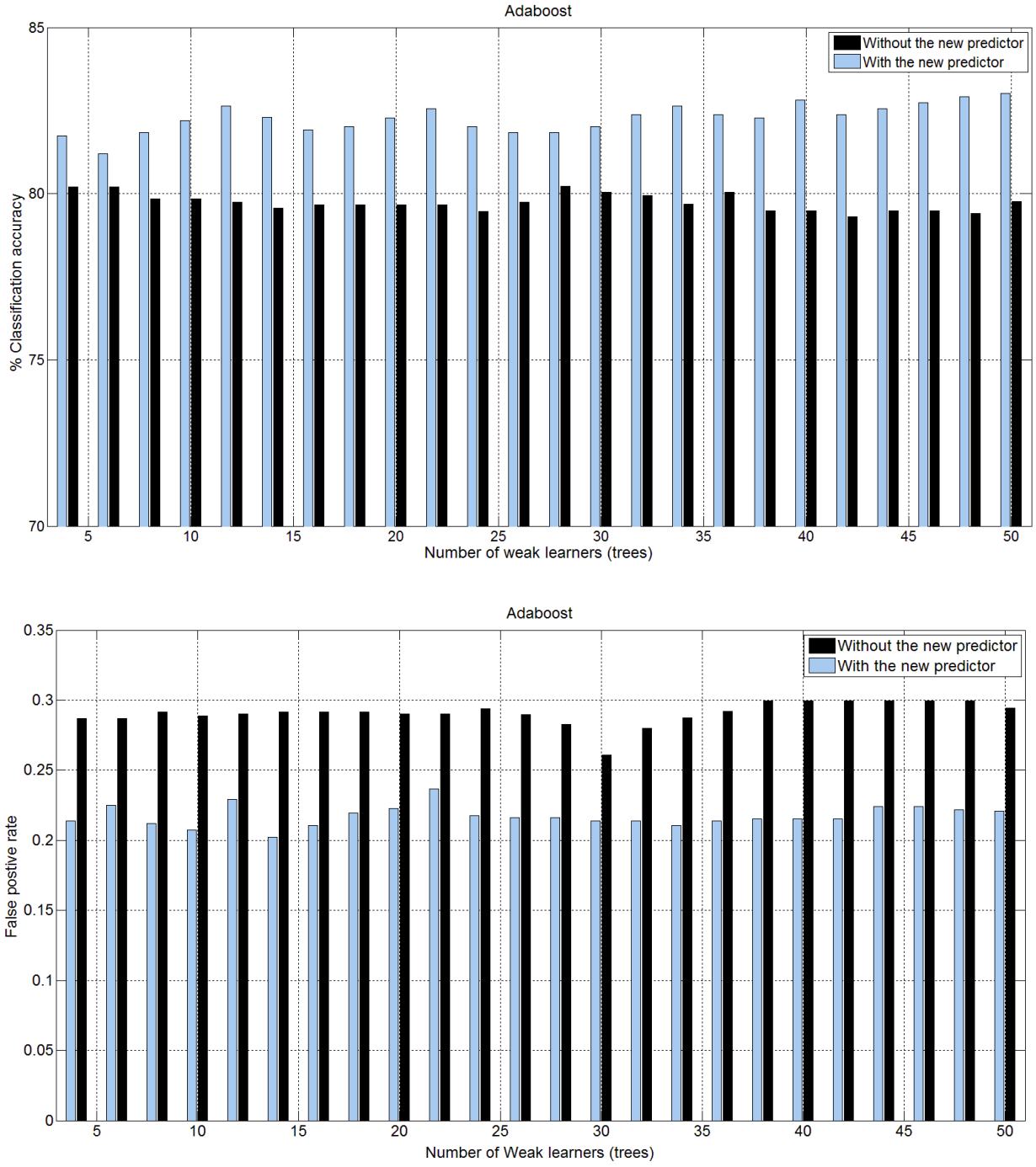


Figure 80: The Classification Accuracy (Above Panel) and the FPR (Bottom Panel) Using Different Number of Weak Learners

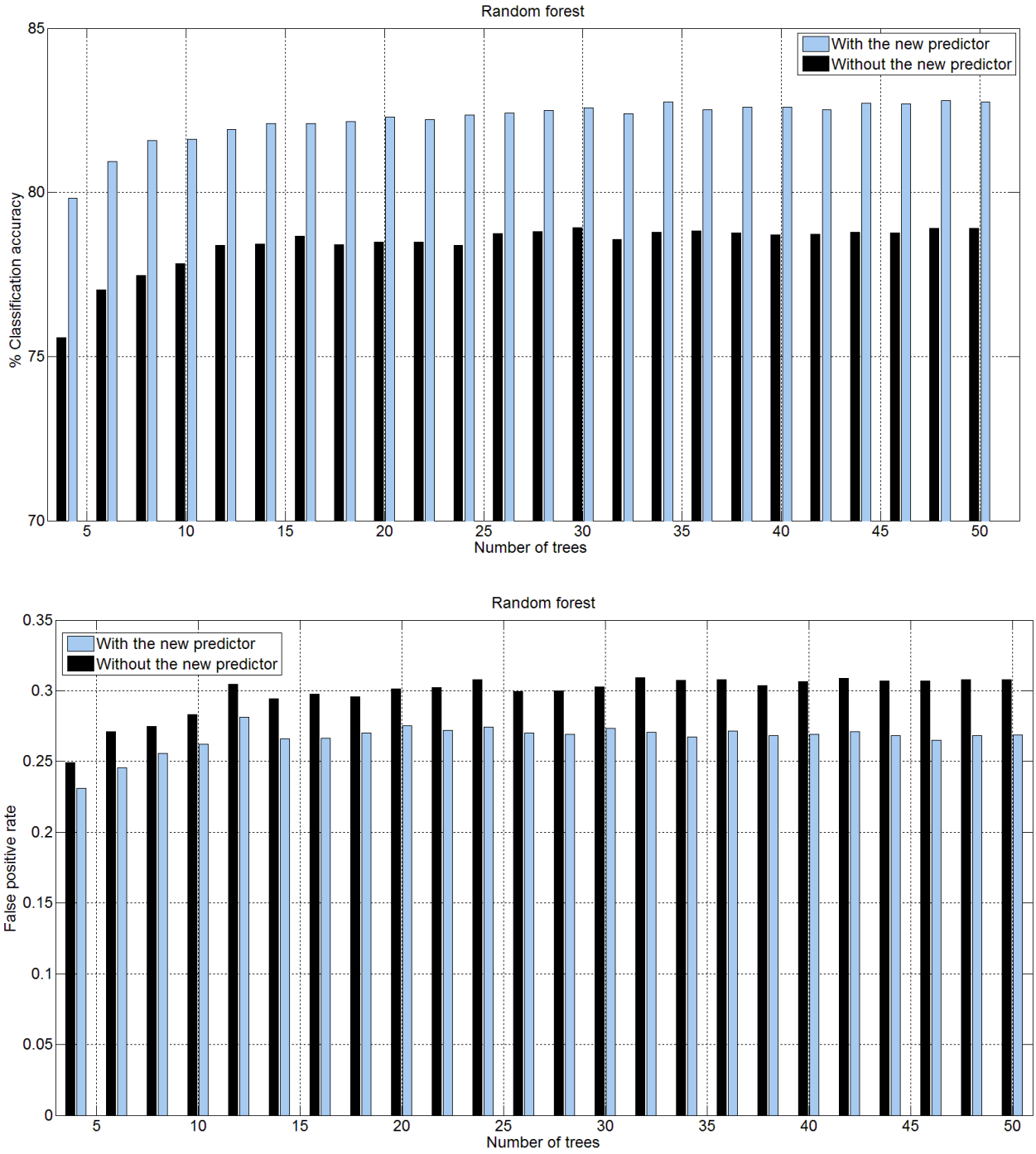


Figure 81: The Classification Accuracy (Above Panel) and the FPR (Bottom Panel) Using Different Number of CART

4. Model Comparison

The four developed classifiers were compared in terms of the false positive rate (false alarms) and the classification accuracy, as shown in Table 37. All four classifiers were comparable with and without the new predictor in terms of classification accuracy. AdaBoost was the best

classifier in terms of low false alarms when compared with the other three classifiers. However, if only the overall classification accuracy is considered, the SVM model with the new predictor achieved the highest accuracy despite the slightly higher false positive rate. The paired t-test was used to test the significance of the improvement in both FPR and classification accuracy of the classifier without the new predictor and after adding the new predictor. In both tests (FPR and classification accuracy tests), our null hypothesis is that the distribution of differences come from distribution with mean equals zero, assuming both classifiers (without and with new predictor) are equivalent and the new predictor does not have any significant effect. The alternative hypothesis is that the differences tend to be different from zero and the new predictor significantly improves the classifier.

Table 37: Comparison between Different Classifiers

Method	Evaluation Measure	Without the new predictor	With the new predictor	p-value
Logistic Regression	Classification accuracy (%)	80.76	84.19	0.0120
	False positive (%)	24.51	24.17	0.9290
AdaBoost	Classification accuracy (%)	79.77	83.00	0.0499
	False positive (%)	29.42	22.06	0.0441
Random Forest	Classification accuracy (%)	78.90	82.77	0.0356
	False positive (%)	30.80	26.88	0.1974
SVM	Classification accuracy (%)	86.46	90.02	0.0187
	False positive (%)	22.34	20.73	0.4219

Table 37 shows the p-values which are extracted from t-distribution with the degree of freedom equals (number of drivers-1). The experimental results show the AdaBoost model with added new predictor has a statistically significant lower FPR and statistically significant higher classification accuracy at 0.05 significance level. The other three classifiers have only one significant improvement in terms of classification accuracy. In other words, including the new predictors for these three models resulted in statistically significant improvement in classification accuracy, but the reduction in FPR was not statistically significant at 0.05 significance level.

Study Conclusions and Future Work

In this paper, we introduce a measure of driver aggressiveness into the modeling of driver stop/run behavior at the onset of a yellow indication. The driver aggressiveness parameter can be estimated by monitoring the driver historical response to yellow indications. The new aggressiveness parameter is based on the count of the number of runs the driver makes when the time to intersection, at the onset of the yellow indicator, is greater than the yellow time and his speed is equal to or greater than the posted speed limit. The parameter can then be added to the model after some period of monitoring. The experimental results demonstrate the ability of the new predictor to explain part of the variability in the driver stop/run decision. Specifically, there is a statistically significant increase in the classification accuracy. the FPR is remarkably reduced, but this reduction is not statistically significant. The study also demonstrates that the AdaBoost machine learning algorithm is the best algorithm in terms of its statistically significant improvement in FPR. However, SVM model had the best performance in terms of the classification accuracy only.

Further enhancements to the model are required to model driver stop/run behavior under more severe inclement weather (such as snow, freezing rain, ice and fog) and road surface

conditions, considering the impact of the vehicle type (bus or truck) on the driver behavior, and developing real-time machine learning techniques that can adapt to changes in driver behavior.

Acknowledgements

The authors acknowledge the support of the Center for Technology Development and Smart Road Operations Group at the Virginia Tech Transportation Institute (VTTI), Ahmed Amer, Huan Li for running the tests. This work was supported in part by grants from the Virginia Center for Transportation Innovation and Research (VCTIR), the Mid-Atlantic Universities Transportation Center (MAUTC), and SAFETEA-LU funding.

References

[1] W. D. Jones, "Keeping cars from crashing," *IEEE Spectrum*, vol. 38, pp. 40–45, 2001.

[2] R. T. Trevor Hastie, Jerome Friedman *The Elements of Statistical Learning Data Mining, Inference, and Prediction*, 2009.

[3] U.S, D. o. Transportation, and F. H. Administration. (2014). *Safety at Signalized Intersections*. Available: http://safety.fhwa.dot.gov/Intersection/signalized/presentations/sign_int_pps051508/long/index.cfm

[4] H. Rakha, I. El-Shawarby, and J. R. Setti, "Characterizing driver behavior on signalized intersection approaches at the onset of a yellow-phase trigger," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 8, pp. 630-640, 2007.

[5] D. Gazis, R. Herman, and A. Maradudin, "The Problem of the Amber Signal Light in Traffic Flow," *Operations Research*, vol. 8, pp. 112-132, 1960.

[6] Y. Sheffi and H. Mahmassani, "A model of driver behavior at high speed signalized intersections," *Transportation Science*, vol. 15, pp. 50-61, 1981.

[7] J. A. Bonneson, D. Middleton, K. Zimmerman, H. Charara, and M. Abbas, "Intelligent detection-control system for rural signalized intersections," Texas Transportation Institute, Texas A&M University System2002.

[8] T. J. Gates, D. A. Noyce, L. Laracuente, and E. V. Nordheim, "Analysis of driver behavior in dilemma zones at signalized intersections," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2030, pp. 29-39, 2007.

[9] P. D. Pant, Y. Cheng, A. Rajagopal, and N. Kashayi, "Field testing and implementation of dilemma zone protection and signal coordination at closely-spaced high-speed intersections," University of Cincinnati2005.

[10] M.-S. Chang, C. J. Messer, and A. J. Santiago, *Timing traffic signal change intervals based on driver behavior*, 1985.

[11] C. V. Zegeer, "GREEN-EXTENSION SYSTEMS AT EHGI-I-SPEED INTERSECTIONS," 1978.

[12] Y. Liu, G.-L. Chang, R. Tao, T. Hicks, and E. Tabacek, "Empirical observations of dynamic dilemma zones at signalized intersections," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2035, pp. 122-133, 2007.

[13] H. Wei, Z. Li, P. Yi, and K. R. Duemmel, "Quantifying Dynamic Factors Contributing to Dilemma Zone at High-Speed Signalized Intersections," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2259, pp. 202-212, 2011.

[14] S. Ghanipoor Machiani and M. Abbas, "Dynamic Driver's Perception of Dilemma Zone: Experimental Design and Analysis of Driver's Learning in a Simulator Study," in *The 93nd Annual Meeting of the Transportation Research Board*, Washington, DC, 2014.

[15] M. Abbas, S. G. Machiani, P. M. Garvey, A. Farkas, and R. Lord-Attivor, "Modeling the Dynamics of Driver's Dilemma Zone Perception Using Machine Learning Methods for Safer Intersection Control," 2014.

[16] S. G. Machiani and M. Abbas, "Predicting Drivers Decision in Dilemma Zone in a Driving Simulator Environment Using Canonical Discriminant Analysis," in *Transportation Research Board 93rd Annual Meeting*, 2014.

[17] J. K. Caird, S. Chisholm, C. J. Edwards, and J. I. Creaser, "The effect of yellow light onset time on older and younger drivers' perception response time (PRT) and intersection behavior," *Transportation research part F: traffic psychology and behaviour,* vol. 10, pp. 383-396, 2007.

[18] I. El-Shawarby, H. A. Rakha, V. W. Inman, and G. W. Davis, "Age and gender impact on driver behavior at the onset of a yellow phase on high-speed signalized intersection approaches," in *Transportation Research Board 86th Annual Meeting*, 2007.

