A Vehicle Classification Algorithm based on Telematics Data

by

Linh Vuong Nguyen

B.S, Massachusetts Institute of Technology (2017)

Submitted to the Department of Electrical Engineering and Computer Science

in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2018

Author . . . .

Department of Electrical Engineering and Computer Science

February 2, 2018

Certified by . . . .

Tomas Palacios

Professor of Computer Science and Engineering

Thesis Supervisor

Certified by . . . .

Hari Balakrishnan

Professor of Computer Science and Engineering

Thesis Supervisor

Certified by . . . .

Bill Bradley

Principal Data Scientist, Cambridge Mobile Telematics

Thesis Supervisor

Accepted by . . . .

Christopher Terman

Chairman, Masters of Engineering Thesis Committee

A Vehicle Classification Algorithm based on Telematics Data

by

Linh Vuong Nguyen

Submitted to the Department of Electrical Engineering and Computer Science
on February 2, 2018, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

In the thesis, I develop an algorithm to identify the vehicle model from telematics data. By extracting the features from the accelerometer and GPS data, we obtain the classification features, which then goes through a multiclass random forest classifier. We apply this results into problems of driver and vehicle identification. The result shows that, while the algorithm could identify the vehicle models to some extent, the dominating signal comes from driving style, and an approach running purely unsupervised learning is harder to achieve good classification results compared to supervised methods.

Thesis Supervisor: Tomas Palacios
Title: Professor of Computer Science and Engineering

Thesis Supervisor: Hari Balakrishnan
Title: Professor of Computer Science and Engineering

Thesis Supervisor: Bill Bradley
Title: Principal Data Scientist, Cambridge Mobile Telematics

Acknowledgments

Not all the moments at MIT are smooth. I deeply thank my parents and sister for supporting me the whole time and reminding me of staying healthy and enjoying life, the habits that I often conveniently forget.

I would like to thank my advisors, Hari Balakrishnan, Tomas Palacios and Bill Bradley, for giving me this challenging problem and mentoring me during the course of my Master program. I learned a lot during this time and this experience really changed the way I think about Computer Science.

My deskmate, Geromino Mirano, for teaching me a ton of cool tricks and techniques for using Python and processing data. I learn a lot from him.

My coworkers and mentors at Cambridge Mobile Telematics: Jun-geun Park, Matthew Levine, Kanak Ksentri, Polina Binder and Ji-yang Wang, for walking me through the first days of "real world" life, bringing me to speed for contribution and allowing me the freedom and independence to carry the research.

Other great professors at MIT: Sam Madden, Srini Devadas and David Jerison, for trusting me unconditionally and listening to my desperate conversations when I was totally lost in life. I will surely miss you and wish to stay in Cambridge forever, if not for the weather.

A special thank to life, for keeping me surprised with new things, even when I thought I have known everything. Being in this life, at this place and this time, is a special gift that I could never forget.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivations

In America, on average people spend more than 290 hours a year driving, logging more than 10500 miles. Vehicle telematics offers a rich source on understanding users' driving behaviors. Recent advances from big data processing, machine learning and sensor networks have allowed for effective telematics data collection and processing, which have not only resolved many traditional problems, but also opened new avenues for studying new questions. Starting from 2006, MIT CarTel project [7] has attempted to collect and analyze telematics data when driving simply by using smartphone devices. Combined with big data processing and analytics, the project has also evaluated users' driving behavior and given suggestions to make them drive better.

With the development of big data techniques, automobile insurance companies are also changing their approach for insurance pricing. Traditional approaches are based on static, easily defined features, such as driver's age, gender, years of experience, as well as vehicle make and model. However, advances in big data has enabled the rise of telemetry-based insurance model, for example the pay as you go model [3]. The new methods take into account extra information, such as vehicle mileage, usage pattern or risky driving behavior, and employ complex machine learning models for risk assessment. This allows for insurance companies to tailor insurance plan for each user. The transition process has led to many interesting questions and forced revision on traditional insurance pricing methods.

1.2 Problem Statement

The focus of this work, is to take the rich telematics data collected on trips to study the question of *vehicle model recognition.* There are multiple variants of the problem; in this instance, we focus on vehicle identification of a user. That is, given driving history of a user consisting of multiple trips, each trip represented by its telematics, we need to identify all available vehicles and cluster the trips based on the vehicle the person is using.

