

© 2016 by Fatemeh Saremi

PARTICIPATORY SENSING FUEL-EFFICIENT NAVIGATION
SYSTEM GREENGPS

BY

FATEMEH SAREMI

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2016

Urbana, Illinois

Doctoral Committee:

Professor Tarek Abdelzaher, Chair
Professor Nitin Vaidya
Professor Klara Nahrstedt
Associate Professor Xue Liu, McGill University

ABSTRACT

The proliferation of smartphones has led to increased interest in mobile participatory sensing. This paradigm enables low cost establishment of a wide range of applications in variety of domains, including environmental monitoring, transportation, safety, healthcare, social networks, urban sensing, etc. This thesis proposes, designs and develops a novel application in this genre, called *GreenGPS*, which owes its practicality to the widespread usage of smart mobile devices. GreenGPS is a navigation service that finds fuel optimal routes, customized to individual drivers and vehicles, between arbitrary end-points.

This thesis studies research challenges revealed in development of GreenGPS on how to build an easy-to-deploy and inexpensive participatory sensing system to support data collection, how to generalize sparse samples of high-dimensional spaces to develop models of complex nonlinear phenomena, how to build a general but personalizable fuel-saving navigation system, how to infer the information on location and type of traffic regulators with low effort and expense, and how to insure reliability of the modeling throughout the lifetime of the service, especially the early deployment stage through which service adoption is sparse and proper modeling facilitates getting the participatory sensing based system off the ground and surviving conditions of sparse deployment.

GreenGPS navigation service is offered in both web-based and smartphone

application forms. To launch GreenGPS, we deployed a medium scaled vehicular participatory sensing system, consisting of 46 user subjects, collecting over 6700 miles of GPS driving data. To provide a testbed for future transportation fuel saving research, we started to deploy GreenGPS on over 100 vehicles of UIUC Facilities and Services fleet. To give the reader a sense of how effective are route choices provisioned by GreenGPS, it was assessed that compared to alternative fastest and shortest routes provided by traditional navigation tools, green routes are respectively 21.5% and 11.2% more fuel economic. The GreenGPS fuel optimal routes were further compared to Garmin ecoRoutes, a well-known commercial GPS product, and presented 8.4% more fuel savings.

In the Name of Allah

*To whom,
the world is waiting for (AJ)*

*To my dear family,
for their unconditional love and sacrifices*

ACKNOWLEDGMENTS

First and foremost, I thank God, the compassionate, the merciful, the all-knowing, for everything.

I would like to thank my advisor Professor Tarek Abdelzaher for his guidance and support throughout my PhD career. I appreciate all his contributions of time and effort. I would like to extend my sincere thanks to my committee members Professor Nitin Vaidya, Professor Klara Nahrstedt, and Professor Xue Liu for their valuable feedback to improve the dissertation.

My appreciation goes to my collaborators Raghu Ganti, Omid Fatemieh, Hossein Ahmadi, Hongyan Wang, Hengchang Liu, Shaohan Hu, Shen Li, Lu Su, Yusuf Sarwar, and Praveen Jayachandran. I am also deeply thankful to my dear friends who stood by my side in times of need.

I would like to express my deepest gratitude to my husband Hasan, for his love and unceasing support. My heartfelt thanks to my parents for their unconditional love and sacrifices; to my mother who has been my symbol of patience and strength, and in memory of my father. I am deeply indebted to them for all I have accomplished. I am very grateful to my sisters, Vida and Azam, and to my brothers, Reza and Meraj, for being very supportive. Words cannot describe how much I appreciate them for being the best family, one can ever have.

This work was funded in part by the National Science Foundation and IBM Research under various grants.

TABLE OF CONTENTS

LIST OF TABLES	ix
LIST OF FIGURES	x
CHAPTER 1 INTRODUCTION	1
1.1 Thesis Statement	4
1.2 Thesis Contributions	5
1.3 Thesis Outline	8
CHAPTER 2 RELATED WORK	10
2.1 Participatory Sensing	10
2.2 Transportation Fuel Saving	11
2.2.1 Routing and Driving Pattern	11
2.2.2 Speed Adjustment and Traffic Signals	12
2.2.3 Fleet Management, Real-time Traffic, Misc.	13
2.3 Traffic Regulator Detection	15
2.3.1 Image Processing	15
2.3.2 Vehicular GPS Traces	15
2.3.3 Traffic Signals Schedule	16
2.3.4 Road Events	16
2.4 Sample Size Planning	17
CHAPTER 3 GREEN NAVIGATION MODELING	19
3.1 GreenGPS System	22
3.2 Derivation of Model Structure	25
3.3 Model Generalization to Predict Green Routes	32
3.3.1 Model Evaluation: One Size Fits All?	33
3.3.2 Model Clustering	36
3.4 Dynamic Traffic Modeling	39
CHAPTER 4 TRAFFIC REGULATOR DETECTION	44
4.1 Modeling Approach	48
4.2 Map-based Inference	50
4.2.1 Intersection Extraction	51
4.2.2 Street Approach Attributes	51

4.2.3	Knowledge Representation	53
4.2.4	Methodology	54
4.2.5	Domain Knowledge	55
4.3	Crowd-sensing Model	55
4.4	Training and Cross-city Applicability	58
4.5	Experimental Evaluation	59
4.5.1	Data Collection and Datasets	59
4.5.2	Results	60
4.6	Discussion	67
CHAPTER 5 PARTICIPATORY SLOW START		70
5.1	Slow Start Transition	72
5.2	GreenGPS Model Transition?	73
5.3	Model Transition Planning	75
5.3.1	Design Criterion	76
5.3.2	Transition Point Derivation	77
5.4	Data Distribution	79
5.5	Multi-level Model Transition	80
5.6	Experimental Evaluation	81
5.6.1	Approaches	81
5.6.2	Experiment Setup	84
5.6.3	Model Transition Point	85
5.6.4	Service Prediction Accuracy	88
5.6.5	Sensing Data Distribution	90
CHAPTER 6 ARCHITECTURE AND IMPLEMENTATION		92
6.1	GreenGPS System Architecture	92
6.1.1	Data Collection	92
6.1.2	Modeling and Generalization	94
6.1.3	Detection of Traffic Signs Location	94
6.1.4	Navigation	95
6.2	A Participatory Sensing System for Data Collection	95
6.2.1	OBD-II Communication	96
6.2.2	Opportunistic Uploading	98
6.2.3	Collaborative Uploading	99
6.2.4	Energy Management	100
6.2.5	Collected Data	102
6.3	Discussion	104
CHAPTER 7 SERVICE EVALUATION AND IMPACT		108
7.1	GreenGPS Application Products	108
7.1.1	Green Navigation Engine Web-based GUI	108
7.1.2	Green Navigation Android Application	109
7.1.3	Data Collection Android Application	110
7.2	Green Navigation Model Accuracy	110

7.3	Regulator Detection Module Impact	117
7.4	Fuel Savings in Urbana-Champaign	118
CHAPTER 8 CONCLUSIONS AND FUTURE WORK		123
8.1	Thesis Summary	123
8.2	Limitations and Future Directions	125
REFERENCES		128

LIST OF TABLES

3.1	The average error percentage (magnitude) for the individual car models and the generalized case when all the data is used to obtain the model	35
3.2	The average error percentage (magnitude) for the cluster-based model constructed based on the optimal generalization order	40
4.1	Sample MUTCD rules and guidelines on the placement of traffic regulators	49
6.1	The vehicle set used and the amount of data collected	103

LIST OF FIGURES

1.1	U.S. energy consumption by sector, 2013	1
1.2	(a) U.S. greenhouse gas emissions by end-use sectors, 2011; (b) U.S. transportation end-use sector greenhouse gas emissions by source, 2011	2
3.1	The user interface of GreenGPS with the most fuel-efficient route between two points for a member’s vehicle	23
3.2	The real mpg distribution of all cars	25
3.3	The free body diagram of a car	27
3.4	The path error percentage distribution for one car	34
3.5	Average error percentage (magnitude) of the models obtained from various clusters	38
4.1	The coverage map of Google street views: covered areas in dark blue	45
4.2	The update map of OpenStreetMap: more recent updates shaded in red and older imports depicted in green and blue	47
4.3	Intersection street approaches and attributes	52
4.4	Detection of traffic regulatory signs modeling	58
4.5	Detection accuracy in: (a) Urbana, IL; (b) Champaign, IL; (c) Los Angeles, CA; (d) Pittsburgh, PA	62
4.6	Detection accuracy at various confidence levels in: (a) Urbana, IL; (b) Champaign, IL	63
4.7	Cross-city detection accuracy in: (a) training in Urbana, IL and testing in Champaign, IL; (b) training in Champaign, IL and testing in Urbana, IL	65
4.8	Cross-city detection accuracy at various confidence levels in: (a) training in Urbana, IL and testing in Champaign, IL; (b) training in Champaign, IL and testing in Urbana, IL	66
4.9	Cross-city detection accuracy when training in Urbana-Champaign, IL (<i>UC</i>) and testing in Los Angeles, CA (<i>LA</i>) or Pittsburgh, PA (<i>Pitt</i>), and vice versa – notation <i>A-B</i> denotes training in <i>A</i> and testing in <i>B</i>	67

4.10	Cross-city detection accuracy at various confidence levels when training in Urbana-Champaign, IL (<i>UC</i>) and testing in Los Angeles, CA (<i>LA</i>) or Pittsburgh, PA (<i>Pitt</i>), and vice versa – notation <i>A-B</i> denotes training in <i>A</i> and testing in <i>B</i>	68
5.1	Impact of model transition on GreenGPS performance	75
5.2	The typical behavior of fuel consumption for cars and light trucks with respect to speed	80
5.3	The number of samples required for reliable model transition for: (a) model with 12 parameters; (b) model with 4 parameters	86
5.4	The number of samples required for reliable model transition for: (a) trip set with 0.5 mile long trips; (b) trip set with 1 mile long trips; (c) trip set with 1.5 mile long trips; (d) trip set with 2 mile long trips	89
5.5	Impact of the width threshold on transition points for: (a) different model sizes; (b) datasets of varying trip set length	89
5.6	(a) Prediction error for models with different number of parameters; (b) Prediction error for sets of trips with varying length	90
5.7	Impact of participatory sensing data distribution on: (a) reliable model transition point; (b) prediction error	91
6.1	GreenGPS system architecture	93
6.2	(a) OBD-II to bluetooth adaptor; (b) Adaptor deployed in a car	96
6.3	Coverage map for the paths on which data were collected	104
6.4	The distribution of trip data collected from all cars: (a) The path distance distribution; (b) The average speed distribution; (c) The average number of stop signs, traffic lights, left turns and right turns per one-mile road segments with respect to the distance of the trips	105
7.1	Distribution of path error percentage for different prediction models: (a) signed error, (b) unsigned error	112
7.2	Mean path error percentage for different prediction models when path length is varied: using (c) original data, (d) synthetic data	114
7.3	Impact of the amount of training data on different prediction models accuracy	116
7.4	Impact of the regulators detection module on GreenGPS	118
7.5	The landmarks and the corresponding shortest (in red), fastest (in blue), Garmin eco (in purple), and GreenGPS green (in green) routes: (a,b): Toyota Camry 2004; (c,d): Nissan Altima 2006; (e,f): Toyota Corolla 2000.	120

7.6	Average normalized fuel consumption for the various trips between different landmarks	121
7.7	Percentage fuel saved by using GreenGPS green routes, relative to the Fastest, Shortest, and Garmin Eco routes . . .	122

CHAPTER 1

INTRODUCTION

The proliferation of smartphones has led to increased interest in mobile participatory sensing as an important paradigm in today’s data-driven marketplace and for a wide range of areas. This thesis approaches energy section and adopts participatory sensing paradigm to reduce energy consumption in transportation through “*navigating the most fuel-efficient or green routes*”.

According to the U.S. Energy Information Administration (EIA) [1], 28% of the energy consumption of the United States comes from transportation sector (Figure 1.1). In addition, according to the U.S. Environmental Protection Agency (EPA) transportation contributes to 27% of total Greenhouse Gas (GHG) emissions, out of which 83% is caused by passenger cars and trucks (Figure 1.2) [2].

The EPA statistics report that over 200 million light vehicles (passenger cars and light trucks) are on the road in the US and each of them is driven,

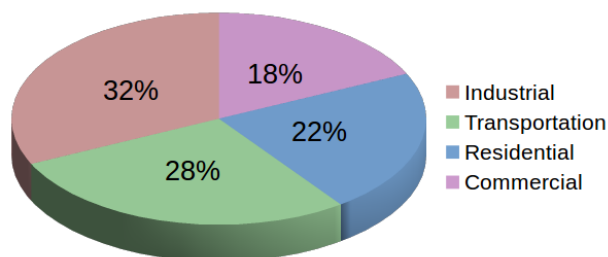


Figure 1.1: U.S. energy consumption by sector, 2013

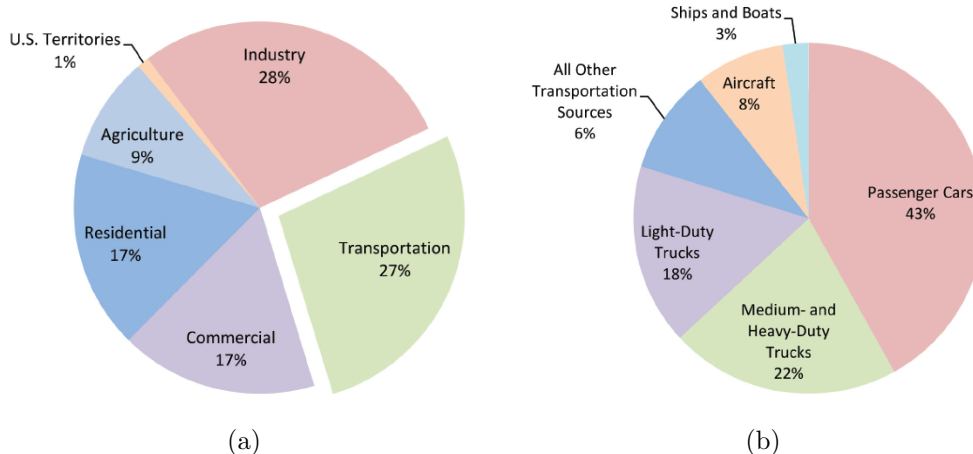


Figure 1.2: (a) U.S. greenhouse gas emissions by end-use sectors, 2011; (b) U.S. transportation end-use sector greenhouse gas emissions by source, 2011

on an average, 12000 miles in a year [3]. With respect to the average mile per gallon (MPG) rating for light vehicles, more than 118 billion gallons of fuel is consumed per year!

This thesis moves toward reducing transportation energy consumption by navigating the most fuel-efficient or green routes. This is as opposed to the traditional navigation tools’ trend which provide either the shortest or fastest routes, such as Google maps [4] and MapQuest [5]. The thesis objective is followed and performed in the context and by the aid of mobile participatory sensing which accommodates efficient modeling of the system, leading to inexpensive construction of the over-arching product of the thesis, called “*GreenGPS*”. Mobile participatory sensing relies on data collected from voluntarily deployed mobile devices that are tasked to gather and share local data in order to enable services in communities’ common interests.

GreenGPS collects and derives necessary information to compute and answer queries on the most fuel-efficient route. The most fuel-efficient route between two points may be different from the shortest and fastest routes. For example, a fastest route that uses a freeway may consume more fuel than the

most fuel-efficient route because fuel consumption increases non-linearly with speed or because it is longer. Similarly, the shortest route that traverses busy city streets may be suboptimal because of downtown traffic.

GreenGPS supports two types of users; members and non-members. Members are those who contribute required participatory data to the GreenGPS repository and register their vehicles used for data collection. Hence, GreenGPS can compute the most fuel-efficient route specifically for the registered vehicle. Non-members can use GreenGPS to query for fuel-efficient routes as well. They may enter their vehicle's brand. Since different vehicles have different fuel consumption characteristics, the car details are used to compute the most fuel-efficient route for the given vehicle type. GreenGPS answers such queries based on the average estimated performance for their vehicle's attributes. The advantage for the users who contribute data is that the system provides better estimates of the most fuel-efficient routes to these individuals, thus allowing them to have higher savings.

The motivation for GreenGPS does not need elaboration. GreenGPS users might be driven by benefits such as savings on fuel or positive impacts on the environment by reducing motor emissions such as CO_x and NO_x air poisoning gases. Further, GreenGPS can be offered to the users at a very low cost. These factors will incentivize the users to adopt it in a large scale; this being the main hurdle in getting a participatory sensing application off the ground.

To estimate the amount of energy savings that can be achieved by GreenGPS on a global scale, we provide approximate calculations based on data from EPA. A study of GreenGPS (Chapter 7) reports, on average, over 16% fuel savings on selected routes, compared to the fastest and shortest alternative routes. Even if 10% of the vehicles adopted GreenGPS and 16% fuel

savings were achieved on only 30% of the routes traveled by each of these vehicles, the amount of overall fuel savings is over 567 million gallons of fuel per year $((12000 / 20.3) \times (0.10 \times 200M) \times 0.16 \times 0.30)$. This translates into over 1.8 billion dollars in savings at the pump (based on the current national average pump prices for a gallon of gasoline [6]). We consider the above prospective savings acceptable.

1.1 Thesis Statement

Green navigation demands for search of green routes whose fuel-efficiency is customized to individual drivers and vehicles. Initially it is impractical to assume that members of the navigation service will measure all city streets and cover all vehicle types. Instead, measurements of members can be used to calibrate generalized fuel-efficiency *prediction models*. The fuel consumption on an arbitrary street is affected by different types of parameters and it is the mathematical model describing the relation between these general parameters and fuel-efficiency that gets estimated from participatory data. Hence, the following major challenges need to be addressed. First, developing a fuel-efficiency prediction model the parameters of which can be easily measured and utilized in navigation of green routes. Second, building a general but personalizable fuel-saving navigation system using the sparse data collected by the participatory sensing system. Third, modeling the effect of dynamically changing traffic and spatio-temporal parameters contributing to the model. Fourth, automatically inferring the location and type of traffic regulators involved in navigation and fuel-efficiency prediction. Finally, addressing lack of reliability brought about by lack of sensing data during *slow start* deployment stage where service adoption is sparse and data is limited

in quantity and distribution.

1.2 Thesis Contributions

In order to navigate fuel optimal routes, fuel efficiency predictive models are derived such that the contributing parameters can either be measured by our participatory sensing system or be automatically inferred for the sake of route computation. The elaborate model designed spans variety of parameter types such as vehicle parameters, static and dynamic road parameters, and route parameters. Among the involved parameters is information on the location of traffic regulatory signs which is not available in public databases and needs to be inferred. We design an efficient and widely applicable inference model to provision such information to GreenGPS.

Dealing with data sparsity in GreenGPS application, an effective model generalization framework is designed to address the sparsity issue in the high dimensional feature space, enabling model extrapolation beyond the current data coverage. However, the elaborate predictive models themselves may be unreliable during early deployment stage in which service adoption is sparse. The reliability of the service is guaranteed by employing multiple layers of modeling and theoretically planning transition to the next level.

The main contributions of the thesis can thus be enumerated as follows.

1. **Generalized Modeling of Sparse High-dimensional Data:** Participatory sensing services based on mobile devices constitute an important growing area of mobile computing. Most services start small and hence are initially sparsely deployed. Unless a mobile service adds value while sparsely deployed, it may not survive conditions of sparse deployment. We offer a generic solution to this problem. Specifically,

when the participatory sensing service is sparsely deployed, we demonstrate a general framework for generalization from sparse collected data to produce models extending beyond the current data coverage. This generalization allows the mobile service to offer value under broader conditions. Namely, it enables GreenGPS to extrapolate from fuel-efficiency data of members' vehicles on some streets to the fuel consumption of arbitrary vehicles on arbitrary streets.

2. **Detection of Traffic Regulators:** Intelligent transportation systems serve as important technologies to improve traffic safety, mobility, cost and environmental sustainability. Towards that end, a variety of applications and driver advisory tools have been developed. One such application is GreenGPS, modeled and developed in this thesis. To work efficiently, many such applications require knowledge of not only street maps but also elements affecting traffic flow. The most obvious elements are traffic lights and stop signs, which we shall henceforth call traffic regulators. Unfortunately, information on traffic regulators is not widely available in public databases such as *OpenStreetMap (OSM)*. We offer a combination of map-based modeling and crowd-sourcing to predict regulator type and locations. The modeling component *reverse engineers* rules for placement of traffic regulators, allowing it to predict their locations and type based on map information. Where available, crowd-sourced vehicular GPS traces are incorporated into the prediction function to improve the results. The approach is evaluated across multiple cities and is shown to outperform both crowd-sourcing alone and map-based modeling alone. It achieves a prediction accuracy level above 97% in detecting the existence and determining the type of traffic

regulators in the cities considered.

3. **Participatory Slow Start:** In participatory sensing applications a service is provided based on data collected by participants. The “*Slow Start Problem*” refers to the initial stage in participatory sensing service deployment, during which service adoption remains sparse and, hence, the collected data does not offer adequate coverage. Predictive models, learned from data, offer a way to generalize from sparse observations, but the models themselves need to be statistically reliable to offer a reliable service. To achieve service reliability, we offer a modeling approach, where simpler models are used initially, gradually transitioning to more elaborate models, when enough data is collected. A key challenge is to time model transitions correctly to provide theoretical guarantees on modeling error. Our technique takes a holistic approach in bounding modeling error as opposed to prior statistical approaches that bound the error of a single model component at a time. We show that our approach significantly reduces prediction error in the initial stages of deployment.
4. **Participatory Sensing Platform:** We developed a participatory sensing platform which can be used for collecting participatory data. The participatory sensing architecture consists of data collection and sharing components on the frontend and data processing, aggregation and modeling on the backend. We deployed the client side module on 46 individual user subjects in order to collect sensing data for the purpose of evaluating our green navigation system. We have started to deploy the system on over 100 vehicles of UIUC Facilities and Services to provide a platform for future studies.

5. **Application Products:** We developed the following system and application components for our green navigation service which can be used by the public.
- (a) Green Navigation Engine: The core navigation engine that finds the most fuel-efficient routes.
 - (b) Web-based Graphical User Interface: A web-based graphical user interface that presents the computed green routes to the users.
 - (c) Data Collection Android Application: An Android application that can be used by GreenGPS members who contribute their fuel consumption data to our system. The application automatically collects users' fuel related data and opportunistically uploads them to our backend server.
 - (d) Green Navigation Android Application: An Android application that can be used by GreenGPS users (both members and non-members) and navigates the most fuel-efficient routes. The computed routes along with voice directions are presented to the user for the sake of driving.

1.3 Thesis Outline

This thesis is organized as follows. Chapter 2 reviews relevant works from the literature. Chapter 3 presents modeling foundation of the green navigation service. Chapter 4 presents detection modeling of traffic regulators' location. Chapter 5 visits participatory slow start problem and presents an effective modeling approach for this stage. Chapter 6 presents the architecture and implementation of the fuel-efficient navigation system. Chapter 7 evaluates

performance of the service and its impact compared to a commercial product. Chapter 8 presents future directions to extend the work and concludes the thesis.

CHAPTER 2

RELATED WORK

Prior work in participatory sensing, transportation fuel saving and emission reduction, traffic regulator detection, and sample size planning are relevant to this thesis work and are reviewed in this chapter.

2.1 Participatory Sensing

The GreenGPS navigation service is an example of participatory sensing services, that have recently become popular in networked sensing. The concept of participatory sensing was introduced in [7]. In participatory sensing, individuals are tasked with data collection which is then shared for a common purpose. A broad overview of such applications is provided in [8], [9], [10]. Several such applications include CenWits [11], a participatory sensing network to search and rescue hikers, CarTel [12], a vehicular sensor network for traffic monitoring, CabSense [13], a cabs sensor network to find best corners to catch taxis depending on the current location and time, BikeNet [14], a bikers sensor network for monitoring popular cyclist routes, ImageScape [15], a cellphone camera network for sharing diet related images, Micro-Blog [16], a content-sharing platform, PEIR [17], a report system enabling individuals to measure and compare their impact on environment as well as their exposure to environmental emissions, Escort [18], an electronic escort system that enables localizing people, MoVi [19], a service for covering social events, [20],

a service to determine buses arrival time, and IndoorCrowd2D [21], a crowdsourcing system to reconstruct the building interior views at large scale. Our application, GreenGPS, introduces a novel example of this genre that enables individuals to compute fuel efficient routes customized for their vehicles.

2.2 Transportation Fuel Saving

There exist a body of work addressing transportation fuel consumption factors to achieve savings in the field.

2.2.1 Routing and Driving Pattern

A comprehensive study that provides optimal route choices for lowest fuel consumption is presented in [22]. In the paper, fuel consumption measurements are made through the extensive deployment of sensing devices (different from the OBD-II) in experimental cars. These fuel consumption measurements are then used to compute the lowest fuel consumption route. In contrast, GreenGPS service uses a sparse deployment to build mathematical models for predicting fuel consumption for other streets and cars.

In [23], the influence of driving patterns of a community on the exhaust emissions and fuel consumption were studied. Feedback was provided to the community regarding the driving patterns to cut back on the fuel consumption and exhaust. A driver support tool, FEST, was developed in [24]. FEST uses sensors installed in the car along with a software to determine the driving behavior of the driver and provide real-time feedback to the individual. An extension to FEST that includes more experiments and further evaluation can be found in [25]. In [26] a driving assistance system is proposed that

provides guidelines to drivers considering the current situation and vehicle specific characteristics to help them save fuel. [27] investigates the driving factors that have the main impact on fuel-economy and optimizes driving styles with respect to those factors in order to provide feedback to drivers.

UbiGreen [28] is a mobile tool that tracks an individual’s personal transportation and provides feedback regarding their CO₂ emissions. [29] proposes to exploit information on surrounding vehicles and road conditions in designing eco-driving systems to achieve higher fuel-saving. The authors in [30] study the effects of adopting eco-routing on fuel and emissions reduction in the scale of metropolitan networks and demonstrate the significant dependence of the potential benefits on the transportation networks configuration. [31] provides a comparative study on the effectiveness of different initiatives on modifying driving behavior. Social drive [32] is a crowdsourcing service that provides feedback to drivers regarding their driving behavior and enables them to share their real-time trip information through social networks, stimulating users to reduce their gas consumption. CarMA [33] provides high-level abstractions for sensing and tuning automobile engine parameters to achieve fuel efficiency. The tuning can be done at the granularity of individual trips.

2.2.2 Speed Adjustment and Traffic Signals

SignalGuru [34], a participatory sensing based system, is a traffic signal schedule advisory application that assists drivers to adjust speed and avoid coming to a complete stop. A feedback control algorithm was developed in [35] that determines speed of automobiles on highways with varying terrain to achieve minimal fuel consumption. An extension to the work in [35] was developed in [36]. In [36], suggestions of driving style were made for

varying road and trip types (e.g. constant grade road, hilly road). The problem was formulated using a control theoretic approach. The authors in [37] investigate the fuel and CO₂ saving that can be achieved by following recommendations of eco-approach technology on drivers' speed. [38] performs a driving optimization that compromises between minimization of fuel consumption and maintaining the recommended speed.

In [39] a simulation study is conducted to investigate the impact of traffic signals placement policies on fuel consumption and emissions. [40] analyzes optimal timing of traffic signals to minimize fuel-consumption. [41] proposes to use RFID-based e-stop signs at unsignalized intersections to alter drivers behavior properly early and achieve potential emissions reduction and fuel-economy improvement. [42] provides a comparative study on time and fuel efficiency of green light optimal speed advisory systems (GLOSA). The authors in [43] propose a mechanism based on communication between traffic light signals and approaching vehicles in which a fuel-optimal speed is computed and sent to the vehicles to reduce fuel consumption.

2.2.3 Fleet Management, Real-time Traffic, Misc.

[44] develops fleet management strategies for the purpose of reducing fuel consumption and gas emissions. In [45] the impact of the sampling frequency of the inertial variables on the estimation of vehicles' fuel consumption is studied. coRide [46], among others, proposes the use of carpooling to share rides and reduce gas consumption. coRide service adopts a fare model that incentivizes both drivers and passengers to participate.

There exist a large category of works, such as VTrack [47], that collect real-time traffic information and provide estimations on road travel times in order

to aid users route around traffic congestion, being a major cause of excess fuel consumption. VTrack utilizes WiFi and GPS sensors of smartphones to perform localization in an energy-aware fashion. Kyun [48] develops a networked sensor based real-time traffic queue monitoring system for developing countries, which can lead to improved automatic traffic signals scheduling in order to reduce fuel inefficiency. An energy-optimization navigation system utilizing real-time traffic information is proposed in [49] for hybrid electric vehicles.

Some other works like PhonePark [50] approach reduction of vehicles gas consumption by detection of available on-street parking spaces which enables users to minimize their travel distance searching for parking. PhonePark uses the GPS and accelerometer sensors of travelers mobile phones.

In a separate study [51], it was shown that rising obesity has a significant impact on the total fuel consumption in the US. Models were developed that studied the impact of obesity on the amount of fuel consumed in passenger vehicles.

In contrast, GreenGPS represents a participatory sensing service that aims at improving fuel consumption and reducing gas emissions through green routing. Using a sparse deployment and the sparse data collected from volunteer participants, models are built and continuously updated that enable vehicle customized navigation on the minimum-fuel route for both members and non-members of the service.

2.3 Traffic Regulator Detection

The topic of traffic regulator detection received much attention in prior literature.

2.3.1 Image Processing

There exists a significant body of image-processing approaches that focus on recognizing various traffic regulatory signs. For a survey, please refer to [52]. These approaches nicely complement our methodology for detection and recognition of traffic regulators. Clearly, cameras offer more reliable information, but are lower in coverage than OpenStreetMap. Hence, we can leverage cameras in detection when available and fall back on our proposed solution when not.

2.3.2 Vehicular GPS Traces

SmartRoad [53] and similar efforts [54, 55] utilize vehicular GPS traces to detect traffic regulators. For example, the authors in [55] formulate the problem in a setting where very simple features are extracted from GPS traces of in-vehicle smartphones. Phones then make an educated guess and send a binary “claim” to a server that signals the existence of a stop sign, or a traffic light. The claim is treated as an unreliable hint. A maximum likelihood estimation technique is used to determine which claims are more likely to be true. This category of prior work depends on the availability of extensive driving traces across each intersection in order to detect the regulator type.