[19] H. Li, H. Rakha, and I. El-Shawarby, "Designing Yellow Intervals for Rainy and Wet Roadway Conditions," *International Journal of Transportation Science and Technology,* vol. 1, pp. 171-190, 06/01/ 2012.

[20] A. Jahangiri, H. Rakha, and T. A. Dingus, "Predicting Red-light Running Violations at Signalized Intersections Using Machine Learning Techniques," 2015.

[21] I. El-Shawarby, A.-S. G. Abdel-Salam, H. Li, and H. Rakha, "Driver Behavior at the Onset of Yellow Indication for Rainy/Wet Roadway Surface Conditions."

[22] Richard C. Coakley and E. R. Stollof, "Intersection Safety Needs Identification Report," 2009.

[23] H. Wei, "Characterize dynamic dilemma zone and minimize its effect at signalized intersections," 2008.

[24] D. Shinar and R. Compton, "Aggressive driving: An observational study of driver, vehicle, and situational variables," *Accident Analysis & Prevention,* vol. 36, pp. 429-437, 2004.

[25] "Empirical Study of Driver Responses during the Yellow Signal Phase at Six Maryland Intersections," *Journal of Transportation Engineering,* vol. 138, pp. 31-42, 2012.

[26] Y. Liu, G.-L. Chang, R. Tao, T. Hicks, and E. Tabacek, "Empirical Observations of Dynamic Dilemma Zones at Signalized Intersections," *Transportation Research Record: Journal of the Transportation Research Board,* vol. 2035, pp. 122-133, 12/01/ 2007.

[27] M. Elhenawy, H. Rakha, and I. El-Shawarby, "Enhanced Modeling of Driver Stop-or-Run Actions at a Yellow Indication," *Transportation Research Record: Journal of the Transportation Research Board,* vol. 2423, pp. 24-34, 12/01/ 2014.

[28] S. S. M. Ali, N. Joshi, B. George, and L. Vanajakshi, "Application of Random Forest Algorithm to Classify Vehicles Detected by a Multiple Inductive Loop System," *2012 15th International Ieee Conference on Intelligent Transportation Systems (Itsc),* pp. 491-495, 2012.

[29] M.-H. Pham, A. Bhaskar, E. Chung, and A.-G. Dumont, "Random Forest Models for Identifying Motorway Rear-End Crash Risks Using Disaggregate Data," presented at the 13th International IEEE Annual Conference on Intelligent Transportation Systems, Madeira Island, Portugal, 2010.

[30] Qingchao Liu, Jian Lu, and S. Chen, "Traffic Incident D etection Using Random Forest," presented at the Transportation Research Board 92nd Annual Meeting, Washington DC.
[31] L. Yulan, M. L. Reyes, and J. D. Lee, "Real-Time Detection of Driver Cognitive Distraction Using Support Vector Machines," *Intelligent Transportation Systems, IEEE Transactions on,* vol. 8, pp. 340-350, 2007.
[32] F. Tango and M. Botta, "Real-Time Detection System of Driver Distraction Using Machine Learning," *Intelligent Transportation Systems, IEEE Transactions on,* vol. 14, pp. 894-905, 2013.
[33] A. Jahangiri and H. Rakha, "Developing a Support Vector Machine (SVM) Classifier for Transportation Mode Identification by Using Mobile Phone Sensor Data," in *Transportation Research Board 93rd Annual Meeting*, 2014.
[34] A. Jahangiri and H. A. Rakha, "Applying Machine Learning Techniques to Transportation Mode Recognition Using Mobile Phone Sensor Data," *Intelligent Transportation Systems, IEEE Transactions on,* vol. PP, pp. 1-12, 2015.
[35] V. Balali and M. Golparvar-Fard, "Scalable Nonparametric Parsing for Segmentation and Recognition of High-quantity, Low-cost Highway Assets from Car-mounted Video Streams," in *Construction Research Congress 2014@ sConstruction in a Global Network*, 2014, pp. 120-129.
[36] S. Ghanipoor Machiani and M. Abbas, "Predicting Drivers Decision in Dilemma Zone in a Driving Simulator Environment using Canonical Discriminant Analysis," in *The 93nd Annual Meeting of the Transportation Research Board*, Washington, DC, 2014.
[37] A. M. M. Amer, H. A. Rakha, and I. El-Shawarby, "A Behavioral Modeling Framework of Driver Behavior at Onset of Yellow a Indication at Signalized Intersections," presented at the Transportation Research Board 89th Annual Meeting, Washington DC, 2010.
[38] L. Breiman, "Random Forests," *Machine Learning,* vol. 45, pp. 5-32, 2001.
[39] L. Breiman, "Random forests–random features," 1999.
[40] L. Breiman, *Classification and regression trees*. Belmont, Calif.: Wadsworth International Group, 1984.
[41] Y. Freund and R. Schapire, "A short introduction to boosting," *Japonese Society for Artificial Intelligence,* vol. 14, pp. 771-780, // 1999.
[42] R. Clarke, H. W. Ressom, A. Wang, J. Xuan, M. C. Liu, E. A. Gehan, *et al.*, "The properties of high-dimensional data spaces: implications for exploring gene and protein expression data," *Nat Rev Cancer,* vol. 8, pp. 37-49, 01//print 2008.
[43] L. Smith. (2002, A Tutorial on Principal Components Analysis.
[44] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural Networks,* vol. 13, pp. 411-430, 6// 2000.
[45] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification," ed, 2003.
[46] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors)," *The annals of statistics,* vol. 28, pp. 337-407, 2000.
[47] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *Neural Networks, IEEE Transactions on,* vol. 13, pp. 415-425, 2002.
[48] H. Rakha, I. Lucic, S. H. Demarchi, J. R. Setti, and M. Van Aerde, "Vehicle dynamics model for predicting maximum truck acceleration levels," *Journal of Transportation Engineering,* vol. 127, pp. 418-425, 2001.

[49] T. Evgeniou, M. Pontil, and A. Elisseeff, "Leave One Out Error, Stability, and Generalization of Voting Combinations of Classifiers," *Machine Learning,* vol. 55, pp. 71-97, 2004/04/01 2004.
[50] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST),* vol. 2, p. 27, 2011.

Chapter 15: Driver Stop/Run Behavior Modeling at the Onset of a Yellow Indication Considering Vehicle Type and Roadway Surface Condition

This chapter is based on
Mohammed Elhenawy, Arash Jahangiri, Hesham Rakha, and Ihab. El-Shawarby, "Classification of driver stop/run behavior at the onset of a yellow indication for different vehicles and roadway surface conditions using historical behavior," presented at the 6th International Conference on Applied Human Factors and Ergonomics, Las Vegas, Nevada, USA, 2015.

Abstract

The ability to classify driver stop/run behavior at signalized intersections considering the vehicle type and roadway surface conditions is critical in the design of advanced driver assistance systems. Such systems can reduce intersection crashes and fatalities by predicting driver stop/run behavior. The research presented in this paper uses data collected from three controlled field experiments and one dataset collected using a truck simulator. The field experiments are done on the Smart Road at the Virginia Tech Transportation Institute (VTTI) to model driver stop/run behavior at the onset of a yellow indication for different roadway surface conditions and different vehicle type. The paper offers two contributions. First, it introduces a new predictor related to driver aggressiveness and demonstrates that this measure enhances the modeling of driver stop/run behavior. Second, it applies well-known Artificial Intelligence techniques including: adaptive boosting (AdaBoost), artificial neural networks (ANN), and Support Vector Machine (SVM) algorithms on the data in order to develop a model that can be used by traffic signal controllers to predict driver stop/run decisions in a connected vehicle environment. The research demonstrates that by adding the driver aggressiveness predictor to the model, the increase in the model accuracy is significant for all models except SVM. However, the reduction in the false alarm rate was not statistically significant when using any of the approaches.

Introduction

In the US, the Department of Transportation (DOT) reported 32,367 fatalities caused by road accidents in 2011 [1]. A significant percentage of these road accidents occurred at signalized intersections as a result of the dilemma zone problem. Rear-end crash and right-angle crash are the two types of dilemma-zone-related crashes. These crashes can be avoided if vehicles know the predicted behavior of surrounding vehicles.

Modeling driver behavior at signalized intersections and more specifically in the dilemma zone has been the focus of many studies [2-6]. The dilemma zone is first examined and modeled as a binary decision problem to either stop or proceed when a yellow indication is triggered in [7]. The dilemma zone is created when the maximum clearing distance is smaller than the minimum stopping distance. The distance required for an approaching vehicle to pass the stop bar before the end of yellow indication time is defined as clearing distance. It is a function of the speed of the approaching vehicle and the duration of the yellow interval. The stopping distance is defined as the distance required for an approaching vehicle to come to a complete stop before the stop bar. It is a function of the vehicle's speed, the driver's perception-reaction time, and an acceptable deceleration rate. On the onset of yellow indication, the approaching drivers have two options either proceed through the intersection before the end of the yellow interval or stop

safely. Incorrect driver decisions may result in either a rear-end crash, if the driver fails to come to a safe stop, or a right-angle crash with side-street traffic, if the driver does not have enough time to safely cross the intersection before the conflicting flow is released.

Drivers approaching signalized intersection need to get many parameters in order to decide crossing the intersection or stopping at the onset of yellow indicator. There are many factors that affect the dilemma zone and the driver decision. These factors are divided into internal factors group and external factor group. The internal factors include driver-related attributes such as age and gender. The external factors include factors such as the traffic, intersection and vehicle type[8]. The intersection factor itself has many attributes such as the number of legs in the intersection and the roadway surface condition[9]. There is another factors group called the intermediate group which includes factors such as perception-reaction time (PRT) and acceptable acceleration/deceleration rates.

Aggressive driving could be critical in modeling driver stop/run behavior at signalized intersections; however measuring driver aggressiveness may not be plausible. The previous research uses five driver actions to measure aggressive driving. These five measures include: short or long honk of the horn, cutting in front of other vehicles in a passing lane maneuver, cutting in front of other vehicles in a multi-lane passing maneuver, and passing one or more vehicles by driving on the shoulder and then cutting in [10]. In our previous study, we propose the frequency of running at yellow indication as a measure of aggressive driving [11]. In the current study, a better sold and formal definition and formulation of the driver aggressiveness is proposed.

Consider a vehicle approaching signalized intersection, our goal is to build a model that uses many predictors such as distance to intersection and driver age to predict whether the driver will stop or run at onset of the yellow indicator. Because, in real life, individual drivers behave differently, we included the proposed predictor to explain some of the variation between drivers based on their history. Such model should be one of the main building blocks in more advanced driver assistance systems. These systems should be able to predict the driver behavior and warn him and the surrounding vehicles if the driver's decision is wrong. The drawback of advanced driver assistance systems warnings is that it may distract the driver. For that, we should be careful when choosing the classifier so that it produces minimum false positives and maximum classification accuracy.

The past two decades have seen numerous research efforts and advances in both machine learning and computers. The available machine learning algorithms, computation power and datasets from fixed detectors or data probes and intelligent transportation systems (ITSs) encourages Transportation engineers to apply machine learning in their field. Recently, some machine learning algorithms were used in the transportation field, including: classifying and counting vehicles detected by multiple inductive loop detectors [12], identifying motorway rear-end crash risks using disaggregate data [13], automatic traffic incident detection [14], real-time detection of driver distraction [15, 16], transportation mode recognition using smartphone sensor data [17], and video-based highway asset segmentation and recognition [18]. Modeling driver stop/run behavior at signalized intersections is very important and is ideal for applying machine learning techniques [19]. At first glance, driver stop/run behavior modeling seems to be a good candidate for straightforward application of machine learning algorithms. Observations of the driver stop/run behavior from naturalized datasets or from controlled field experiment datasets can be used to train machine learning algorithms. The trained models can then be used to predict future driver decisions for implementation in in-vehicle safety systems. However, machine

learning modeling of driver stop/run behavior faces some challenges including the need for large labeled datasets, driver stop/run behavior drift, and computational complexity.

In this paper, we introduce a new parameter related to the driver aggressiveness. This new predictor can be observed directly from stop/run historical behavior. By using this new predictor, we demonstrate that the modeling of driver stop/run behavior can be enhanced. The use of such models can then be integrated with in-vehicle safety systems to predict the action of a driver and thus warn other drivers or take action to ensure that no collisions occur.

Methods

In this section, a brief introduction to the modeling techniques used in this paper are presented to familiarize readers with them. The modeling techniques represent a variety of machine learning techniques. The used algorithms demonstrate the wide variety of algorithms that can be used by transportation practitioners.

1. Adaptive Boosting Algorithm

The Adaptive Boosting (AdaBoost) is a machine learning algorithm that is based on the idea of incremental contribution [20]. AdaBoost uses a set of weak classifiers, each is trained using the same training dataset but with a different weight distribution. Each of the weak learners focuses on the instances that are misclassified by the previous learner. The output of AdaBoost is the weighted average of all weak learner outputs. To describe the AdaBoost algorithm, let us assume the training set consists of n instances $T = \{(x_1, y_1), \dots, (x_n, y_n)\}$ where, x_i is the vector of predictors that can be represented by a point in the multidimensional feature (predictor) space and $y_i = \{-1, +1\}$ is the corresponding label. After training T weak learners the model is ready to predict the label for test instance (unseen) x_{test}. The label of the test instance is defined using Equation (1)

$$\text{sign}\left(\sum_{t=1}^{T} \propto_t L_t(x_{test})\right) \quad (1)$$

Where $\propto_t$ is trustiness level of learner L_t. The label is set equal to 1 if the output of Equation (1) is positive and -1 if the output is negative.

2. Artificial Neural Networks

In machine learning, artificial neural networks (ANN) are used to estimate or approximate unknown linear and non-linear functions that depend on a large number of inputs. Artificial neural networks can compute values or return labels using inputs.

An ANN consists of several processing units, called neurons, which are arranged in layers. In this paper, we used the multi-layered feed-forward ANN's which is commonly used for classification analysis. In multi-layered feed-forward ANN, the neurons are connected by directed connections which allow information flow in the direction from the input layer to output layer. A neuron k in layer m receive an input x_j from each neuron j at layer $m - 1$. The neuron adds the weighted sum of its inputs to a bias term; then apply the whole thing to a transfer function and pass the result to its output toward downstream layer. In general the ANN requires the definition of the number of layers, number of neurons in each layer and the neuron's transfer function. Given the training dataset the ANN can use learning algorithm such as back propagation to learn the weights and biases for each single neuron[21].

3. Support Vector Machine

Support Vector Machine (SVM) is a rather complex machine learning technique that can be employed in classification problems. SVM is known as a large margin classifier which means

that while this method attempts to find decision boundaries between different classes, it tries to maximize the gap or margin between classes.