There are multiple applications from solving this problem. For example, determining which vehicle driven by an user enables analytic and behavioral study on their driving behavior and helps making suggestion to improve their driving. From insurance companies' perspective, this enables them to study large scale behavior of users, for example, which vehicle model is more prone for unsafe driving behavior.

1.3 Challenges

Like many data problems, the main issues lie in data qualify. In telematics, the data quality problem is amplified by a wide variety of causes. Since all data is recorded in open road condition, such data can be affected by external factor, such as road bumps, traffic or pitch elevations. Such external factors could at best add noise into measurements, and at worst corrupt recorded data (for example, driving through tunnel makes GPS data become unavailable). The difficulty also comes from the unpredictable nature from human input, which is often case specific. Smartphone position, if data is recorded from the smartphone, can also add noise to the measurement. The low sampling rate also limits the ability to extract more granular features, which adds difficulty into designing good features that could differentiate different vehicle models.

Practical demand requires us to focus on two important conditions that allow the algorithm to be easily applied in real world: *granularity* (able to identify vehicle type, not just generic transportation mode like train, car or walking) and *ubiquity* (require only smartphone sensors, and data is collected on open road condition versus controlled environment such as closed circuit and wind tunnel). Choosing the right abstraction and performing controlled feature engineering are crucial for the classifier to be able to correctly classify trips.

1.4 Related Work

Many previous works have focused on various aspects of vehicle classification under different measurement conditions. The theory of vehicle modeling is documented in [5] [12]. Traditionally, most measurements are done in a controlled environment, with the vehicle is in factory condition and runs on closed circuit track, or require expensive preparation such as wind tunnel and various custom-made sensors. Such assumption is generally not applicable in real life condition, where external effect and driving characteristics can affect the measurements.

Nevertheless, more recent works have attempted to develop algorithms under general conditions, using only measurements from smartphone. [6] employs smartphone accelerometer to detect transportation mode. Many of their idea are adapted in this thesis. [9] used vertical acceleration to estimate vehicle's weight.

Telematics data belongs to the class of time series data, hence many techniques to extract features from time series data are relevant, such as statistical features, time-dependent features and spectral analysis. [11] gives an overview on feature extraction techniques and their application in music fingerprinting.

A similar problem is classifying trips with respect to driving style, in which [2] has proposed a deep learning solution. Our problem, although on a similar nature, has remarked difference. Since telematics data is dominantly influenced from driving input, which is heavily driver dependent, it is not clear how to extract invariant, vehicle-based features that does not depend on driving style.

1.5 Results Summary

In this thesis, we develop an algorithm for classifying vehicle model, and subsequently apply into user vehicle identification. If sufficient labeled data for each driver is available, we could build a per-user classifier that can classify about 90 percent of trips with their correct vehicle model. In addition, we also develop a classifier that utilizes data aggregated trips from multiple drivers and from popular vehicle models. With such framework, the classifier could classify about 45 percent of trips to their correct vehicle type (SUV, compact or sedan), compared to 33 percent with random guessing. All the classifiers are built using strictly telematics data, that is, no metadata information about trip or user is considered. However, combining with heuristics derived from user metadata, the classifier can be used for classify trips of drivers having no prior labels, achieving robust accuracy, albeit lesser compared to strictly supervised methods. We also study the features that could effectively discriminate different vehicle models.

1.6 Organization

Chapter 2 discusses the main technical part of the thesis, containing details on data collection, feature engineering and algorithm. With large amount of concepts introduced, we choose to present the idea in Chapter 2 with the goal to be self-explanatory rather than strictly precise. Chapter 3 presents the experiment setup and results. Chapter 4 discusses implications of the results and future extensions. The Appendix complements the main chapters by providing formal derivation of technical concepts we use in this thesis.

1.7 Disclosure

The work requires access to a large quantity of telematic data. The data is provided by Cambridge Mobile Telematics (CMT), which has allowed the author to use the data and general analysis pipeline to perform the work.

Chapter 2

Description of Approach

2.1 Data Collection

The data is recorded either from user's smartphone or from a customized hardware device designed by CMT attached to the vehicle, referred here simply as **tag**. Trips are recorded across many countries from 2013 to 2017. Various sensors record data at different sampling rate, but for simplification we assume all sensors sample at a fixed rate, with downsampling on sensors with higher sampling rate and linear interpolation on sensors with lower sampling rate. Table 2-1 lists all available sensors, corresponding with their notation that remain consistent in the thesis.