In contrast, the solution proposed in this thesis for the detection and recognition of traffic regulators builds on a map-based inference core, enabling it to

predict the existence and type of traffic regulators even at those intersections from which no GPS traces are available.

2.3.3 Traffic Signals Schedule

Another category of related work concentrates on finding the exact schedule of traffic signals to help drivers improve various performance metrics such as total trip time and fuel efficiency by planning their traversal schedules through signalized intersections. For example, CityDrive [56] utilizes phone sensors and GPS in vehicles to build a topology of road intersections and infer the schedule of traffic signals. The resulting information is then used to suggest to drivers the most appropriate driving speed, while approaching traffic lights. In TLCorA [57, 58] vehicles share their traces with a cloud. The traces are then used to draw correlations among traffic signals, which are intentionally introduced so that efficient driving patterns are enabled. Signalguru [59] is a participatory sensing system that leverages windshield-mounted phone cameras to opportunistically sense and detect traffic signals. The results are then shared with nearby vehicles to collectively derive the schedule of the signals in order to predict their future timing.

2.3.4 Road Events

In a different vein, there has been interest in extracting related road network semantics or events. Among such efforts is Dejavu [60], which performs outdoor localization based on a dead-reckoning technique. It uses an array of phone inertial sensors (compass, gyroscope, and accelerometer) to create unique signatures of landmarks that are then used to reset the error accumulation in the dead-reckoning displacement. Dejavu is later used in

Map++ [61] to provide low-power location information. Map++ utilizes crowd-sourced phone sensor traces from pedestrians and vehicle passengers to detect landmarks such as tunnels, bridges, stop signs and traffic lights. CARLOC [62] utilizes crowd-sourced landmark estimates and built-in vehicle sensors to improve position estimate of vehicles. Nericell [63] uses accelerometer, microphone, GSM radio, and/or GPS sensors in mobile smartphones to monitor road and traffic conditions and detect potholes, bumps, braking, and honking in the complex settings of developing countries, where traffic is not smooth.

In comparison with that literature, the detection module proposed in this thesis leverages broadly available map data to detect and identify traffic regulators. Our approach exploits vehicular GPS traces, where available, but is not dependent on their availability to perform the detection task.

2.4 Sample Size Planning

The literature on statistical sample size planning is relevant to our model transition component. In behavioral and social sciences the conclusions from a conducted experiment or study may not be valid with an inadequate sample size. On the other hand, an excessive sample size will waste the resources. Hence the sample size needs to be carefully determined.

The required sample size has long been obtained from a power analytic perspective, e.g. [64], [65], [66], [67], [68]. In a different vein, a category of works applicable to educational, behavioral and social sciences mainly concentrate on accuracy in parameter estimation, when planning the sample size, e.g. [69], [70], [71], [72]. The latter category is more relevant to our model transition planning. The work in [69] derives sample size for multiple regres-

sion through bounding the likely confidence interval widths. The authors in [70] extend the work in [69] and address unstandardized regression coefficients as well. The authors in [71] demonstrate how a Monte Carlo study can be used to calculate an adequate sample size. The work [72] explains the use of a Monte Carlo study in sample size planning using the statistical programming language of R.

In contrast to these quantitative approaches, qualitative research has taken an approach based on the concept of theoretical saturation and the sample size is considered adequate when further samples do not yield additional insight into the problem being addressed, e.g. [73], [74], [75], [76], [77].

In participatory sensing paradigm that we are dealing with in this thesis, data is collected in a passive manner and application designers often do not have fine-grained control over where (in the high-dimensional feature space) the data samples are collected. This is in contrast to what is dealt with in literature on experiment design (e.g. [78], [79]) and active learning and adaptive sampling (e.g. [80], [81], [82]), in which the optimal location of data samples is specified in an offline or online fashion, respectively. Hence we adapt the model transition planning to this setting and following the so far collected data distribution determine the transition point for each multi-dimensional feature subspace.

CHAPTER 3

GREEN NAVIGATION MODELING

Mobile participatory sensing relies on user devices that are on the move to obtain sensing coverage of large areas for purposes of interest to the mobile service [7], [8], [9], [10]. Examples include mapping of physical phenomena or computing community statistics of interest [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21]. An inherent challenge in such a service is therefore to handle conditions of sparse deployment, where coverage is small. Clearly, a mobile participatory sensing service must offer value to customers even when sparsely deployed. Otherwise, it may not survive to see a larger deployment. The fundamental way to improve value under conditions of sparse deployment is to develop models for generalization from sparse data. This chapter describes a general approach for such generalization and applies it to the specific context of GreenGPS, a novel navigation service that finds the most *fuel-efficient* (hence, *green*) routes for drivers [83], as opposed to the traditional shortest or fastest routes, offered by such services as Google maps [4] and MapQuest [5]. GreenGPS collects the necessary information to compute and answer queries on the most fuel-efficient route. We show that we are successful at generalizing from sparse data and are able to offer value (i.e., fuel savings) in conditions of sparse deployment.

A GreenGPS client is offered as an Android application that can be installed on participants' smartphones. The application collects data parameters involved in engine fuel consumption, vehicle speed and location. Fuel

consumption parameters are provided by the *On-Board Diagnostic* (OBD-II) interface of the vehicles, standardized in all vehicles sold in the United States since 1996. The OBD-II is a diagnostic system that monitors the health of the automobile using sensors that measure approximately 100 different engine parameters. Other examples of monitored measurements include engine RPM, coolant temperature, vehicle speed, and engine idle time. A comprehensive list of measured parameters can be obtained from standard specifications as well as manufacturers of OBD-II scanners.

There exist several commercial OBD-II scanner tools [84], [85], [86], [87], that can read and record the sensor values. Apart from such scanners, remote diagnostic systems such as GM's OnStar, BMW's ConnectedDrive, and Lexus Link are capable of monitoring the car's engine parameters from a remote location (e.g. home of driver of the car). With respect to the increase in the use of bluetooth devices (e.g., cell-phones), GreenGPS utilizes a typical OBD-II to bluetooth adaptor in conjunction with its participatory data collection framework. This enables GreenGPS to be offered at a very low price. For example, in our deployment we use ELM327 OBD-II bluetooth wireless transceiver dongle [88] which is available for less than \$10 at the time of writing. The fuel consumption data, read via the adaptor, are wirelessly transmitted to the user-side hub of sensing, the phone application, upon request. The application combines the OBD-II data with other sensory data and opportunistically uploads them to an aggregation and modeling backend upon availability of WiFi Internet connectivity.

The general challenge in participatory sensing applications addressed in this chapter is the sparsity of their high dimensional data space. To address the data sparsity challenge, GreenGPS exploits prediction models that enable it to extrapolate from fuel-efficiency data of some vehicles on some streets to

the fuel consumption of arbitrary vehicles on arbitrary streets. The developed generalization methodology employed by GreenGPS can be adopted by a variety of other participatory sensing applications as well, where data follows discoverable models. The constructed prediction models in GreenGPS abstract vehicles and routes by a set of parameters such that fuel efficiency can be computed simply by plugging in the parameters of the right car and route.

Thanks to its generalization methodology, GreenGPS offers value even when sparsely deployed. Sparse deployment, here, refers to the deployment of the GreenGPS application, not deployment of OBD-II measurement devices (as those are abundant in modern cars). One specific instance of generalization in GreenGPS in the sparse deployment scenario is to support two types of users; members and non-members. Members are those who contribute their data to the GreenGPS repository from the OBD-II sensors described above. They have GreenGPS accounts and benefit from more accurate estimates of route fuel-efficiency, customized to the performance of their individual vehicles. Non-members can use GreenGPS to query for fuel-efficient routes as well. Since GreenGPS does not have measurements from their specific vehicles, it answers queries based on the average estimated performance for their vehicle's attributes such as make, model, year and class (or some subset thereof, as available). GreenGPS also allows members to get navigation advice on routes they had never driven before using models developed from data collected on other routes.

In summary, this chapter demonstrates how sparse samples of high-dimensional spaces can be generalized to develop models of complex nonlinear phenomena, where one size (i.e., model) does not fit all. The rest of the chapter is structured as follows. Section 3.1 presents an overview of our green

navigation service. Section 3.2 and Section 3.3 elaborate on fuel consumption modeling and model generalization, respectively. Section 3.4 presents how the impact of dynamic traffic on fuel consumption is modeled.

3.1 GreenGPS System

The service provided by GreenGPS is similar to a regular map application, such as Google maps [4] or MapQuest [5]. Google maps and MapQuest provide the shortest or fastest routes between two points, whereas GreenGPS computes the most fuel-efficient route. A snapshot of the Web-based GreenGPS's user interface is shown in Figure 3.1 along with the most fuel efficient route between two points for a member vehicle.

Individuals who want to compute the most fuel-efficient route between two points enter the source and destination address via the interface provided by GreenGPS. Members of GreenGPS (i.e., those individuals who contributed participatory data) can register their vehicles that were used for data collection. Hence, GreenGPS can compute the route specifically for the registered vehicle. Other users may enter their vehicle's make, model, and year of manufacture. Since different vehicles have different fuel consumption characteristics, these car details are used to compute the most fuel-efficient route for the given vehicle brand.

It is impractical to assume that GreenGPS members will measure all city streets and cover all vehicle types. Instead, measurements of GreenGPS members are used to calibrate generalized fuel-efficiency *prediction models*. These models, discussed in Section 3.3, show that the fuel consumption on an arbitrary street can be predicted accurately from a set of *static* street parameters (e.g., the number of traffic lights, the number of stop signs, and

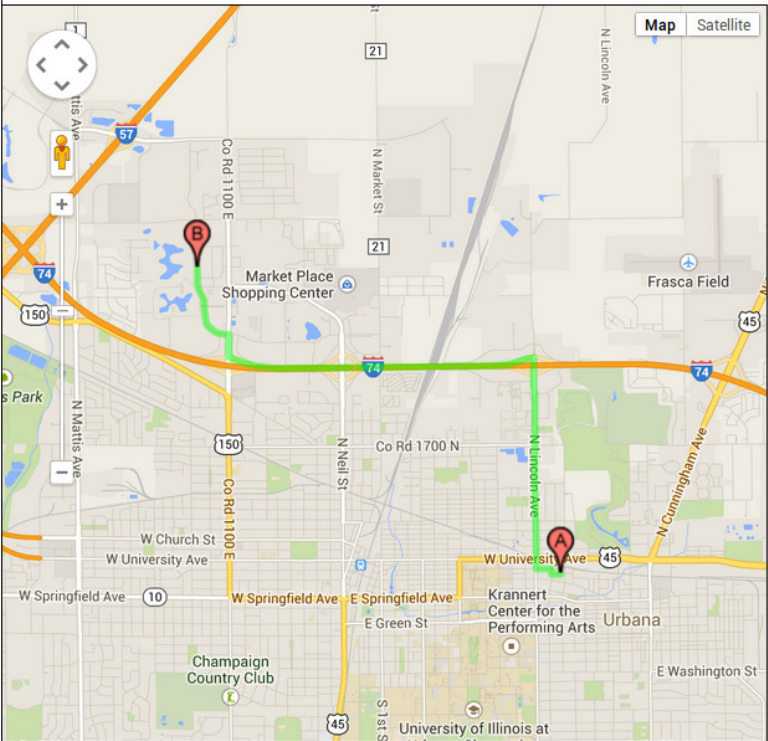
Start Address: Street City State

End Address: Street City State

Member
 IMEI

Non-Member
 Make: Model: Year: Class:

Not a GreenGPS member? [Register here](#)
 Would like to see your detailed fuel consumption info? [Login here](#)



- Shortest**
 Distance (Mi): 4.1
 Time (Min): 13
 Fuel (Gal): 0.16
 MPG: 25.1
- Fastest**
 Distance (Mi): 6.2
 Time (Min): 9
 Fuel (Gal): 0.20
 MPG: 31.6
- Green**
 Distance (Mi): 4.4
 Time (Min): 10
 Fuel (Gal): 0.14
 MPG: 30.6

Figure 3.1: The user interface of GreenGPS with the most fuel-efficient route between two points for a member's vehicle

the slope of the roads) and a set of *dynamic* street parameters (such as the average speed on the street or the average congestion level), plus the route parameters (such as the number of left turns and right turns), the vehicle parameters (e.g., weight and frontal area) and the driving behavior (e.g., making high acceleration or hard breaking). It is the mathematical model describing the relation between these general parameters and fuel-efficiency that gets estimated from participant data. Hence, the larger and more diverse is the set of participants, the better the generalized model.

For most streets, static street parameters can be obtained from traffic databases. (We show in Chapter 4 how to estimate static parameters not in databases, such as locations of traffic lights and stop signs.) Dynamically changing parameters such as the congestion levels or average speed should be obtained as well. In larger cities, real-time traffic monitoring services can supply these parameters [89], [90], [4]. Many GPS device vendors, such as Garmin and TomTom, also collect and provide congestion information. In GreenGPS, speed information can be obtained from collected data using our participatory sensing infrastructure described in Section 6.2.

Finally, note that the increasing availability of vehicular fuel efficiency measurements to drivers in modern vehicles is not a substitute for green navigation. In order to exploit fuel efficiency measurements, a driver who wants to find a most fuel-efficient route to a given destination would have to drive on all the different alternative routes to that destination multiple times and note the average fuel consumption over a statistically significant number of trips on each route, then decide (for future reference) which route was better. In contrast, our service computes the answer automatically from a model trained using other trips on other routes that the driver already drove. This highlights the benefits of our generalization models over present

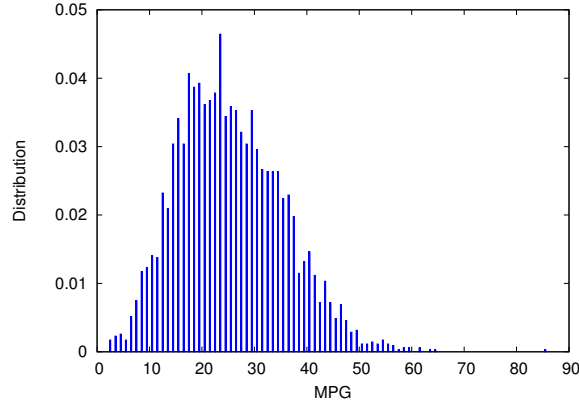


Figure 3.2: The real mpg distribution of all cars

affordances of modern cars.

3.2 Derivation of Model Structure

The first part of data generalization is to derive a model structure. In this section, we derive the fuel consumption model structure.

To motivate the need for modeling, we plot the distribution of miles per gallon (mpg) for all the data collected in Figure 3.2. We observe from this figure that the distribution spans a wide range of values between 2 and over 60. The standard deviation of the mpg distribution is 9.4 miles per gallon, which is pretty high. Hence, an appropriate model is needed to estimate the fuel consumption on various segments.

The difference from the models in the literature [91], [92], [93] lies in that we are interested in developing a model whose parameters can be easily measured by our participatory sensing system and later utilized in the route navigation phase. This imposes restrictions on what parameters can be used which makes it different from developing first-principle models whose goal is simply to understand the physics.

Several factors affect the fuel consumption on streets. We classify these

parameters into five categories, which are (i) *static street parameters*, (ii) *dynamic street parameters*, (iii) *route parameters*, (iv) *car specific parameters*, and (v) *personal parameters*. Static street parameters model the street characteristics and do not change (or change with a very high time constant) over a period of time. For example, the speed limits of streets change much less frequently and the number of traffic lights on the street (in a given stretch) remain more or less constant. The dynamic street parameters are characteristics that change with time, for example, the congestion levels on a street or the average speed on a street. The static and dynamic street parameters together determine the fuel efficiency of a particular street. The fuel usage is also affected by the number of left turns and right turns through the route. Hence, route parameters are parameters that depend on the shape of the overall route (such as turns), as opposed to the individual street segments. Other variations in the fuel consumption can occur due to the type of car being driven and the nature of the person's driving. For example, a big SUV may consume more fuel than a small sedan or a person who is aggressive (making higher acceleration or hard braking) is likely to consume more fuel than a sluggish driver. These parameters account for the variation in fuel consumption due to the route parameters, the car type and the driver behavior.

The inputs to our prediction model include street segment parameters, route parameters, and car parameters. We do not consider driver factors in the model; those can be explored in future work. Note that, we are interested in predicting long-term fuel consumption only. While actual savings of a user on individual commutes to work may vary, the user might be more concerned with their net long-term savings. Hence, it is important to capture only the statistical averages of fuel consumption. As long as the errors have near zero

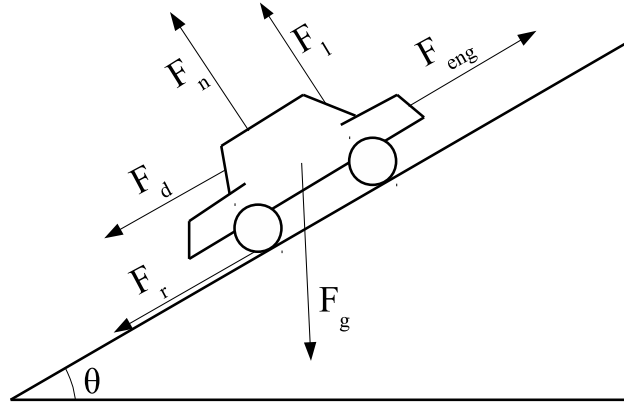


Figure 3.3: The free body diagram of a car

mean, the savings are accurate in the long term. As a given user drives more segments, a value of interest is the end-to-end prediction error that results, which improves over time and represents how far we are off in our estimate of total fuel consumption.

The free body diagram of a car is given in Figure 3.3. Assuming that the car is on an upslope, the final force acting on the car is given by the following equation:

$$F_{car} = F_{eng} - F_d - F_r - F_{g_x} \quad (3.1)$$

where F_{eng} is the engine force, F_d is the air resistance force (drag), F_r is the rolling resistance force, and F_{g_x} is the gravitational force acting on the car. These forces will be elaborated on in the following.

Assuming that the engine RPM is ω , the torque generated by the engine is $\tau(\omega)$, the k -th gear ratio is r_{gk} , the differential ratio is r_d , the transmission efficiency is e_t and the radius of the tire is r , then the engine force F_{eng} is given by the following equation:

$$F_{eng} = \frac{\tau(\omega) \cdot r_{gk} \cdot r_d \cdot e_t}{r} \quad (3.2)$$

The force due to air resistance, F_d , is given by the following equation:

$$F_d = \frac{1}{2} \cdot \rho \cdot c_d \cdot A \cdot v^2 \quad (3.3)$$

In the above equation, ρ is the air density, c_d is the drag coefficient, A is the frontal area of the car, and v is the instantaneous speed of the car. The drag coefficient quantifies the resistance in a fluid environment (air). For example, for a streamlined body the coefficient is about 0.05, for a regular sedan is about 0.4-0.5, and for a truck could be about 1.

The rolling resistance force F_r is characterized by the instantaneous speed of the car, the normal force, and the corresponding coefficients as:

$$F_r = c_{r1} \cdot v + c_{r2} \cdot F_n \quad (3.4)$$

in which F_n is the normal force given by:

$$F_n = F_{g_y} - F_l \quad (3.5)$$

wherein F_{g_y} is the gravitational force acting on the car and F_l is the lift force.

The F_{g_y} is given as follows:

$$F_{g_y} = m \cdot g \cdot \cos(\theta) \quad (3.6)$$

where m is the mass of the car, g is the gravitational acceleration, and θ is the slope of the road. The F_l is given as follows:

$$F_l = \frac{1}{2} \cdot \rho \cdot c_l \cdot A \cdot v^2 \quad (3.7)$$

The gravitational force due to the slope, F_{g_x} , is given by the following equation:

$$F_{g_x} = m \cdot g \cdot \sin(\theta) \quad (3.8)$$

In order to obtain a relation between the fuel consumed and the above forces, we note that the fuel consumed is related to the power generated by the engine at any instance of time t . If f_r is the fuel rate (fuel consumption at a given time instance) and P is the instantaneous power, then $f_r \propto P$. Power is related to the torque function and engine RPM as follows:

$$P = 2 \cdot \pi \cdot \omega \cdot \tau(\omega) \quad (3.9)$$

Hence, we obtain,

$$f_r = \beta \cdot \omega \cdot \tau(\omega) \quad (3.10)$$

In the above equation, β is a constant. Further, we also have the following relationship from rotational dynamics:

$$v = r \cdot \frac{\omega}{r_{gk} \cdot r_d} \quad (3.11)$$

Substituting for ω in Equation 3.10 from Equation 3.11 and for $\tau(\omega)$ in Equation 3.2 from Equation 3.10, F_{eng} can be written as:

$$F_{eng} = \frac{e_t f_r}{\beta v} \quad (3.12)$$

Subsequently, substituting Equation 3.12 and Equations 3.3 to 3.8 in Equation 3.1 gives the following:

$$\begin{aligned}
F_{car} &= ma \\
&= \frac{e_t f_r}{\beta v} - \frac{1}{2} \rho c_d A v^2 - c_{r1} v - c_{r2} m g \cos(\theta) \\
&\quad + \frac{1}{2} c_{r2} \rho c_l A v^2 - m g \sin(\theta)
\end{aligned} \tag{3.13}$$

where a is the instantaneous acceleration of the car.

From the above equation, we obtain the fuel consumption rate as a function of the forces acting on the car shown below:

$$\begin{aligned}
f_r &= k_0 m a v + k_1 c_d A v^3 + k_2 v^2 + k_3 m v \cos(\theta) \\
&\quad + k_4 A v^3 + k_5 m v \sin(\theta)
\end{aligned} \tag{3.14}$$

wherein k_0, \dots, k_5 are constant coefficients.

In order to further derive a model that can be used for regression analysis, we will detail the various components that are part of the fuel consumption of a car. As shown in the above equation, a moving car at a constant speed on a straight road which does not encounter any stop signs, traffic lights or turns will only need to overcome the frictional forces caused by the air, the road, and gravity. These are represented by $k_1 c_d A v^3$, $k_2 v^2 + k_3 m v \cos(\theta) + k_4 A v^3$, and $k_5 m v \sin(\theta)$, respectively. On the other hand, the first component $k_0 m a v$ can be broken down further into two components, one is the extra fuel rate due to congestion, and the second one is the extra fuel rate due to encountering stop signs (ST), traffic lights (TL), left turns (LT) and right

turns (RT). Hence, the previous equation now becomes the following:

$$\begin{aligned}
f_r &= k_1 c_d A v^3 + k_2 v^2 + k_3 m v \cos(\theta) \\
&+ k_4 A v^3 + k_5 m v \sin(\theta) + k_{00} m a v \\
&+ (k_{01} + k_{02} m a v) (\nu'_1 n_{ST} + \nu'_2 n_{TL} + \nu'_3 n_{LT} + \nu'_4 n_{RT})
\end{aligned} \tag{3.15}$$

where ν'_1 , ν'_2 , ν'_3 and ν'_4 are constant coefficients, n_{ST} , n_{TL} , n_{LT} and n_{RT} are the number of stop signs, traffic lights, left turns and right turns, respectively. In the above equation, the last component represents the fuel rate during the idle time and consequent acceleration when encountering traffic signals, stops and turns.

Finally, we can obtain the equation for the consumed fuel, f_c , by integrating the rate of the fuel consumption with respect to time:

$$f_c = \int_{t_{ini}}^{t_{fin}} f_r(t) dt \tag{3.16}$$

in which t_{ini} denotes the time a new trip is initiated, t_{fin} denotes the time the trip is finished.

If we assume the road gradient θ remains constant, for each road segment i replace v with \bar{v}_i , the segment average speed, and consider $a = dv/dt$, we can further simplify the above integral to the following equation for the purpose of regression analysis:

$$\begin{aligned}
f_c = & k_1 c_d A \sum_{i=1}^n \bar{v}_i^2 \Delta L_i + k_2 \sum_{i=1}^n \bar{v}_i \Delta L_i + k_3 m L \cos(\theta) \\
& + k_4 A \sum_{i=1}^n \bar{v}_i^2 \Delta L_i + k_5 m L \sin(\theta) + k_6 m (v_{fin}^2 - v_{ini}^2) \\
& + k_7 (\nu_1 n_{ST} + \nu_2 n_{TL} + \nu_3 n_{LT} + \nu_4 n_{RT}) \\
& + k_8 m (\nu_1 \sum_{i=1}^{n_{ST}} \bar{v}_i^2 + \nu_2 \sum_{i=1}^{n_{TL}} \bar{v}_i^2 + \nu_3 \sum_{i=1}^{n_{LT}} \bar{v}_i^2 + \nu_4 \sum_{i=1}^{n_{RT}} \bar{v}_i^2)
\end{aligned} \tag{3.17}$$

wherein k_1, \dots, k_8 are regression coefficients, n is the total number of road segments along the trip, L is the trip distance, and ν_1, ν_2, ν_3 and ν_4 are constant coefficients. In the equation, \bar{v}_i denotes the speed of the segment immediately following the traffic signals, stops or turns which lays on the path. Note that at the beginning of such street segment $v_{ini} = 0$ as the vehicle has come to stop at the intersection.

In section 3.3.1, we show that the coefficients of our model, k_1, \dots, k_8 differ among different vehicles making it harder to generalize from vehicles we have data for to those we do not.

3.3 Model Generalization to Predict Green Routes

In this section, we demonstrate the foundations of one of the key mechanisms in participatory sensing applications that are tolerant to conditions of sparse deployment; namely, the generalization from sparse multidimensional data. The generalization mechanism solves a key problem at a critical phase of most newly deployed systems, which makes it important. Such generalization is complicated by the fact that, in high-dimensional datasets, one size does not

fit all. Hence, for example, developing a single regression model to represent all data is highly suboptimal. In the case of GreenGPS, the data contributed by users of our participatory sensing application will be a sparse sampling of routes and cars. Hence, we aim to use data collected by a smaller population to build models capable of predicting the fuel consumption characteristics of a larger population.

3.3.1 Model Evaluation: One Size Fits All?

Regression analysis is a standard technique for estimating coefficients of models with known structure. In this section, we demonstrate that a single regression model is a bad fit for our data. Said differently, while a regression model that accurately predicts fuel consumption can be found for each car from data of that one car, the model found from the collective data pool of all cars is not a good predictor for single vehicles. Hence, in a sparse dataset (where data is not available/sufficient for all cars) it is not trivial to generalize. We illustrate that challenge by first evaluating the performance of car models obtained from their own data (which is good), then comparing it to the trivial generalization approach: one that finds a single model based on all car data then uses it to predict fuel consumption of other cars. A solution to the challenge is presented in the next section.

We evaluate the accuracy of models derived from vehicle data according to a cross validation approach. We predict fuel consumption of a randomly chosen trip using a model trained based on data from other trips. We distinguish models based on other trips of the same car from models based on data from other cars as well in predicting the fuel consumption of the one trip. The fourth and fifth columns of Table 3.1 summarize the resulting errors,

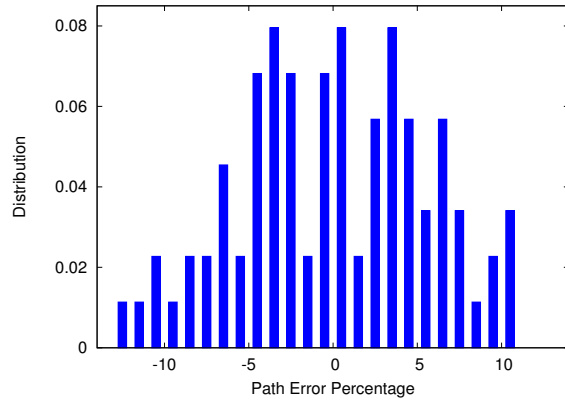


Figure 3.4: The path error percentage distribution for one car

respectively, for the set of cars used. More specifically, to compute the error of a particular trip, the trip is removed and a model is trained based on other trips of the same car which is then utilized to predict fuel consumption for the trip. Using the collected actual fuel consumption of the trip, the relative prediction error percentage is then computed. This is repeated for all trips in the dataset. The average error percentage across all trips of the same car (i.e., the summation of all trips' absolute errors divided by the number of trips) is considered as Individual error percentage. As for the General error percentage, when training the model, the trips of other cars are included in the training dataset as well. The errors reported here are for trips from four miles up to ten miles; the errors for shorter and longer trips will be presented later in Chapter 7.

We also plot the error distribution for individual trips (for one car) in Figure 3.4. We observe that the distribution is near normal and the mean is near zero (-0.14%). We observe a similar distribution for other cars too.

We also observe from Table 3.1 that the prediction errors of the single model computed from the data of all cars are significantly (over several times) worse than those of the models obtained from each individual car. This

Table 3.1: The average error percentage (magnitude) for the individual car models and the generalized case when all the data is used to obtain the model

Car Make	Car Model	Car Year	Individual Error %	General Error %
Toyota	Camry	2004	1.55	8.44
Chevrolet	Impala	2002	3.02	17.16
Ford	Ranger	2008	0.89	25.26
Toyota	Corolla	2000	6.06	10.68
Buick	LeSabre	2002	3.38	7.46
Ford	E-250	2011	3.59	7.93
Toyota	Corolla	2010	4.31	18.47
Toyota	Celica	2001	4.94	11.69
Nissan	Altima	2006	3.83	7.04
Subaru	Impreza	2010	0.09	3.82
Toyota	Corolla	2004	3.67	13.59
Mazda	Mazda6	2003	3.94	18.5
Audi	A4	2005	6.86	14.58
Toyota	Camry	2012	4.96	7.59
Subaru	Impreza	2010	9.22	15.47
Hyundai	Santa-Fe	2001	3.3	17.92
Ford	Taurus	2002	4.01	5.51
Mitsubishi	Eclipse	2002	5.32	15.91
Nissan	Altima	2010	2.44	9.59
Mitsubishi	Galant	2002	4.45	12.19
Toyota	Celica	2000	6.24	8.74
Toyota	Camry	2004	0.73	13.76
Average Error Percentage:			4.91	11.33

suggests the existence of non-trivial bias in the error of the former model that does not cancel out by aggregation. In the next section, we propose a way to mitigate this problem based on grouping cars into clusters, such that prediction can be done based on other *similar* cars by some metric of similarity.

3.3.2 Model Clustering

The above discussion and results suggest a need for better generalization over vehicle data. Different car types behave differently. Even though the model is parameterized by factors such as car weight and frontal area, they are not enough to account for differences among cars. This is a common problem in high-dimensional datasets collected in participatory sensing applications. The question becomes, if we cannot generalize over the whole set, can we generalize over a subset of dimensions?