The objective function of the SVM formulation and the associated constraints are presented below in Equation (2) through Equation (4) [22]. The sum of two terms are minimized in the objective function; minimizing the first term is basically equivalent to maximizing the margin between classes, and the second term consists of an error term multiplied by the regularization (penalty) parameter denoted by C. The regularization is designed to deal with the issue of overfitting. The value of the C parameter should be adjusted to obtain the best possible performance.

$$\min_{w,b,\xi}\left(\frac{1}{2}w^{T}w + C\sum_{n=1}^{K}\xi_n\right) \quad (2)$$

Subject to:

$$y_n(w^{T}\phi(x_n) + b) \geq 1 - \xi_n\,, n = 1, \dots, K \quad (3)$$

$$\xi_n \geq 0\,, n = 1, \dots, K \quad (4)$$

where,

- w Parameters to define decision boundary between classes
- C Regularization (or penalty) parameter
- ξ_n Error parameter to denote margin violation
- b Intercept associated with decision boundaries
- $\phi(x_n)$ Function to transform data from X space into some Z space

When using SVM, the data are transformed from the X space to the Z space using some function $\phi(x_n)$. However, in solving the problem, there is no need to actually do the transformation. Instead, some other functions, known as kernels, are adopted. The kernels, which appear in the dual formulation of the problem, correspond to the vector inner product in the Z space. To construct the model, kernel type should be selected (e.g. linear, polynomial, Gaussian).

Proposed Driver Aggressiveness Predictor

Thanks to advances in telecommunication and computation power, connected vehicles have become a reality in which vehicles can exchange information with each other (V2V) and with the traffic signal controller (V2I). We can use this technology advantage to allow vehicles to learn the behavior of its driver by maintaining a small record describing many behavioral related parameters. These parameters can describe the level of aggressiveness of the driver. One intuitive parameter we propose is the probability of running at the onset of a yellow indication at signalized intersections when stopping is a better decision.

We propose a new predictor that can be used as a measure of the driver's aggressiveness. The new measure is based on the count of the number of runs the driver makes when the time to intersection at the onset of the yellow indicator is greater than the yellow time and his speed is equal or greater than the maximum posted speed. The value of new predictor θ_j for driver j, can be estimated based on the Bayesian approach by finding the posterior density distribution, as shown in equation (5),

$$f(\theta_j|y_{j1}, y_{j2}, \dots, y_{jn}) \propto f(y_{j1}, y_{j2}, \dots, y_{jn}|\theta_j)f(\theta_j) \quad (5)$$

where, $f(y_{j1}, y_{j2}, \dots, y_{jn}|\theta_j)$ is the sampling distribution and $f(\theta_j)$ is the prior distribution.

The distribution of the $f(y_{ji}|\theta_j)$ is Bernoulli because the random variable is either one or zero. The new predictor can take any value between zero (stop) and one (run). This means the domain of the $f(\theta_j)$ is from zero to one. So that $f(\theta_j)$ is distributed according to Beta distribution and the whole problem can be viewed as Beta-Bernoulli model, as shown in equation (6),

$$f(y_{ji}|\theta_j) \sim \text{Bernoulli}\ (\theta_j)$$
$$\theta_j \sim \text{Beta}\ (a, b) \quad (6)$$
$$a = 1 \text{ and } b = 1000$$
$$f(\theta_j|y_{j1}, y_{j2}, \dots, y_{jn}) \propto \{\prod_{i=1}^{n} \theta_j^{y_{ji}}(1-\theta_j)^{1-y_{ji}}\} \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \theta_j^{a-1}(1-\theta_j)^{b-1}$$

by removing the constants from the above equation and naming the distribution we find,

$$f(\theta_j|y_{j1}, y_{j2}, \dots, y_{jn}) \sim \text{Beta}\ (\sum_{i=1}^{n} y_i + a\ , n - \sum_{i=1}^{n} y_i + b) \quad (7)$$

where n is the total number of stops and runs when the time to the intersection is greater than yellow time and the driver's speed is equal or greater than the maximum posted speed.

From equation (7) we can estimate the expectation $E[f(\theta_j|y_{j1}, y_{j2}, \dots, y_{jn})]$ and use it as the aggressiveness measure for driver j

$$E[f(\theta_j|y_{j1}, y_{j2}, \dots, y_{jn})] = \frac{\sum_{i=1}^{n} y_i + a}{a+b+n} \quad (8)$$

If the value of the new predictor is close to one that means the driver rarely stops and thus is more aggressive than the driver who has a smaller value of the new predictor. This new predictor is important because it captures the stop/run tendencies of that specific driver. It is envisioned that the computation of this new predictor can be done through some form of infrastructure-to-vehicle (I2V) communication in which the vehicle receives Signal Phasing and Timing (SPaT) information to identify the indication of the traffic signal. Moreover, Vehicle-to-vehicle (V2V) communication would be required to exchange information with surrounding vehicles to identify if the driver was not forced to stop because the vehicle ahead of it stopped. Using the SPaT and surrounding vehicle information the vehicle would count the number of times the driver stopped and ran the yellow indication when s/he has the freedom to proceed.

Data Description

The data used in this paper was collected from three different field experiments and one truck simulator study. The field experiments were conducted at the Virginia Department of Transportation's (VDOT) Smart Road facility, located at the Virginia Tech Transportation Institute (VTTI). The length of the Smart Road is a 3.5 km (2.2 mile). It is a two-lane road with one four-way signalized intersection[23].

1. Dry Roadway Surface Field Experiment

Three vehicles were used in this experiment, one was driven by test participants and accompanied with in-vehicle experimenter [24]. The other two vehicles were driven by trained experimenters who were involved in the study. One of them is following the test vehicle, whereas the other vehicle was crossing the intersection from the conflicting approach when the traffic light was green. The test vehicle is equipped with a real-time data acquisition system (DAS), differential Global Positioning System (GPS) unit, a longitudinal accelerometer, sensors for accelerator position and brake application, and a computer to run the different experimental scenarios.

The vehicle data stream is synchronized with changes in the traffic signal controller by the communication channel that links the data recording equipment to the intersection signal control box. The phase change is trigger by the test vehicle using GPS unit that determine the distance from the intersection. Twenty-four licensed drivers were recruited in three equal age groups (under 40-years-old, 40 to 59-years-old, and 60-years-old or older); each group is male-female balanced.

Participants were asked to follow all normal traffic rules and to obey all traffic laws while their driving. They drove loops on the Smart Road at 72.4 km/h (45 mi/h) instructed speed, crossing the four-way signalized intersection 24 times for a total of 48 trials, where a trial consists of one approach to the intersection. Among the 48 trials, a 4-second yellow indication at the 72 km/h (45 mi/h) instructed speed were triggered for a total of 24 times (four repetitions at six distances). The yellow indications were triggered when the front of the test vehicle was 40.2, 54.3, 62.5, 70.4, 76.5, and 82.6 m (132, 178, 205, 231, 251, and 271 ft) from the intersection for the 72 km/h (45 mi/h) instructed speed to ensure that the entire dilemma zone was within the range. On the remaining 24 trials, the signal indication remained green.

2. Rainy/wet Roadway Surface Field Experiment

Two vehicles were used in this study, one was driven by test participants (accompanied by the in-vehicle experimenter) and the other vehicle was driven by a trained research assistant to simulate real-world conditions[25]. The confederate vehicle crossed the intersection from the side street when the signal was red for the test vehicle. The participants were asked to follow all normal traffic rules and to obey all traffic laws.

The test vehicle was equipped with a differential Global Positioning System (GPS), a real-time Data Acquisition System (DAS), and a computer to run the different experimental scenarios. A communications link to the intersection signal control box was used by the data recording equipment to synchronize the vehicle data stream with changes in the traffic. The two vehicles were equipped with a communications system between vehicles, operated by the research assistants.

Twenty-six drivers were recruited in three age groups (under 40-years-old, 40 to 59-years-old, and 60 years of age or older), equal number of male and female participants were assigned to each group.

The experiment was only run in rainy weather and wet pavement surface condition. During the experiment, the participants pass the intersection 48 times. A 4-second yellow interval at the 72.4 km/h instructed speed was triggered for a total of 24 times (four repetitions at six distances). The yellow indications were triggered when the front of the test vehicle was 54.3, 62.5, 70.4, 76.5, 82.6, and 92.7 m (178, 205, 231, 251, 271, and 304 ft) from the intersection to ensure that the entire dilemma zone was within the range.

3. Bus Field Experiment

The vehicle used in this experiment is a 1990 Blue Bird East school bus which is driven by participant bus drivers (accompanied by a research assistant). The bus was equipped with a Differential Global Positioning System (DGPS), a real-time data acquisition system (DAS). Two video cameras were used as well: one records the front view of the test vehicle and the other continuously records the participant's foot movements.). The trials and road scenarios are controlled using a laptop installed with VTTI proprietary programs.

There were thirty-six participants who are employed as a bus driver and have a valid Class-B commercial driver's license (CDL). The participant drove 24 loops around the instructed test area, passing the intersection 48 times and was instructed to cruise at a speed of 56 km/h (35 mi/h) while approaching the signalized intersection and to obey all traffic laws.

The experiment was balanced where 24 trials out of the 48 trials were yellow and 24 trials were green. The signal was triggered to switch to yellow at 6 different distances to the intersection, 4 times each. The yellow indications were triggered when the front of the test vehicle was (120, 150, 180, 200, 220, and 250 ft.) from the intersection.

4. Truck Simulator Experiment

This experiment is done using the Commercial Testing and Prototyping Simulator (CTAPS) at the Virginia Tech Transportation Institute (VTTI) in Blacksburg, VA. The participant in this experiment were required to be male, hold a valid, Class-A commercial driver's license (CDL), be between the ages of 21 to 55, be able to drive a standard 10-speed manual transmission with no assistive devices, hold a valid Department of Transportation (DOT) medical examination, and operate a truck a minimum of 2 to 4 times per week.

After completing the orientation drives, drivers moved on to eight scenarios study trials. In each of these eight scenarios, the same six traffic signals were considered. These signals were triggered to change to yellow at different six distances (224, 264, 284, 304, 330, and 363 feet) from the intersection. The instructed speed was 45 MPH (72.42 KPH).

The experiment started with twenty-five drivers participated. Four participant developed simulator sickness at some point during the experiment and two drivers produced data that was corrupt/incomplete. Additionally several interactions with the signals malfunctioned during participant sessions and these records also were not included in the analysis. A total of 910 records were collected in this experiment.

Results

This section presents the classification results of the machine learning algorithms (i.e. AdaBoost, ANN, and SVM). Using the datasets collected in the previous studies, as described in the above section, to find the best classifier and show how the new proposed predictor improve the classifiers' performance. There are eight predictors used to classify the driver: gender, age, time-to-intersection, approaching speed, roadway surface condition, vehicle type and the new proposed driver aggressiveness predictor.

The machine learning models are evaluated using both the classification accuracy and the false positive rate. Our goal is to get the most accurate model with the minimum false positive rate. In other words, we want to give the driver the best safety with minimum false alarm and distraction. For each classifier, the average of classification accuracy and FPR of each set of trials are calculated using the leave-one-out (LOO) cross-validation method [26].

1. Artificial Neural Network

We used a feedforward neural network with one hidden layer and sigmoid transfer function. The output layer consists of one neuron with sigmoid function as well. Four networks with different neurons in the hidden layers are trained and tested to find the best number of neurons. Using 6 neurons in the hidden layer, the classification accuracy of 82.65 and 86.20 were obtained without and with the new predictor, respectively. Similarly, false positive rate of 30.98 and 26.74 were obtained without and with the new predictor, respectively

2. AdaBoost

We modeled the driver stop/run behavior using AdaBoost with and without the new predictor at different number of learners. We observed the reduction in the FPR and the increase in the classification accuracy compared to the model which does not use the new predictor. Using 20 trees, the classification accuracy of 84.62 and 85.67 were obtained without and with the new predictor, respectively. Similarly, false positive rate of 30.85 and 27.12 were obtained without and with the new predictor, respectively

3. Support Vector Machine

To implement SVM, the LibSVM library of SVMs was applied [27]. With regard to the size of the data, Gaussian kernel was selected to adopt for model development [28]. Furthermore, complete model selection was conducted by changing the regularization parameter and the Gaussian parameter to achieve the highest performance. Classification accuracy of 92.9% and 91.7% were obtained with and without including the new predictor, respectively. Moreover, the SVM model resulted in FPR of about 5% and 4% with and without the new predictor, respectively. As mentioned earlier, LOO cross-validation technique was applied to assess the model.

4. Model Comparison

The four used classifier were compared in terms of false positive (false alarm) and the classification accuracy, as shown in Table 38. The first three classifiers were comparable with and without the new predictor in terms of classification accuracy. SVM was found to be the best classifier in terms of low false alarm and higher classification accuracy followed by ANN. The paired t-test was used to test the significance of the improvement in both FPR and classification accuracy of the classifier without the new predictor and after adding the new predictor.

Table 38: Comparison between Different Classifiers

Method	Evaluation measure	Without the new predictor	With the new predictor	p-value
Adaboost	Classification accuracy (%)	84.62	85.67	0.0036
	False positive (%)	30.85	27.12	0.0613
ANN	Classification accuracy (%)	82.65	86.20	0.0008
	False positive (%)	30.98	26.74	0.1007
SVM	Classification accuracy (%)	91.7	92.9	0.0675
	False positive (%)	22.34	20.73	0.4219

In both tests (FPR and classification accuracy tests), our null hypothesis is that the distribution of differences come from distribution with mean equal zero which means both classifiers (without and with new predictor) are equivalent and the new predictor does not have any significant effect. The alternative hypothesis is that the differences tend to be different from zero and the new predictor significantly improves the classifier. The experimental results show that the improvement in the classification accuracy of the first three classifiers is statistically significant. In other words, including the new predictor for three four models resulted in statistically significant improvement in classification accuracy, but the reduction in FPR was not statistically significant according to the p-values in case of all models. In case of the SVM, which resulted in the highest accuracy and lowest false positive rate, the addition of the new predictor did not have any statistically significant effect according to the p-values.

Study Conclusions and Future Work

In this paper, we introduce a measure of driver aggressiveness into the modeling of driver stop/run behavior at the onset of a yellow indication. The driver aggressiveness parameter can be estimated by monitoring the driver historical response to yellow indications. The new aggressiveness parameter is based on the count of the number of runs the driver makes when the

time to intersection, at the onset of the yellow indicator, is greater than the yellow time and his speed is equal or greater than the maximum posted speed. The parameter can then be added to the model after some period of monitoring. The experimental results demonstrated the ability of the new predictor to explain part of the variability in the driver stop/run decision. Specifically, the addition of this predictor significantly increased the classification accuracy in case of three models (i.e. ANN, AdaBoost, and Random Forest). However, the reduction in FPR was not statistically significant after adding the new predictor when using all models. In case of the SVM, which resulted in the highest accuracy and lowest false positive rate, the addition of the new predictor did not have any statistically significant effect. The paper also considers different types of vehicles and dry/wet road surface as factors to explain the response variability.

Further enhancements to the model are required to model driver stop/run behavior under more severe inclement weather (such as snow, freezing rain and ice) and road surface conditions, and developing real-time machine learning techniques that can adapt to changes in driver behavior.

References

[1] R. T. Trevor Hastie, Jerome Friedman *The Elements of Statistical Learning Data Mining, Inference, and Prediction*, 2009.

[2] H. Rakha, A. Amer, and I. El-Shawarby, "Modeling driver behavior within a signalized intersection approach decision-dilemma zone," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2069, pp. 16-25, 2008.