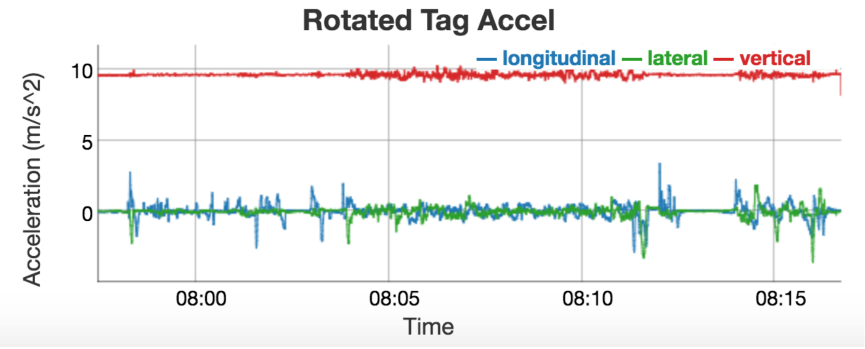

Figure 2-1: An instance of recorded data of a given trip.

The device records data in raw form and accounts for all the external factors that can affect the measurement. For example, gravitational force causes a constant down-

Measurements	**Sensor used**
Longitudial (a_x), lateral (a_y) and vertical acceleration (a_z)	Accelerometer
Position and velocity (v)	GPS
Roll, pitch and yaw	Gyroscope
Road pitch	Barometer
Vehicle orientation	Magnometer

Table 2.1: List of available measurements and corresponding sensors

ward acceleration in the vertical direction of the accelerometer. Road bumps or poor weather conditions can also affect the quality of the device's reading. A processing algorithm subsequently filters such external effects and aligns the measurements to be road-oriented, whose details is outside the scope of the thesis. For many trips, there is label of vehicle make and model, which we will take as ground truth data. However, the label is made at user basis, meaning for many users there is no information about their vehicles. There are 30 million such labeled trips, and 90 million unlabeled trips in our dataset.

Some metadata that are also useful for analysis includes trip information (trip start/end timestamp, start and end location, duration and distance) and anonymized user ID.

2.2 Intuition for the Approach

We will approach the problem with a semi-supervised learning algorithm. We build a classifier on *vehicle type* (such as SUV, compact or sedan) using data from many trips of many users, then apply the classifier to predict the vehicle type on trips by a particular user. Finally, we apply heuristics on vehicle usage pattern to group certain trips into the same vehicle classes.

Note that, although the problem is formulated as a clustering problem, we do not implement any clustering algorithm here. The reason is that, clustering algorithms suffer inherent difficulty, such as requiring a notion of similarity, and in some algorithms requiring the number of clusters in advance. For most cases, results obtained from clustering algorithms are hard to interpret and there is no obvious strategy on how to improve the results beside feature engineering, which is often trial and error process. Furthermore, in this problem, large amount of labeled data can enable semi-supervised approaches, if interpreted correctly.

Algorithms that rely on global features (for example, global analysis throughout the trip) suffer from the lack of discriminable features and noises incurred by various factors from the trip, such as traffic condition.