A solution is borrowed from the general literature on data cubes [94]. Data cubes are structures for Online Analytical Processing (OLAP) that are widely used for multidimensional data analysis. They group data using multiple attributes and extract similarities within each group. For example, previous work showed how to efficiently construct regression models for various subsets of data [95]. The data cube framework can thus help compute the optimal generalization order in that it can help generalize data based on those dimensions that result in the minimum modeling error.

We consider four major attributes (data dimensions) of a given car: *make*, *model*, *year* and *class*¹. Based on these four attributes, data can be grouped in 16 ways, out of which 6 are redundant since vehicle model specifies make

¹Other vehicle attributes can be employed as well, for example, *city mpg*, *highway mpg*, *mpg difference* (the difference between highway mpg and city mpg) and *mpg ratio* (the ratio of highway mpg to city mpg).

and class as well. At one extreme, all cars may be grouped together, thus producing a single regression model (which we have shown is not acceptable). At the other extreme, cars can be partitioned into clusters based on their four attributes. Intermediate clusters are constructed based on a subset of these attributes. A separate model is derived for each cluster. One should note that in cluster (model, year) for example, a Camry 2004 is modeled differently from a Camry 2012 and a Civic 2004.

Between the two extremes, to find out which clustering scheme gives the best accuracy, we obtain the average percentage error for each scheme. The results, summarized in Figure 3.5, show that different generalizations have different quality. These generalizations are better than using all cars data lumped together. While our dataset is small to make general conclusions, as more data is collected in our deployed participatory sensing infrastructure (e.g., say deployment reaches 100s of cars), progressively better generalizations can be attained. In the figure it can be observed that some of the clusters present quite similar accuracy. This behavior is induced due to limited vehicle type overlap in our dataset and the performance of the intermediate clusters is not well differentiated thereof. Specifically, these clusters end up having several single vehicle groups in common. To draw general conclusions, a further scaled vehicle set with adequate vehicle overlap with respect to the considered attributes is required.

To use results of Figure 3.5, one would build models for each cluster shown in the Figure 3.5 which has sufficient data for reliable modeling. Chapter 5 elaborates on how the reliability of a model can be inferred. To model a car, an instantiated cluster with the same attributes as the car is utilized that has the least error. If a car is encountered for which none of the clusters match the car, we have no recourse but to use the model computed from all data. That

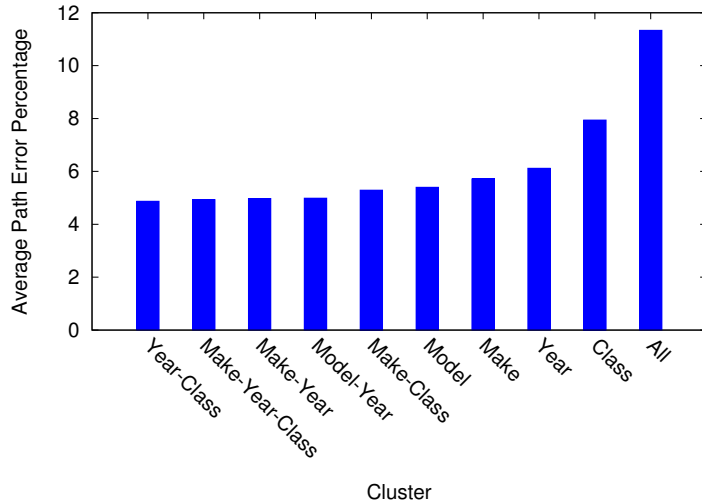


Figure 3.5: Average error percentage (magnitude) of the models obtained from various clusters

is, the clusters in Figure 3.5 are traversed sequentially, from the most accurate to the least accurate, until a cluster containing sufficient data is reached. We evaluate the performance of the *Cluster-based* modeling technique by measuring how accurately an individual car can be modeled using the data from cars with similar attributes. Specifically, we construct the model cluster while removing data of a certain car trip. We use the model cluster to estimate the fuel consumption for the given car trip. This is done for all car trips. The resulting average error percentage is presented in Table 3.2. As it can be observed from the table, the Cluster-based modeling technique has led to significant accuracy improvements compared to the General model. In a few cases, such as the second vehicle in the table (Chevrolet Impala 2002 Large) the error has reduced even over the Individual model. This is because the individual vehicles involved did not collect representative enough data to generate an accurate model. Hence, improvements are achieved from grouping of this vehicle and Buick LeSabre 2002 Large into the same cluster (i.e., Year-Class) that results in reducing the errors even over the Individual

model for both vehicles.

3.4 Dynamic Traffic Modeling

Our experience reveals, not surprisingly, that the degree of traffic congestion plays the largest role in accounting for fuel consumption variations among individual trips of the same vehicle. To model the effect of dynamically changing traffic, the street segments real-time speed should be used as the speed rating in the fuel consumption model presented in equation 3.17. However, it should be noticed that the current speed at distant locations would become obsolete when the vehicle arrives there. Therefore, for distant areas the future traffic status should be predicted, to be used in the model. Here we address such spatio-temporal parameters contributing to the model.

Let the overall speed of a street segment at location x at time t be denoted by $v_{x,t}$ and defined as:

$$v_{x,t} = \mu_{x,t} + \gamma_{x,t} \quad (3.18)$$

wherein $\mu_{x,t}$ represents the speed mean value and $\gamma_{x,t}$ represents the deviation from the mean. The former, $\mu_{x,t}$, is calculated through a weighted average over the past speed values taken from traffic history for street segment located at x . In the calculation higher weights are given to the more recent speed values. The latter, $\gamma_{x,t}$, can be modeled as a stationary process with mean zero, modeled using an autoregressive moving average (ARMA) model as follows:

$$\gamma_{x,t} = \sum_{l=1}^p \phi_l \gamma_{x,t-l} + e_{x,t} - \sum_{l=1}^q \theta_l e_{x,t-l} \quad (3.19)$$

Table 3.2: The average error percentage (magnitude) for the cluster-based model constructed based on the optimal generalization order

Car Make	Car Model	Car Year	Cluster Error %
Toyota	Camry	2004	1.72
Chevrolet	Impala	2002	2.48
Ford	Ranger	2008	5.26
Toyota	Corolla	2000	6.01
Buick	LeSabre	2002	2.45
Ford	E-250	2011	3.59
Toyota	Corolla	2010	9.32
Toyota	Celica	2001	4.94
Nissan	Altima	2006	3.83
Subaru	Impreza	2010	4.74
Toyota	Corolla	2004	3.67
Mazda	Mazda6	2003	3.94
Audi	A4	2005	6.86
Toyota	Camry	2012	4.96
Subaru	Impreza	2010	8.23
Hyundai	Santa-Fe	2001	3.3
Ford	Taurus	2002	5.06
Mitsubishi	Eclipse	2002	5.32
Nissan	Altima	2010	2.44
Mitsubishi	Galant	2002	8.11
Toyota	Celica	2000	6.06
Toyota	Camry	2004	2.21
Average Error Percentage:			5.07

where the first p terms correspond to the autoregressive terms and the last q terms correspond to the moving average terms. The coefficients ϕ_1, \dots, ϕ_p and $\theta_1, \dots, \theta_q$ are the model parameters. The subscript l denotes the time lag and $t - l$ means l time units before the current time t . The $e_{x,t}$'s are independent, identically distributed random variables, each with mean zero and variance σ_e^2 .

However, it is evident that there exists spatial correlation in road traffic, that is, the traffic status at some street depends on that of the neighboring streets as well. In order to incorporate the spatial correlation into the model, let the spatial correlation matrix be denoted as $\Pi^{(\tau)} = [\pi_{x,x'}^{(\tau)}]_{N \times N}$ where $x, x' \in \{1 \dots N\}$ and N denotes the number of street segments. The entry $\pi_{x,x'}^{(\tau)} \in \mathbb{N}$ specifies the number of time units needed for the traffic at street segment x' to influence the traffic at x according to the average historical speed of the area. Note that $\pi_{x,x'}^{(\tau)} = 0$ implies $x = x'$. Also that, when there is no spatial correlation between x and x' at time interval τ , $\pi_{x,x'}^{(\tau)} = \infty$. The superscript τ will be described shortly.

The spatial correlation is then reflected in the model as follows:

$$\begin{aligned} \gamma_{x,t} = & \sum_{l=1}^p \sum_{x'=1}^N \phi_l I(\pi_{x,x'}^{(\tau)} \leq p - l + 1) \gamma_{x',t-l} + e_{x,t} \\ & - \sum_{l=1}^q \sum_{x'=1}^N \theta_l I(\pi_{x,x'}^{(\tau)} \leq q - l + 1) e_{x',t-l} \end{aligned} \quad (3.20)$$

Thus, to predict the future street speed, the model expression includes not only the impact of the traffic history at the same location x , but also the effect of the traffic at nearby correlated streets as well. To make the model expression concise, let $\Gamma_t = [\gamma_{1,t} \dots \gamma_{N,t}]^t$, $e_t = [e_{1,t} \dots e_{N,t}]^t$, $\Upsilon_p = [I(\pi_{x,x'}^{(\tau)} \leq$

$p - l + 1)$] $_{N \times N}$ and $\Upsilon_q = [I(\pi_{x,x'}^{(\tau)} \leq q - l + 1)]_{N \times N}$. The model can thus be rewritten as:

$$\Gamma_t = \sum_{l=1}^p \phi_l \Upsilon_p \Gamma_{t-l} + e_t - \sum_{l=1}^q \theta_l \Upsilon_q e_{t-l} \quad (3.21)$$

To compute the most fuel-efficient route the speed values in equation 3.17 are computed as follows. The real-time speed $V_t = \mathcal{M}_t + \Gamma_t$, where $V_t = [v_{1,t} \cdots v_{N,t}]^t$ and $\mathcal{M}_t = [\mu_{1,t} \cdots \mu_{N,t}]^t$, is used for the speed of the street segments up to 5 minutes (one time unit) away from the source address. For streets $t + 5n$ to $t + 5(n + 1)$ minutes away, where $n \in \{1 \cdots 11\}$, the predicted speed value $V_{t+5n} = \mathcal{M}_{t+5n} + \Gamma_{t+5n}$ is utilized. To calculate Γ_{t+5n} , $n > 1$, first the future speed Γ_{t+5} is computed through equation 3.21 and using the real-time speed Γ_t and the speed values from history, Γ_{t-l} . The predicted speed Γ_{t+5} is then used in the prediction of the Γ_{t+10} . The computation continues until Γ_{t+5n} is calculated. Finally, for streets more than one hour away, the average historical speed \mathcal{M}_t is utilized. Note that, utilizing the predicted speed values the approximate time that the vehicle reaches each street segment along the path can be computed.

The computed most fuel-efficient route is updated every 5 minutes using the most recent traffic information. This calls for the speed predictions to be performed every 5 minutes, however, the spatial correlation matrix is computed once. To compute $\Pi^{(\tau)}$, we divide the time horizon based on the time of the day and the day of the week, and then for each time period, referred to by τ , the spatial correlation matrix is computed accordingly. For example, for Friday 3pm to 8pm $\Pi^{(Fri\ 3pm-8pm)}$ is computed once. For holidays a separate time period can be considered.

It should be mentioned that the results reported in Chapter 7 are based on

data collected in the area of Urbana-Champaign. The county is almost never congested and has very low traffic variability that renders the extensions mentioned in this section unnecessary. The approach can be used in larger cities, where savings will likely be higher than those reported in this thesis due to the larger variability in traffic conditions that could be taken advantage of, and because of the larger connectivity which offers more alternatives in the choice of route. The growth in GreenGPS service adoption enables the system to acquire real-time traffic data, the timing characteristics of which can be analyzed by an approach similar to [96]. Currently, Google maps [4], INRIX [97], Nokia Here [98], Microsoft Bing [99], MapQuest [5], PeMS [90] and 511NY [100] are example traffic data providers that offer real-time and/or historical traffic information.

CHAPTER 4

TRAFFIC REGULATOR DETECTION

This chapter is motivated by developing tools to support intelligent transportation systems for improving transportation safety, mobility, cost and environmental sustainability. Many driver support tools have already been developed to achieve these goals. To be successful, tools that support intelligent transportation require knowledge of not only street maps but also factors that affect traffic flow. A key factor is the location of traffic regulators (traffic lights and stop signs). Traffic regulators significantly influence fuel consumption, pollution, delay and safety issues. Navigation systems can take into account the impact of traffic regulators on travel time, fuel consumption and gas emissions in searching for appropriate routes.

Unfortunately information on the location of traffic regulators is not widely available. Either there is no information at all, beyond paper archives, in some areas (the authors had several unsuccessful attempts to access such data in different cities), or such information is fragmented into municipalities and is hard to integrate. To recognize traffic regulators one may try to decide upon exploiting the following intuition: community roads use stop signs and mid to large roads use traffic lights. Investigating such idea we found a poor performance almost equivalent to random guessing: in a collected dataset consisting of 3780 intersection samples it resulted in an accuracy of 59%! Thus neither cases offers a solution that can help, for example, in developing a national service for intelligent navigation. This chapter addresses the prob-

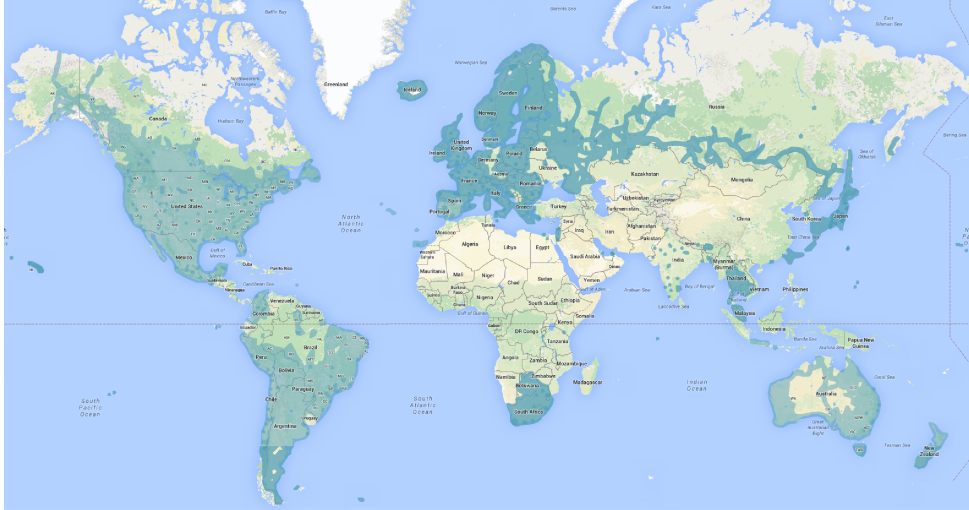


Figure 4.1: The coverage map of Google street views: covered areas in dark blue

lem of indirectly extracting traffic regulator information automatically from widely available and up-to-date worldwide data sources designed for other purposes [101].

One solution to the problem could be to employ vision algorithms to process Google street views [102] to extract traffic regulators and their locations. However, Google street views coverage is far from complete and has faced privacy concerns and legal issues in some countries, such as Germany and Australia, stopping it from achieving world-wide coverage. Figure 4.1 shows the coverage map of Google street views with covered areas denoted in dark blue. It can be observed that most of the world not covered: missing parts include almost all of Asia, Russia, Middle East and Africa, the majority of Canada, and parts of Europe, South America, and Australia. In addition, extensive use of Google street views API is not free. Therefore, such solution is not open (outside Google).

To address the problem, this chapter proposes a novel method based on

combined map-based inference and crowd-sensing. The solution approach is based on the fact that the placement of traffic regulators obeys rules and guidelines specified by the authorities, as opposed to being completely random. Hence, we *reverse engineer* the locations of traffic regulators. To this end, we aim at leveraging broadly available map data in order to construct models capturing regulator placement policies. Specifically, the proposed approach builds a map-based inference model using a machine learning method. The constructed inference model is enhanced with crowd-sourced vehicular GPS traces, where available. The model integrates the power of map-based inference (exploiting *static* attributes) and crowd-sensing based inference (exploiting *dynamic* attributes), thereby improving prediction accuracy compared to either of the approaches alone. The approach does not require the presence of driving traces (although can use that information to improve inference). Hence, it can be broadly employed even when Google street view or GPS traces are not available.¹

To provide the information required for our inference model, we use the widely available free world map of *OpenStreetMap (OSM)* [103]. OSM data are collected from various sources, such as the US TIGER database [104], Landsat 7 [105], and user contributed GPS data. OSM has about 2 million registered contributors at the time of writing this manuscript, and enjoys a very good coverage across the world. The update map of OSM is presented in Figure 4.2. More recent updates are shaded in red and older imports are depicted in green and blue.

There are at least two main categories of tools that directly require the work

¹In this work, we ultimately conducted the study at locations where Google street view coverage is *available*, but that was motivated by the ease of collecting ground truth on the performance of our algorithms. The real value, of course, lies in applying the same at locations where street view is not available.

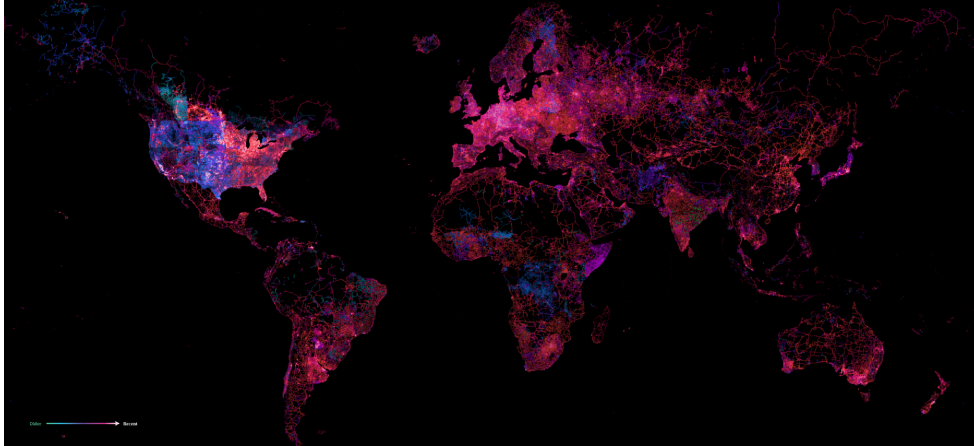


Figure 4.2: The update map of OpenStreetMap: more recent updates shaded in red and older imports depicted in green and blue

presented in this chapter:

- *Navigation and Route Planning Systems:* One of the most important factors impacting the quality of navigated routes is the number of traffic regulators along the path. Traffic regulators impose stop and go driving mode and engine idling time intervals which cause excess travel time, fuel consumption, and gas emissions. To produce efficient routes, navigation engines and route planning systems, like GreenGPS, must account for the impact of traffic regulators.
- *Driver Advisory Tools:* Driver advisory tools need to inform drivers of the presence of traffic regulators in advance. Doing so allows drivers to take appropriate actions when approaching traffic regulators to reduce chances of accidents. Hence, knowledge of location of traffic regulators is required.

The rest of the chapter is structured as follows. Section 4.1 describes the inference approach. Section 4.2 presents map-based inference model and

Section 4.3 presents crowd sensing based model construction. Section 4.4 describes general applicability property. Section 4.5 explains the evaluation methodology and presents performance results. Section 4.6 presents discussion.

4.1 Modeling Approach

Traffic regulators are not placed at random in a city. There exist rules and guidelines explaining where traffic signs should be installed. In the United States, the federal highway administration (FHWA) in the transportation department defines such standards. The standards are developed and published in the *Manual on Uniform Traffic Control Devices* (MUTCD) and used by road managers on installment and maintenance of traffic control devices on all highways, public streets, private roads open to public travel, and bike-ways [106]. A few sample guidelines for locating traffic regulatory signs are reported in Table 4.1.

The criteria established in the MUTCD are known as *warrants*. Warrants include a variety of parameters such as traffic volumes on all approaches, approach speeds, delay experienced by crossing traffic and pedestrians, existence of school crossings, the number and angle of approaches, proximity to the intersection of a grade crossing, sight distance available at each approach, and reported crash experience.

The fact that there exist rules and guidelines behind the placement of traffic regulators implies that the intersection street approaches facing signs could be identified through the evaluation of the specified warrants. However, as specified by the FHWA, the decision on the location of traffic regulatory signs are in essence based on engineering judgment and most of the warrant

Table 4.1: Sample MUTCD rules and guidelines on the placement of traffic regulators

- ▷ *Engineering judgment should be used to establish intersection control. The following factors should be considered:*
 - A. *Vehicular, bicycle, and pedestrian traffic volumes on all approaches;*
 - B. *Number and angle of approaches;*
 - C. *Approach speeds;*
 - D. *Sight distance available on each approach; and*
 - E. *Reported crash experience*

- ▷ *Once the decision has been made to control an intersection, the decision regarding the appropriate roadway to control should be based on engineering judgment. In most cases, the roadway carrying the lowest volume of traffic should be controlled.*

- ▷ *The use of STOP signs on the minor-street approaches should be considered if engineering judgment indicates that a stop is always required because of one or more of the following conditions:*
 - A. *The vehicular traffic volumes on the through street or highway exceed 6000 vehicles per day;*
 - B. *A restricted view exists that requires road users to stop in order to adequately observe conflicting traffic on the through street or highway; and/or*
 - C. *Crash records indicate that three or more crashes that are susceptible to correction by the installation of a STOP sign have been reported within a 12-month period, or that five or more such crashes have been reported within a 2-year period. Such crashes include right-angle collisions involving road users on the minor-street approach failing to yield the right-of-way to traffic on the through street or highway.*

parameters are not accessible either. As a result the location of traffic signs cannot be inferred in a deterministic manner. Instead, we devise a machine learning approach to construct classifiers for identifying the location and type of traffic regulators on city streets. We call them *detection models*. In the following subsections, we discuss how detection models are constructed.

4.2 Map-based Inference

To model the placement of the traffic regulators, rather than modeling the complex detailed placement guidelines, we take the simple but effective approach of investigating the final outcome of the procedure to build the detection model. To acquire the attributes of the intersections' street approaches, OpenStreetMap provides a reasonably rich resource. OpenStreetMap is the equivalent of Wikipedia for maps and the data are provided from various free sources such as the US TIGER database [104], Landsat 7 satellite imagery [105], and user contributed GPS data.

An editable map of a given area is created in XML format using the three data primitives of OSM: *nodes*, *ways* and *relations*. A node represents a specific point of interest defined by its latitude and longitude (e.g. junction of roads, a hospital, etc). A way is an ordered list of nodes representing linear features or boundaries of areas (e.g. roads, rivers, park or building boundaries). A relation models relationship between two or more data objects (i.e. nodes, ways, other relations) and can be used for different purposes, for example for defining routes (e.g. highways, bus routes) and enforcing restrictions (e.g. no turn from one way into another one). All types of data objects can have *tags* which define the meaning of the particular corresponding object and describe its geographic attributes. For example, a “type” tag

can be used to define a relation as turn restriction. Consequently, tags “to” and “from” define *roles* of the two contributing ways (i.e., relation *members*) to which the restriction is applied. Tags are presented in key-value format.

Enjoying coverage across the world and being a free map, OSM creates an appropriate resource for the extraction and/or computation of various attributes of street approaches. Thus, we model the outcome of the task done by the engineering judgment teams in terms of plausible related street attributes extracted from OSM. The procedure follows.

4.2.1 Intersection Extraction

First, intersections are extracted using OSM maps. To find intersections, as they are not directly provided by OSM, initially nodes which are present in two or more ways are extracted. However, considering that ways are not merely used for representing street segments, some filtering and cleaning phases are performed to remove invalid intersection candidates.

The extracted intersections are then decomposed into multiple *street approaches*, depending on the number of street segments joining at the intersection point. For example, 3-way and 4-way intersections are decomposed into three and four street approaches, respectively. Figure 4.3 shows a 4-way intersection “*S. Gregory St. & W. Oregon St.*”, denoted by n , which is composed of four approaches of α_0 , α_{90} , α_{180} , and α_{270} .

4.2.2 Street Approach Attributes

Inspired by the warrants established in the MUTCD by the FHWA, the following street attributes are selected and derived from OSM maps for the intersection street approaches:



Figure 4.3: Intersection street approaches and attributes

- *Speed Rating*: The average rated velocity of vehicles traveling along each street segment is extracted from the OSM maps.
- *End-to-End Distance*: For a given intersection street approach, we define End-to-End Distance to denote the end-to-end length of the road containing the particular street segment present at the intersection point. Figure 4.3 clarifies the definition. For the intersection of South Gregory street and West Oregon street, there are two end-to-end distances associated. First, the end-to-end distance for the north-south direction which is the length of the South Gregory street from end-point e_0 to end-point e_{180} . Second, the end-to-end distance for the east-west direction for which the length of the West Oregon street between two end-points e_{90} and e_{270} is computed.
- *Semi Distance*: For a given intersection street approach, Semi Distance denotes the length of the road containing the particular intersection street segment from the intersection point towards the end of the road on the same approach. In Figure 4.3, there are four semi distances

associated with the intersection n corresponding to the four approaches α_0 , α_{90} , α_{180} , and α_{270} . The semi distance for the approaches α_0 , α_{90} , α_{180} , and α_{270} equals to the length of the street between the intersection point n and the end-points e_0 , e_{90} , e_{180} , and e_{270} , respectively.

- *Closest Intersection Distance*: For a given intersection street approach, the distance between the intersection and the nearest intersection along the approach is considered. In Figure 4.3 for the intersection under consideration, n_0 , n_{90} , n_{180} , and n_{270} are the closest intersections along the approaches α_0 , α_{90} , α_{180} , and α_{270} , respectively, and the corresponding distance is computed for each street approach.
- *Category*: Category of a street segment denotes the type and importance of the street segment in the road network. Ranging from the most to the least important it could be *motorway*, *trunk*, *primary*, *secondary*, *tertiary*, *motorway link*, *primary link*, *unclassified*, *road*, *residential*, or *service*².

4.2.3 Knowledge Representation

Every intersection approach is used as a sample in computing the model. The aforementioned attributes are extracted for each street included in the intersection and represented in the sample corresponding to each contributing street approach. More specifically, the intersection of South Gregory street and West Oregon street in Figure 4.3 is represented by four samples as follow:

$$\triangleright \langle \alpha_0, v_0, v_{90}, v_{180}, v_{270}, L_0, L_{90}, L_{180}, L_{270}, l_0, l_{90}, l_{180}, l_{270}, \\ d_0, d_{90}, d_{180}, d_{270}, c_0, c_{90}, c_{180}, c_{270} \rangle$$

²The road classification system used in OSM is British English, hence, for example the *motorway* tag is equivalent to the US interstate highway.

- ▷ $\langle \alpha_{90}, v_{90}, v_{180}, v_{270}, v_0, L_{90}, L_{180}, L_{270}, L_0, l_{90}, l_{180}, l_{270}, l_0, d_{90}, d_{180}, d_{270}, d_0, c_{90}, c_{180}, c_{270}, c_0 \rangle$
- ▷ $\langle \alpha_{180}, v_{180}, v_{270}, v_0, v_{90}, L_{180}, L_{270}, L_0, L_{90}, l_{180}, l_{270}, l_0, l_{90}, d_{180}, d_{270}, d_0, d_{90}, c_{180}, c_{270}, c_0, c_{90} \rangle$
- ▷ $\langle \alpha_{270}, v_{270}, v_0, v_{90}, v_{180}, L_{270}, L_0, L_{90}, L_{180}, l_{270}, l_0, l_{90}, l_{180}, d_{270}, d_0, d_{90}, d_{180}, c_{270}, c_0, c_{90}, c_{180} \rangle$

in which α denotes the approach identifier, v denotes the speed rating, L denotes the end-to-end distance, l denotes the semi distance, and c denotes the category.

4.2.4 Methodology

The machine learning technique of Random Forests (RF) [107] is used to train a classifier modeling the placement of traffic regulators in terms of the attributes of the intersection approaches. In the RF technique, a set of decision trees, grown in an automated manner (in randomly selected subspaces of data) form the model and will be exploited for drawing inferences on the location of traffic signs. Specifically, the learned model determines the presence and type of traffic regulators at a given location; namely, it denotes the existence of a *traffic light (TL)* or a *stop sign (ST)*, or the *absence of traffic regulators (None)* at a given intersection approach. Random Forests presented a superior performance compared to several other learning techniques.

4.2.5 Domain Knowledge

As explained before, each intersection is decomposed into its corresponding street approaches and then the set of street approaches is used as training samples for building detection models. The following domain knowledge rule triggered by jointly considering approaches associated with the same intersection is missed out thereof. Hence at the end of the modeling task we employ the rule:

- Either all or none of the approaches contributed to the same intersection have a traffic light. This implies that when the classifier labels some of the approaches of an intersection, but not all of them, with TL , the predicted label should be revised. The revision makes either all or none of the intersection approaches have a traffic light, this being decided upon utilizing probabilities computed based on the fraction of decision trees voting for the approaches' alternative labels.

4.3 Crowd-sensing Model

Apart from map-originated information, crowd-sourced data could serve as a resource assisting with detection of traffic regulators. In particular, crowd-sourced vehicular GPS traces have the potential to help in improving the detection and discrimination task when properly integrated with the map-based inference model. The widespread use of GPS-enabled devices such as smartphones facilitates the collection of vehicular GPS traces. The goal here is to exploit the discrimination potential of GPS driving traces whose provisioned information is complementary to that provided by OSM maps. Hence, where such traces are available, they are integrated with the map-based inference model, resulting in the construction of more accurate models

(compared to map-based inference models or crowd-sensing based inference models alone).

To integrate crowd-sourced data, as opposed to employing it as a separate layer, we decide to view both the map data and crowd-sourced data together and develop only one modeling layer. To achieve this purpose, we propose various attributes which could be drawn from GPS driving traces and help in classifying traffic regulators. Compared to map-derived attributes, which constitute *static* map features, the attributes derived from vehicular GPS traces capture *dynamic* behaviors around traffic regulators, and hence are complementary to the first category of attributes. Therefore, trace-based attributes are selected so as to reflect the differences in driving behavior when facing different types of traffic regulators.