[3] H. Rakha, I. El-Shawarby, and J. R. Setti, "Characterizing driver behavior on signalized intersection approaches at the onset of a yellow-phase trigger," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 8, pp. 630-640, 2007.

[4] S. Ghanipoor Machiani and M. Abbas, "Dynamic Driver's Perception of Dilemma Zone: Experimental Design and Analysis of Driver's Learning in a Simulator Study," in *The 93nd Annual Meeting of the Transportation Research Board*, Washington, DC, 2014.

[5] P. Papaioannou, "Driver behaviour, dilemma zone and safety effects at urban signalised intersections in Greece," *Accident Analysis & Prevention*, vol. 39, pp. 147-158, 2007.

[6] A. Sharma, D. Bullock, and S. Peeta, "Estimating dilemma zone hazard function at high speed isolated intersection," *Transportation research part C: emerging technologies*, vol. 19, pp. 400-412, 2011.

[7] D. Gazis, R. Herman, and A. Maradudin, "The Problem of the Amber Signal Light in Traffic Flow," *Operations Research*, vol. 8, pp. 112-132, 1960.

[8] T. Gates and D. Noyce, "Dilemma Zone Driver Behavior as a Function of Vehicle Type, Time of Day, and Platooning," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2149, pp. 84-93, 12/01/ 2010.

[9] I. El-Shawarby, A.-S. G. Abdel-Salam, H. Li, and H. Rakha, "Driver Behavior at the Onset of Yellow Indication for Rainy/Wet Roadway Surface Conditions."

[10] D. Shinar and R. Compton, "Aggressive driving: An observational study of driver, vehicle, and situational variables," *Accident Analysis & Prevention*, vol. 36, pp. 429-437, 2004.

[11] M. Elhenawy, H. Rakha, and I. El-Shawarby, "Enhancing Driver Stop/Run Modeling at the Onset of a Yellow Indication using Historical Behavior and Machine Learning Techniques " presented at the 93th TRB annual meeting 2014.

[12] S. S. M. Ali, N. Joshi, B. George, and L. Vanajakshi, "Application of Random Forest Algorithm to Classify Vehicles Detected by a Multiple Inductive Loop System," *2012 15th International Ieee Conference on Intelligent Transportation Systems (Itsc)*, pp. 491-495, 2012.

[13] M.-H. Pham, A. Bhaskar, E. Chung, and A.-G. Dumont, "Random Forest Models for Identifying Motorway Rear-End Crash Risks Using Disaggregate Data," presented at the 13th International IEEE Annual Conference on Intelligent Transportation Systems, Madeira Island, Portugal, 2010.

[14] Qingchao Liu, Jian Lu, and S. Chen, "Traffic Incident D etection Using Random Forest," presented at the Transportation Research Board 92nd Annual Meeting, Washington DC.

[15] L. Yulan, M. L. Reyes, and J. D. Lee, "Real-Time Detection of Driver Cognitive Distraction Using Support Vector Machines," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 8, pp. 340-350, 2007.

[16] F. Tango and M. Botta, "Real-Time Detection System of Driver Distraction Using Machine Learning," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 14, pp. 894-905, 2013.

[17] A. Jahangiri and H. Rakha, "Developing a Support Vector Machine (SVM) Classifier for Transportation Mode Identification by Using Mobile Phone Sensor Data," in *Transportation Research Board 93rd Annual Meeting*, 2014.

[18] V. Balali and M. Golparvar-Fard, "Scalable Nonparametric Parsing for Segmentation and Recognition of High-quantity, Low-cost Highway Assets from Car-mounted Video Streams," in *Construction Research Congress 2014@ sConstruction in a Global Network*, 2014, pp. 120-129.

[19] S. Ghanipoor Machiani and M. Abbas, "Predicting Drivers Decision in Dilemma Zone in a Driving Simulator Environment using Canonical Discriminant Analysis," in *The 93nd Annual Meeting of the Transportation Research Board*, Washington, DC, 2014.

[20] Y. Freund and R. Schapire, "A short introduction to boosting," *Japonese Society for Artificial Intelligence*, vol. 14, pp. 771-780, // 1999.

[21] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Cognitive modeling*, vol. 5.

[22] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *Neural Networks, IEEE Transactions on*, vol. 13, pp. 415-425, 2002.

[23] H. Rakha, I. Lucic, S. H. Demarchi, J. R. Setti, and M. Van Aerde, "Vehicle dynamics model for predicting maximum truck acceleration levels," *Journal of Transportation Engineering*, vol. 127, pp. 418-425, 2001.

[24] A. M. M. Amer, H. A. Rakha, and I. El-Shawarby, "A Behavioral Modeling Framework of Driver Behavior at Onset of Yellow a Indication at Signalized Intersections," presented at the Transportation Research Board 89th Annual Meeting, Washington DC, 2010.

[25] I. El-Shawarby, Abdel-Salam G. Abdel-Salam, H. Li, and H. Rakha, "Driver Behavior at the Onset of Yellow Indication for Rainy/Wet Roadway Surface Conditions," presented at the Transportation Research Board 91st Annual Meeting, 2012.

[26] T. Evgeniou, M. Pontil, and A. Elisseeff, "Leave One Out Error, Stability, and Generalization of Voting Combinations of Classifiers," *Machine Learning*, vol. 55, pp. 71-97, 2004/04/01 2004.

[27] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, p. 27, 2011.

[28] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification," ed, 2003.

Chapter 16: A Game-Theory-Based Algorithm for Traffic Control at Uncontrolled Isolated Intersections for Connected Vehicles Environments

This chapter is based on
Mohammed Elhenawy, Ahmed Abdelnaeim Elbery, Abdallah Hassan Mahmoud, and Hesham. Rakha, "A Game-Theory-Based Algorithm for Traffic Control at Uncontrolled Isolated Intersections for Connected Vehicles Environments," under review paper.

Abstract

Urban traffic congestion is a growing problem that we experience every day. Intersections are of the major factors contributing to urban traffic congestion. Traditional traffic control methods, such as traffic signal and stop sign control are not optimal as demonstrated in the literature. Recently, many research efforts proposed Intelligent Transportation Systems (ITS) to enhance capacity at intersections and hence to reduce congestion. In this paper we propose a game-theory-based algorithm for controlling vehicle movements which are equipped with Cooperative Adaptive Cruise Control (CACC) systems at uncontrolled intersections. The goal of this research effort is to develop an algorithm capable to use the future advanced vehicles' capabilities to replace the usual state-of-the-practice control systems at intersections (e.g. stop sign, yield signs, etc.). The proposed algorithm is chicken-game inspired and is efficient for application in real time. It assumes vehicles can communicate with a central agent at the intersection to provide their speeds and locations. The proposed algorithm assumes the vehicles obey the Nash equilibrium solution of the game as well. The simulation results show in average 49% saving in travel time.

Introduction

The state of the practice control systems at road intersections are stop signs, yield signs or traffic signals. The main design goal of these control systems is to manage traffic as well as to improve safety at intersections. Recently, some issues are raised related to the efficiency and safety of the intersections using these control systems. The intersection safety needs identification report published by federal highway administration in July 2009 showed that in 2007, 22% of the total fatal crashes were intersection-related with an estimated cost of 27.8 US billion dollars. Also, 44.8% of the total injury crashes were also intersection-related with an estimated cost of 51.3 US billion dollars [1]. Texas Transportation Institute published the 2011 Urban Mobility Report [2], which showed that the average commuter experienced 34 hours of delay in 2010 with an estimated cost of $100.9 US billion dollars. A later report in 2012 published by the same institute [3], showed how the problem is getting worse as the amount of delay experienced by the average commuter in 2011 increased to 38 hours with an estimated cost of $121.2 US billion dollars.

The above problems need innovative solutions, especially, when we know that the number of operating vehicles in the world will, at least, double by 2050 [4]. The use of autonomous vehicles is one of the promising innovative solutions, its idea dates back to 1939 when General Motors (GM) presented its vision of driverless vehicles at the World's Fairs in New York. Back then, even before appearance of computers, the aim was to reach fully-automated vehicles controlled by mechanical systems and radio controls. The appearance of

computers encouraged GM and US. Department of Transportation (USDOT) to introduce the fully-automated highway concept [5].

The Automated Highway Systems (AHS) program was established by USDOT to reduce delay and to improve safety of traffic networks using automated vehicle control [5]. At the time of establishment of AHS program, the existing technology was not mature enough, so the program was not able to continue. Yet, the AHS project paved the way for many driver's assistant systems existing in today's market.

The advances in wireless technologies and positioning systems make it possible to establish communication links between vehicles, traffic environment, and the control system. Moreover, new opportunities are introduced such as cooperative driving for the lane changing and merging in platoon [6], collision free movements of vehicles through non-signalized intersections (blind crossing) [7], etc.

Recently, the concept of Cooperative Adaptive Cruise Control (CACC) systems became feasible. CACC is an improvement of the Adaptive Cruise Control (ACC) which uses forward ranging sensors to get distance and approaching rate to the leading vehicle. ACC needs heavily processing to filter the input signals of the sensors from noise and interference. This signal filtering introduces delays which limits the ability of the ACC to follow other leading vehicle accurately. CACC overcome this limitation by getting additional information communicated over a wireless data link. CACC can get information through vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication and fuses it with the sensed information to make better and quicker decisions and to be able to follow the leading vehicle with higher accuracy. The decision taken by the CACC after fusing the information can either be to accelerate, to decelerate, or to maintain the current speed. The current generation of the CACC is not responsible for maneuverability.

The rest of this paper is organized as following; section II briefly discusses some related work in literature and the two main classes of methods for scheduling vehicles at uncontrolled intersections. Section III provides an overview of the chicken game and section IV describes the proposed algorithm. The results of the experimental work are shown in section V and then conclusions and future work are discussed in section VI.

Related Work

The main objective of the algorithms for uncontrolled intersections is to provide an efficient way for crossing vehicles to negotiate and to cooperate to get the right to access the conflict zones of uncontrolled intersections. A conflict zone in an intersection is defined as an area of the intersection where two different crossing vehicles try to access during the same time interval. The algorithms for uncontrolled intersection can be classified into two broad categories.

One category of methods use centralized control. Vehicles approaching the intersection communicate with a central controller at the intersection. The central controller responds with directions needed for every vehicle to safely cross the intersection. Dresner and Stone proposed a simple centralized framework based on First In First Out (FIFO) priorities [8]. They used a multi-agent time reservation system consisting of an intersection manager (controller) and vehicle agents. When a vehicle approaches the intersection it requests time-space slot to cross the intersection. Upon receiving the driver agent request, the controller simulates the vehicle crossing the intersection and based on the output trajectories, the controller makes decisions that avoid conflicts.

There are several research efforts that treated safe-crossing of vehicles at uncontrolled intersection as a scheduling problem. Colombo et al. solved this problem by finding the maximal controlled invariant set and checking membership in this set using an algorithm that approximated the solution [9]. This approximate solution was used to design the controller for collision avoidance. The above scheduling approach is limited because it requires a perfect state information and absence of any disturbances. Bruni et al. improved the design of the controller by removing the limitations above and designed a controller that can deal with imperfect state information and input uncertainties [10]. Both of the above controllers assumed all vehicles are equipped with driver assistance systems (controlled vehicles). Ahn et al. proposed an inserted idle-time (IIT) scheduling approach [11] to enable the design of controller in presence of multiple uncontrolled vehicles [12]. Arora et al. modeled the collision avoidance between two vehicles at intersection as a two-player zero-sum game problem [13]. In their algorithm, each vehicle was modeled using a state space model at the continuous level. The two players in this game were the control action of the first vehicle and the velocity disturbance in the second vehicle. Another game theory framework was proposed by Zohdy and Rakha to develop a heuristic optimization algorithm for vehicles equipped with CACC [4]. The proposed algorithm was centralized where automated vehicles were modeled as reactive agents communicating with a manager agent at the intersection and following its directions.

The other category of methods use distributed (decentralized) control. Vehicles approaching an intersection directly negotiate with each other and decide which vehicle gets the right-of-way at which time. Guangquan et al. defined a set of rules used to prioritize passing vehicles through an intersection [14]. Following these rules was shown to resolve conflict problems and help avoid collisions. Based on the rules, each approaching car exchanges information with other vehicles and then decides whether to preempt or to yield other cars. Makarem and Gillet proposed a decentralized algorithm using a navigation function [15]. Their algorithm prioritized vehicles based on several factors to optimize on-board energy. VanMiddlesworth et al. proposed another decentralized algorithm which required peer-to-peer communication to replace stop signs and schedule the crossing vehicles in small intersections [16]. The drawback of that algorithm is the requirement that each vehicle communicates with all other vehicles at each time step. Hassan and Rakha proposed a fully distributed algorithm which is more scalable because it required communication only between neighboring vehicles [17]. A complete intersection utilization schedule is formed after the leading vehicles on all approach lanes share information with each other. The algorithm minimized the overall intersection delay by favoring vehicles coming from heavier lanes.

Chicken Game Background

The game of chicken [18] is a non-zero game that models two conflicting drivers. Both drivers are approaching a single lane bridge from opposite directions. Drivers are competing for the right to access the bridge first. The driver, who decides to swerve away and yield the bridge to the other opposing driver, loses the game and is called the chicken while the other wins the game. If both drivers decide to go straight, they would crash and both will lose. The payoff matrix for this game is shown in Figure 82. The payoffs are chosen to show the players' preferences. The higher the payoff, the higher the player's preference. The most preferred outcome is to win and the least preferred outcome is to crash.

The Proposed Game for Isolated Intersections

In this section, we will describe our algorithm (game) to resolve the conflict between crossing vehicles at intersections. Any game consists of three basic elements, players, players' actions, and utilities (payoffs). The following subsections define these three elements and set up the game for an intersection consisting of four single-lane approaches.

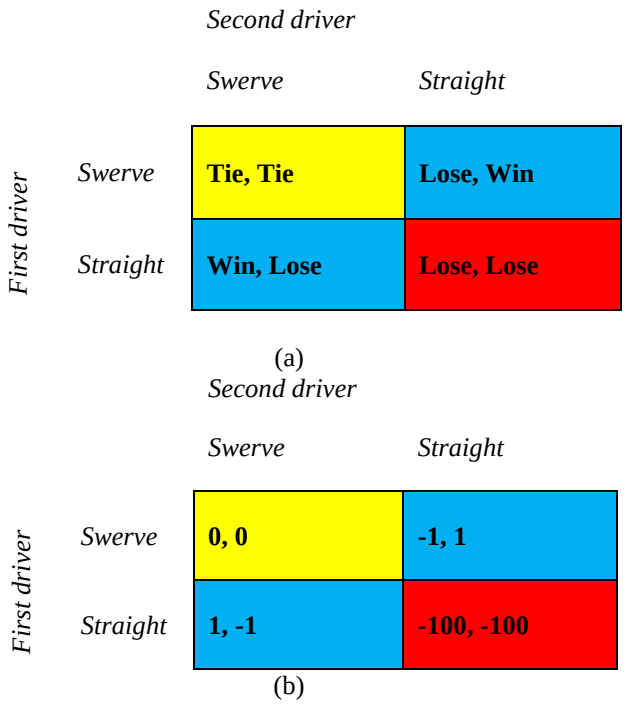


Figure 82: Chicken Game Matrix (A) Payoff Matrix (B) Numerical Payoff Matrix

1. Players

This game has only two players. The first player decides the actions of vehicles 2 and 4 while the second player decides the actions of vehicles 1 and 3 as shown in

Figure 83. Each player wants to control the vehicles in such a way to minimize their delay at the intersection and to guarantee they safely cross the intersection without getting in a crash with any conflicting vehicles.

2. Players' Actions

Depending on the speed of a vehicle, each player has three possible actions at most: to accelerate, to decelerate, or to continue at current speed. For example, if the speed of a vehicle is less than the maximum speed and greater than zero, the player can assign it one of the above three actions. When a vehicle runs at the maximum posted speed, the player has only two actions

(decelerate and constant) because the player cannot violate the law. Since each player has two vehicles and each vehicle has an action set, the player's actions are the cross product of the actions of his vehicles. For example, if the actions of car #1 are {accelerate, constant} and the actions of car #3 are {accelerate, constant, decelerate}, then player #2 actions are {accelerate, constant}X{accelerate, constant, decelerate}={(accelerate, accelerate),(constant, accelerate), (decelerate, accelerate), (accelerate, constant),(constant, constant), (decelerate, constant)}.

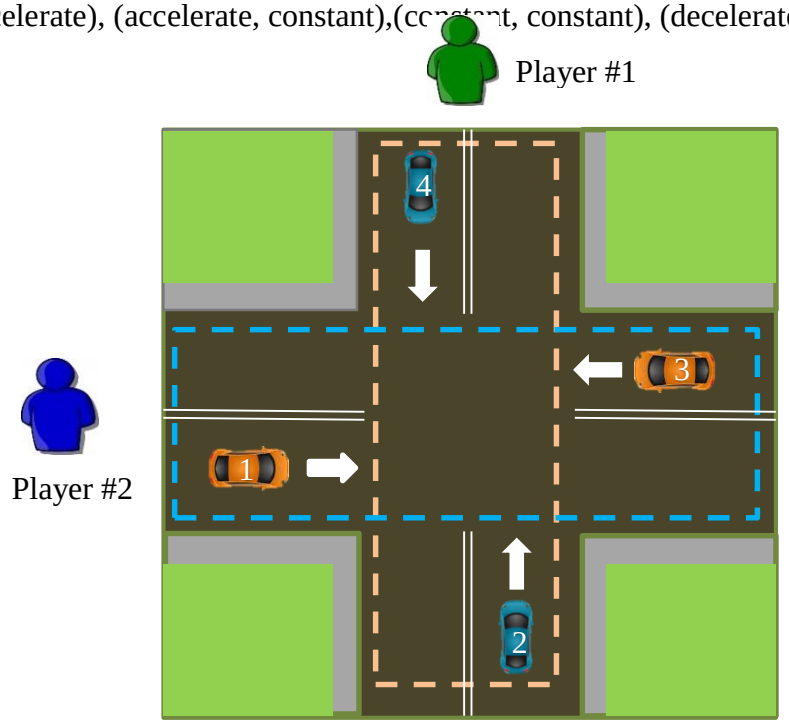


Figure 83: Illustration of Players in Proposed Game

3. Payoffs

We set up the following utilities for each player's action as shown in Table 39. The payoff utilities are chosen such that the most preferred action of the player is the action which guarantees safe crossing and minimum delay.

Table 39: Game Payoffs for One Player

Player's Action	Utility if no conflict	Utility if conflict
(accelerate, accelerate)	4	-100
(constant, accelerate)	3	-100
(decelerate, accelerate)	2	-100
(accelerate, constant)	3	-100
(constant, constant)	2	-100
(decelerate, constant)	1	-100
(accelerate, decelerate)	2	-100
(constant, decelerate)	1	-100
(decelerate, decelerate)	0	-100

4. Playing a Game to Choose the Best Players' Action

1. Whenever a vehicle gets close to the central controller agent (200m from the center of the intersection), it sends its current speed and position to the controller.

2. The controller chooses the nearest vehicle in each approach to the stop line and based on their speeds it finds the set of action for each vehicle.
3. The controller gets each player's actions by cross multiplying its vehicles' actions.
4. The controller sets up a game matrix for the current four vehicles.
5. The controller scans the matrix and for each action set of player #1 and player #2, it runs a simulation. Based on the simulation result (collision/no collision) and the game payoff values shown in Table 39, the payoffs are assigned to each player. Figure 84 shows an example of a game matrix for two players, each having four actions only.
6. The controller solves the game matrix and gets the Nash equilibrium.
7. The controller sends back to each vehicle its optimum action.

		Player #2			
		(accelerate, accelerate)	(constant, accelerate)	(accelerate, constant)	(constant, constant)
Player #1	(accelerate, accelerate)	4,4	-100,-100	4,3	4,2
	(constant, accelerate)	-100,-100	-100,-100	-100,-100	3,2
	(accelerate, constant)	-100,-100	-100,-100	-100,-100	3,2
	(constant, constant)	-100,-100	-100,-100	-100,-100	-100,-100

Figure 84: The Payoff Matrix for the Game When Player #1 Has Four Actions and Player #2 Has Four Actions. Each Cell in This Matrix Shows the Payoff For Each Player If Their Actions Did Not Cause Conflict.

Simulated Experiments

To test the proposed algorithm, a simulated study is held assuming an intersection consisting of four single lane approaches, as shown in

Figure 83. The speed limit of each approach is set to 25 mph. For each experiment, this study considers a single vehicle arriving at each approach. For each experiment, the proposed game-theory-based intersection manager is compared to an all-way stop sign controlled intersection. For both scenarios, the entrance time and the speed of each vehicle is randomly generated. The average travel time is computed for all four automated vehicles. This procedure is repeated 30000 times using a Monte Carlo simulation and the average travel time was recorded for each simulation. Figure 85 shows the percentage of reduction in the average travel time when using the proposed game theory algorithm compared to the use of the conventional stop sign control scheme.

Conclusion and Future Work

This paper proposes an algorithm for traffic control at uncontrolled intersections. The proposed algorithm is inspired by the famous chicken game. The proposed algorithm assumes perfect communication between vehicles and the controller and those vehicles are equipped with CACC. The results of the simulated experiments show that the proposed algorithm achieves 49% reduction in average delay on average when compared to the all-way stop sign controlled intersection.

Currently work is performed in extending the proposed algorithm beyond the isolated intersection case. Our future version of this algorithm aims to resolve the conflict of crossing vehicles while taking into account the traffic status at the downstream intersections. Another future extension to the work introduced in this paper is playing another game on each approach such that vehicles are platooned to improve the average delay experienced by each vehicle.

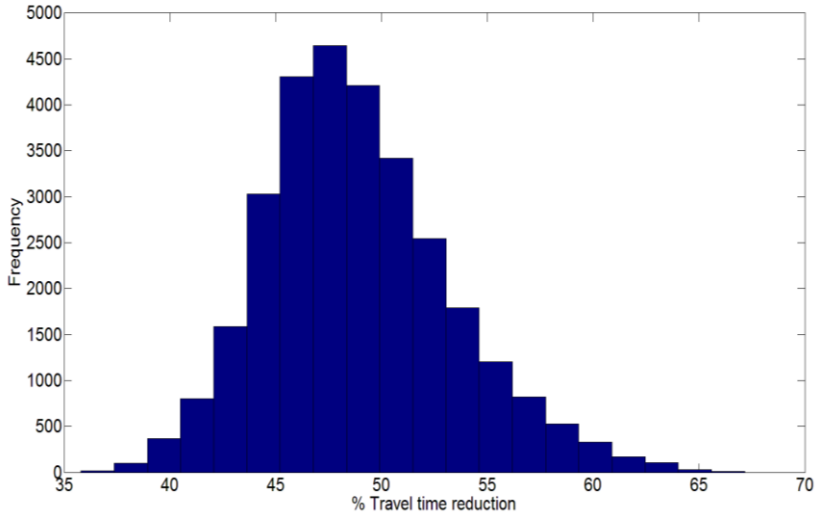


Figure 85: The Histogram of the Reduction in Travel Time When Using the Game Theory Proposed Algorithm.

References

[1] Richard C. Coakley and E. R. Stollof, "Intersection Safety Needs Identification Report," 2009.

[2] D. Schrank, T. Lomax, and B. Eisele, "2011 urban mobility report," 2011.

[3] D. Schrank, B. Eisele, and T. Lomax, "TTI's 2012 urban mobility report," *Texas A&M Transportation Institute. The Texas A&M University System*, 2012.

[4] I. H. Zohdy and H. Rakha, "Game theory algorithm for intersection-based cooperative adaptive cruise control (CACC) systems," in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, 2012, pp. 1097-1102.

[5] S. Cheon, "An overview of automated highway systems (AHS) and the social and institutional challenges they face," *University of California Transportation Center*, 2003.

[6] S. Tsugawa, "Inter-vehicle communications and their applications to intelligent vehicles: an overview," in *Intelligent Vehicle Symposium, 2002. IEEE*, 2002, pp. 564-569 vol.2.

[7] L. Li and F.-Y. Wang, "Cooperative driving at blind crossings using intervehicle communication," *Vehicular Technology, IEEE Transactions on*, vol. 55, pp. 1712-1724, 2006.

[8] K. Dresner and P. Stone, "Traffic intersections of the future," 2006.
[9] A. Colombo and D. Del Vecchio, "Efficient algorithms for collision avoidance at intersections," in *Proceedings of the 15th ACM international conference on Hybrid Systems: Computation and Control*, 2012, pp. 145-154.
[10] L. Bruni, A. Colombo, and D. Del Vecchio, "Robust multi-agent collision avoidance through scheduling."
[11] J. J. Kanet and V. Sridharan, "Scheduling with inserted idle time: problem taxonomy and literature review," *Operations Research*, vol. 48, pp. 99-110, 2000.
[12] H. Ahn, A. Colombo, and D. Del Vecchio, "Supervisory control for intersection collision avoidance in the presence of uncontrolled vehicles," in *American Control Conference (ACC), 2014*, 2014, pp. 867-873.
[13] S. Arora, A. K. Raina, and A. K. Mittal, "Collision avoidance among AGVs at junctions," in *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE*, 2000, pp. 585-589.
[14] L. Guangquan, L. Lumiao, W. Yunpeng, Z. Ran, B. Zewen, and C. Haichong, "A rule based control algorithm of connected vehicles in uncontrolled intersection," in *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, 2014, pp. 115-120.
[15] L. Makarem and D. Gillet, "Fluent coordination of autonomous vehicles at intersections," in *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on*, 2012, pp. 2557-2562.
[16] M. VanMiddlesworth, K. Dresner, and P. Stone, "Replacing the stop sign: Unmanaged intersection control for autonomous vehicles," in *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 3*, 2008, pp. 1413-1416.
[17] A. A. Hassan and H. A. Rakha, "A Fully-Distributed Heuristic Algorithm for Control of Autonomous Vehicle Movements at Isolated Intersections," *International Journal of Transportation Science and Technology*, vol. 3, pp. 297-310, 2014.
[18] Z. Han, D. Niyato, W. Saad, and T. Başar, *Game Theory in Wireless and Communication Networks: Theory, Models, and Applications* Cambridge University Press, January 2012.

Chapter 17: Research Conclusions and Recommendations for Future Work

The main goal of this research effort was to develop innovative solutions to address traffic congestion and driver safety problems. Specifically, the following objectives were addressed:

1. Develop automatic congestion identification and prediction algorithms that are suitable for real-time applications.
2. Integrate weather and visibility conditions into the automatic congestion identification and prediction algorithms.
3. Develop travel time models using machine learning and statistical learning techniques that are suitable for real-time applications.
4. Use machine learning techniques to model driver stop-run behavior at the onset of a traffic signal yellow indication to enhance the safety of traffic signal timings.
5. Develop real-time algorithms to control the movement of autonomous/automated vehicles at intersections.

The contributions of the dissertation include developing novel and innovative algorithms suitable for real-time transportation applications. These contributions are summarized as follows:

1- We developed and formulated a novel algorithm that predicts the speed of all segments of a roadway stretch (stretch-wide). The proposed algorithm determines two spaces for the multivariate predictors and multivariate responses, respectively, such that the projections of predictors (scores) and responses are equal. These new spaces are obtained through a series of matrix factorizations and multiplications. The predicted speed is used to construct the traveler trajectory and then calculate the predicted travel time. Our novel algorithm is compared with the partial least square regression (PLSR) the well-known multivariate algorithm and the outputs are quite similar, however; our novel algorithm is 4509 times faster than the PLSR in the case of the large parameters values.

2- We proposed a new measure for driver aggressiveness at traffic signalized intersections. The new measure (predictor) is based on the fact that at the onset of a yellow light, the decision whether to run the light or to stop depends on the yellow time and the time to intersection. It is clear that at the onset of a yellow light stopping is not always the best decision. Our proposed aggressiveness parameter is not simply the percentage of time a driver stops/runs during the yellow interval. The new measure is based on the number of runs the driver makes when both the time-to-intersection at the onset of the yellow indication is greater than the yellow time and the vehicular speed is equal to or greater than the posted speed limit. This number is incorporated into a Bernoulli-Beta model to estimate the driver aggressiveness.

3- We considered driver aggressiveness as a predictor to develop driver stop-run models. Adding this predictor increases the classification accuracy and reduces the false alarm which will help increase user's acceptance of the advanced driver assistance system.

4- We developed a novel game theory framework algorithm for connected vehicles equipped with Cooperative Adaptive Cruise Control (CACC) to control vehicles passing through uncontrolled intersections. Our algorithm is inspired by the chicken game to model the intersection and to define the players, strategies and payoffs relevant to the intersection dynamics.

5- To the best of our knowledge, we developed the first automatic congestion identification algorithm that incorporates weather conditions and visibility levels.
6- We developed a unified model for congestion identification applicable to any highway road system in the United States.
7- We exploited the Adaptive Boosting machine learning algorithm (AdaBoost) to develop an automatic congestion prediction algorithm which is considered one of a few research efforts to tackle this important problem.
8- We modeled travel time using a mixture of linear regression and historical data. The resultant model can be used to predict travel time and travel time reliability. Other algorithms divide the day into different time slots and fit the data of these slots to a mixture of components. Our resultant model includes means and proportional parameters that are functions of the speeds inside a window from the departure time and backwards.
9- Machine learning techniques are exploited in modeling the stop-run behavior of the driver at the onset of a yellow indication for various roadway surface conditions and different vehicle types.

The conclusions and recommendations for further research are summarized in this chapter.

Conclusions

The conclusions of the research are summarized for each of the three areas of research.

1. Congestion Identification and Prediction

Traffic congestion is one of the problems we face on a daily basis. Automatic traffic congestion identification and prediction is one of the existing innovative solutions to overcome this problem. We propose four algorithms to identify congestion.