With respect to the fact that GPS traces are not as openly available (for privacy reasons) compared to the widespread availability of maps data, developing a single model would face *missing attribute values*. To address the issue, we take the approach of *Reduced Models* and develop models in data subspaces. Here, there are only two different patterns of missing features (either all or none of the dynamic attributes are present), wiping out the drawback of reduced models (being generally exponentially expensive) and leading to an efficient solution. On the other hand, reduced-feature models have been shown to outperform other approaches such as imputation-based methods in which missing values are replaced with estimation of the values or the corresponding distributions [108].

To this end, we propose to use the following vehicular traces-derived attributes in the detection of traffic regulatory signs:

- *Traverse Speed*: The Traverse Speed shows the smallest instantaneous velocity of the vehicle when traversing an intersection while approach-

ing it along the given intersection approach.

- *Number of Stops*: The Number of Stops denotes the number of time intervals the vehicle has stopped and been idling when passing through the last street segment along a given intersection street approach.
- *Stop Duration*: The Stop Duration is the length of the latest time interval the vehicle has stopped and been idling.

For each intersection approach covered by GPS traces, the attributes' distribution parameters are used for modeling; namely, the average, the minimum, the maximum, and the standard deviation of the attribute values are employed.

The reason for proposing the above attributes is due to noticeable differences in their expected values when facing different types of traffic regulators. Specifically, while the intersection traverse speed is high at uncontrolled intersection approaches, it is about zero at stop-sign regulated intersection approaches and zero or high at signal-regulated intersection approaches (depending on the light being green or red). There is no stop at uncontrolled intersection approaches, while the number of stops of a vehicle at traffic lights is normally either zero or one and the number of stops at stop signs could be one or more (when there are other vehicles backed up in front of the vehicle). The stop duration is zero at uncontrolled intersection approaches, short at stop signs, and zero or long at traffic lights. These differences can be exploited in drawing patterns and performing detection.

The overview of the detection of traffic regulators is presented in Figure 4.4.

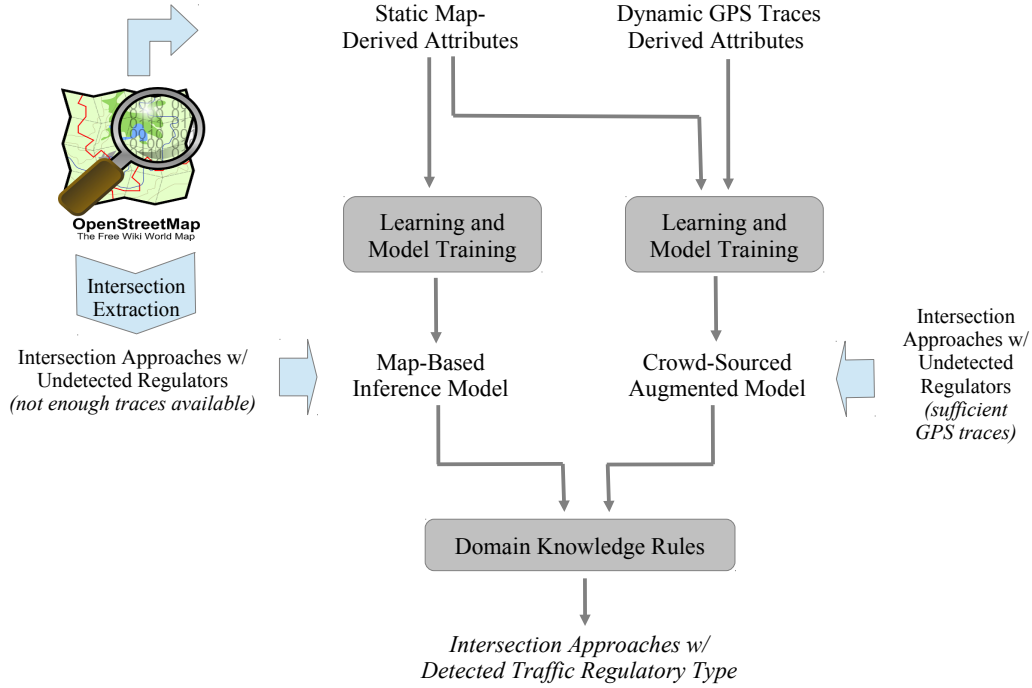


Figure 4.4: Detection of traffic regulatory signs modeling

4.4 Training and Cross-city Applicability

One important question regarding the designed traffic regulators detection methodology could be whether or not the training phase is needed each time the technique is to be employed in a new city or area for which a detection model has not already been constructed. More specifically, can we train the detection model based on data acquired from one city and then apply it for predicting the location and type of traffic regulators in a different city? Let us call this property *cross-city applicability*.

In Section 4.5, through an experimental evaluation, it is shown that the proposed detection approach has this property and can lead to sufficiently accurate results. The reason is that underlying the placement of traffic regulators in different regions are rather similar policies, as opposed to drastically different fundamental rules. In the next section, we evaluate the accuracy of

the proposed detection algorithm.

4.5 Experimental Evaluation

In this section, we evaluate the performance of the developed traffic regulator detection techniques. The datasets collected for evaluating the approach are described in Section 4.5.1 and the performance results are presented in Section 4.5.2. The impact of the detection techniques on GreenGPS is assessed later in Chapter 7.

4.5.1 Data Collection and Datasets

To evaluate the performance of our detection model, we collected datasets in multiple cities: the city of Urbana, IL, the city of Champaign, IL, part of the city of Los Angeles, CA, and part of the city of Pittsburgh, PA. Both *static datasets* and *dynamic datasets* were collected in the cities of Urbana, Champaign, and Pittsburgh. In the city of Los Angeles only static dataset was collected.

The static datasets include map information and attributes of intersection approaches, extracted from OSM maps. They also include ground-truth information for these intersections, collected from Google street views. The following data was collected:

- a total of 3691 intersection approaches was collected in the city of Urbana;
- a total of 2803 intersection approaches was collected in the city of Champaign (mostly covered);

- a total of 7561 intersection approaches was collected in part of the city of Los Angeles – the covered area ranged in latitude from $34^{\circ} 11' 38.0400''$ to $34^{\circ} 15' 25.9200''$ and in longitude from $-118^{\circ} 32' 25.4400''$ to $-118^{\circ} 26' 33.3600''$;
- a total of 1032 intersection approaches was collected in part of the city of Pittsburgh – within an area ranged in latitude from $40^{\circ} 26' 13.5600''$ to $40^{\circ} 28' 14.8800''$ and in longitude from $-079^{\circ} 59' 07.4400''$ to $-079^{\circ} 53' 34.4400''$.

Dynamic datasets contain crowd-sensed vehicular GPS traces. In order to collect dynamic datasets, we used our participatory sensing platform. In total, over 6700 miles of vehicular GPS traces were collected by a total of 46 subjects over the course of several months. The minimum, the first, second, and third quartiles, and the maximum number of traversals per intersection street approach covered by the GPS traces in the cities of Urbana, Champaign, and Pittsburgh are $\{1, 2, 5, 24, 189\}$, $\{1, 2, 4, 15, 223\}$, and $\{1, 3, 5, 6, 13\}$, respectively.

4.5.2 Results

The collected datasets span a small campus town (Urbana-Champaign), an average city (Pittsburgh), and a major metropolis (Los Angeles). We assess the accuracy of our predictions on the presence and type of traffic regulators under the two following conditions. First, we evaluate prediction accuracy when both training and testing are conducted in the same city. That is, the data collected in a given city is divided in two parts, one part is used for training and the other part is used for testing. Second, we evaluate prediction accuracy when training and testing are conducted in different cities.

The following performance metrics are considered: the *overall accuracy*, the *average per-class true positive (TP) rate*, and the *average per-class false positive (FP) rate*. The overall accuracy shows the fraction of all predicted values which correctly match the ground-truth. The two other metrics denote the average of the true positive and false positive rates, respectively, corresponding to the three traffic regulator classes, namely, *TL*, *ST*, and *None*.

To implement the Random Forests classifier, we use the statistical tool R “randomForest” package [109]. In each run 500 trees are constructed and the number of variables randomly sampled as candidates at each split is set to the square root of the number of attributes.

Figure 4.5 shows performance results with 80% confidence level in predictions for the four cities of Urbana, Champaign, Los Angeles and Pittsburgh, when training and testing are done in the same city. Results are shown for three prediction models, (i) a model learned solely from map-based attributes, (ii) a model learned from vehicular GPS traces, where available, and (iii) the combined model. They are denoted in the figure as *Static*, *Dynamic*, and *Static-Dynamic*, respectively. For the city of Los Angeles, due to unavailability of vehicular GPS traces, only the static model is presented. It can be observed that the approach works accurately in all cities. The worst-case accuracy across all cities is 97%, 95%, and 91% for the *Static-Dynamic*, *Static*, and *Dynamic* inference models, respectively. Note that, as expected, the combined approach outperforms both of the map-based inference model and crowd-sensing based inference model, reducing misclassifications from 5%-9% to 3%. This is approximately 40%-66% reduction in error. For the same setting the overall accuracy for various confidence levels is presented in Figure 4.6. The depicted minimum confidence level of “any” means all

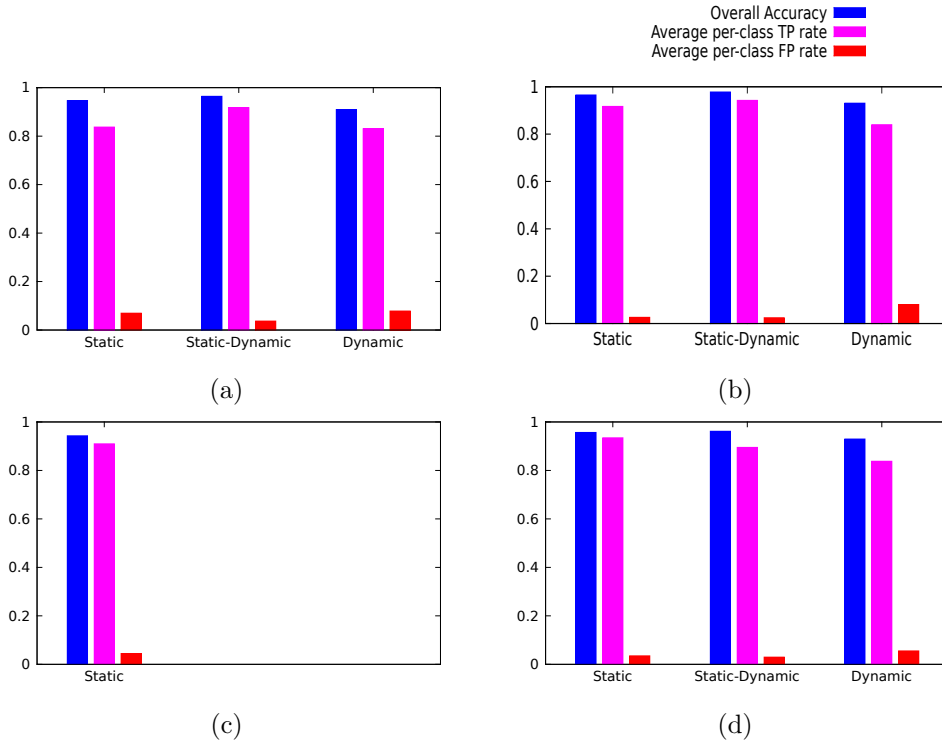
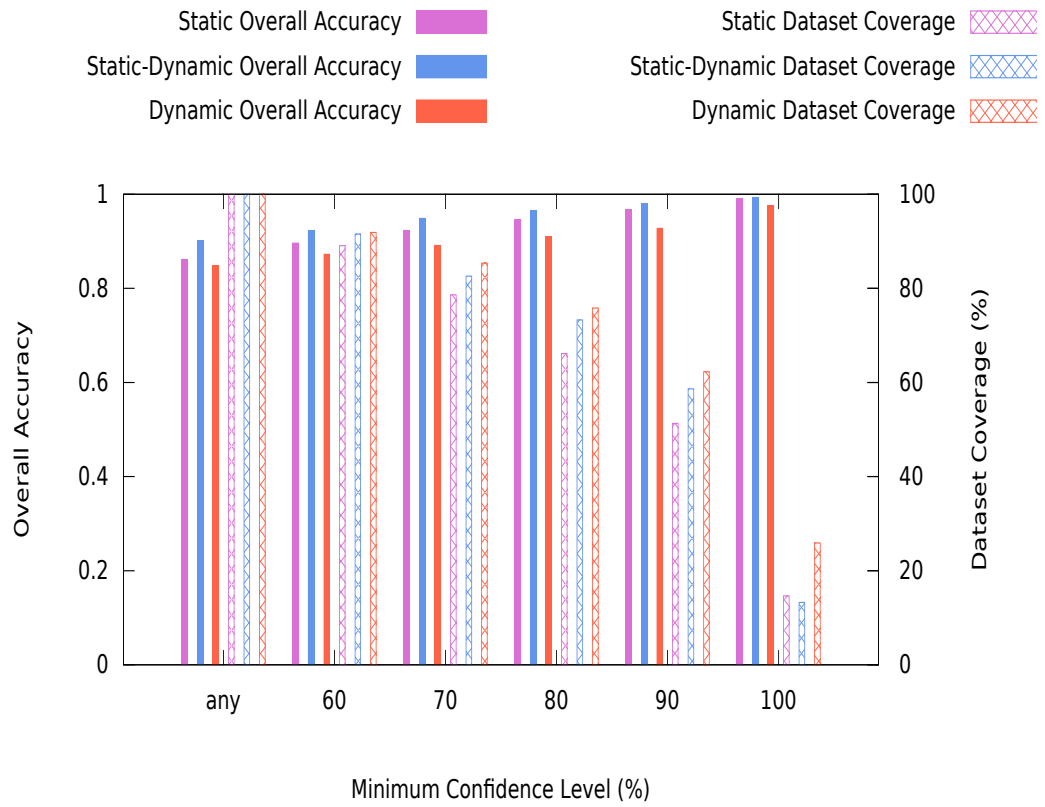


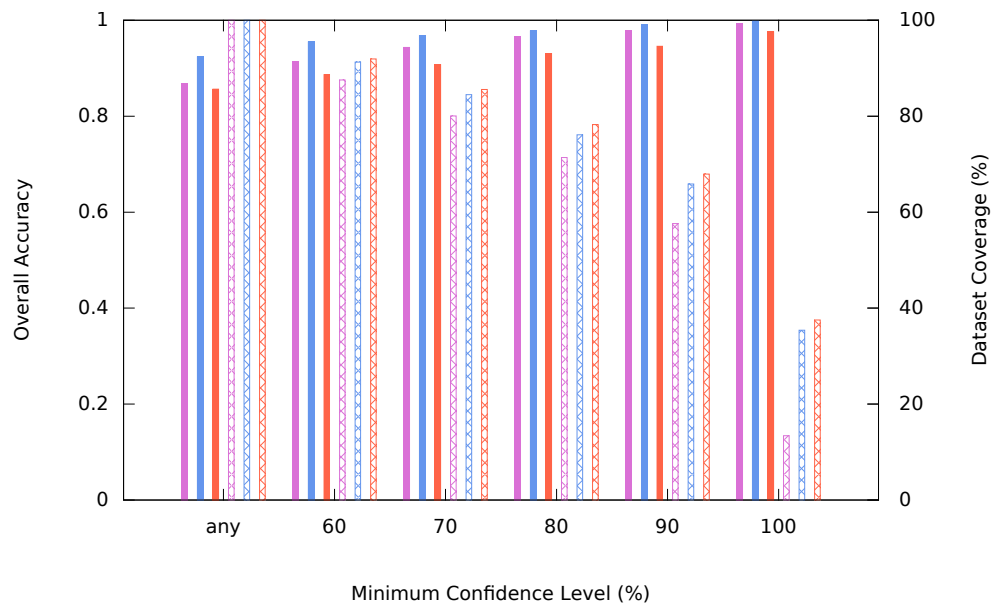
Figure 4.5: Detection accuracy in: (a) Urbana, IL; (b) Champaign, IL; (c) Los Angeles, CA; (d) Pittsburgh, PA

dataset predictions are used in computing the overall accuracy no matter how confident the classifier is in predictions. The fraction of the dataset covered at each confidence level is also denoted in the figure.

Cross-city testing results (when training in one city and testing in another) with 80% confidence level are shown in Figure 4.7. We first show cross-city testing results for a case where the cities involved are very similar. Specifically, the dataset for the campus town of Urbana is used for training a model, which is then used to detect traffic regulators in the campus town of Champaign. Results are shown in Figure 4.7a. Next we train based on data from Champaign and test for regulators in Urbana. Results are reported in Figure 4.7b. The results show that the approach can make sufficiently accurate predictions in both directions. The accuracy in both cases remains above

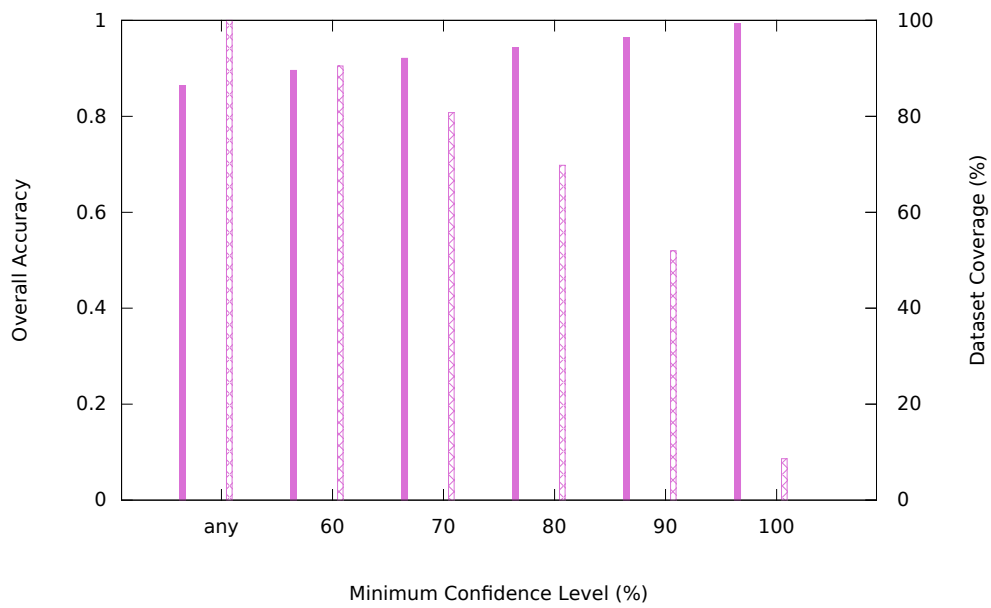


(a)

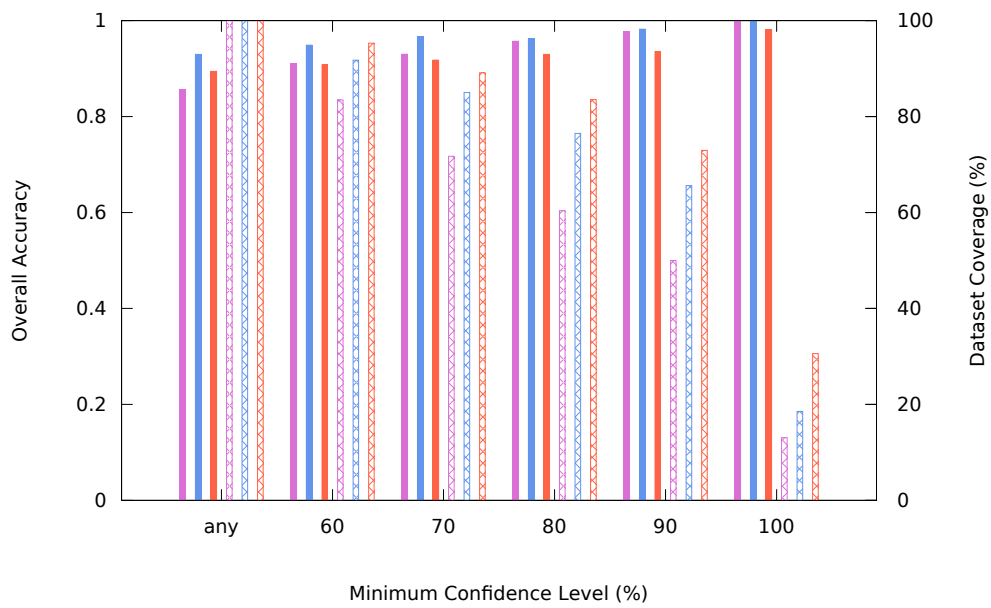


(b)

Figure 4.6: Detection accuracy at various confidence levels in: (a) Urbana, IL; (b) Champaign, IL



(c)



(d)

Figure 4.6: (cont'd) Detection accuracy at various confidence levels in: (c) Los Angeles, CA; (d) Pittsburgh, PA

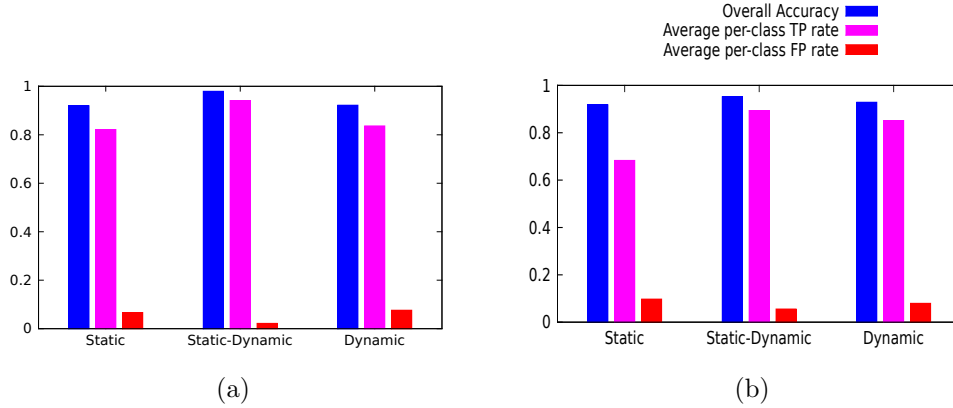
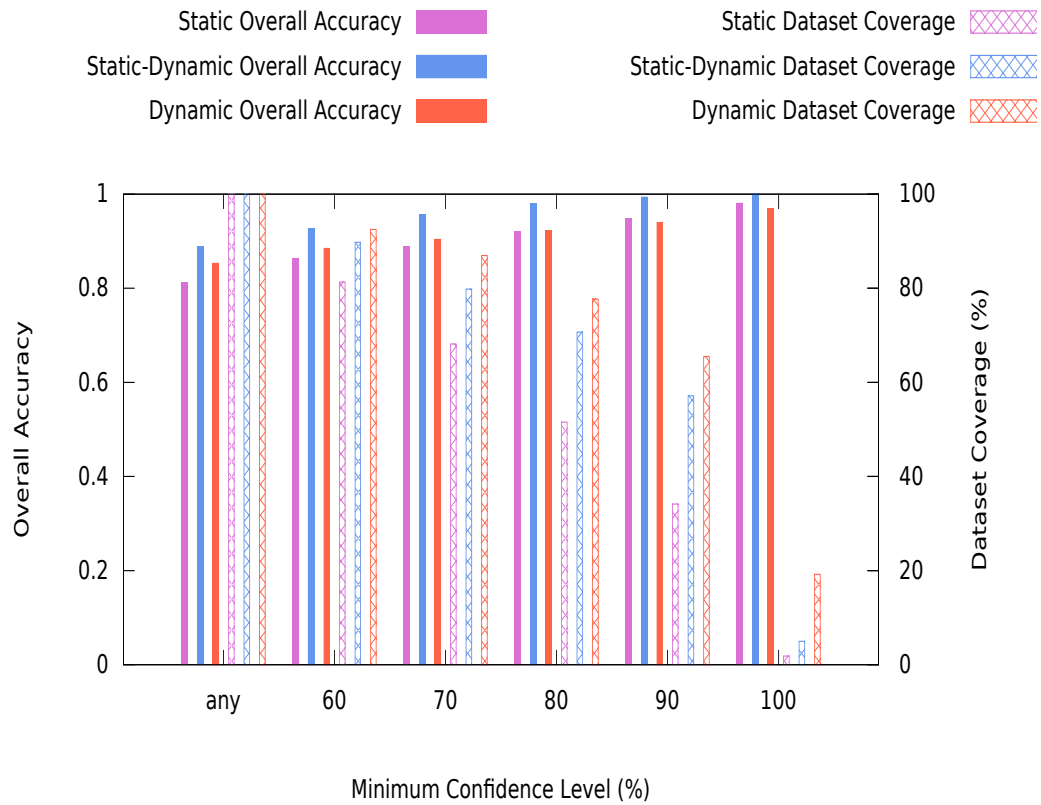


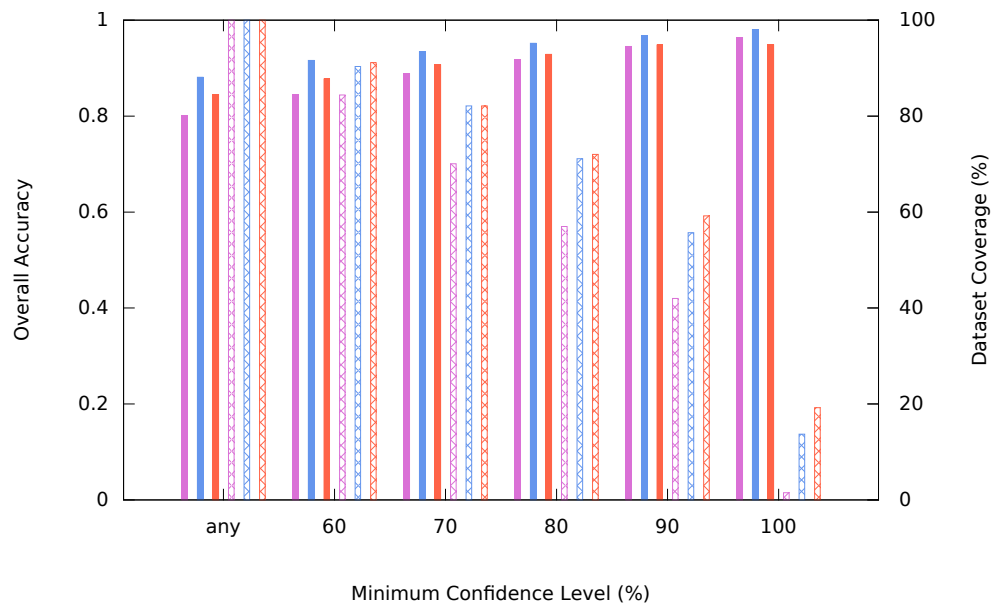
Figure 4.7: Cross-city detection accuracy in: (a) training in Urbana, IL and testing in Champaign, IL; (b) training in Champaign, IL and testing in Urbana, IL

96%, 92%, and 93% for the *Static-Dynamic*, *Static*, and *Dynamic* inference models, respectively. In other words, misclassifications are reduced from 7%-8% to 4%, a reduction in the range of approximately 42%-50%. The overall accuracy at various confidence levels is depicted in Figure 4.8.

Next, we repeat cross-city testing across very different cities. Specifically, we use the data collected in Urbana and Champaign (both being small campus towns) to compute the model. We then test it in the much larger cities of Los Angeles and Pittsburgh. The results of this experiment for both directions (i.e., training in Urbana-Champaign and testing in Los Angeles and Pittsburgh and vice versa) are reported in Figure 4.9. The results for the city of Los Angeles and Pittsburgh come from the static and static-dynamic models, respectively. Interestingly, it can be noticed that in all cases an accuracy level of above 91% is achieved, presenting sufficiently accurate predictions compared to a model computed based on data from the same city. This observation is especially important because it implies that we do not need city-by-city training to achieve reasonable performance. Results collected from a small number of representative cities can be applied to a large



(a)



(b)

Figure 4.8: Cross-city detection accuracy at various confidence levels in: (a) training in Urbana, IL and testing in Champaign, IL; (b) training in Champaign, IL and testing in Urbana, IL

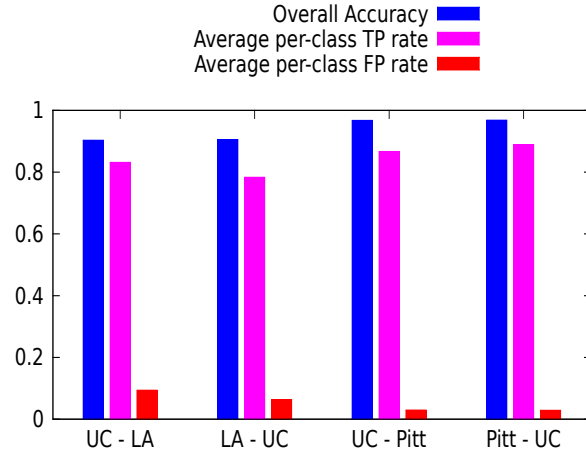


Figure 4.9: Cross-city detection accuracy when training in Urbana-Champaign, IL (*UC*) and testing in Los Angeles, CA (*LA*) or Pittsburgh, PA (*Pitt*), and vice versa – notation *A-B* denotes training in *A* and testing in *B*

number of other cities. Figure 4.10 depicts the overall accuracy at various confidence levels.