The first algorithm is based on the one sample t-test. This algorithm uses speed measurements over short temporal and spatial intervals/segments to identify the status of the center point using t-test. The experimental results based on archived data from the northbound Interstate 5 (I-5) corridor in the Portland, Oregon, metropolitan region demonstrates significant improvements over the Chen et al. bottleneck identification algorithms. The drawback of this algorithm is its need of future speed reading and the multiple testing corrections.

The second algorithm fits a mixture of two components using historical dataset. The mixture is used to calculate a threshold to separate between free-flow and congested traffic. If the speed of a segment is greater than this threshold the segment is considered in free-flow state and is considered congested otherwise. Two datasets were used to justify the two-component assumption and to compare between the different models using normal, lognormal, and gamma distributions. The experimental results show that the lognormal mixture model is the best model if the data is skewed. The proposed algorithm overcomes the model deficiencies of ASBIA. The proposed algorithm overcomes the normality problem by using the lognormal distribution to model skewed data. It is suitable for online (real-time) application because it does not require knowledge of future speed readings. This algorithm does not consider any weather conditions or visibility level.

The third proposed algorithm uses mixture of two linear regressions to model the speed distributions at different traffic conditions including free-flow and congested traffic. To the best of our knowledge we are the first to integrate the weather conditions and the visibility level into the automatic congestion algorithm. The speed of each regime is modeled as a normal distribution its mean is function of weather condition and visibility level. The threshold that separates between free-flow and congested traffic becomes a function of weather and visibility

level as well. The proposed model overcomes the problem of limited data for some weather condition and visibility level by pooling the data during fitting the model but overestimates the thresholds separating the free-flow and congested regime

The final proposed algorithm overcomes the overestimation problem that was happening because of the assumption of two components; congestion and free-flow. This algorithm assumes that the speed distribution consists of three components; free-flow, speed at capacity, and congestion. Therefore, the algorithm uses a mixture of three linear regressions to model the speed. The mixture of three linear regressions using three datasets from Texas, Virginia, and California builds a unified model that can be used at any highway system in the United States. The results obtained using the unified model are sensible, because all weather groups have thresholds lower or equal to the clear group. Moreover, the threshold increases as visibility increases.

We proposed a bank of Adaptive Boosting (AdaBoost) machine learning classifiers to predict the status of roadway segments as either congested or free-flow. The algorithm is tested using a dataset of 24 days; it resulted in true positive rate slightly higher than 0.99 and a false positive rate slightly less than 0.001. The congestion prediction is done for 100 minutes into the future along the 22-mile test section using 20 weak learners for each AdaBoost classifier.

2. Travel Time Modeling

We propose several algorithms for travel time modeling. The proposed algorithms can be divided into two categories. The first category consists of the algorithms that take the travel time as its response. The output of these algorithms is the travel time prediction. The mixture of linear regression, random forests and genetic programming based algorithms are examples of this category. These algorithms predict the travel time directly when the input predictors are submitted to them. The problem of this category is that it gives the whole expected travel time, but the travel time to intermediate destinations on the road can't be known. The second category of algorithms predicts the speed on all the segments of the road stretch up to few hours in the future. Predicted speeds are used to construct the trajectory from any origin to any destination on the road stretch. PLSR and matrix projection based algorithms are examples of the second category. In the following paragraphs, we summarize the algorithms used to model travel time.

A genetic programming algorithm for predicting dynamic travel times is proposed. During training phase, the algorithm clusters the training data then it builds a model for each data partition. During the testing phase, the algorithm assigns the incoming observation to one cluster and hence it predicts travel time using the model corresponding to this cluster. The models obtained by the proposed algorithm are simple, computational efficiency, and interpretable. The experimental results show that the proposed algorithm achieves more than a 25 percent and 76 percent reduction in the prediction error over the instantaneous and historical average, respectively on congested days. This algorithm uses bagging to predict travel time interval. The experimental results show that the mean width of the travel time interval is less than 5 minutes for the 37-mile trip.

A random forest based algorithm is proposed to model travel time. The proposed algorithm utilizes congestion probability and spatiotemporal speed measurements as input predictors. A random forest is built to model the relationship between the predictors and the travel time. Consequently, the random forest can predict the travel time by propagating the new features vector through all trees. Besides predicting the travel time, the proposed algorithm has the advantage of giving the travel time reliability without any extra processing. The experimental results using a 37-mile freeway section in Virginia show that the proposed algorithm achieves

more than a 38 percent reduction in the prediction error on congested days compared to the state-of-practice instantaneous algorithm and 28 percent reduction when compared to the genetic programming travel time prediction algorithm. Moreover, the computed travel time confidence limits show that the mean width of the travel time interval is less than 10 minutes for the 37-mile trip.

A new algorithm based on PLSR is developed to predict the speed across road segments. This algorithm has the same advantage as the matrix projection algorithm where the predicted speeds can be used to construct the traveler trajectory from and to any origin/destination on the roadway and to calculate the corresponding travel time. The travel times predicted by this algorithm are compared to travel times predicted using pattern recognition, K-NN, and historical average methods. The proposed algorithm gives better predictions in terms of smaller MAE and MAPE. Yet, building the models is time demanding because PLSR is computationally expensive.

A novel algorithm based on matrix projection is developed to predict the spatiotemporal evolution of roadway speeds. The predicted speeds can be used to construct the traveler trajectory from and to any origin/destination on the roadway and to calculate the corresponding travel time. The main idea of this algorithm is to find two spaces with different loadings that produce identical scores for the predictor and response matrices. The proposed algorithm is compared to the K-Nearest Neighbor and to a pattern recognition algorithm developed earlier. The results demonstrate that the proposed algorithm outperforms both approaches. Moreover, our novel algorithm is compared with the PLSR and the outputs are quite similar, however; our novel algorithm is 4509 times faster than the PLSR in the case of the large parameters values. The algorithm based on matrix projection is fast because both the model building and the prediction involve only matrix multiplications.

Mixture of two linear regressions is used to formulate a travel time model capturing the stochastic nature of the travel time. The model assigns one normal component for free-flow regime and another normal component for the congested regime. The means of the components are function in the input predictors which are chosen using a random forest algorithm. The proposed model can be used to predict the travel time as well as the upper and lower bounds for the travel time. Moreover, the proposed model can be used to give the travel time reliability information at anytime of any day. The same technique is used to build travel time model but with log-normal distribution and a different set of predictors.

3. Enhanced Safety Modeling at Signalized Intersections and Intersection Control of Autonomous Vehicles

Four datasets collected from four controlled experiments are used to model stop-run behavior of the driver at the onset of the yellow indicators. Several machine learning techniques are exploited to get the best possible model in terms of high classification accuracy and low false positive rate.

We introduce a measure of driver aggressiveness into the modeling of driver stop/run behavior at the onset of a yellow indication. The driver aggressiveness parameter can be estimated by monitoring the driver historical response to yellow indications. The new aggressiveness parameter is based on the count of the number of runs the driver makes when the time to intersection, at the onset of the yellow indicator, is greater than the yellow time and his speed is equal to or greater than the posted speed limit. The parameter can then be added to the model after some period of monitoring.

We studied the effect of the new aggressiveness parameter at different road condition. The experimental results demonstrate the ability of the new predictor to explain part of the variability in the driver stop/run decision. Specifically, there is a statistically significant increase in the classification accuracy. The FPR is remarkably reduced but this reduction is not statistically significant. The study also demonstrates that the AdaBoost machine learning algorithm is the best algorithm in terms of its statistically significant improvement in FPR. However, SVM model had the best performance in terms of the classification accuracy only.

Vehicles type factor is added and we studied the effect of the new aggressiveness parameter at different road condition and different vehicles. Similarly, the experimental results show that, the addition of this predictor significantly increased the classification accuracy in case of three models (i.e. ANN, AdaBoost, and Random Forest). However, the reduction in FPR was not statistically significant after adding the new predictor when using all models. In case of the SVM, which resulted in the highest accuracy and lowest false positive rate, addition of the new predictor did not have any statistically significant effect.

A game theory based framework is proposed to handle the crossing of the vehicles through uncontrolled intersections. A four legged intersection is modeled using a chicken game inspired algorithm. The result of the Monte Carlo simulation of the proposed algorithm shows 49% reduction in the travel time compared to traditional 4-ways stop sign control.

Recommendations for Future Work

1. Congestion Identification and Prediction

Further improvements are needed for the congestion identification algorithms to extend them to major arterial roadways and to consider incidents and work zones.

Regarding the prediction algorithm, we propose several enhancements for future work. The proposed enhancements include the use of different datasets, involving other factors such as weather, work zones and incidents. The algorithm can also be modified to work on major arterial roadways.

2. Travel Time Modeling

Weather conditions, work zones and incidents are three factors to be included in future models of travel time. The problems of these factors are the lack of availability of datasets and that some interesting conditions are very rare to happen. Hence, a technique used to model travel time in presence of these factors should be chosen very carefully.

Another interesting future research work is modeling travel time at arterial roads. It is a challenging task because of stochastic delays introduced by traffic lights and congestion.

3. Enhanced Safety Modeling at Signalized Intersections and Intersection Control of Autonomous Vehicles

The driver stop-run behavior models include only three types of vehicles and two road surface conditions. Future work is to include severe weather conditions and more types of vehicles. Another factor to include in the model is the loaded/unloaded condition of a vehicle.

Extending the proposed game beyond the isolated intersection case is one of future work. The future version of the game should take into account the traffic status at the downstream intersections. Playing another game on each approach to platoon the vehicles is another extension to be considered in the future work.

Appendix H IRB letters

VirginiaTech

Office of Research Compliance
Institutational Review Board
North End Center, Suite 4120, Virginia Tech
300 Turner Street NW
Blacksburg, Virginia 24061
540/231-4606 Fax 540/231-0959
email irb@vt.edu
website http://www.irb.vt.edu

MEMORANDUM

DATE:	June 26, 2014
TO:	Hesham A Rakha, Ihab E Elshawarby, Mohammed Mamdouh Elhenawy, Boon Teck Ong, Arash Jahangiri
FROM:	Virginia Tech Institutional Review Board (FWA00000572, expires April 25, 2018)
PROTOCOL TITLE:	Signalized Intersection Red Light Running and Dilemma Zone Study
IRB NUMBER:	**08-240**

Effective June 26, 2014, the Virginia Tech Institution Review Board (IRB) Chair, David M Moore, approved the Amendment request for the above-mentioned research protocol.

This approval provides permission to begin the human subject activities outlined in the IRB-approved protocol and supporting documents.

Plans to deviate from the approved protocol and/or supporting documents must be submitted to the IRB as an amendment request and approved by the IRB prior to the implementation of any changes, regardless of how minor, except where necessary to eliminate apparent immediate hazards to the subjects. Report within 5 business days to the IRB any injuries or other unanticipated or adverse events involving risks or harms to human research subjects or others.

All investigators (listed above) are required to comply with the researcher requirements outlined at:

http://www.irb.vt.edu/pages/responsibilities.htm

(Please review responsibilities before the commencement of your research.)

PROTOCOL INFORMATION:

Approved As:	**Expedited, under 45 CFR 46.110 category(ies) 6,7**
Protocol Approval Date:	**April 15, 2014**
Protocol Expiration Date:	**April 14, 2015**
Continuing Review Due Date*:	**March 31, 2015**

*Date a Continuing Review application is due to the IRB office if human subject activities covered under this protocol, including data analysis, are to continue beyond the Protocol Expiration Date.

FEDERALLY FUNDED RESEARCH REQUIREMENTS:

Per federal regulations, 45 CFR 46.103(f), the IRB is required to compare all federally funded grant proposals/work statements to the IRB protocol(s) which cover the human research activities included in the proposal / work statement before funds are released. Note that this requirement does not apply to Exempt and Interim IRB protocols, or grants for which VT is not the primary awardee.

The table on the following page indicates whether grant proposals are related to this IRB protocol, and which of the listed proposals, if any, have been compared to this IRB protocol, if required.

Date*	OSP Number	Sponsor	Grant Comparison Conducted?
03/18/2013	08009707	Virginia Center for Transportation Innovation & Research	Not required (Not federally funded)

* Date this proposal number was compared, assessed as not requiring comparison, or comparison information was revised.

If this IRB protocol is to cover any other grant proposals, please contact the IRB office (irbadmin@vt.edu) immediately.

VirginiaTech

Office of Research Compliance
Institutational Review Board
North End Center, Suite 4120, Virginia Tech
300 Turner Street NW
Blacksburg, Virginia 24061
540/231-4606 Fax 540/231-0959
email irb@vt.edu
website http://www.irb.vt.edu

MEMORANDUM

DATE: March 21, 2014

TO: Hesham A Rakha, Ihab E Elshawarby, Mohammed Mamdouh Elhenawy, Boon Teck Ong

FROM: Virginia Tech Institutional Review Board (FWA00000572, expires April 25, 2018)

PROTOCOL TITLE: Signalized Intersection Red Light Running and Dilemma Zone Study

IRB NUMBER: **08-240**

Effective March 20, 2014, the Virginia Tech Institution Review Board (IRB) Chair, David M Moore, approved the Continuing Review request for the above-mentioned research protocol.

This approval provides permission to begin the human subject activities outlined in the IRB-approved protocol and supporting documents.

Plans to deviate from the approved protocol and/or supporting documents must be submitted to the IRB as an amendment request and approved by the IRB prior to the implementation of any changes, regardless of how minor, except where necessary to eliminate apparent immediate hazards to the subjects. Report within 5 business days to the IRB any injuries or other unanticipated or adverse events involving risks or harms to human research subjects or others.

All investigators (listed above) are required to comply with the researcher requirements outlined at:

http://www.irb.vt.edu/pages/responsibilities.htm

(Please review responsibilities before the commencement of your research.)

PROTOCOL INFORMATION:

Approved As:	**Expedited, under 45 CFR 46.110 category(ies) 6,7**
Protocol Approval Date:	**April 15, 2014**
Protocol Expiration Date:	**April 14, 2015**
Continuing Review Due Date*:	**March 31, 2015**

*Date a Continuing Review application is due to the IRB office if human subject activities covered under this protocol, including data analysis, are to continue beyond the Protocol Expiration Date.

FEDERALLY FUNDED RESEARCH REQUIREMENTS:

Per federal regulations, 45 CFR 46.103(f), the IRB is required to compare all federally funded grant proposals/work statements to the IRB protocol(s) which cover the human research activities included in the proposal / work statement before funds are released. Note that this requirement does not apply to Exempt and Interim IRB protocols, or grants for which VT is not the primary awardee.

The table on the following page indicates whether grant proposals are related to this IRB protocol, and which of the listed proposals, if any, have been compared to this IRB protocol, if required.

Date*	OSP Number	Sponsor	Grant Comparison Conducted?
03/18/2013	08009707	Virginia Center for Transportation Innovation & Research	Not required (Not federally funded)

* Date this proposal number was compared, assessed as not requiring comparison, or comparison information was revised.

If this IRB protocol is to cover any other grant proposals, please contact the IRB office (irbadmin@vt.edu) immediately.

VirginiaTech

Office of Research Compliance
Institutional Review Board
North End Center, Suite 4120, Virginia Tech
300 Turner Street NW
Blacksburg, Virginia 24061
540/231-4606 Fax 540/231-0959
email irb@vt.edu
website http://www.irb.vt.edu

MEMORANDUM

DATE: September 10, 2014

TO: Hesham A Rakha, Ihab E Elshawarby, Mohammed Mamdouh Elhenawy, Arash Jahangiri

FROM: Virginia Tech Institutional Review Board (FWA00000572, expires April 25, 2018)

PROTOCOL TITLE: Inclement Weather Yellow Timing

IRB NUMBER: 09-712

Effective September 10, 2014, the Virginia Tech Institution Review Board (IRB) Chair, David M Moore, approved the Amendment request for the above-mentioned research protocol.

This approval provides permission to begin the human subject activities outlined in the IRB-approved protocol and supporting documents.

Plans to deviate from the approved protocol and/or supporting documents must be submitted to the IRB as an amendment request and approved by the IRB prior to the implementation of any changes, regardless of how minor, except where necessary to eliminate apparent immediate hazards to the subjects. Report within 5 business days to the IRB any injuries or other unanticipated or adverse events involving risks or harms to human research subjects or others.

All investigators (listed above) are required to comply with the researcher requirements outlined at:

http://www.irb.vt.edu/pages/responsibilities.htm

(Please review responsibilities before the commencement of your research.)

PROTOCOL INFORMATION:

Approved As:	**Expedited, under 45 CFR 46.110 category(ies) 6,7**
Protocol Approval Date:	**October 2, 2013**
Protocol Expiration Date:	**October 1, 2014**
Continuing Review Due Date*:	**September 17, 2014**

*Date a Continuing Review application is due to the IRB office if human subject activities covered under this protocol, including data analysis, are to continue beyond the Protocol Expiration Date.

FEDERALLY FUNDED RESEARCH REQUIREMENTS:

Per federal regulations, 45 CFR 46.103(f), the IRB is required to compare all federally funded grant proposals/work statements to the IRB protocol(s) which cover the human research activities included in the proposal / work statement before funds are released. Note that this requirement does not apply to Exempt and Interim IRB protocols, or grants for which VT is not the primary awardee.

The table on the following page indicates whether grant proposals are related to this IRB protocol, and which of the listed proposals, if any, have been compared to this IRB protocol, if required.

Date*	OSP Number	Sponsor	Grant Comparison Conducted?
08/21/2012	06207204	Virginia Center for Transportation Innovation & Research	Compared on 06/29/2010
08/21/2012	10043708	Virginia Center for Transportation Innovation & Research	Not required (Not federally funded)

* Date this proposal number was compared, assessed as not requiring comparison, or comparison information was revised.

If this IRB protocol is to cover any other grant proposals, please contact the IRB office (irbadmin@vt.edu) immediately.

VirginiaTech

Office of Research Compliance
Institutational Review Board
North End Center, Suite 4120, Virginia Tech
300 Turner Street NW
Blacksburg, Virginia 24061
540/231-4606 Fax 540/231-0959
email irb@vt.edu
website http://www.irb.vt.edu

MEMORANDUM

DATE: June 26, 2014

TO: Hesham A Rakha, Ihab E Elshawarby, Michael Baird, Craig William Bryant, Mohammed Mamdouh Elhenawy, Boon Teck Ong, Arash Jahangiri

FROM: Virginia Tech Institutional Review Board (FWA00000572, expires April 25, 2018)

PROTOCOL TITLE: Inclement Weather Yellow Timing

IRB NUMBER: 09-712

Effective June 26, 2014, the Virginia Tech Institution Review Board (IRB) Chair, David M Moore, approved the Amendment request for the above-mentioned research protocol.

This approval provides permission to begin the human subject activities outlined in the IRB-approved protocol and supporting documents.

Plans to deviate from the approved protocol and/or supporting documents must be submitted to the IRB as an amendment request and approved by the IRB prior to the implementation of any changes, regardless of how minor, except where necessary to eliminate apparent immediate hazards to the subjects. Report within 5 business days to the IRB any injuries or other unanticipated or adverse events involving risks or harms to human research subjects or others.

All investigators (listed above) are required to comply with the researcher requirements outlined at:

http://www.irb.vt.edu/pages/responsibilities.htm

(Please review responsibilities before the commencement of your research.)

PROTOCOL INFORMATION:

Approved As:	**Expedited, under 45 CFR 46.110 category(ies) 6,7**
Protocol Approval Date:	**October 2, 2013**
Protocol Expiration Date:	**October 1, 2014**
Continuing Review Due Date*:	**September 17, 2014**

*Date a Continuing Review application is due to the IRB office if human subject activities covered under this protocol, including data analysis, are to continue beyond the Protocol Expiration Date.

FEDERALLY FUNDED RESEARCH REQUIREMENTS:

Per federal regulations, 45 CFR 46.103(f), the IRB is required to compare all federally funded grant proposals/work statements to the IRB protocol(s) which cover the human research activities included in the proposal / work statement before funds are released. Note that this requirement does not apply to Exempt and Interim IRB protocols, or grants for which VT is not the primary awardee.

The table on the following page indicates whether grant proposals are related to this IRB protocol, and which of the listed proposals, if any, have been compared to this IRB protocol, if required.

Date*	OSP Number	Sponsor	Grant Comparison Conducted?
08/21/2012	06207204	Virginia Center for Transportation Innovation & Research	Compared on 06/29/2010
08/21/2012	10043708	Virginia Center for Transportation Innovation & Research	Not required (Not federally funded)

* Date this proposal number was compared, assessed as not requiring comparison, or comparison information was revised.

If this IRB protocol is to cover any other grant proposals, please contact the IRB office (irbadmin@vt.edu) immediately.

VirginiaTech

Office of Research Compliance
Institutational Review Board
North End Center, Suite 4120, Virginia Tech
300 Turner Street NW
Blacksburg, Virginia 24061
540/231-4606 Fax 540/231-0959
email irb@vt.edu
website http://www.irb.vt.edu

MEMORANDUM

DATE: September 5, 2013

TO: Hesham A Rakha, Ihab E Elshawarby, Michael Baird, Craig William Bryant, Mohammed Mamdouh Elhenawy, Boon Teck Ong

FROM: Virginia Tech Institutional Review Board (FWA00000572, expires April 25, 2018)

PROTOCOL TITLE: Inclement Weather Yellow Timing

IRB NUMBER: 09-712

Effective September 4, 2013, the Virginia Tech Institution Review Board (IRB) Chair, David M Moore, approved the Continuing Review request for the above-mentioned research protocol.

This approval provides permission to begin the human subject activities outlined in the IRB-approved protocol and supporting documents.

Plans to deviate from the approved protocol and/or supporting documents must be submitted to the IRB as an amendment request and approved by the IRB prior to the implementation of any changes, regardless of how minor, except where necessary to eliminate apparent immediate hazards to the subjects. Report within 5 business days to the IRB any injuries or other unanticipated or adverse events involving risks or harms to human research subjects or others.

All investigators (listed above) are required to comply with the researcher requirements outlined at:

http://www.irb.vt.edu/pages/responsibilities.htm

(Please review responsibilities before the commencement of your research.)

PROTOCOL INFORMATION:

Approved As:	**Expedited, under 45 CFR 46.110 category(ies) 6,7**
Protocol Approval Date:	**October 2, 2013**
Protocol Expiration Date:	**October 1, 2014**
Continuing Review Due Date*:	**September 17, 2014**

*Date a Continuing Review application is due to the IRB office if human subject activities covered under this protocol, including data analysis, are to continue beyond the Protocol Expiration Date.

FEDERALLY FUNDED RESEARCH REQUIREMENTS:

Per federal regulations, 45 CFR 46.103(f), the IRB is required to compare all federally funded grant proposals/work statements to the IRB protocol(s) which cover the human research activities included in the proposal / work statement before funds are released. Note that this requirement does not apply to Exempt and Interim IRB protocols, or grants for which VT is not the primary awardee.

The table on the following page indicates whether grant proposals are related to this IRB protocol, and which of the listed proposals, if any, have been compared to this IRB protocol, if required.

Date*	OSP Number	Sponsor	Grant Comparison Conducted?
08/21/2012	06207204	Virginia Center for Transportation Innovation & Research	Compared on 06/29/2010
08/21/2012	10043708	Virginia Center for Transportation Innovation & Research	Not required (Not federally funded)

* Date this proposal number was compared, assessed as not requiring comparison, or comparison information was revised.

If this IRB protocol is to cover any other grant proposals, please contact the IRB office (irbadmin@vt.edu) immediately.

VirginiaTech

Office of Research Compliance
Institutational Review Board
North End Center, Suite 4120, Virginia Tech
300 Turner Street NW
Blacksburg, Virginia 24061
540/231-4606 Fax 540/231-0959
email irb@vt.edu
website http://www.irb.vt.edu

MEMORANDUM

DATE: April 23, 2013

TO: Hesham A Rakha, Ihab E Elshawarby, Michael Baird, Craig William Bryant, Mohammed Mamdouh Elhenawy

FROM: Virginia Tech Institutional Review Board (FWA00000572, expires May 31, 2014)

PROTOCOL TITLE: Inclement Weather Yellow Timing

IRB NUMBER: 09-712

Effective April 23, 2013, the Virginia Tech Institution Review Board (IRB) Chair, David M Moore, approved the Amendment request for the above-mentioned research protocol.

This approval provides permission to begin the human subject activities outlined in the IRB-approved protocol and supporting documents.

Plans to deviate from the approved protocol and/or supporting documents must be submitted to the IRB as an amendment request and approved by the IRB prior to the implementation of any changes, regardless of how minor, except where necessary to eliminate apparent immediate hazards to the subjects. Report within 5 business days to the IRB any injuries or other unanticipated or adverse events involving risks or harms to human research subjects or others.

All investigators (listed above) are required to comply with the researcher requirements outlined at:

http://www.irb.vt.edu/pages/responsibilities.htm

(Please review responsibilities before the commencement of your research.)

PROTOCOL INFORMATION:

Approved As:	**Expedited, under 45 CFR 46.110 category(ies) 6,7**
Protocol Approval Date:	**October 2, 2012**
Protocol Expiration Date:	**October 1, 2013**
Continuing Review Due Date*:	**September 17, 2013**

*Date a Continuing Review application is due to the IRB office if human subject activities covered under this protocol, including data analysis, are to continue beyond the Protocol Expiration Date.

FEDERALLY FUNDED RESEARCH REQUIREMENTS:

Per federal regulations, 45 CFR 46.103(f), the IRB is required to compare all federally funded grant proposals/work statements to the IRB protocol(s) which cover the human research activities included in the proposal / work statement before funds are released. Note that this requirement does not apply to Exempt and Interim IRB protocols, or grants for which VT is not the primary awardee.

The table on the following page indicates whether grant proposals are related to this IRB protocol, and which of the listed proposals, if any, have been compared to this IRB protocol, if required.

Date*	OSP Number	Sponsor	Grant Comparison Conducted?
08/21/2012	06207204	Virginia Center for Transportation Innovation & Research	Compared on 06/29/2010
08/21/2012	10043708	Virginia Center for Transportation Innovation & Research	Not required (Not federally funded)

* Date this proposal number was compared, assessed as not requiring comparison, or comparison information was revised.

If this IRB protocol is to cover any other grant proposals, please contact the IRB office (irbadmin@vt.edu) immediately.

VirginiaTech

Office of Research Compliance
Institutional Review Board
North End Center, Suite 4120, Virginia Tech
300 Turner Street NW
Blacksburg, Virginia 24061
540/231-4606 Fax 540/231-0959
email irb@vt.edu
website http://www.irb.vt.edu

MEMORANDUM

DATE: October 22, 2014

TO: Hesham A Rakha, Ihab E Elshawarby, Craig William Bryant, Boon Teck Ong, Mohammed Mamdouh Elhenawy, Arash Jahangiri

FROM: Virginia Tech Institutional Review Board (FWA00000572, expires April 25, 2018)

PROTOCOL TITLE: Truck Simulator Dilemma Zone

IRB NUMBER: 13-374

Effective October 22, 2014, the Virginia Tech Institution Review Board (IRB) Chair, David M Moore, approved the Amendment request for the above-mentioned research protocol.

This approval provides permission to begin the human subject activities outlined in the IRB-approved protocol and supporting documents.

Plans to deviate from the approved protocol and/or supporting documents must be submitted to the IRB as an amendment request and approved by the IRB prior to the implementation of any changes, regardless of how minor, except where necessary to eliminate apparent immediate hazards to the subjects. Report within 5 business days to the IRB any injuries or other unanticipated or adverse events involving risks or harms to human research subjects or others.

All investigators (listed above) are required to comply with the researcher requirements outlined at:

http://www.irb.vt.edu/pages/responsibilities.htm

(Please review responsibilities before the commencement of your research.)

PROTOCOL INFORMATION:

Approved As:	**Expedited, under 45 CFR 46.110 category(ies) 6,7**
Protocol Approval Date:	**April 8, 2014**
Protocol Expiration Date:	**April 7, 2015**
Continuing Review Due Date*:	**March 24, 2015**

*Date a Continuing Review application is due to the IRB office if human subject activities covered under this protocol, including data analysis, are to continue beyond the Protocol Expiration Date.

FEDERALLY FUNDED RESEARCH REQUIREMENTS:

Per federal regulations, 45 CFR 46.103(f), the IRB is required to compare all federally funded grant proposals/work statements to the IRB protocol(s) which cover the human research activities included in the proposal / work statement before funds are released. Note that this requirement does not apply to Exempt and Interim IRB protocols, or grants for which VT is not the primary awardee.

The table on the following page indicates whether grant proposals are related to this IRB protocol, and which of the listed proposals, if any, have been compared to this IRB protocol, if required.

Date*	OSP Number	Sponsor	Grant Comparison Conducted?
04/10/2013	06207204	Virginia Center for Transportation Innovation & Research	Compared on 04/10/2013

* Date this proposal number was compared, assessed as not requiring comparison, or comparison information was revised.

If this IRB protocol is to cover any other grant proposals, please contact the IRB office (irbadmin@vt.edu) immediately.

VirginiaTech

Office of Research Compliance
Institutional Review Board
North End Center, Suite 4120, Virginia Tech
300 Turner Street NW
Blacksburg, Virginia 24061
540/231-4606 Fax 540/231-0959
email irb@vt.edu
website http://www.irb.vt.edu

MEMORANDUM

DATE: September 17, 2014

TO: Hesham A Rakha, Ihab E Elshawarby, Craig William Bryant, Boon Teck Ong, Mohammed Mamdouh Elhenawy

FROM: Virginia Tech Institutional Review Board (FWA00000572, expires April 25, 2018)

PROTOCOL TITLE: Truck Simulator Dilemma Zone

IRB NUMBER: 13-374

Effective September 17, 2014, the Virginia Tech Institution Review Board (IRB) Chair, David M Moore, approved the Amendment request for the above-mentioned research protocol.

This approval provides permission to begin the human subject activities outlined in the IRB-approved protocol and supporting documents.

Plans to deviate from the approved protocol and/or supporting documents must be submitted to the IRB as an amendment request and approved by the IRB prior to the implementation of any changes, regardless of how minor, except where necessary to eliminate apparent immediate hazards to the subjects. Report within 5 business days to the IRB any injuries or other unanticipated or adverse events involving risks or harms to human research subjects or others.