4.6 Discussion

This chapter presented a step towards a general service for locating traffic regulators to aid with various intelligent transportation applications. An important advantage of our approach lies in the fact that it does not require training and testing to occur in the same city. One can train our traffic regulator detector in one city and use it in several others. This generalizability is a key advantage that distinguishes this work from several prior approaches. The generalizability of our framework, however, hinges on the observations that guidelines for regulator placement are the same everywhere. Hence, once those are “reverse engineered” by the classifier, it is possible to apply the same rules elsewhere. Unfortunately, this may not hold across different countries. We tested our framework only within the US. Although sites

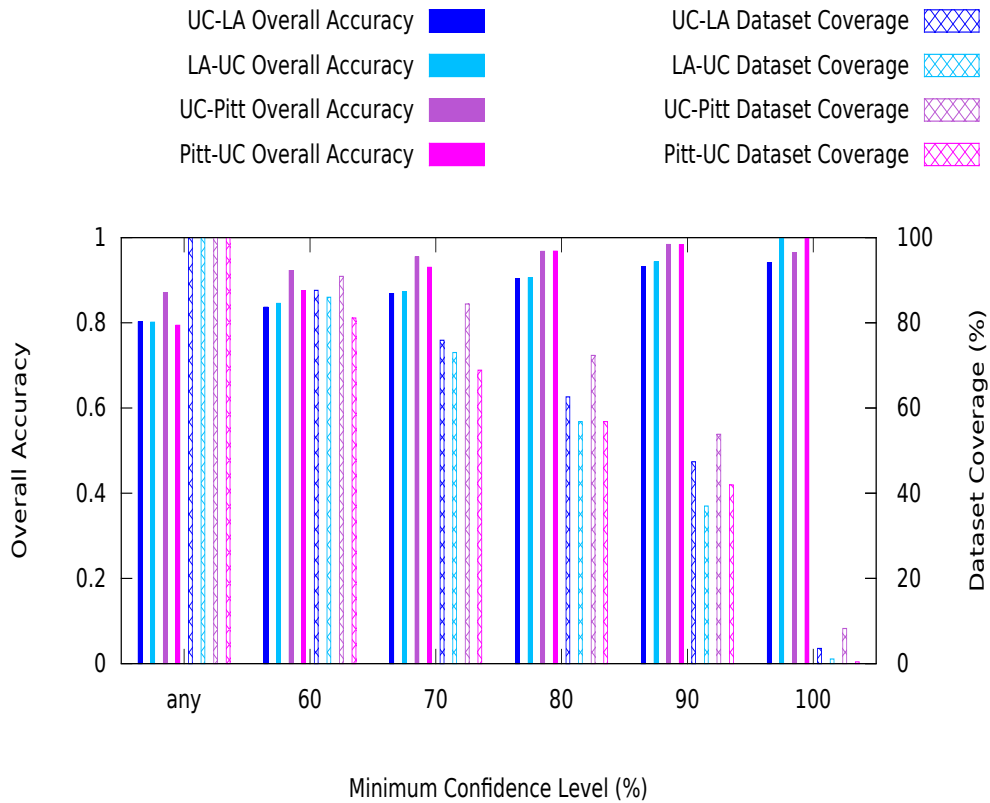


Figure 4.10: Cross-city detection accuracy at various confidence levels when training in Urbana-Champaign, IL (*UC*) and testing in Los Angeles, CA (*LA*) or Pittsburgh, PA (*Pitt*), and vice versa – notation *A-B* denotes training in *A* and testing in *B*

with different characteristics and types were selected (spanning small campus towns and larger cities), drawing general conclusions on the extensibility and cross-city applicability of the approach would need an evaluation outside the United States as well. It is likely that if we trained our classifier in a US city and applied it, say, in North Africa, we would have obtained significantly worse results. Indeed, one would need to train our method in each country in which it may be applied. We believe, however, that this is not a serious limitation and cross-city applicability should play its role inside each country or region whose traffic regulators placement policies are the same.

CHAPTER 5

PARTICIPATORY SLOW START

This chapter addresses the category of participatory sensing applications where sensing data is collected to *model* phenomena of interest. The constructed models are used in making future predictions, thereby constituting a service that benefits communities and/or individual users. An example of such a service is GreenGPS, where data collected from participants' vehicles is used to model their engines' fuel consumption, thereby predicting the amount of fuel used on different roads under different conditions. On top of that predictive model, a navigation service is built that can compute the most fuel-efficient route to a destination of choice.

In the early deployment stages of a new participatory sensing service adoption may be sparse. Thus, collected data may be restricted in quantity and distribution. We call this stage, the *slow start* stage [110]. Data sparsity creates difficulties getting the participatory sensing application off the ground, because the models built from such limited data may be unreliable.

To survive conditions of sparse deployment, a participatory sensing service must necessarily adopt simpler models in the beginning. The less complex models, though generally less accurate, are more reliable during slow start, because they need less training data. One must balance inaccuracies brought about by model (over)simplification, against those brought about by lack of reliability due to inadequate training data. While initially, simplification is unavoidable, the challenge becomes to find a good time to switch models,

when enough data is collected.

Let us consider GreenGPS again. To find the most fuel-efficient route, the fuel consumption of vehicles is modeled in terms of various relevant parameters. It is shown that the properly designed elaborate model leads to accurate fuel consumption predictions, and hence fulfills the demands of the GreenGPS navigation service (Chapter 7). However, in the early deployment stage, limited data impairs prediction accuracy, making the computed model unreliable. During this stage, one can utilize simpler models such as MPG-based approaches that use vehicles' published MPG ratings in order to approximately predict fuel consumption. These models are inaccurate (because they constitute average behavior, not performance on specific routes under specific conditions), but may initially do better when data is too sparse to train the elaborate models.

As *sufficient* data is collected, a transition from simple models to more elaborate models is made. The question is, when model transition should take place? If model transition is excessively postponed, the application may unnecessarily use less accurate models when it could have switched to better ones. However, if the switch is done prematurely, lack of sufficient training data may cause the more elaborate models to misbehave, generating even more error than the simpler ones. Therefore, it is important to carefully plan when the transition should take place.

The problem is not entirely straightforward. A person who travels mostly on freeways might collect a lot of data on freeway fuel consumption, but not enough downtown. Conversely, a person who only travels on downtown city streets, might not have enough observations on freeway data. Hence, to model that person's vehicle with adequate (guaranteed) reliability, one must consider not only how much data is collected but also the distribution of

collected data.

This chapter is arranged as follows. Section 5.1 describes the slow start transition problem. Section 5.2 elaborates on necessity of model transition during slow start. The model transition planning and theoretical analysis are presented in Section 5.3. Section 5.4 explains the impact of sensing data distribution on the model transition point, and describes how it is addressed. Section 5.5 addresses planning model transition when multiple modeling layers are employed. Finally, Section 5.6 evaluates the methodology using our collected participatory sensing dataset of GreenGPS.

5.1 Slow Start Transition

In participatory sensing applications, data of interest is collected by participants and commonly transferred to backend servers for analysis and modeling. In applications considered in this chapter, collected data forms the basis for building empirical models of a phenomenon of interest. The amount of collected data plays an important role in determining the constructed model reliability. In the early stages of launching an application, the collected data is limited, making reliability low.

We assume the existence of a simplified initial model and a full-fledged mature model (that needs more data collection to be reliable). Instead of planning for a specific data sensing and collection design, which is not feasible in most participatory sensing applications, we track the so far collected sensing data distribution, and decide at what point reliable transition to the full-fledged model can be made. In order to plan model transition, we take a quantitative approach and propose a technique that fits participatory sensing applications.

A typical and prevailing modeling technique is regression based modeling, that we consider here. Our goal in planning the model transition is the reliability of future predictions. The prediction error in regression modeling comes from two sources: disturbance or noise, and modeling error. The first type is inherent in the model and fundamentally cannot be reduced. Thus, we concentrate on the second source type, that is, modeling error or error in regression coefficients.

We propose a technique to address the problem of planning model transition in participatory sensing applications. The technique derives a reliable transition point and provides probabilistic guarantees on the ensuing modeling error bound. The resulted transition point can be plugged into the service implementation for making an automatic transition when appropriate. The transition point is derived based on Chebyshev's inequality and hence we call it "*Chebyshev's based*" derivation technique. In contrast to statistical quantitative approaches designed for computing the number of samples required in educational, behavioral and social sciences studies that take into account a single model coefficient at a time, our proposed approach is able to take all model coefficients simultaneously into account, and derive the transition point. Taking a holistic approach in planning model transition fits participatory sensing applications better. In addition, distribution imperfections prevailing in participatory sensing data is addressed in planning model transition.

5.2 GreenGPS Model Transition?

It is shown in Chapter 7 that the designed fuel consumption model in Equation 3.17 leads to sufficiently accurate predictions; however, during the slow

start phase it turns out to be quite inaccurate, due to the insufficient amount of driving data available. We therefore adopt simpler models initially, which tend to be more accurate and reliable during the slow start phase. For example, a simple MPG-based fuel prediction model can be utilized:

$$f_c = \frac{l}{mpg} \quad (5.1)$$

where l denotes the distance of a given trip and mpg denotes the mile per gallon (MPG) rating of the vehicle. For mpg , the officially published factory MPG ratings could be utilized, or an estimated value could be obtained from a set of collected data.

Figure 5.1 confirms the necessity and impact of model transition on the accuracy level of GreenGPS. The x axis shows the fraction of the dataset used (in percentage) and the y axis presents the average relative prediction error in percentage. The “*Full-fledged model*” denotes the predictive fuel consumption model offered by Equation 3.17 and the simple models carry out the fuel consumption prediction according to Equation 5.1. As for the mpg value, the first simple model uses a mile per gallon rating computed from the driving data collected for each vehicle, the second simple model uses the average of city MPG and highway MPG ratings for each vehicle, officially reported by car manufacturers, and the third simple model uses either city MPG rating or highway MPG rating, depending on the type of street segment the fuel consumption of which is to be predicted.

As reported in the figure, the full-fledged model suffers poor performance when data is limited, however, simple models experience a reasonable accuracy level during that period. Hence, initially simpler models must be employed, and then, as sufficient driving data is collected, a transition from

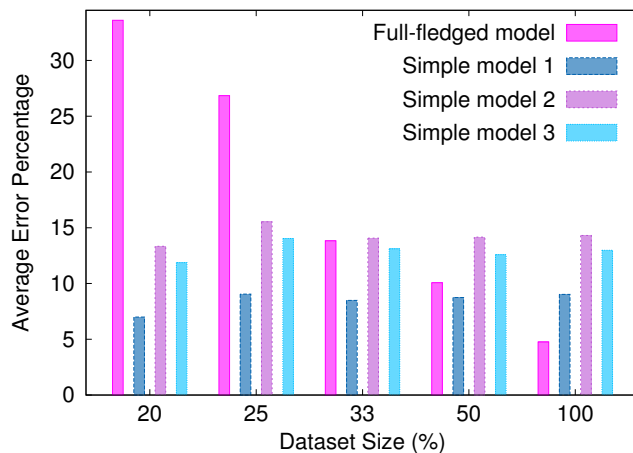


Figure 5.1: Impact of model transition on GreenGPS performance

MPG-based models to the full-fledged fuel consumption model is made. With transition planned at the appropriate time, the reliability of the service is improved.

5.3 Model Transition Planning

We assume the existence of two models; a simple one (e.g., rated-MPG based fuel consumption prediction) that can be used initially, and a full-fledged one (taking into account more attributes such as specific road conditions) that is more accurate but needs more training data to compute. During the slow start phase, which may be quite long, the simpler model is used. The full-fledged model can make better predictions, but only when the underlying training data becomes of sufficient size. A transition to the full-fledged model must be made when *sufficient* data is collected. Section 5.3.1 describes the planning design metric and Section 5.3.2 derives the model transition point, so that it is probabilistically guaranteed that the holistic modeling error is within a predetermined bound, defined and demanded by the application.

5.3.1 Design Criterion

A popular and common modeling technique is linear regression, which is extensively used as an analysis tool in various fields and applications. We design a transition planning approach for this general category of models. The design criterion is reducing prediction error by bounding the resulting modeling error.

Let a two-dimensional matrix X , of size $n \times p$, represent the set of n sensed data samples (e.g., road segments), each of p features. Let vector Y , of size $n \times 1$, represent the vector of outcomes (e.g., measured fuel consumption on each segment). The multiple linear regression can be written as $Y = f(X) + \epsilon = X\beta + \epsilon$, where vector β of size $p \times 1$ represents the vector of regression coefficients and ϵ represents a zero mean noise with variance σ^2 that is not correlated with X . The prediction is carried out as $\hat{Y} = \hat{f}(X) = X\hat{\beta}$, where \hat{Y} and $\hat{\beta}$ are estimations of the outcome variables and regression coefficients. The expected squared prediction error at data sample x can be written as $Err(x) = E[(y - \hat{y})^2]$, which can then be rewritten as:

$$Err(x) = E[(\hat{f}(x) - E[\hat{f}(x)])^2] + (f(x) - E[\hat{f}(x)])^2 + \sigma^2 \quad (5.2)$$

wherein the first term represents the variance of $\hat{f}(x)$ and the second term represents the squared bias of $\hat{f}(x)$ (which for an unbiased estimator such as least squares is zero). Hence, the prediction error in essence comes from two sources: the error in the regression model and the inherent noise. The second error type cannot fundamentally be controlled. Hence, we concern ourselves with modeling error only.

5.3.2 Transition Point Derivation

In this section, the model transition point in participatory sensing applications is derived, such that probabilistic guarantees are provided on the holistic model error bound to stay within an application defined threshold. We call our proposed approach the “*Chebyshev’s based*” derivation technique.

In order to derive a holistic error bound, we aim at confining the L_2 norm of the vector of regression coefficient error terms, namely $\|\hat{\beta} - \beta\|$. Notice that the probability of the L_2 norm of a vector being larger than or equal to a threshold ω is less than or equal to the probability of the absolute value of one of the vector components being larger than or equal to ω divided by the square root of the vector size. Applying this property to the vector of regression coefficient error terms yields the following inequality:

$$P(\|\hat{\beta} - \beta\| \geq \omega) \leq P\left(|\hat{\beta}_j - \beta_j| \geq \frac{\omega}{\sqrt{p}}\right) \quad (5.3)$$

By employing the least squares estimator for regression coefficients and considering that it is an unbiased estimator, that is, $E[\hat{\beta}] = \beta$, the right side of the Inequation 5.3 can be bounded by the use of Chebyshev’s inequality as:

$$P\left(|\hat{\beta}_j - \beta_j| \geq \frac{\omega}{\sqrt{p}}\right) \leq \frac{p}{\omega^2} \sigma_{\hat{\beta}_j}^2 \quad (5.4)$$

in which $\sigma_{\hat{\beta}_j}^2$ denotes the variance of $\hat{\beta}_j$. The variance of $\hat{\beta}$ is $\sigma_{\hat{\beta}}^2 = \sigma^2(X^T X)^{-1}$, and hence [111],

$$\sigma_{\hat{\beta}_j}^2 \leq \sigma^2 \lambda_{max}^{(X^T X)^{-1}} = \frac{\sigma^2}{\lambda_{min}^{X^T X}} \quad (5.5)$$

where λ_{max}^A and λ_{min}^A represent the largest and smallest eigenvalue of matrix

A , respectively.

Putting relations 5.3 to 5.5 together, the following inequality results:

$$P(\|\hat{\beta} - \beta\| \geq \omega) \leq \frac{p\sigma^2}{\omega^2\lambda_{min}^{X^T X}} \quad (5.6)$$

in which σ^2 , the variance of the model inherent error, can be estimated by $\frac{rss}{n-p}$, where rss denotes the residuals sum of squares (i.e. $\sum_i(\hat{y}_i - y_i)^2$).

In order to plan the model transition point, the following inequality shall be solved for the dataset of size n :

$$\frac{p \cdot rss}{\omega^2\lambda_{min}^{X^T X}(n-p)} \leq \rho \quad (5.7)$$

from which the smallest required sample size n_ρ must be obtained to ensure that, with $100(1-\rho)$ percent probability, the error bound is within threshold ω . This is done as follows:

$$n_\rho = p + \frac{p \cdot rss}{\rho\omega^2\lambda_{min}^{X^T X}} \quad (5.8)$$

In order to compute n_ρ for a participatory sensing application, rss and $\lambda_{min}^{X^T X}$ are obtained from the preliminary sensing data collected *so far* during the slow start phase. The transition point n_ρ is subsequently computed using Equation (5.8).

If the preliminary data used in planning the transition point properly covers the application's data space, then n_ρ truly denotes the point where reliable model transition can take place. However, if the application data space is not thoroughly covered (e.g., all collected data is from freeways, not downtown city streets), the derived value, n_ρ , would not represent a reliable transition point. The following section addresses this issue.

5.4 Data Distribution

The model transition point strongly depends on many factors including the size of the model and the data distribution. The preliminary sensing data used for computing the transition point may not thoroughly cover the application data space. Hence simply satisfying the requirement on transition point, posed by Equation (5.8), does not imply guarantees for the entire multi-dimensional data space.

To address that challenge, our data space is divided into application dependent subspaces. Within each, local models are constructed. The transition point is then computed for each subspace and the model transition in a given subspace is carried out when the corresponding transition point requirement is satisfied. This is performed to ensure collection of sufficient sensing data throughout the entire application data space rather than intensively covering some data regions and ignoring other regions.

A proper partitioning of the application data space insures fulfilling the goal of reasonably covering the entire data space, while preventing creation of unnecessary partitions. The partitioning is application-dependent and depends on the nature of the underlying nonlinearities. For example, for GreenGPS, a good way to partition data is by vehicle speed. Fuel consumption models at high speed differ from those at low speed. Hence, to fully model the vehicle, data must be collected for model construction within each of multiple speed ranges.

More specifically, the behavior for a typical vehicle is shown in Figure 5.2¹. With respect to this behavior, we partition the data space into regions: $v < 30$, $30 \leq v < 40$, $40 \leq v < 60$, and $v \geq 60$, where v denotes the average

¹<http://www.nrcan.gc.ca/energy/efficiency/transportation/cars-light-trucks/fuel-efficient-driving-techniques/7513>

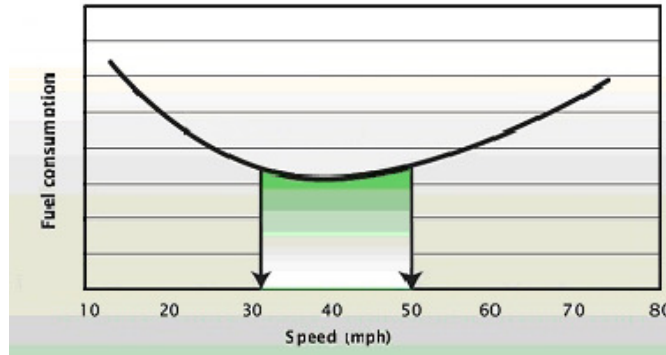


Figure 5.2: The typical behavior of fuel consumption for cars and light trucks with respect to speed

speed value. The transition point is then computed for each fuel consumption model constructed locally within a data region. In principle, the partitioning can be done along any dimension and for as many number of partitions as needed, however, creating excessive number of partitions may postpone model transition unnecessarily.

5.5 Multi-level Model Transition

The above discussion assumed only two models: an initial simple model and a full-fledged model. However, the slow start phase may, in general, take quite a long time. Hence, more than two models can be designed, featuring intermediate levels of accuracy and maturity. Multiple transitions can then be planned. For each transition point, the target model in the next layer should be considered in the analysis. Multiple transition points can thus be accommodated following the technique described above, applied separately to each model.

As an example, in the design of the GreenGPS application, there are two layers of models, besides the initial one; namely: an *individual training model* and a *cluster-based training model*. In the individual training model, in or-

der to model a given vehicle, data collected from only that vehicle is used. In contrast, in the cluster-based training model, data from other similar vehicles (e.g., vehicles of the same make and model) can also be utilized. The cluster-based training itself builds a hierarchy of models using different dataset aggregation strategies. Among the models devised, the individual training approach is the most accurate. However, it commonly takes quite long time until sufficient data is available for each and every individual vehicle. By grouping data from multiple vehicles, cluster-based techniques may accumulate enough data sooner. Planning multiple transition points assists GreenGPS in exploiting each model as soon as sufficient data is present.

5.6 Experimental Evaluation

This section evaluates the proposed transition planning technique. Section 5.6.1 presents the state of the art approaches that are compared with our proposed model transition methodology. Section 5.6.2 summarizes the implementation and the configuration of the experiments. The transition points for various configurations are planned using the proposed technique and are compared with other approaches in Section 5.6.3. Section 5.6.4 presents the service prediction error. Finally, Section 5.6.5 evaluates the role of sensing data distribution in model transition.

5.6.1 Approaches

The model transition planning approach proposed in this chapter is compared with two statistical approaches for determining sample size in social and behavioral sciences. A brief description of each approach follows in order to enable a proper comparison.

5.6.1.1 CI-based Derivation

We compare our proposed technique with a statistical state of the art approach for computing the required sample size in educational, behavioral and social sciences studies [69, 70]. The approach derives the required sample size using the confidence interval formula for a single regression coefficient and hence we call it *Confidence Interval (CI) based* technique. The CI-based approach captures the impact of a single regression coefficient at a time. In contrast, our approach is able to simultaneously take the impact of all the multiple regression coefficients into analysis.

The CI-based approach exploits the $100(1 - \alpha)$ percent confidence interval for a single regression coefficient β_j to find the required sample size with $100(1 - \rho)$ percent assurance that the regression coefficient estimation $\hat{\beta}_j$ will be within a specified width w from its true value β_j :

$$n_\rho = \left(\frac{\chi_{1-\rho, n-1}^2}{n-p} \right) \left(\frac{z_{1-\alpha/2}}{w} \right)^2 \left(\frac{1-R^2}{1-R_{X_j X_{-j}}^2} \right) + p \quad (5.9)$$

where $\chi_{1-\rho, n-1}^2$ is the $1 - \rho$ quantile of a chi-squared distribution with $n - 1$ degrees of freedom, $z_{1-\alpha/2}$ is the $1 - \alpha/2$ quantile of a standard normal distribution, R^2 denotes the multiple correlation coefficient, $R_{X_j X_{-j}}^2$ denotes the multiple correlation coefficient when X_j is regressed on the remaining predictors (denoted X_{-j}), and n is computed as:

$$n = \left(\frac{z_{1-\alpha/2}}{w} \right)^2 \left(\frac{1-R^2}{1-R_{X_j X_{-j}}^2} \right) + p \quad (5.10)$$

The CI-based formula demands for the knowledge of multiple correlation coefficients to compute the required sample size. We use the preliminary data to obtain the demanded parameters. Note that width w above corresponds to the specified bound for a single regression coefficient while width ω in

Equation 5.8 concerns the holistic model error bound.

In order to compare the required sample sizes computed by the CI-based approach with the model transition points computed by our proposed (Chebyshev's based derivation) technique, we take and use the largest sample size among the CI-based values obtained in regard with each single regression coefficient. We *emphasize* that doing so *does not* provide the promised guarantees for single regression coefficients simultaneously for all of them together, as the CI-based approach has not been designed for that purpose. This is in contrast to our Chebyshev's based derivation technique that provides holistic guarantees and more appropriately fits participatory sensing applications.

5.6.1.2 Monte Carlo Simulation

A different vein than providing theoretical guarantees is to acquire the model transition point through simulation studies [71, 72]. *Monte Carlo Simulation* is a technique that puts its foundation in the law of large numbers and thus approaches problems via generation of a large number of data samples randomly drawn from a population with hypothesized parameters. According to the law, the empirical mean of the parameter of interest approaches the expected value of the parameter as the number of trials increases. For a large number of repetitions, Monte Carlo can achieve a stable estimation of the target parameter.

The number of data samples required before making the model transition can be obtained using Monte Carlo simulations. To this end, regression model, regression coefficients, variables correlations, and multiple correlation coefficients should be determined. We obtain these parameters from the preliminary collected data.

Given the model and the population parameters, a tentative number of

samples n is randomly drawn for a large number of times m , and the average confidence interval width is computed for each regression coefficient across the m repetitions. The number of samples n is changed and the simulation is repeated until a minimum dataset size n_ρ is found for which the half width confidence interval empirical mean is less than w for all the regression coefficients in $100(1 - \rho)$ percent of the m repetitions. The procedure can be repeated for different random seed numbers as needed to achieve stable results.

5.6.2 Experiment Setup

For the implementation of the approaches we used the statistical tool R [112]. In order to investigate the impact of the preliminary collected dataset size on the computed transition point, we draw subsets of our collected participatory sensing dataset with gradual increase in their size. Specifically, initially 20 sample trips are randomly selected to form a trip set of size 20. To create a trip set of size 40, the first 20 trips are retained and another 20 data samples are randomly selected from the original dataset excluding the 20 previously drawn samples. That is, each larger size trip set subsumes the smaller sized sets. Larger sized trip sets are created in the same manner (e.g. 60, 80, 100, 200, etc.). The procedure is iterated 100 times, that is, 100 different sets are created for each trip set size.

In the experiments, the level of assurance for the transition point planning approaches is set to 95% ($\rho = 0.05$). The confidence level in the Confidence Interval Based approach is set to 95% ($\alpha = 0.05$), and the number of repetitions in the Monte Carlo Simulation is set to 500 (m) and confirmed to be sufficiently large. It was verified that using a single seed number was enough

to result in stable results.

The width threshold w is chosen to be 0.15 for the standardized regression coefficients in the simulation study, and equivalently $0.15 \times \frac{\sigma_Y}{\sigma_{\beta_j}}$ for the unstandardized regression coefficients in CI-based approach. For Chebyshev’s based derivation technique that targets holistic regression model error, simultaneously considering all the regression coefficients, the width threshold ω is considered to be the L_2 norm of a vector with p elements, each element being equal to the threshold value for an unstandardized regression coefficient.

5.6.3 Model Transition Point

The impact of the preliminary dataset size and the model size on the computed transition point using Chebyshev’s based approach, CI-based approach, and Monte Carlo simulations is reflected in Figure 5.3. The results for fuel consumption prediction models of size 12 parameters and 4 parameters are shown in Figure 5.3a and Figure 5.3b, respectively.

GreenGPS application is deployed in a flat region (the area of Urbana-Champaign, IL) and hence the fifth component in the fuel consumption model presented in Equation 3.17 is zero ($\sin(\theta_i) = 0$). In addition, in the experiments, a trip is defined to start from when a vehicle’s engine is turned on until the time it is turned off. Hence the speed of the vehicle at the beginning and at the end of the trips is zero, resulting in the sixth component in the fuel consumption model being zero. The resulting model thus has 12 components or equivalently 12 regression coefficients and is referred to as the “12-parameter model”. A different sized model is also considered, consisting of the first four components in Equation 3.17, and is referred to as the “4-parameter model”.

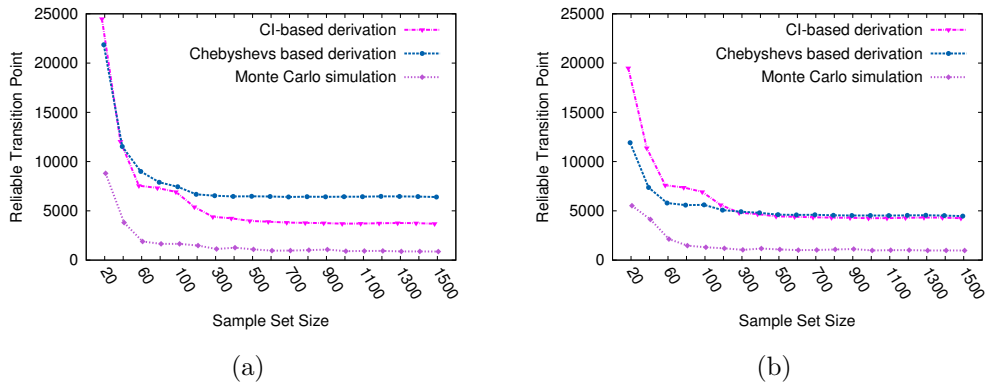


Figure 5.3: The number of samples required for reliable model transition for: (a) model with 12 parameters; (b) model with 4 parameters

The x axes in the figures show the preliminary sample sets of various sizes. It can be observed that for the three approaches and both model sizes, the computed transition points stabilize at about 500 sample set size. This implies that small preliminary participatory sensing datasets would suffice to lead to stable computations of the transition points.

To provide probabilistic guarantees on the holistic model error bound, a smaller transition point is deemed to realize for models with fewer parameters which is confirmed for Chebyshev's based approach. However, for CI-based approach that focuses on a single regression coefficient, it is not the case. Instead the opposite may occur due to the presence of $1 - R^2$ in the numerator of the formula in equation 5.9, which tends to be larger for models with fewer parameters.

The transition points reported in the figure show smaller numbers for CI-based approach compared to Chebyshev's based approach. However, note that the reported values for CI-based approach do not imply the same guarantees as provided with Chebyshev's based approach. As explained before, Chebyshev's based reported values indicate a point at which probabilistic guarantees are provisioned with the holistic model error, but the CI-based

reported values imply a weaker guarantee concerning a single regression parameter. Monte Carlo simulation-based reported values are smaller in comparison, rooting in the fact that provision of theoretical guarantees comes at a cost: larger transition points are resulted with the theoretical approaches. It is worth noticing that the simulation-based approach takes longer in computation time compared to the theoretic approaches.

Figure 5.4 investigate the impact of dataset characteristics on the computed transition point. To this end, trips in the original dataset are decomposed into shorter trips, and new datasets of approximately the same trip length (i.e. 0.5, 1, 1.5 and 2 miles) are constructed. As a result, 7000 sample trips of 0.5 mile length, 4000 sample trips of 1 mile length, 3000 sample trips of 1.5 mile length, and 2500 sample trips of 2 mile length are generated, the results for which are presented in Figure 5.4a, Figure 5.4b, Figure 5.4c, and Figure 5.4d, respectively. Thus the constructed datasets are different from the original dataset in the characteristic modeling attribute of *trip distance*. Following the same procedure as mentioned above, preliminary datasets of different sizes are formed for each trip set category (0.5, 1, 1.5 and 2 mile trip sets) and for each preliminary dataset the transition points are computed using the three approaches.

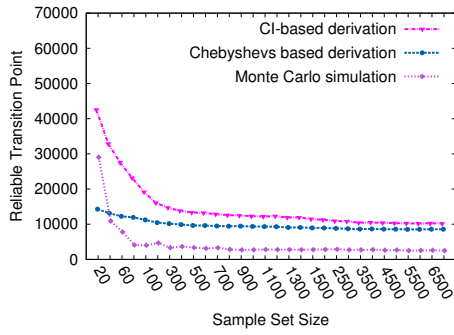
It is evident that compared to the original dataset a significant difference in the transition points is resulted. It can be observed that for longer trips the number of required data samples is smaller. The reason lies in the large variability of model attributes for shorter trips and also that for longer trips the errors tend to be aggregated and canceled out. Hence, for longer trips a smaller number of data samples is required to reach the same level of assurance and reliability.

Figure 5.5 plots the impact of the width threshold on the computed reliable transition point. The threshold for a single standardized regression coefficient is varied from 0.10 to 0.30². The figure presents the results for Chebyshev’s based derivation approach and the preliminary sample set size equal to 500. Figure 5.5a presents the results for different model sizes and Figure 5.5b presents the results for constructed datasets with varying trips length. As expected, the more restricted the error bound, the larger the reliable transition point throughout different model sizes and varying trip sets length. The same trend remains true for the other two approaches, hence their results are not reported here.