All investigators (listed above) are required to comply with the researcher requirements outlined at:

http://www.irb.vt.edu/pages/responsibilities.htm

(Please review responsibilities before the commencement of your research.)

PROTOCOL INFORMATION:

Approved As:	**Expedited, under 45 CFR 46.110 category(ies) 6,7**
Protocol Approval Date:	**April 8, 2014**
Protocol Expiration Date:	**April 7, 2015**
Continuing Review Due Date*:	**March 24, 2015**

*Date a Continuing Review application is due to the IRB office if human subject activities covered under this protocol, including data analysis, are to continue beyond the Protocol Expiration Date.

FEDERALLY FUNDED RESEARCH REQUIREMENTS:

Per federal regulations, 45 CFR 46.103(f), the IRB is required to compare all federally funded grant proposals/work statements to the IRB protocol(s) which cover the human research activities included in the proposal / work statement before funds are released. Note that this requirement does not apply to Exempt and Interim IRB protocols, or grants for which VT is not the primary awardee.

The table on the following page indicates whether grant proposals are related to this IRB protocol, and which of the listed proposals, if any, have been compared to this IRB protocol, if required.

Date*	OSP Number	Sponsor	Grant Comparison Conducted?
04/10/2013	06207204	Virginia Center for Transportation Innovation & Research	Compared on 04/10/2013

* Date this proposal number was compared, assessed as not requiring comparison, or comparison information was revised.

If this IRB protocol is to cover any other grant proposals, please contact the IRB office (irbadmin@vt.edu) immediately.

VirginiaTech

Office of Research Compliance
Institutional Review Board
North End Center, Suite 4120, Virginia Tech
300 Turner Street NW
Blacksburg, Virginia 24061
540/231-4606 Fax 540/231-0959
email irb@vt.edu
website http://www.irb.vt.edu

MEMORANDUM

DATE: March 19, 2015

TO: Hesham A Rakha, Ihab E Elshawarby, Craig William Bryant, Boon Teck Ong, Mohammed Mamdouh Elhenawy, Arash Jahangiri

FROM: Virginia Tech Institutional Review Board (FWA00000572, expires April 25, 2018)

PROTOCOL TITLE: Truck Simulator Dilemma Zone

IRB NUMBER: 13-374

Effective March 19, 2015, the Virginia Tech Institution Review Board (IRB) Chair, David M Moore, approved the Continuing Review request for the above-mentioned research protocol.

This approval provides permission to begin the human subject activities outlined in the IRB-approved protocol and supporting documents.

Plans to deviate from the approved protocol and/or supporting documents must be submitted to the IRB as an amendment request and approved by the IRB prior to the implementation of any changes, regardless of how minor, except where necessary to eliminate apparent immediate hazards to the subjects. Report within 5 business days to the IRB any injuries or other unanticipated or adverse events involving risks or harms to human research subjects or others.

All investigators (listed above) are required to comply with the researcher requirements outlined at:

http://www.irb.vt.edu/pages/responsibilities.htm

(Please review responsibilities before the commencement of your research.)

PROTOCOL INFORMATION:

Approved As:	**Expedited, under 45 CFR 46.110 category(ies) 6,7**
Protocol Approval Date:	**April 8, 2015**
Protocol Expiration Date:	**April 7, 2016**
Continuing Review Due Date*:	**March 24, 2016**

*Date a Continuing Review application is due to the IRB office if human subject activities covered under this protocol, including data analysis, are to continue beyond the Protocol Expiration Date.

FEDERALLY FUNDED RESEARCH REQUIREMENTS:

Per federal regulations, 45 CFR 46.103(f), the IRB is required to compare all federally funded grant proposals/work statements to the IRB protocol(s) which cover the human research activities included in the proposal / work statement before funds are released. Note that this requirement does not apply to Exempt and Interim IRB protocols, or grants for which VT is not the primary awardee.

The table on the following page indicates whether grant proposals are related to this IRB protocol, and which of the listed proposals, if any, have been compared to this IRB protocol, if required.

Date*	OSP Number	Sponsor	Grant Comparison Conducted?
04/10/2013	06207204	Virginia Center for Transportation Innovation & Research	Compared on 04/10/2013

* Date this proposal number was compared, assessed as not requiring comparison, or comparison information was revised.

If this IRB protocol is to cover any other grant proposals, please contact the IRB office (irbadmin@vt.edu) immediately.

VirginiaTech

Office of Research Compliance
Institutational Review Board
North End Center, Suite 4120, Virginia Tech
300 Turner Street NW
Blacksburg, Virginia 24061
540/231-4606 Fax 540/231-0959
email irb@vt.edu
website http://www.irb.vt.edu

MEMORANDUM

DATE: June 26, 2014

TO: Hesham A Rakha, Ihab E Elshawarby, Mohammed Mamdouh Elhenawy, Arash Jahangiri

FROM: Virginia Tech Institutional Review Board (FWA00000572, expires April 25, 2018)

PROTOCOL TITLE: Driver Behavior Modeling at Signalized Intersections

IRB NUMBER: 13-447

Effective June 26, 2014, the Virginia Tech Institution Review Board (IRB) Chair, David M Moore, approved the Amendment request for the above-mentioned research protocol.

This approval provides permission to begin the human subject activities outlined in the IRB-approved protocol and supporting documents.

Plans to deviate from the approved protocol and/or supporting documents must be submitted to the IRB as an amendment request and approved by the IRB prior to the implementation of any changes, regardless of how minor, except where necessary to eliminate apparent immediate hazards to the subjects. Report within 5 business days to the IRB any injuries or other unanticipated or adverse events involving risks or harms to human research subjects or others.

All investigators (listed above) are required to comply with the researcher requirements outlined at:

http://www.irb.vt.edu/pages/responsibilities.htm

(Please review responsibilities before the commencement of your research.)

PROTOCOL INFORMATION:

Approved As:	**Expedited, under 45 CFR 46.110 category(ies) 5**
Protocol Approval Date:	**May 1, 2014**
Protocol Expiration Date:	**April 30, 2015**
Continuing Review Due Date*:	**April 16, 2015**

*Date a Continuing Review application is due to the IRB office if human subject activities covered under this protocol, including data analysis, are to continue beyond the Protocol Expiration Date.

FEDERALLY FUNDED RESEARCH REQUIREMENTS:

Per federal regulations, 45 CFR 46.103(f), the IRB is required to compare all federally funded grant proposals/work statements to the IRB protocol(s) which cover the human research activities included in the proposal / work statement before funds are released. Note that this requirement does not apply to Exempt and Interim IRB protocols, or grants for which VT is not the primary awardee.

The table on the following page indicates whether grant proposals are related to this IRB protocol, and which of the listed proposals, if any, have been compared to this IRB protocol, if required.

Date*	OSP Number	Sponsor	Grant Comparison Conducted?

* Date this proposal number was compared, assessed as not requiring comparison, or comparison information was revised.

If this IRB protocol is to cover any other grant proposals, please contact the IRB office (irbadmin@vt.edu) immediately.

VirginiaTech

Office of Research Compliance
Institutional Review Board
North End Center, Suite 4120, Virginia Tech
300 Turner Street NW
Blacksburg, Virginia 24061
540/231-4606 Fax 540/231-0959
email irb@vt.edu
website http://www.irb.vt.edu

MEMORANDUM

DATE: September 18, 2014

TO: Hesham A Rakha, Ihab E Elshawarby, Boon Teck Ong, Mohammed Mamdouh Elhenawy

FROM: Virginia Tech Institutional Review Board (FWA00000572, expires April 25, 2018)

PROTOCOL TITLE: Bus Dilemma Zone

IRB NUMBER: 13-835

Effective September 17, 2014, the Virginia Tech Institution Review Board (IRB) Chair, David M Moore, approved the Continuing Review request for the above-mentioned research protocol.

This approval provides permission to begin the human subject activities outlined in the IRB-approved protocol and supporting documents.

Plans to deviate from the approved protocol and/or supporting documents must be submitted to the IRB as an amendment request and approved by the IRB prior to the implementation of any changes, regardless of how minor, except where necessary to eliminate apparent immediate hazards to the subjects. Report within 5 business days to the IRB any injuries or other unanticipated or adverse events involving risks or harms to human research subjects or others.

All investigators (listed above) are required to comply with the researcher requirements outlined at:

http://www.irb.vt.edu/pages/responsibilities.htm

(Please review responsibilities before the commencement of your research.)

PROTOCOL INFORMATION:

Approved As:	**Expedited, under 45 CFR 46.110 category(ies) 6,7**
Protocol Approval Date:	**October 3, 2014**
Protocol Expiration Date:	**October 2, 2015**
Continuing Review Due Date*:	**September 18, 2015**

*Date a Continuing Review application is due to the IRB office if human subject activities covered under this protocol, including data analysis, are to continue beyond the Protocol Expiration Date.

FEDERALLY FUNDED RESEARCH REQUIREMENTS:

Per federal regulations, 45 CFR 46.103(f), the IRB is required to compare all federally funded grant proposals/work statements to the IRB protocol(s) which cover the human research activities included in the proposal / work statement before funds are released. Note that this requirement does not apply to Exempt and Interim IRB protocols, or grants for which VT is not the primary awardee.

The table on the following page indicates whether grant proposals are related to this IRB protocol, and which of the listed proposals, if any, have been compared to this IRB protocol, if required.

Date*	OSP Number	Sponsor	Grant Comparison Conducted?
10/03/2013	06207204	Virginia Center for Transportation Innovation & Research	Compared on 10/03/2013

* Date this proposal number was compared, assessed as not requiring comparison, or comparison information was revised.

If this IRB protocol is to cover any other grant proposals, please contact the IRB office (irbadmin@vt.edu) immediately.

VirginiaTech

Office of Research Compliance
Institutional Review Board
North End Center, Suite 4120, Virginia Tech
300 Turner Street NW
Blacksburg, Virginia 24061
540/231-4606 Fax 540/231-0959
email irb@vt.edu
website http://www.irb.vt.edu

MEMORANDUM

DATE: September 10, 2014

TO: Hesham A Rakha, Ihab E Elshawarby, Boon Teck Ong, Mohammed Mamdouh Elhenawy

FROM: Virginia Tech Institutional Review Board (FWA00000572, expires April 25, 2018)

PROTOCOL TITLE: Bus Dilemma Zone

IRB NUMBER: 13-835

Effective September 10, 2014, the Virginia Tech Institution Review Board (IRB) Chair, David M Moore, approved the Amendment request for the above-mentioned research protocol.

This approval provides permission to begin the human subject activities outlined in the IRB-approved protocol and supporting documents.

Plans to deviate from the approved protocol and/or supporting documents must be submitted to the IRB as an amendment request and approved by the IRB prior to the implementation of any changes, regardless of how minor, except where necessary to eliminate apparent immediate hazards to the subjects. Report within 5 business days to the IRB any injuries or other unanticipated or adverse events involving risks or harms to human research subjects or others.

All investigators (listed above) are required to comply with the researcher requirements outlined at:

http://www.irb.vt.edu/pages/responsibilities.htm

(Please review responsibilities before the commencement of your research.)

PROTOCOL INFORMATION:

Approved As:	**Expedited, under 45 CFR 46.110 category(ies) 6,7**
Protocol Approval Date:	**October 3, 2013**
Protocol Expiration Date:	**October 2, 2014**
Continuing Review Due Date*:	**September 18, 2014**

*Date a Continuing Review application is due to the IRB office if human subject activities covered under this protocol, including data analysis, are to continue beyond the Protocol Expiration Date.

FEDERALLY FUNDED RESEARCH REQUIREMENTS:

Per federal regulations, 45 CFR 46.103(f), the IRB is required to compare all federally funded grant proposals/work statements to the IRB protocol(s) which cover the human research activities included in the proposal / work statement before funds are released. Note that this requirement does not apply to Exempt and Interim IRB protocols, or grants for which VT is not the primary awardee.

The table on the following page indicates whether grant proposals are related to this IRB protocol, and which of the listed proposals, if any, have been compared to this IRB protocol, if required.

Date*	OSP Number	Sponsor	Grant Comparison Conducted?
10/03/2013	06207204	Virginia Center for Transportation Innovation & Research	Compared on 10/03/2013

* Date this proposal number was compared, assessed as not requiring comparison, or comparison information was revised.

If this IRB protocol is to cover any other grant proposals, please contact the IRB office (irbadmin@vt.edu) immediately.

Office of Research Compliance
Institutional Review Board
North End Center, Suite 4120, Virginia Tech
300 Turner Street NW
Blacksburg, Virginia 24061
540/231-4606 Fax 540/231-0959
email irb@vt.edu
website http://www.irb.vt.edu

MEMORANDUM

DATE: October 22, 2014

TO: Hesham A Rakha, Ihab E Elshawarby, Boon Teck Ong, Mohammed Mamdouh Elhenawy, Arash Jahangiri

FROM: Virginia Tech Institutional Review Board (FWA00000572, expires April 25, 2018)

PROTOCOL TITLE: Bus Dilemma Zone

IRB NUMBER: 13-835

Effective October 22, 2014, the Virginia Tech Institution Review Board (IRB) Chair, David M Moore, approved the Amendment request for the above-mentioned research protocol.

This approval provides permission to begin the human subject activities outlined in the IRB-approved protocol and supporting documents.

Plans to deviate from the approved protocol and/or supporting documents must be submitted to the IRB as an amendment request and approved by the IRB prior to the implementation of any changes, regardless of how minor, except where necessary to eliminate apparent immediate hazards to the subjects. Report within 5 business days to the IRB any injuries or other unanticipated or adverse events involving risks or harms to human research subjects or others.

All investigators (listed above) are required to comply with the researcher requirements outlined at:

http://www.irb.vt.edu/pages/responsibilities.htm

(Please review responsibilities before the commencement of your research.)

PROTOCOL INFORMATION:

Approved As:	**Expedited, under 45 CFR 46.110 category(ies) 6,7**
Protocol Approval Date:	**October 3, 2014**
Protocol Expiration Date:	**October 2, 2015**
Continuing Review Due Date*:	**September 18, 2015**

*Date a Continuing Review application is due to the IRB office if human subject activities covered under this protocol, including data analysis, are to continue beyond the Protocol Expiration Date.

FEDERALLY FUNDED RESEARCH REQUIREMENTS:

Per federal regulations, 45 CFR 46.103(f), the IRB is required to compare all federally funded grant proposals/work statements to the IRB protocol(s) which cover the human research activities included in the proposal / work statement before funds are released. Note that this requirement does not apply to Exempt and Interim IRB protocols, or grants for which VT is not the primary awardee.

The table on the following page indicates whether grant proposals are related to this IRB protocol, and which of the listed proposals, if any, have been compared to this IRB protocol, if required.

Date*	OSP Number	Sponsor	Grant Comparison Conducted?
10/03/2013	06207204	Virginia Center for Transportation Innovation & Research	Compared on 10/03/2013

* Date this proposal number was compared, assessed as not requiring comparison, or comparison information was revised.

If this IRB protocol is to cover any other grant proposals, please contact the IRB office (irbadmin@vt.edu) immediately.