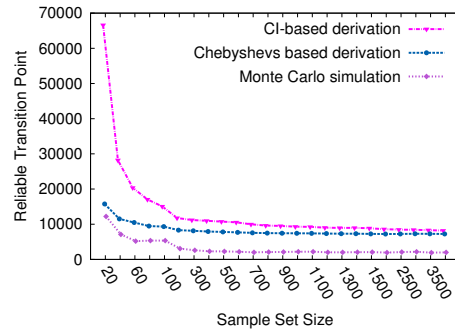
5.6.4 Service Prediction Accuracy

Figure 5.6 depicts the impact of increasing sample set size on service prediction error. Figure 5.6a presents the impact for different model sizes and Figure 5.6b shows that for datasets of varying trip set length. The sample set size is denoted on axis x and the root mean square error (RMSE) is denoted on axis y . The error is computed using the leave-one-out cross validation technique in the following manner. For a given trip, a model is constructed using other trips in the dataset, which is then used for predicting the fuel consumption of the trip and consequently obtaining the trip’s prediction squared error. The root mean square error across all trips is subsequently determined. The error is then scaled and divided by the average fuel consumption across all trips in the dataset (normalized). It can be seen that throughout all configurations the prediction error improves by increasing the number of data samples utilized for construction of the models.

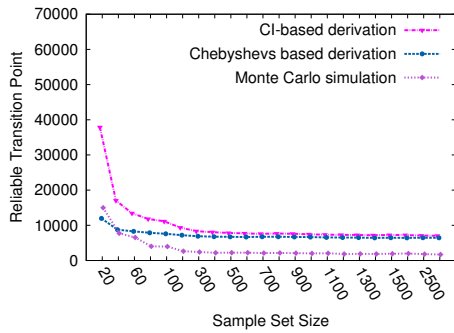
²The threshold is scaled for an unstandardized regression coefficient and the vector of regression coefficients accordingly as described previously.



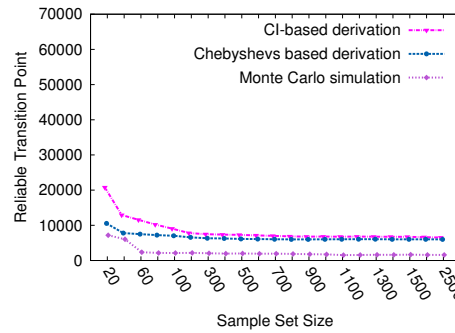
(a)



(b)

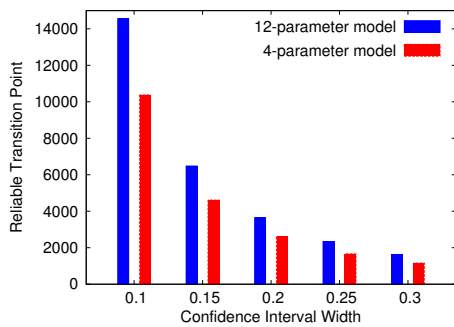


(c)

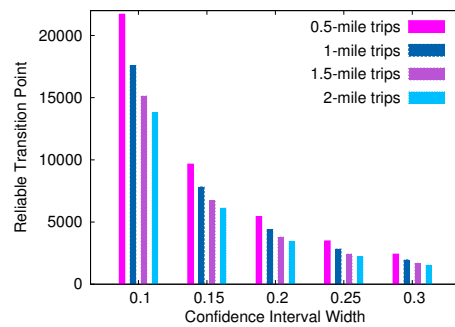


(d)

Figure 5.4: The number of samples required for reliable model transition for: (a) trip set with 0.5 mile long trips; (b) trip set with 1 mile long trips; (c) trip set with 1.5 mile long trips; (d) trip set with 2 mile long trips



(a)



(b)

Figure 5.5: Impact of the width threshold on transition points for: (a) different model sizes; (b) datasets of varying trip set length

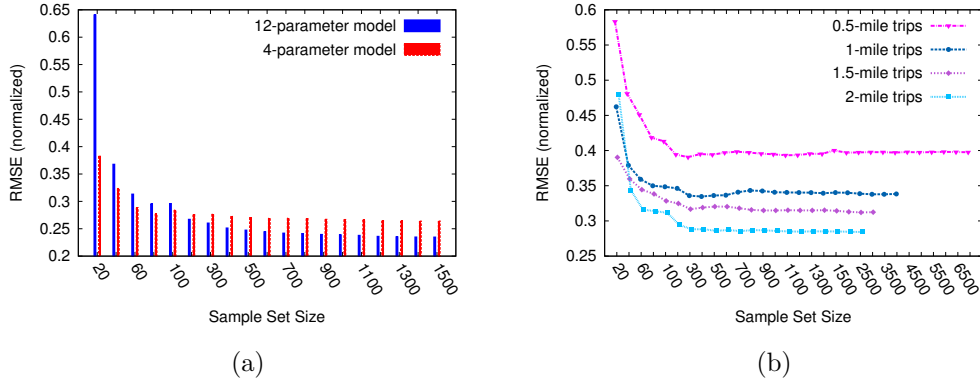


Figure 5.6: (a) Prediction error for models with different number of parameters; (b) Prediction error for sets of trips with varying length

5.6.5 Sensing Data Distribution

This section evaluates the sensing data distribution impact. To address that, GreenGPS application domain space is partitioned into subspaces along the vehicle speed dimension, namely, subspaces with the average trip speed value $v < 30$, $30 \leq v < 40$, $40 \leq v < 60$, and $v > 60$. Then the reliable transition point is computed for the dataset in each region using the proposed approach Chebyshev's based derivation technique.

The computed transition points for the original dataset and for those of the partitioned regions are depicted in Figure 5.7a. Our collected dataset did not contain any trips with average trip speed beyond 60 mile per hour; hence the corresponding region is not reported in the figures. The figure indicates that in order to reliably adopt the locally constructed elaborate models within each subspace, how many data samples from each region are required. Note that, the total number of samples required for all regions is above that of the original dataset for the application domain. Besides, it is now imposed that every specified number of samples should come from a particular region, insuring a demanded proper coverage of the application domain and the reliability of the local model transitions.

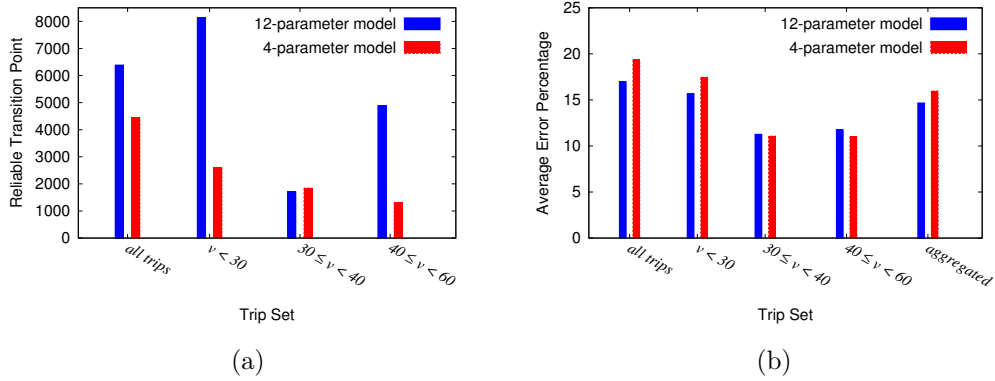


Figure 5.7: Impact of participatory sensing data distribution on: (a) reliable model transition point; (b) prediction error

Figure 5.7b presents the average relative error percentage for fuel prediction of the original dataset and individual local datasets from the partitioned regions. The trips' prediction errors are computed using the leave-one-out cross validation technique. The figure also shows the error for the aggregated dataset, which is equivalent to the original dataset but each trip's fuel consumption is predicted using a locally constructed model for the corresponding region. As it can be observed, local models improve the prediction accuracy of the service. Specifically, about 14% and 18% improvement in the accuracy level is achieved for the 12 and 4 parameter models, respectively.

CHAPTER 6

ARCHITECTURE AND IMPLEMENTATION

This chapter presents the architecture and implementation of the fuel-efficient navigation system GreenGPS. Section 6.1 describes the system architecture, Section 6.2 presents the system participatory sensing platform, and Section 6.3 presents the lessons learned.

6.1 GreenGPS System Architecture

The GreenGPS server combines several developed software modules and open source software services to provide the fuel-efficient route computation service. The various modules that are part of the GreenGPS implementation are depicted in Figure 6.1. The GreenGPS source code is available in [113].

6.1.1 Data Collection

We implement the user-facing participatory sensing module as an Android application in Java that runs on users' smartphones. This application gathers fuel consumption and speed information data from the car's engine, combines that with location data gathered using phone's GPS, and opportunistically uploads the data to the backend aggregation server.

Further details about the implementation are presented in Section 6.2.

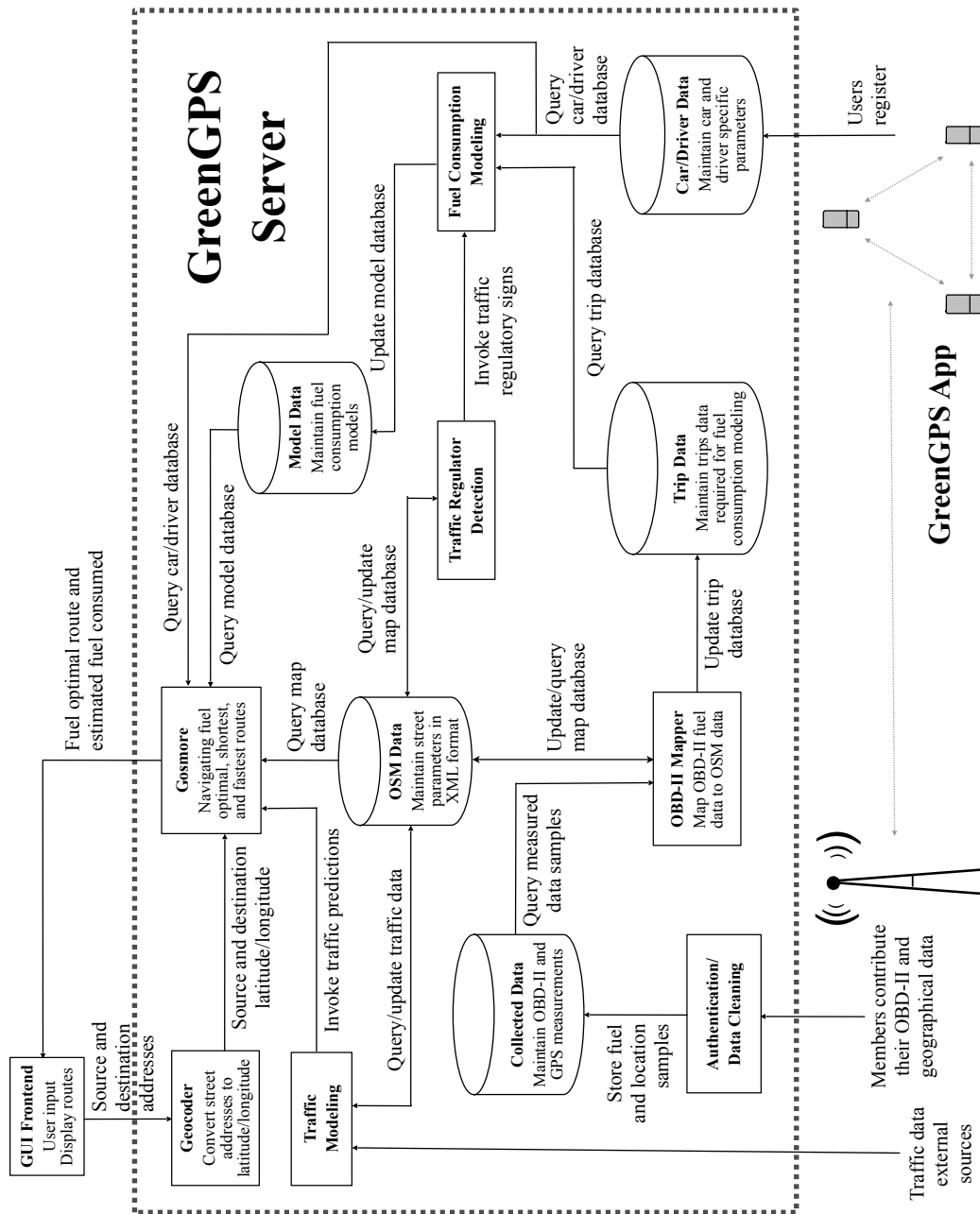


Figure 6.1: GreenGPS system architecture

6.1.2 Modeling and Generalization

The OBD-II data shared by individuals is used to compute regression models that predict the fuel consumption on specific streets given the car details (e.g. make, model, age, category). The regression variables are stored in the Trip Database, whereas the car and driver specific variables are stored in a similar database. The modeling module queries the database and constructs the prediction models which are stored in the Model Database. The Model Database is later queried by the navigation engine in order to compute fuel consumption on a given way for a given car and driver.

Each trip is organized as a row in the database where 14 of its attributes are the values of the physical model parameters in Equation 3.17 and are used for regression. Four other attributes (Make, Model, Year, Class) are used for grouping. After computing the regression models for all clusters, search for a specific 4-tuple of (Make, Model, Year, Class) is done according to the optimal generalization in Figure 3.5. The first regression model that matches the query is used for prediction.

6.1.3 Detection of Traffic Signs Location

To implement the traffic signs location detection module, we built our Random Forest based classifier using the “randomForest” package [109] in the statistical tool *R*. The classifier was trained using a dataset collected in part of the city of Los Angeles and used to predict the traffic signs at each intersection in the area of Urbana-Champaign, needed for evaluating the performance of GreenGPS in Chapter 7.

6.1.4 Navigation

GreenGPS maintains the map of a given area as an OSM map. Navigation is achieved in GreenGPS by customizing the open source routing software, Gosmore [114]. Gosmore is a C++ based implementation of a generic routing algorithm that provides shortest and fastest routes between two arbitrary end-points. Gosmore uses OSM XML map data for doing routing. Gosmore’s routing algorithm, A*, by default computes the shortest route. This routing algorithm works on the OSM map, where the nodes of the graph are OSM nodes and the edges of the graph are OSM ways and the weights of the edges are the lengths (distance) of the ways. The fastest route is then computed by multiplying the distance by an inverse speed factor (thus giving lower weights to faster ways). Our fuel-optimal routing algorithm multiplies the distance by an inverse mpg metric that results in lower weights for fuel-optimal ways.

6.2 A Participatory Sensing System for Data Collection

In this section, we present the participatory sensing framework that we utilize for data collection and sharing. We implement a client-side interface for data collection that automatically uploads all data to a central server, called GreenGPS *aggregation server*. An individual who wishes to share their OBD-II sensor and location data simply downloads our client-side software, publicly available as an Android application on Google Play Store, and uses it to automatically upload their data to the aggregation server. The aggregation server uses the data to calibrate models that relate street and vehicle parameters to fuel-efficiency and offers the GreenGPS navigation interface for fuel-efficient routes.

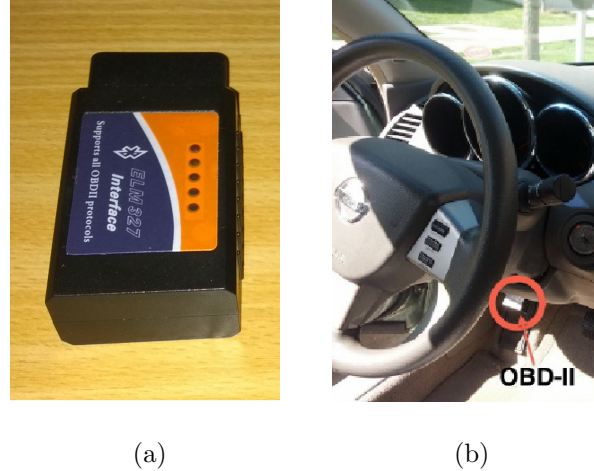


Figure 6.2: (a) OBD-II to bluetooth adaptor; (b) Adaptor deployed in a car

Individuals who wish to contribute OBD-II data to GreenGPS, install an off-the-shelf and inexpensive OBD-II to bluetooth adapter in their vehicle (Figure 6.2). The GreenGPS phone application communicates with the vehicle OBD-II via bluetooth to obtain the engine fuel consumption data. The data is then timestamped and stored in a small database on the phone. The parameters obtained from the car and the GPS sensor on the phone include instantaneous vehicle speed, mass air flow, command equivalence ratio, engine rpm, throttle position, latitude, longitude, altitude, bearing, time and phone IMEI.

6.2.1 OBD-II Communication

We sample fuel parameters from the OBD-II unit using the OBD-II to bluetooth adaptor. The key parameters, namely mass air flow, speed, command equivalence ratio, engine rpm, and throttle position are queried in sequential order. The sequential sampling provides better overall response rate as we discovered that frequently querying the OBD-II for all the parameters (at the same time) resulted in response gaps. For example, for the majority of the

vehicles, if we query for all five parameters at once, the likelihood of receiving all five responses before reaching our timeout is low. However, if we query for parameter values one at a time, the likelihood of all values being present is very high.

The sampling is ordered in the sequence described above to minimize the timing differences when calculating fuel rate and fuel economy. Since we only calculate two fields, we try to group the sampling parameters together so that the values used for fuel equations are closer in time.

- (a) Fuel Rate uses 2 queries, mass air flow (MAF) and command equivalence ratio (EQV), and is calculated in gallons per second as,

$$FuelRate = \frac{MAF}{(14.7 \times EQV) \times 454.0 \times 6.17} \quad (6.1)$$

wherein MAF is in grams per second, 14.7 is grams of air to 1 gram of gasoline (ideal air to fuel ratio), $|EQV| \leq 1$, 454.0 is grams per pound, and 6.17 is pounds per gallon of gasoline.

- (b) Fuel Economy needs 3 queries, MAF, EQV, and vehicle speed (VSS), and is calculated in miles per gallon as,

$$FuelEconomy = \frac{(14.7 \times EQV) \times 454.0 \times 6.17}{MAF} \times \frac{VSS \times 0.621371}{3600} \quad (6.2)$$

wherein VSS is in kilometers per hour, 0.621371 is kilometers per hour to miles per hour conversion ratio, and 3600 is seconds per hour.

The engine rpm and throttle position are collected for future uses.

We try to generate samples as quickly as possible, however, the sampling rate is not constant across all vehicles. More specifically, the sampling rate

varies with the OBD protocol being used, the age of the vehicle and its OBD-II unit, and the version of the OBD-II to bluetooth adaptor (newer models support higher data transfer rates), please see [88] page 61 for more details.

6.2.2 Opportunistic Uploading

One of the design goals of the GreenGPS's participatory sensing framework was to eliminate the need for cellular data connections for data collection. This helps to avoid imposing communication overhead of data collection on users, for which they may be reluctant to use their own data plans (as opposed to the route navigation step that they experience immediate benefit return and would be willing to utilize their cellular data connections). This extends the reach and availability of users who either do not have cellular data connections or prefer not to use it for GreenGPS. Achieving this design goal imposes several practical requirements. In particular, collected data must be stored on the phone in persistent storage, the amount of data stored has to be minimized, data has to be appropriately compressed for storage and transmission, and the backend has to be optimized to quickly absorb large volumes of data being rapidly uploaded. The vehicles in our testbed at the University campus presented DTN-like mobility patterns. Because individual devices had a low probability of coming into contact with the wireless access points located around campus, we embraced the notion of opportunistic uploading.

We begin by storing generated samples in a small database on the phone. Once our application sends its samples to the data storage server, it clears out the delivered samples to free up resources within the database. We reduced the amount of characters per transfer by replacing parameter names

with numeric constants. The numeric representations (e.g. $2 = OBDSpeed$) provide easy decoding of messages as well as save storage space on the mobile devices. On the back-end, we speed up data entry and lookups using unique composite indexing on the tables. Multi-column indexes provide an additional use in that we can filter out duplicate samples as we insert them into the database.

6.2.3 Collaborative Uploading

GreenGPS provides collaborative uploading capability for the purposes of deployment on a fleet of vehicles as well. We assume that users typically do not want the mobile sensing applications to use their 3G communication for altruistic raw data upload to the server, since unlimited data plans are not prevalent.

On the other hand, WiFi based store and forward of sensed data may result in a large latency, which motivates optimizing data transfer among vehicles and between vehicles and the infrastructure for faster offloading. Current communication techniques on smartphones that support peer-to-peer sharing, such as WiFi ad-hoc mode and WiFi Direct, have significant limitations and are not directly usable for mobile sensing. WiFi ad-hoc is not supported on most popular phones unless rooted or jailbroken and will probably not be in the near future due to economic and political issues. WiFi Direct was not designed with opportunistic networking in mind, but tries to connect WiFi enabled devices such as printers and cameras in a secure way and as easily as possible. User involvement is mandatory for WiFi Direct for security reasons.

In contrast, we utilize a WiFi hotspot switching approach that is com-

patible with existing access points and does not need to root or jailbreak smartphones. Two phones can establish connections when one of them is in the hotspot mode and the other in the peer mode, and a phone can offload data to access points when in the peer mode. Moreover, it requires neither involvement of participants nor changes to existing wireless infrastructure and protocols.

The collaborative uploading component in GreenGPS works as follows. After a smartphone joins the vehicle network, it enters the peer mode, in which it searches for available communication opportunities. Here, the opportunity refers to either another phone in the hotspot mode or a WiFi access point. Meanwhile, a timer starts to record how long it has stayed in the peer mode. When the timer expires, the phone enters the hotspot mode, turns itself into a WiFi hotspot, and continues looking for communication opportunities. Now, the opportunity means another phone in the peer mode. Similarly, a second timer is started, and the phone goes back to the peer mode when the timer expires. When the phone is in peer or hotspot modes and a communication opportunity appears, the corresponding timer is paused, the phone goes to the transfer mode, and starts exchanging data with the other phone or offloading data to the backend server. When the communication is terminated because the transmission is finished or wireless connection is out of range, the phone goes back to its last mode before this communication, and the timer is resumed.

6.2.4 Energy Management

Designed for installment on a fleet of vehicles as well, the GreenGPS participatory sensing application is able to function in conditions where the phone

is left in a car for multiple days. This design choice was made in part to accommodate fleet users that contribute to GreenGPS using a phone other than their primary phone. Examples include the fleet of UIUC Facilities and Services and our case study volunteer participants. These phones will be connected to the charger, however, their battery will not be charged when the vehicle is turned off. Therefore, seamless energy management is important, particularly when a car is not started for multiple days (as it frequently happens with some of the University Facilities and Services fleet).

In view of the above requirements, we developed the Energy Management component of GreenGPS as follows. We monitor the battery status every six seconds and turn off GPS, Bluetooth, and Wifi when these components are not needed. When we detect that the phone is not charging, we immediately shut down any sensors and force the Bluetooth and Wifi communication threads into a clean exit. Regardless of whether or not we can detect the OBD-II unit (which may or may not be detected when the car is turned off), we cannot tolerate battery drain in the battery discharging state.

When we detect the device to be charging, there is no immediate concern for the battery to last. In this state, if disconnected, we turn on Bluetooth and attempt to connect to the OBD-II unit every three minutes. Success results in turning on all the sensors and networking components for a newly-labeled car trip. Failure turns Bluetooth off, until the next attempt in three minutes. We tolerate potential data loss to reduce car battery drain, since our device only charges by car battery.

6.2.5 Collected Data

We conducted a study involving 22 users (with different cars) over the course of several months.¹ A total of over 3200 miles was driven by our users to construct the initial models. Figure 6.3 shows a partial map of the paths on which data was collected. The details of the car make, model, year, class, and the number of miles of data collected for each car are summarized in Table 6.1. The distribution for the trips distance is depicted in Figure 6.4a. It can be observed that the majority of the trips are very short. In particular, about 70% of the trips are less than 4 miles long and the remaining 30% are from 4 to 10 miles long. The speed distribution for various one-mile road segments driven is plotted in Figure 6.4b and represents a mixture of two normal distributions. The distribution denotes that most of the road segments are low speed (less than 45 miles per hour) and that is due to the type of streets in the town in which exist only few highways. Figure 6.4c presents the average number of stop signs, traffic lights, left turns and right turns per one-mile road segments with respect to the distance of the trips. It is denoted that, as path length increases, the average number of stop signs per segment shows an overall decreasing trend while the average number of traffic lights, left turns and right turns do not exhibit such overall trend change. This is expected considering that short trips are mostly the ones driven in campus and in low speed streets that an intersection appears almost at every block.

¹Data collection involving human subjects were approved by UIUC IRB (protocol number 10092).

Table 6.1: The vehicle set used and the amount of data collected

Car Make	Car Model	Car Year	Car Class	City MPG	Hwy MPG	Miles Driven
Toyota	Camry	2004	Mid-Size	24	33	80
Chevrolet	Impala	2002	Large	21	32	69
Ford	Ranger	2008	Van	15	19	29
Toyota	Corolla	2000	Compact	31	38	259
Buick	LeSabre	2002	Large	20	29	54
Ford	E-250	2011	Van	13	17	99
Toyota	Corolla	2010	Compact	26	35	53
Toyota	Celica	2001	Sub-Compact	28	34	497
Nissan	Altima	2006	Compact	24	31	95
Subaru	Impreza	2010	Sub-Compact	19	24	26
Toyota	Corolla	2004	Compact	32	40	141
Mazda	Mazda6	2003	Mid-Size	23	29	62
Audi	A4	2005	Compact	22	31	88
Toyota	Camry	2012	Mid-Size	25	35	90
Subaru	Impreza	2010	Sub-Compact	19	24	69
Hyundai	Santa-Fe	2001	Sport-Utility	21	28	87
Ford	Taurus	2002	Mid-Size	20	28	65
Mitsubishi	Eclipse	2002	Sub-Compact	23	30	184
Nissan	Altima	2010	Mid-Size	23	32	103
Mitsubishi	Galant	2002	Mid-Size	21	28	112
Toyota	Celica	2000	Compact	28	34	882
Toyota	Camry	2004	Mid-Size	24	33	57

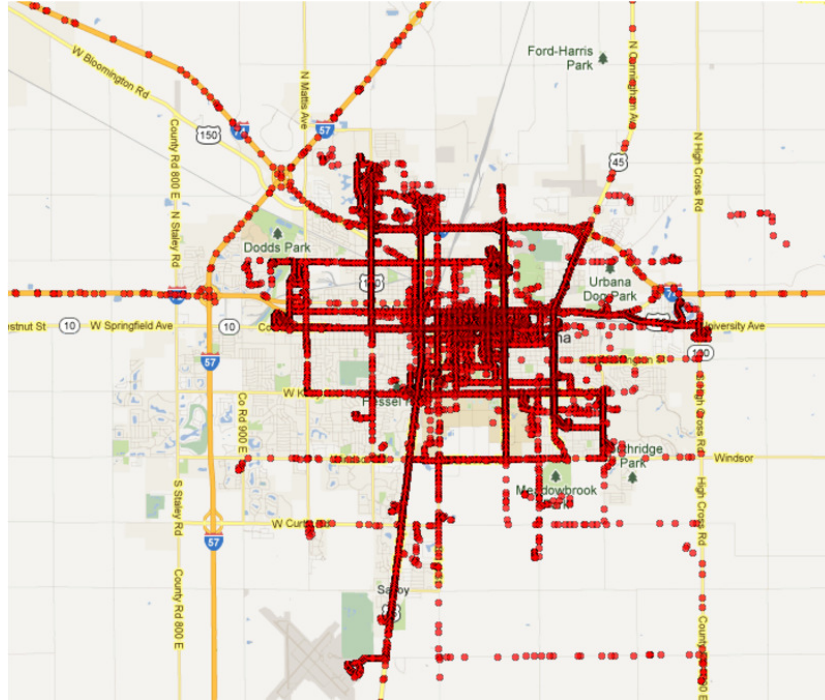


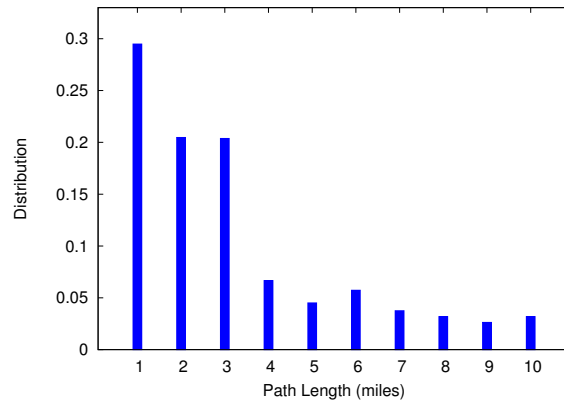
Figure 6.3: Coverage map for the paths on which data were collected

6.3 Discussion

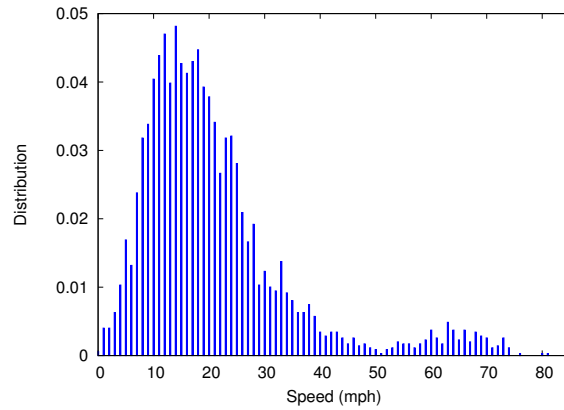
This section presents a brief discussion of lessons learned and experiences with the GreenGPS service and its components, as a participatory sensing application using a mobile platform.

Data Cleaning: We observed that data cleaning is an important problem and it is application dependent. We had several occasions when several fields were missing from the data (e.g., some OBD parameters were empty due to timing subtleties). A simple scheme was used to filter complete datasets from those that were missing values.

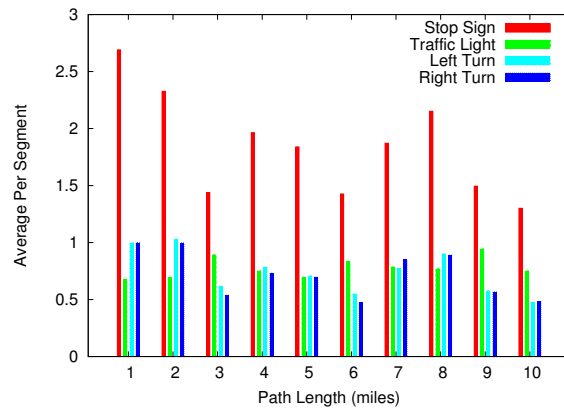
Heterogeneity: An application-specific challenge was observed due to the variations in the OBD-II standards among different cars. It was experienced that some car manufacturers use non-standard OBD-II parameter identifiers



(a)



(b)



(c)

Figure 6.4: The distribution of trip data collected from all cars: (a) The path distance distribution; (b) The average speed distribution; (c) The average number of stop signs, traffic lights, left turns and right turns per one-mile road segments with respect to the distance of the trips

(PIDs). A few such examples we encountered in our initial deployment include Honda Civic 2004, Honda Accord 2005 and General Motors Sonoma 2002. As a result we had to discard data from those vehicles due to missing fuel parameters. This suggests that participatory sensing applications involve a large number of heterogeneous components (e.g., different car types in GreenGPS) that one should take into account and resolve before scaled deployments.

Utility of Generalization: The utility of the generalization methodology described in Chapter 3 is not compromised by the increasing prevalence of fuel-efficiency measurements in modern cars. This is because modern cars measure fuel efficiency on routes they traverse. Cars do not predict fuel efficiency before route traversal. Hence, the only way drivers can compare gas consumption on different routes at present would be to drive all of them and compare results. In contrast, GreenGPS predicts the final answer without the driving. The contribution here is thus complementary to (and *not* subsumed by) affordances offered in modern vehicles.

Privacy: In participatory sensing systems, privacy challenges come to the forefront. A large class of participatory sensing systems monitor location information continuously, which poses significant privacy issues. Simple anonymization of data will not work in such situations, as the GPS traces can lead to privacy breaches (e.g., reveal the home location of the user and thus uncover their identity). Techniques such as the one proposed in [115] can be used to preserve privacy, while still allowing accurate modeling. In [115], measurement samples are first integrated into, so called, *segments* in order to remove correlation. The uncorrelated segments are then converted into some *neutral features* appropriate to be used in modeling the phenomena

(vehicles fuel consumption) while preserving the users privacy. The privacy preserving methodology has been applied to our green navigation service as a case study in the paper. In the current study, individual users simply switch off data collection application when they feel the need for privacy. The latter is simple and fast, however, the participatory sensing service employing it may be permitted for gathering data only intermittently. Nevertheless, the former approach and data perturbation-based approaches such as [116] and [117] enable perpetual privacy-preserving data collection for a reasonable extra computation cost.

Long Term Investment: As expected, the main factors affecting fuel consumption of a vehicle on a path are the average speed, the speed variability (estimated by averaging the speed squared), and the engine idle time (estimated from the number of stop signs, traffic lights and turns on the path). Rather than exploring the use of real-time traffic conditions, we opted to use statistical averages of speed, speed variability and idle time. It is easy to see how such statistical averages can be computed for different hours of the day and different days of the week given a sufficient amount of historical data, yielding expected fuel consumption (in the statistical sense of expectation). The outcome is that individual trips may differ significantly from the statistical expectation. However, by consistently following routes that have a lower expected fuel consumption, savings will accumulate in the long term. Drivers may think of GreenGPS as a long-term investment. Short-term results may vary, but long-term expectations should tend to come true.

CHAPTER 7

SERVICE EVALUATION AND IMPACT

This chapter evaluates the performance of GreenGPS and describes the application products developed in order to assist individuals to find the most fuel-efficient routes. The green navigation performance is evaluated in three stages. First, the performance of the GreenGPS fuel consumption model is evaluated by using it to predict the end-to-end fuel consumption for long routes. Second, the impact of the traffic regulators detection module to GreenGPS is assessed. Third, the potential fuel savings of an individual using GreenGPS is evaluated.

7.1 GreenGPS Application Products

On the client-side of the GreenGPS three application products have been developed and offered to the users. The first two applications are for the sake of navigation of green routes and can be used by GreenGPS members as well as non-members. The third application is for the purpose of collection of sensing data and can be used by volunteer participants (i.e. GreenGPS members). The products source code is available in [113].

7.1.1 Green Navigation Engine Web-based GUI

A web-based frontend graphical user interface is developed to present navigated fuel-optimal routes to the users; a snapshot is shown in Figure 3.1.

When a query is posed to GreenGPS for the fuel-optimal route between the source address and destination address provided by the user inputs, the addresses are first translated into latitude/longitude pairs using the open source geocoding perl module, Geo::Coder::US. This module is used for geocoding US addresses only. Geocoding is the process of finding corresponding latitude/longitude data given a street address, intersection, or zipcode.

After the source and destination addresses are geocoded into their corresponding latitude and longitude pairs using the geocoder module, the latitude and longitude pairs are fed to the navigation module which computes the fuel-optimal route (along with the shortest and fastest routes) using the OSM XML database and the prediction models of fuel consumption on streets (computed from the OBD-II sensor data contributed by users). The computed routes are then displayed on the GUI frontend along with the estimated fuel consumption for the given routes. The GUI frontend to display the routes utilizes Google maps. Routes are color coded and rendered as *polylines* on Google maps. For example, in Figure 3.1 the fuel-optimal route is a “green” color polyline.

7.1.2 Green Navigation Android Application

An Android application has been developed and offered as an alternative to the web-based GUI in order to assist drivers to find the fuel-optimal routes. The users can set their vehicle characteristics (i.e. make, model, age, and category) and other options under the applications’ preferences menu. The fuel-optimal route is computed for the given destination address and visual mapping of the route plus voice directions are provided to the driver. The route characteristics (i.e. distance, time, and fuel consumption) are provi-

sioned as well.

7.1.3 Data Collection Android Application

Another client-side Android application has been developed and offered for the collection of sensing data by volunteers (i.e. GreenGPS members). The application collects fuel related data from the vehicle OBD-II port through the bluetooth adaptor and stores them in a local database along with several GPS parameters obtained from the phone GPS. The application then manages to upload the stored data to the GreenGPS backend server in an automatic and opportunistic manner.

7.2 Green Navigation Model Accuracy

In this section, we evaluate the accuracy of the GreenGPS prediction model in estimating fuel consumption on long routes. For that, the attributes contributed to each trip in the collected driving dataset in the Urbana-Champaign, called for by Equation 3.17, are extracted and/or computed for each corresponding path.

In the experimental evaluation, the number and location of stop signs and traffic lights along each path is predicted using our Random Forest based classifier. The classifier is trained using a dataset collected from part of the city of Los Angeles (and not from Urbana or Champaign). It was tested in Urbana-Champaign to demonstrate cross-city generalizability. When testing, street features were extracted from OSM maps for each intersection then input to the classifier. Ground truth (for both training and testing) was collected using GoogleStreetView. As mentioned earlier in Chapter 4, the LA-based classifier achieved an accuracy level of 91% in predicting the

existence and types of traffic regulators on the streets of Urbana-Champaign. The next question was: given the imperfect prediction of traffic regulators, what is the accuracy in predicting fuel consumption?

The accuracy of our green navigation service is measured using path-based cross validation in which the fuel consumption along one path is predicted using the models trained based on data collected along other paths. The prediction error for the path is then obtained. This is repeated for all paths.

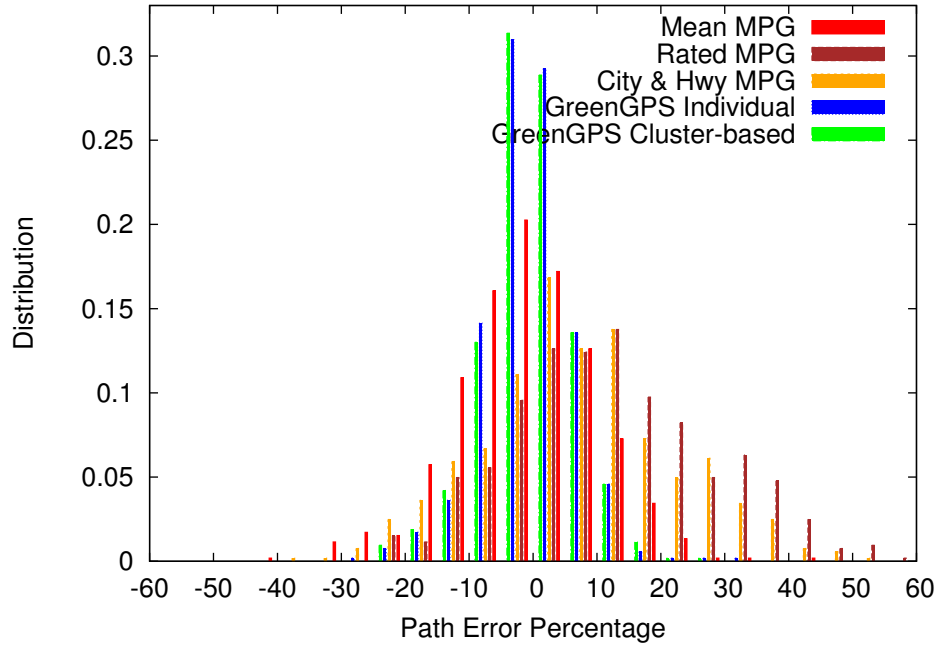
The path error distribution corresponding to the above experiment when prediction for each car is done based on data of the same car (on other paths) is shown in Figure 7.1a as “GreenGPS Individual”. We observe that the path error distribution is nearly normal and that the mean of this distribution is near zero (-0.28%).

We conduct a similar experiment to derive the path error distribution that is achieved by employing Cluster-based training such that fuel consumption of a car trip is predicted from the model trained based on trips of other cars in the nearest cluster as well, as described in Section 3.3.2. The prediction error for each path is computed as before and the distribution is presented in the figure as “GreenGPS Cluster-based”. Again, a normal distribution of the path errors is observed with near zero mean (-0.25%).

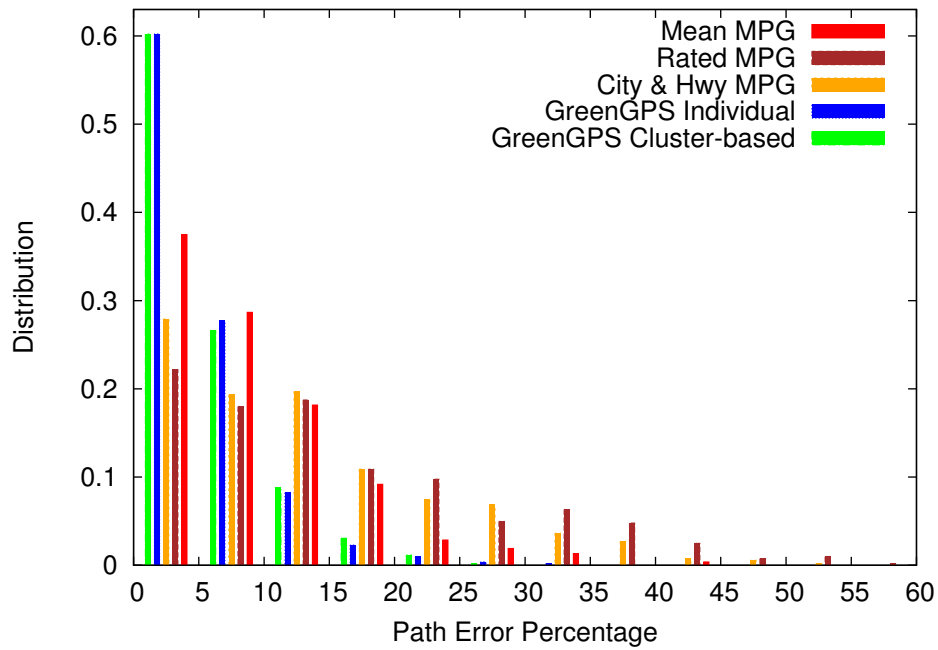
In order to compare the accuracy of our technique, three other fuel prediction approaches are evaluated in Figure 7.1a in which mpg values are the basis of the prediction. In these approaches the fuel consumption along a path is estimated using:

$$f_c^{mpg} = \frac{L}{MPG} \quad (7.1)$$

in which L is the length of the path and MPG is the mpg of the car. In



(a)



(b)

Figure 7.1: Distribution of path error percentage for different prediction models: (a) signed error, (b) unsigned error

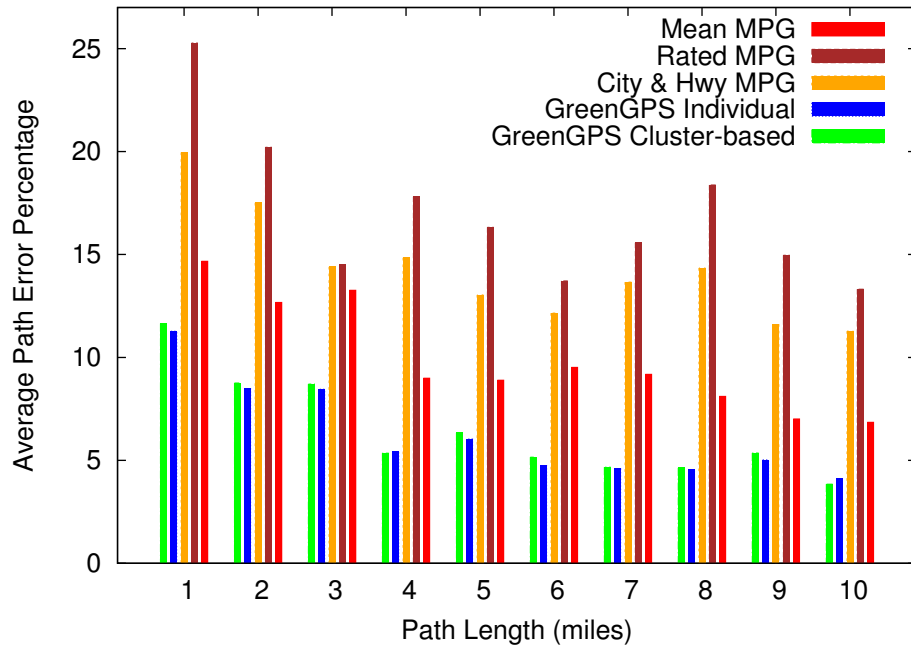
Mean MPG approach, the *MPG* is the average mpg computed from data of the car. In *Rated MPG* approach, the *MPG* is computed as the average of rated city mpg and rated highway mpg for the car. In the last approach, *City & Hwy MPG*, for each individual road segment along a path, depending on the road segment type either city mpg or highway mpg is used for fuel prediction.

In order to compare the approaches more clearly, the distribution of the corresponding unsigned error is shown in Figure 7.1b. As depicted in the figure, GreenGPS approach outperforms the other prediction methods. It is observed in the figure that GreenGPS Individual and Cluster-based training approaches differ only slightly in accuracy. The reason lies behind the lack of overlap among car types in our vehicle set. As a result, for most of the cars the nearest cluster in Cluster-based training becomes a cluster with one single car—the car for which prediction accuracy is being calculated. Therefore it should be emphasized that these two approaches may significantly differ from each other for a different dataset; this is explained later in Figure 7.3.

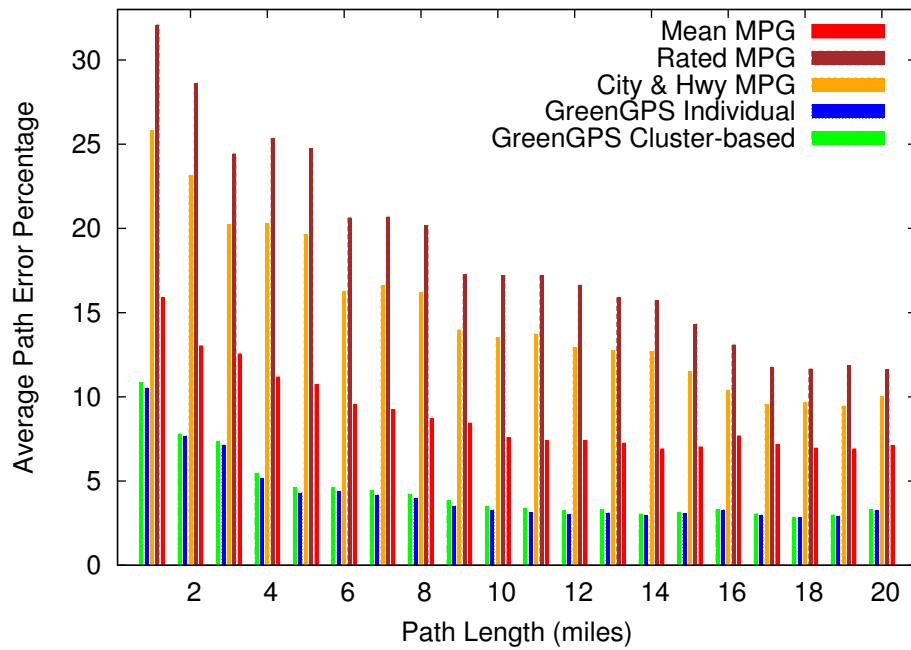
It is worth noticing that, as expected, the Mean MPG approach beats the other mpg-based approaches in Figure 7.1b. This is because the Mean MPG approach uses the collected data to compute cars' mpgs as opposed to considering a predetermined fixed constant.

In order to understand how path errors vary with path lengths, we bin the paths based on their length and compute the average of the absolute path errors as a function of path length. We repeat this experiment for the case where models are derived for each car individually and the case where models are derived for clusters and the nearest cluster is used. We plot the mean of the absolute path errors for varying path lengths in Figure 7.2a.

We observe from Figure 7.2a that the error decreases with increasing path



(a)



(b)

Figure 7.2: Mean path error percentage for different prediction models when path length is varied: using (c) original data, (d) synthetic data

length for both GreenGPS and mpg-based approaches, which is what we want. In order to show the performance of these approaches for longer routes beyond ten miles, the trips in our original dataset are concatenated to form longer trips. We concatenate every up to ten chronologically consecutive trips (timestamped based on start and finish time) together and form longer trips. The features of the new trips (such as distance and the number of traffic regulators) are computed based on those of the original constituting trips. We then added the new longer trips to the original set of trips. Figure 7.2b presents the accuracy results on the new dataset. As expected, the decreasing trend of the prediction errors continues for trips beyond ten miles long as well. The average percentage error for the dataset is 4.74% and for trips longer than four and ten miles is 3.67% and 3.08%, respectively.

We have not explored if the progressively improving accuracy of the approaches with respect to the trip distance holds true when the commutes have large dynamics in speeds, such as in larger cities. The current dataset is limited in that it was collected in a fairly quiet town.

The accuracy of our approach depends on the amount of training data. Figure 7.3 presents the impact of the training dataset size on the performance of fuel prediction approaches. The 100% point denotes using the whole dataset, 50% denotes using half of the dataset, and so on. The dataset down-scaling was performed in an alternate manner on the set of all chronologically ordered trips that were grouped based on the contributing vehicles. For example, for the 50% dataset size, one out of every two consecutive trips in the list was selected, for the 33% dataset size, one out of every three consecutive trips was selected, and so on and so forth.

As depicted in the figure, as the training dataset becomes quite small, the GreenGPS Individual training becomes inaccurate. This is while the ac-

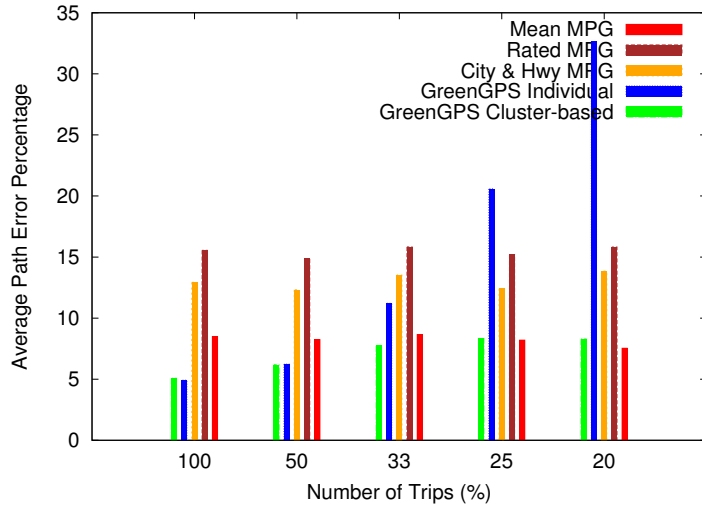


Figure 7.3: Impact of the amount of training data on different prediction models accuracy

accuracy of the Cluster-based approach slightly decreases and it significantly outperforms Individual training approach for small datasets. Hence as the dataset becomes smaller, the performance gap between the Individual training and the Cluster-based training increases. At the same time, the accuracy of the mpg-based approaches remains nearly constant. This suggests to adopt an mpg-based approach at the very beginning of the deployment phase (when there is no or very limited data collected) and then shift to GreenGPS train-based approach as sufficient data for constructing reliable models is collected. The figure also depicts the GreenGPS potential for further increase in precision (compared to the results presented here) through collection of more driving data.

From the perspective of building participatory sensing applications, the above suggests the importance of finding models that do not have *biased error*. Since the models often try to predict aggregate or long-term behavior (such as long term exposure to pollutants, annual cost of energy consumption, eventual weight-loss on a given diet, etc.), if the error in day-by-day predic-

tions is normally distributed with zero mean, the long-term estimates will remain accurate. Hence, rather than worrying about exact models, GreenGPS attempts to find *unbiased* models, which is easier.

One should add that our evaluation is not intended to be a definitive study on vehicular fuel consumption. For example, we evaluate fuel consumption in Urbana-Champaign only, which is quite flat. Hence, $\theta = 0$ is a good approximation. Furthermore, the range of cars used in the study is rather skewed towards sedans, and hence not representative of the diversity of cars on the streets. Fortunately, even this rather homogeneous dataset was sufficient to show that the generalization challenge is hard.

With the above caveats, we believe that the study remains of interest in that it explores problems typical to many participatory sensing applications, such as overcoming conditions of sparse deployment, adjusting to heterogeneity, and living with large day-to-day errors towards estimating cumulative properties. The GreenGPS study could therefore serve as an example of what to expect in building similar services, as well as a recipe for some of the solutions.

7.3 Regulator Detection Module Impact

In this section, we evaluate the impact of the regulator detection module on the performance of the GreenGPS application. The placement of traffic regulators is one of the most important factors significantly affecting estimated fuel consumption on different routes. The proposed approach enables GreenGPS and similar applications to predict this placement automatically from a small amount of data collected from a *small subset* of covered cities,

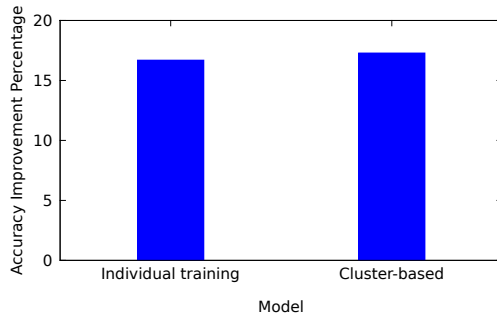


Figure 7.4: Impact of the regulators detection module on GreenGPS

hence obviating expensive efforts to manually collect traffic regulator placement information in all covered cities.

We assess the impact of the proposed regulator detection methodology on the accuracy of fuel consumption estimation in GreenGPS. Specifically we compare the GreenGPS fuel estimation accuracy with and without the benefit of the traffic regulator prediction module. For this purpose, we constructed the model for regulator placement using data collected in the city of Los Angeles, and then used the model to detect and recognize traffic regulators along each path driven by our user subjects in the area of Urbana and Champaign (i.e. cross-city modeling). The results are depicted in Figure 7.4 and show a 17% and 18% improvement in the estimation accuracy of the fuel consumption for members and non-members, respectively.

7.4 Fuel Savings in Urbana-Champaign

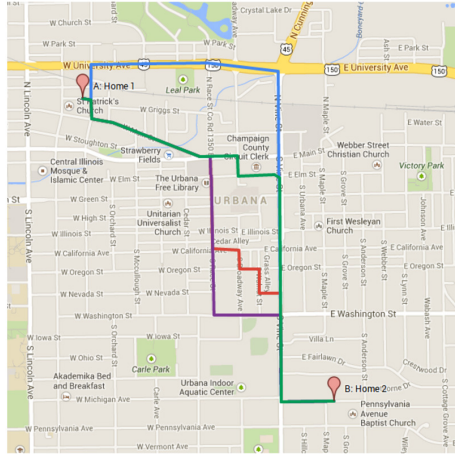
In this section, we evaluate the fuel savings achieved when using the GreenGPS system. To evaluate fuel savings, we chose landmarks in the city of Urbana-Champaign that are regularly visited in our commutes, such as library, the university health center, stadium, frequently visited restaurants and parks, and shopping complexes. Then the shortest, the fastest, the Garmin eco-

route, and the GreenGPS green routes were looked up for each pair of landmarks. Each person selected two pairs of landmarks and for each of which drove twenty round trips (of approximately 15-35 minutes each): five on the shortest route, five on the fastest route, five on the Garmin eco-route, and five on the GreenGPS green route. The actual fuel consumption for each trip was recorded. The landmarks together with the shortest, fastest, Garmin eco, and GreenGPS green routes are shown in Figure 7.5. The routes for the trips in the opposite direction (i.e., driving from point B to point A) are very similar to the ones presented in the figures for forward direction and are thus omitted.

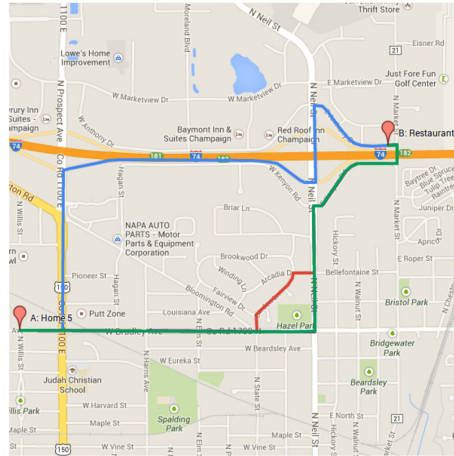
We observe from Figure 7.5 that the fuel-optimal route for the source-destination pair in the b , c , and e were similar to the shortest route and in the d it was the fastest route, whereas, in the a and f the fuel-optimal route was neither the shortest, nor the fastest. Hence, picking the shortest or fastest routes consistently is not optimal.

The average fuel consumption for the trips in the experiment are shown in Figure 7.6. It can be observed that the GreenGPS, except for the trip (f) – *Forward*, consistently finds the most fuel-efficient route. To confirm that the differences in fuel consumption between the compared routes are not due to measurement noise, we tested the statistical significance of the difference in means using the two-way ANOVA. The test yielded that the differences are statistically significant with a confidence level of 95%.

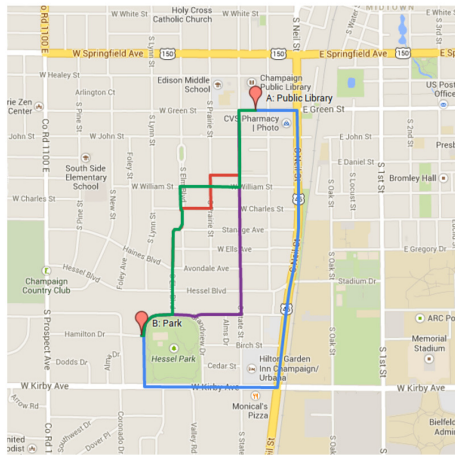
The average fuel saving percentage achieved by following the GreenGPS green routes as opposed to the fastest, the shortest, and the Garmin eco routes is presented in Figure 7.7. The results report that the GreenGPS routes can lead to fuel savings of on average, 21.5% over the fastest routes, 11.2% over the shortest routes, and 8.4% over the Garmin eco routes. Al-



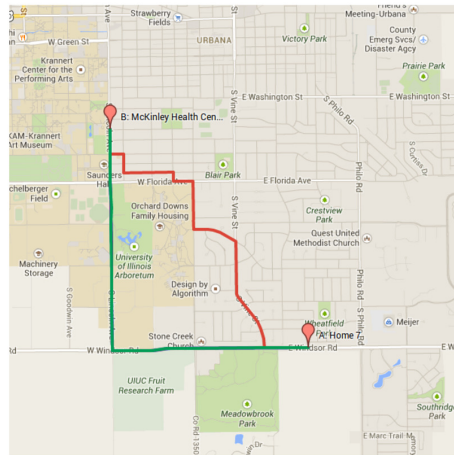
(a)



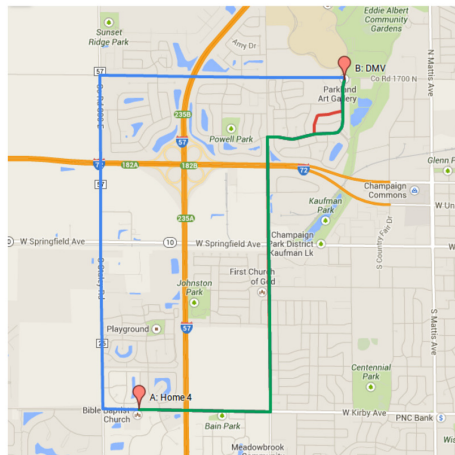
(b)



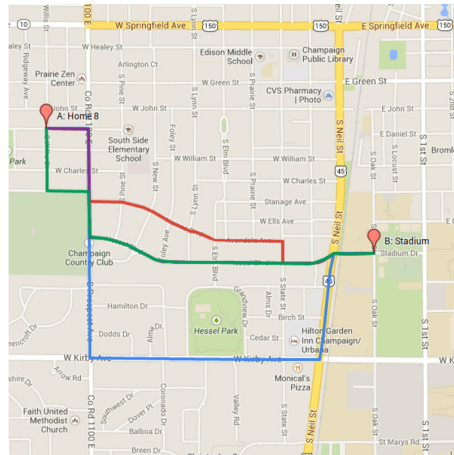
(c)



(d)



(e)



(f)

Figure 7.5: The landmarks and the corresponding shortest (in red), fastest (in blue), Garmin eco (in purple), and GreenGPS green (in green) routes: (a,b): Toyota Camry 2004; (c,d): Nissan Altima 2006; (e,f): Toyota Corolla 2000.

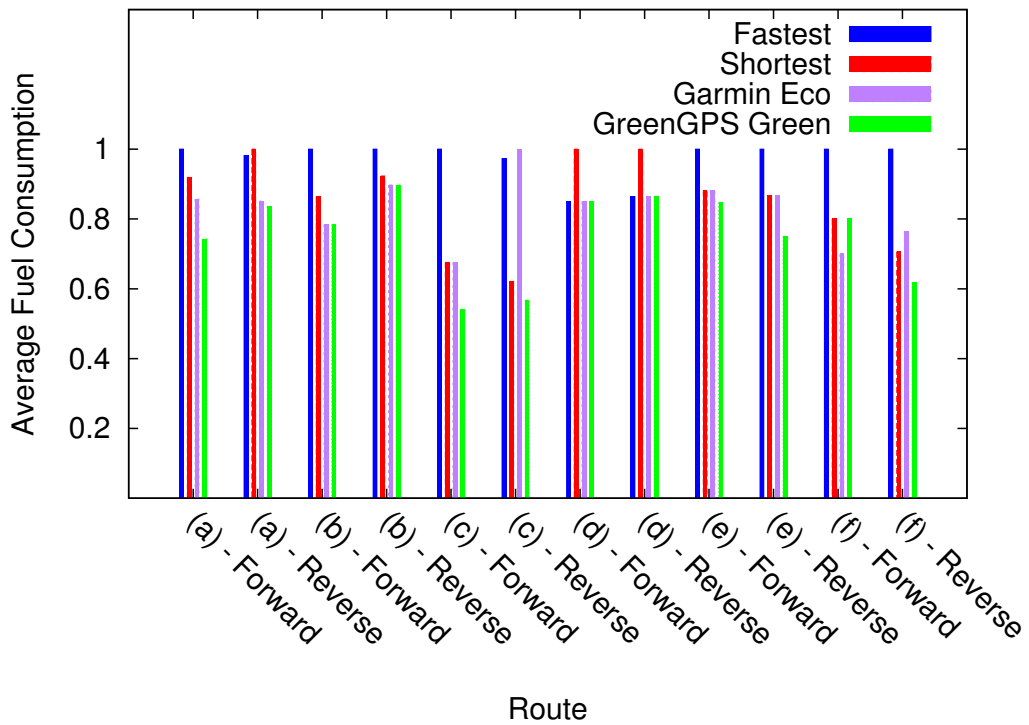


Figure 7.6: Average normalized fuel consumption for the various trips between different landmarks

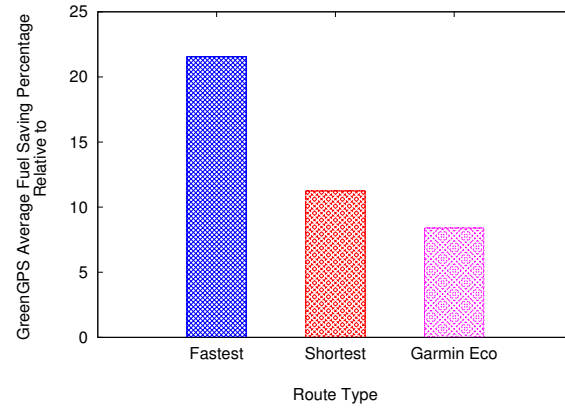


Figure 7.7: Percentage fuel saved by using GreenGPS green routes, relative to the Fastest, Shortest, and Garmin Eco routes

though only a handful of routes were used in the experiments above, it nevertheless shows promise as a proof of concept.

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

In this chapter we conclude the dissertation. First a summary of the thesis is provided and then future directions for extending the thesis work is presented.

8.1 Thesis Summary

We presented GreenGPS, an end-to-end automated participatory sensing navigation service that finds fuel optimal routes. GreenGPS is offered as a phone application and can be easily deployed and used by individuals. The required data is collected from the engine OBD-II of members' vehicles and processed on the backend server in an end-to-end automated manner. GreenGPS enables users including non-members to acquire the most fuel-efficient routes customized for their vehicles between any arbitrary endpoints. To survive conditions of sparse deployment, GreenGPS exploits a sparse data generalization technique from data mining literature to construct reliable fuel prediction models.

In order to detect and recognize the type of traffic regulators a novel combined map-based inference and crowd-sensing methodology is proposed. The approach reverse engineers the placement of traffic regulators, established by respective authorities. To that end, we leverage the power of road network maps and model the process in terms of broadly available map data. The map-based inference model is enhanced with crowd-sourced vehicular

data, where available, and is shown to outperform both the map-based and crowd-sensing based inference models alone.

A major hurdle in getting a participatory sensing system like GreenGPS off the ground is to provide the right incentives to the individuals (who are part of the system) [118]. The low price of the GreenGPS was one of our main design targets in order to incentivize users to adopt the service. We believe that, in addition, the initial deployment which tends to be sparse, should be carefully designed in order to provide incentives for larger adoption. It should therefore be useful from the very early stages. At the early deployment stage, called slow start, during which service adoption is sparse, collected data may not be sufficient for building reliable models for most cars. Hence, apart from the full-fledged model (taking into account more attributes such as specific road conditions) that is more accurate but needs more training data to compute, simpler models (e.g., rated-MPG based fuel consumption prediction) are proposed to be constructed and exploited during the slow start phase which may be quite long.

The full-fledged model can make better predictions, but only when the underlying training data becomes adequate. A transition to the full-fledged model must be made when *sufficient* data is collected. The challenge is that the adequacy of data cannot simply be decided upon using basic guidelines and rules of thumb. The transition point depends on many factors, including model type and complexity, and the collected data distribution in its multi-dimensional feature space. We proposed a theoretical approach to address the problem in participatory sensing applications. The application domain is divided into multi-dimensional subspaces, based on the inherent nonlinearities in the target function (i.e. fuel consumption), within each local model transition is planned. The proposed technique reliably plans transition to

carefully designed elaborate application models. Satisfying an intended criterion on the holistic modeling error bound, a reliable transition point is derived and theoretical guarantees are provided.

We designed and deployed a vehicular participatory sensing platform that can be used for future research. Lessons were presented that extrapolate from experiences with our deployed service to broad issues with participatory sensing service design in general. We utilized the participatory sensing platform to conduct a user subject study and evaluate the GreenGPS service and its components. A moderate sized sensing dataset was collected. Our experimental results show that significant fuel savings can be achieved by using GreenGPS, which not only reduces the cost of fuel, but also has a positive impact on the environment by reducing engine emissions of air poisoning gases. Importantly, the results demonstrate the feasibility of generalization from sparse deployment data, the effectiveness of the regulators detection and recognition module and its cross-city applicability, and the benefits of the slow start model transition planning in terms of service reliability.

We offer the GreenGPS service in the form of a web-based navigation engine and a smartphone navigation application (under Android). The sensing data collection is also performed with the aim of a developed Android application. We are currently in the process of deploying our service on over 100 vehicles of UIUC Facilities and Services fleet.

8.2 Limitations and Future Directions

The service designed and developed in this thesis can be extended in several directions. The first direction is related to driving behavior, an important parameter impacting the fuel consumption: an aggressive driver making higher

acceleration or hard braking is likely to consume more fuel than a sluggish driver. The work presented in this thesis can be extended to account for the impact of driving factors such as speed level and hardness of acceleration and braking, which should be modeled and progressively updated to reflect recent changes in driving habits.

With the deployment of GreenGPS on over 100 vehicles of UIUC Facilities and Services (in progress), in the second direction, the fuel optimal route navigation service can be extended for a fleet of vehicles. Such a service brings new challenges to be addressed, because individual navigation of green routes would be suboptimal for a fleet of vehicles. Instead, routes need to be planned and optimized with respect to available fleet vehicles. The University Facilities and Services fleet provides an opportunity to study fuel consumption characteristics of a fleet, which in turn can influence its management.

The third direction pertains to the methodology proposed for detection and recognition of traffic regulators, which can be extended to detect other traffic signs (beyond traffic lights, stop signs, and not controlled) such as yield signs, left lights, no turn on red signs, etc. The impact of these signs on fuel consumption and route navigation can then be taken into account. The analysis and study in this thesis shows promise for fulfilling the goal.

The experimental study of our green navigation service in this thesis was conducted in the area of Urbana-Champaign, IL, which is quite lightly congested and flat. However, traffic congestion induces a large variation in the fuel consumption of trips in larger cities, and road slope is a main player in the fuel consumption of heavily loaded trucks. Although real-time traffic information and road grade has been taken into the modeling of GreenGPS, the respective performance has not been evaluated. Hence, in the fourth di-

rection, the evaluation and analysis of the service can be extended to conduct studies in larger congested cities and hilly areas and analyze the performance of the system. To look up road grade for the sake of route navigation, map database of GreenGPS and/or existing external elevation maps can be used.

REFERENCES

- [1] EIA, “U.S. Energy Information Administration,” <http://www.eia.gov>.
- [2] EPA, “U.S. Environmental Protection Agency,” <http://www.epa.gov>.
- [3] EPA, “Emission facts: Greenhouse gas emissions from a typical passenger vehicle,” <http://www.epa.gov/OMS/climate/420f05004.htm>.
- [4] Google Maps, <http://maps.google.com>.
- [5] MapQuest, <http://www.mapquest.com>.
- [6] AAA, “National average gas prices,” <http://www.fuelgaugereport.com>.
- [7] J. A. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava, “Participatory sensing,” Workshop on World-Sensor-Web, co-located with ACM SenSys, 2006.
- [8] T. Abdelzaher, Y. Anokwa, P. Boda, J. Burke, D. Estrin, L. Guibas, A. Kansal, S. Madden, and J. Reich, “Mobiscopes for human spaces,” *Pervasive Computing, IEEE*, vol. 6, no. 2, pp. 20–29, 2007.
- [9] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, R. A. Peterson, H. Lu, X. Zheng, M. Musolesi, K. Fodor, and G.-S. Ahn, “The rise of people-centric sensing,” *IEEE Internet Computing*, 2008.
- [10] M. Srivastava, T. Abdelzaher, and B. Szymanski, “Human-centric sensing,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 2012.
- [11] J.-H. Huang, S. Amjad, and S. Mishra, “Cenwits: a sensor-based loosely coupled search and rescue system using witnesses,” in *Proceedings of the 3rd international conference on Embedded networked sensor systems (SenSys)*. ACM, 2005, pp. 180–191.
- [12] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden, “Cartel: a distributed mobile sensor computing system,” in *Proceedings of the 4th international conference on Embedded networked sensor systems (SenSys)*. ACM, 2006, pp. 125–138.

- [13] Sense Networks, “Cab sense,” <http://www.cabsense.com>.
- [14] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell, “The bikenet mobile sensing system for cyclist experience mapping,” in *Proceedings of the 5th international conference on Embedded networked sensor systems (SenSys)*, ser. SenSys ’07. ACM, 2007, pp. 87–101.
- [15] S. Reddy, A. Parker, J. Hyman, J. Burke, D. Estrin, and M. Hansen, “Image browsing, processing, and clustering for participatory sensing: lessons from a dietsense prototype,” in *Proceedings of the 4th workshop on Embedded networked sensors*. ACM, 2007, pp. 13–17.
- [16] S. Gaonkar, J. Li, R. R. Choudhury, L. Cox, and A. Schmidt, “Microblog: sharing and querying content through mobile phones and social participation,” in *Proceedings of the 6th international conference on Mobile systems, applications, and services (MobiSys)*. ACM, 2008, pp. 174–186.
- [17] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda, “Peir, the personal environmental impact report, as a platform for participatory sensing systems research,” in *Proceedings of the 7th international conference on Mobile systems, applications, and services (MobiSys)*. ACM, 2009, pp. 55–68.
- [18] I. Constandache, X. Bao, M. Azizyan, and R. R. Choudhury, “Did you see bob?: human localization using mobile phones,” in *Proceedings of the sixteenth annual international conference on Mobile computing and networking*. ACM, 2010, pp. 149–160.
- [19] X. Bao and R. Roy Choudhury, “Movi: mobile phone based video highlights via collaborative sensing,” in *Proceedings of the 8th international conference on Mobile systems, applications, and services (MobiSys)*. ACM, 2010, pp. 357–370.
- [20] P. Zhou, Y. Zheng, and M. Li, “How long to wait?: predicting bus arrival time with mobile phone based participatory sensing,” in *Proceedings of the 10th international conference on Mobile systems, applications, and services (MobiSys)*. ACM, 2012, pp. 379–392.
- [21] S. Chen, M. Li, K. Ren, X. Fu, and C. Qiao, “Rise of the indoor crowd: Reconstruction of building interior view via mobile crowdsourcing,” in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems (SenSys)*. ACM, 2015, pp. 59–71.

- [22] E. Ericsson, H. Larsson, and K. Brundell-Freij, “Optimizing route choice for lowest fuel consumption—potential effects of a new driver support tool,” *Transportation Research Part C: Emerging Technologies*, vol. 14, no. 6, pp. 369–383, 2006.
- [23] K. Brundell-Freij and E. Ericsson, “Influence of street characteristics, driver category and car performance on urban driving patterns,” *Transportation Research Part D: Transport and Environment*, vol. 10, no. 3, pp. 213–229, 2005.
- [24] M. Van der Voort, “Fest - a new driver support tool that reduces fuel consumption and emissions,” pp. 90–93, 2001.
- [25] M. van der Voort, M. S. Dougherty, and M. van Maarseveen, “A prototype fuel-efficiency support tool,” *Transportation Research Part C: Emerging Technologies*, vol. 9, no. 4, pp. 279–296, 2001.
- [26] T. Guan and C. W. Frey, “Fuel efficiency driver assistance system for manufacturer independent solutions,” in *Proceedings of the 15th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2012, pp. 212–217.
- [27] J. P. Aguilar, “Optimization of driving styles for fuel economy improvement,” Oak Ridge National Laboratory (ORNL); National Transportation Research Center, Tech. Rep., 2012.
- [28] J. Froehlich, T. Dillahunt, P. Klasnja, J. Mankoff, S. Consolvo, B. Harrison, and J. A. Landay, “Ubigreen: investigating a mobile tool for tracking and supporting green transportation habits,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2009, pp. 1043–1052.
- [29] Y. Chen, D. Zhang, and K. Li, “Enhanced eco-driving system based on v2x communication,” in *Proceedings of the 15th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2012, pp. 200–205.
- [30] K. Ahn and H. A. Rakha, “Network-wide impacts of eco-routing strategies: A large-scale case study,” *Transportation Research Part D: Transport and Environment*, vol. 25, pp. 119–130, 2013.
- [31] H. K. Strömberg and I. Karlsson, “Comparative effects of eco-driving initiatives aimed at urban bus drivers—results from a field trial,” *Transportation Research Part D: Transport and Environment*, vol. 22, pp. 28–33, 2013.

- [32] X. Hu, V. Leung, K. G. Li, E. Kong, H. Zhang, N. S. Surendrakumar, and P. TalebiFard, “Social drive: a crowdsourcing-based vehicular social networking system for green transportation,” in *Proceedings of the third ACM international symposium on Design and analysis of intelligent vehicular networks and applications*. ACM, 2013, pp. 85–92.
- [33] T. Flach, N. Mishra, L. Pedrosa, C. Riesz, and R. Govindan, “Carma: towards personalized automotive tuning,” in *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems (SenSys)*. ACM, 2011, pp. 135–148.
- [34] E. Koukoumidis, M. Martonosi, and L.-S. Peh, “Leveraging smartphone cameras for collaborative road advisories,” *IEEE Transactions on Mobile Computing*, vol. 11, no. 5, pp. 707–723, 2012.
- [35] A. Schwarzkopf and R. Leipnik, “Control of highway vehicles for minimum fuel consumption over varying terrain,” *Transportation Research*, vol. 11, no. 4, pp. 279–286, 1977.
- [36] J. Hooker, “Optimal driving for single-vehicle fuel economy,” *Transportation Research Part A: General*, vol. 22, no. 3, pp. 183–201, 1988.
- [37] H. Xia, K. Boriboonsomsin, F. Schweizer, A. Winckler, K. Zhou, W.-B. Zhang, and M. Barth, “Field operational testing of eco-approach technology at a fixed-time signalized intersection,” in *Proceedings of the 15th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2012, pp. 188–193.
- [38] I. Nagy, E. Suzdaleva, and T. Mlynarova, “Optimization of driving based on currently measured data,” in *Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2013.
- [39] A. Corti, V. Manzoni, and S. Savaresi, “Simulation of the impact of traffic lights placement on vehicles energy consumption and co2 emissions,” in *Proceedings of the 15th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2012.
- [40] T.-Y. Liao, “A fuel-based signal optimization model,” *Transportation Research Part D: Transport and Environment*, vol. 23, pp. 1–8, 2013.
- [41] F. Qiao, J. Wang, X. Wang, J. Jia, and L. Yu, “A rfid based e-stop sign and its impacts to vehicle emissions,” in *Proceedings of the 15th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2012, pp. 206–211.

- [42] M. Seredynski, B. Dorronsoro, and D. Khadraoui, "Comparison of green light optimal speed advisory approaches," in *Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2013.
- [43] M. Alsabaan, K. Naik, T. Khalifa, and A. Nayak, "Optimization of fuel cost and emissions with vehicular networks at traffic intersections," in *Proceedings of the 15th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2012, pp. 613–619.
- [44] X. Ma, "Towards intelligent fleet management: Local optimal speeds for fuel and emissions," in *Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2013.
- [45] A. Corti, V. Manzoni, and S. Savaresi, "Vehicle's energy estimation using low frequency speed signal," in *Proceedings of the 15th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2012.
- [46] D. Zhang, Y. Li, F. Zhang, M. Lu, Y. Liu, and T. He, "coride: carpool service with a win-win fare model for large-scale taxicab networks," in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems (SenSys)*. ACM, 2013.
- [47] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson, "Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys)*. ACM, 2009, pp. 85–98.
- [48] R. Sen, A. Maurya, B. Raman, R. Mehta, R. Kalyanaraman, N. Vankadhara, S. Roy, and P. Sharma, "Kyun queue: a sensor network system to monitor road traffic queues," in *Proceedings of the 10th ACM Conference on Embedded Networked Sensor Systems (SenSys)*. ACM, 2012, pp. 127–140.
- [49] A. Hrazdira, A. Cela, R. Hamouche, A. Reama, S. Niculescu, B. Rezende, and C. Villedieu, "Optimal real-time navigation system: Application to a hybrid electrical vehicle," in *Proceedings of the 15th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2012, pp. 409–414.
- [50] B. Xu, O. Wolfson, J. Yang, L. Stenneth, P. S. Yu, and P. C. Nelson, "Real-time street parking availability estimation," in *IEEE 14th International Conference on Mobile Data Management (MDM)*, vol. 1. IEEE, 2013, pp. 16–25.

- [51] S. H. Jacobson and L. A. McLay, “The economic impact of obesity on automobile fuel consumption,” *The Engineering Economist*, vol. 51, no. 4, pp. 307–323, 2006.
- [52] A. Mogelmosse, M. M. Trivedi, and T. B. Moeslund, “Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1484–1497, 2012.
- [53] S. Hu, H. Liu, L. Su, H. Wang, and T. Abdelzaher, “Smartroad: A mobile phone based crowd-sourced road sensing system,” UIUC Technical Report, 2013.
- [54] R. Carisi, E. Giordano, G. Pau, and M. Gerla, “Enhancing in vehicle digital maps via gps crowdsourcing,” in *Wireless On-Demand Network Systems and Services (WONS), 2011 Eighth International Conference on*. IEEE, 2011, pp. 27–34.
- [55] D. Wang, T. Abdelzaher, L. Kaplan, R. Ganti, S. Hu, and H. Liu, “Exploitation of physical constraints for reliable social sensing,” in *Proceedings of the 34th IEEE Real-Time Systems Symposium (RTSS)*. IEEE, 2013, pp. 212–223.
- [56] Y. Zhao, Y. Zhang, T. Yu, T. Liu, X. Wang, X. Tian, and X. Liu, “Citydrive: A map-generating and speed-optimizing driving system,” in *INFOCOM, 2014 Proceedings IEEE*. IEEE, 2014, pp. 1986–1994.
- [57] M. Kerper, C. Wewetzer, and M. Mauve, “Analyzing vehicle traces to find and exploit correlated traffic lights for efficient driving,” in *2012 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2012, pp. 310–315.
- [58] M. Kerper, C. Wewetzer, A. Sasse, and M. Mauve, “Learning traffic light phase schedules from velocity profiles in the cloud,” in *2012 5th International Conference on New Technologies, Mobility and Security (NTMS)*. IEEE, 2012, pp. 1–5.
- [59] E. Koukoumidis, L.-S. Peh, and M. R. Martonosi, “Signalguru: leveraging mobile phones for collaborative traffic signal schedule advisory,” in *Proceedings of the 9th international conference on Mobile systems, applications, and services (MobiSys)*. ACM, 2011, pp. 127–140.
- [60] H. Aly and M. Youssef, “Dejavu: An accurate energy-efficient outdoor localization system,” in *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2013, pp. 154–163.

- [61] H. Aly, A. Basalamah, and M. Youssef, "Map++: A crowd-sensing system for automatic map semantics identification," in *Sensing, Communication, and Networking (SECON), 2014 Eleventh Annual IEEE International Conference on*. IEEE, 2014, pp. 546–554.
- [62] Y. Jiang, H. Qiu, M. McCartney, G. Sukhatme, M. Gruteser, F. Bai, D. Grimm, and R. Govindan, "Carloc: Precise positioning of automobiles," in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems (SenSys)*. ACM, 2015, pp. 253–265.
- [63] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: rich monitoring of road and traffic conditions using mobile smartphones," in *Proceedings of the 6th ACM conference on Embedded network sensor systems*. ACM, 2008, pp. 323–336.
- [64] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, 2nd ed. Lawrence Erlbaum Associates, 1988.
- [65] W. D. Dupont and W. D. Plummer, "Power and sample size calculations: a review and computer program," *Controlled clinical trials*, vol. 11, no. 2, pp. 116–128, 1990.
- [66] W. D. Dupont and W. D. Plummer, "Power and sample size calculations for studies involving linear regression," *Controlled clinical trials*, vol. 19, no. 6, pp. 589–601, 1998.
- [67] P. Dattalo, *Determining sample size: Balancing power, precision, and practicality*. Oxford University Press, 2008.
- [68] T. P. Ryan, *Sample size determination and power*. John Wiley & Sons, 2013.
- [69] K. Kelley and S. E. Maxwell, "Sample size for multiple regression: obtaining regression coefficients that are accurate, not simply significant," *Psychological methods*, vol. 8, no. 3, pp. 305–321, 2003.
- [70] K. Kelley and S. E. Maxwell, "Sample size planning with applications to multiple regression: Power and accuracy for omnibus and targeted effects," *The Sage handbook of social research methods*, pp. 166–192, 2008.
- [71] L. K. Muthén and B. O. Muthén, "How to use a monte carlo study to decide on sample size and determine power," *Structural Equation Modeling*, vol. 9, no. 4, pp. 599–620, 2002.
- [72] A. A. Beaujean, "Sample size determination for regression models using monte carlo methods in r," *Practical Assessment, Research & Evaluation*, vol. 19, no. 12, pp. 1–16, 2014.

- [73] M. Sandelowski, "Sample size in qualitative research," *Research in nursing & health*, vol. 18, no. 2, pp. 179–183, 1995.
- [74] G. Guest, A. Bunce, and L. Johnson, "How many interviews are enough? an experiment with data saturation and variability," *Field methods*, vol. 18, no. 1, pp. 59–82, 2006.
- [75] G. A. Bowen, "Naturalistic inquiry and the saturation concept: a research note," *Qualitative research*, vol. 8, no. 1, pp. 137–152, 2008.
- [76] J. J. Francis, M. Johnston, C. Robertson, L. Glidewell, V. Entwistle, M. P. Eccles, and J. M. Grimshaw, "What is an adequate sample size? operationalising data saturation for theory-based interview studies," *Psychology and Health*, vol. 25, no. 10, pp. 1229–1245, 2010.
- [77] M. Mason, "Sample size and saturation in phd studies using qualitative interviews," in *Forum Qualitative Sozialforschung/Forum: Qualitative Social Research*, vol. 11, no. 3, 2010.
- [78] J. Kiefer and J. Wolfowitz, "Optimum designs in regression problems," *The Annals of Mathematical Statistics*, pp. 271–294, 1959.
- [79] J. Kiefer, "Optimum designs in regression problems, ii," *The Annals of Mathematical Statistics*, pp. 298–325, 1961.
- [80] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," *Journal of artificial intelligence research*, 1996.
- [81] R. Castro, R. Willett, and R. Nowak, "Faster rates in regression via active learning," in *Advances in Neural Information Processing Systems*, 2005.
- [82] R. Willett, A. Martin, and R. Nowak, "Backcasting: adaptive sampling for sensor networks," in *The 3rd international symposium on Information processing in sensor networks*. ACM, 2004.
- [83] F. Saremi, O. Fatemieh, H. Ahmadi, H. Wang, T. Abdelzaher, R. Ganti, H. Liu, S. Hu, S. Li, and L. Su, "Experiences with greengps – fuel-efficient navigation using participatory sensing," *IEEE Transactions on Mobile Computing*, 2015.
- [84] Actron, "Elite AutoScanner," http://www.actron.com/product_category.php?id=249.
- [85] Auterra, "DashDyno," <http://www.auterraweb.com/dashdynoseries.html>.
- [86] AutoTap, "AutoTap Reader," <http://www.autotap.com/products.asp>.

- [87] AutoXRay, “EZ-Scan,” http://www.autoxray.com/product_category.php?id=338.
- [88] ELM, “ELM327,” <http://elmelectronics.com/DSheets/ELM327DS.pdf>.
- [89] Traffic, “Real-time traffic conditions,” <http://www.traffic.com>.
- [90] California Department of Transportation, “PeMS,” <http://pems.dot.ca.gov>.
- [91] D. M. Bevly, R. Sheridan, and J. C. Gerdes, “Integrating ins sensors with gps velocity measurements for continuous estimation of vehicle sideslip and tire cornering stiffness,” in *Proceedings of the American Control Conference*, vol. 1. IEEE, 2001, pp. 25–30.
- [92] J. Farrelly and P. Wellstead, “Estimation of vehicle lateral velocity,” in *Proceedings of the IEEE International Conference on Control Applications*. IEEE, 1996, pp. 552–557.
- [93] H. E. Tseng, “Dynamic estimation of road bank angle,” *Vehicle System Dynamics*, vol. 36, no. 4-5, pp. 307–328, 2001.
- [94] J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow, and H. Pirahesh, “Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals,” *Data Mining and Knowledge Discovery*, vol. 1, no. 1, pp. 29–53, 1997.
- [95] Y. Chen, G. Dong, J. Han, J. Pei, B. W. Wah, and J. Wang, “Regression cubes with lossless compression and aggregation,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 12, pp. 1585–1599, 2006.
- [96] M. Y. S. Uddin, F. Saremi, and T. Abdelzaher, “End-to-end delay bound for prioritized data flows in disruption-tolerant networks,” in *Real-Time Systems Symposium (RTSS), 2010 IEEE 31st*. IEEE, 2010, pp. 305–316.
- [97] INRIX, <http://www.inrix.com>.
- [98] Nokia Here, <http://developer.here.com>.
- [99] Microsoft Bing, <http://msdn.microsoft.com/en-us/library/hh441725.aspx>.
- [100] 511NY, <http://511ny.org>.
- [101] F. Saremi and T. Abdelzaher, “Combining map-based inference and crowd-sensing for detecting traffic regulators,” in *the 12th IEEE international conference on Mobile Ad-hoc and Sensor Systems*, 2015.

- [102] Google Street View, <https://www.google.com/maps/views/streetview?gl=us>.
- [103] OpenStreetMap, <http://wiki.openstreetmap.org>.
- [104] US Census Bureau, “Tiger database,” <http://www.census.gov/geo/www/tiger>.
- [105] National Aeronautics and Space Administration (NASA), “Landsat Data,” <http://landsat.gsfc.nasa.gov/data>.
- [106] Manual on Uniform Traffic Control Devices (MUTCD), <http://mutcd.fhwa.dot.gov>.
- [107] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [108] M. Saar-Tsechansky and F. Provost, “Handling missing values when applying classification models,” *Journal of Machine Learning Research*, 2007.
- [109] A. Liaw and M. Wiener, “Classification and regression by randomforest,” *R News*, vol. 2, no. 3, pp. 18–22, 2002. [Online]. Available: <http://CRAN.R-project.org/doc/Rnews>
- [110] F. Saremi and T. Abdelzaher, “Slow start transition in participatory sensing applications,” in *the 13th IEEE international conference on Mobile Ad-hoc and Sensor Systems*, 2016.
- [111] C. R. Rao, H. Toutenburg, Shalabh, and C. Heumann, *Linear Models and Generalizations – Least Squares and Alternatives*. Springer-Verlag Berlin Heidelberg, 2008.
- [112] W. N. Venables, D. M. Smith, and the R Core Team, “An Introduction to R: A Programming Environment for Data Analysis and Graphics, Version 3.2.3,” <https://cran.r-project.org>.
- [113] F. Saremi et al., “GreenGPS Source Code,” http://greengps.cs.illinois.edu/greengps_code.zip.
- [114] Nic Roets, “Gosmore,” <http://wiki.openstreetmap.org/wiki/Gosmore>.
- [115] H. Ahmadi, N. Pham, R. Ganti, T. Abdelzaher, S. Nath, and J. Han, “Privacy-aware regression modeling of participatory sensing data,” in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (SenSys)*. ACM, 2010, pp. 99–112.

- [116] R. K. Ganti, N. Pham, Y.-E. Tsai, and T. F. Abdelzaher, “Poolview: stream privacy for grassroots participatory sensing,” in *Proceedings of the 6th ACM conference on Embedded networked sensor systems (SenSys)*. ACM, 2008, pp. 281–294.
- [117] N. Pham, R. K. Ganti, Y. S. Uddin, S. Nath, and T. Abdelzaher, “Privacy-preserving reconstruction of multidimensional data maps in vehicular participatory sensing,” in *Wireless Sensor Networks*. Springer, 2010, pp. 114–130.
- [118] S. Reddy, D. Estrin, and M. Srivastava, “Recruitment framework for participatory sensing data collections,” in *Pervasive Computing*. Springer, 2010, pp. 138–155.