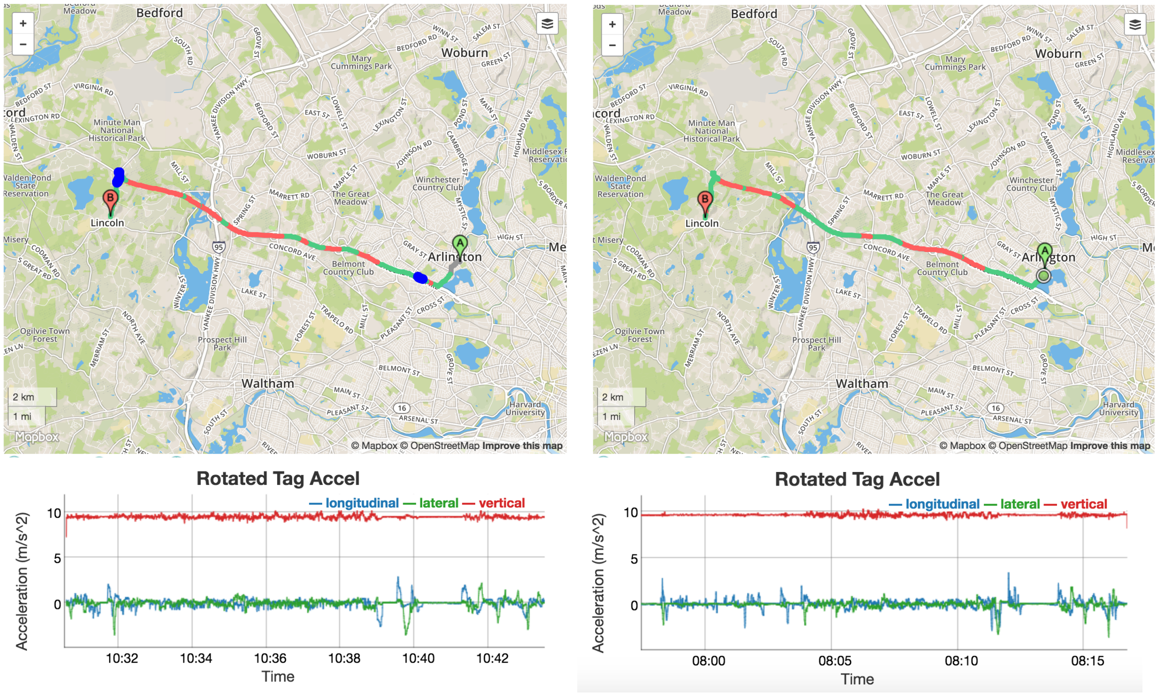

Figure 2-2: Global comparison between two different trips driven by different vehicle models traversing on the same route.

As shown here, in the long run, trip trajectory becomes the discriminative factor, dominating the local difference stemming from driving different vehicles. Therefore, the classification algorithm needs to exploit the local structures of the time series data where it suffices to discriminate different vehicle models. We accept to some extent features that are affected by drivers, since driving behaviors are governed by vehicle characteristics. Road condition, weather or traffic, on the other hand, should be excluded.

Techniques from machine learning suggest to collect locally based characteristics as the features, such as accelerating, engine characteristics, suspensions, steering and cornering. Various work from physics and mechanical engineering give initial intuition for constructing such models, but there are two departure from traditional engineering models. On one hand, the goal here is to *reconstruct* the model based on empirical data instead of confirming the validity of the model under road test. On the other hand, measurement error, limited sampling rate and open road condition may cause deviation from the ideal model, and it is likely that one sometimes needs to work with a more abstract or simplified model for the sake of computational efficiency. We will follow this line of idea, adding justification for abstract models upon necessary.

Although sampling rate limits the ability to obtain the precise values of the parameters, in practice, we don't need such precision. Since the same feature from different trips in the dataset is computed using the same algorithm, as long as the feature extraction function is reasonably well defined and continuous, small adjustments to the function would result to small change in the feature values, which retains their classification ability.

Since the classifier is inevitably noisy, there will be error on classifying user's trips. Therefore, we apply heuristic correction, which looks at trip history as sequence of points and find correlations between some pairs of trips. Those correlations allow us to put trips into the same vehicle type where the generic classifier cannot decide with certainty.

To summarize, our approach consists of three steps:

1. Build a classifier on vehicle type, using trips having labeled data.

2. For each user, use the classifier to classify unlabeled trips.

3. Apply subsequent heuristic correction to group certain trips into the same cluster, and output the final clusters.

2.3 Proposed Feature Engineering

The rest of the chapter explains how to obtain the features, their justification and their characteristics on classification.

2.3.1 Prelude: what to extract from timeseries data?

Unlike typical high dimensional data, timeseries data often comes at different dimensions and different channels, making typical feature extraction or dimensional reduction approaches such as Principal Component Analysis (PCA) or Singular Value Decomposition (SVD) difficult or not feasible. In this work, we approach from three directions:

- Extracting *statistical features* after removing invalid data points in the data. The selected features consist of mean, standard deviation, skew, kurtosis; 25, 50, 75 percentile, and minimum/maximum value. This approach ignores the time-dependent nature of the data; however, we find that its simplicity can essentially capture the nature of the time series, directly relate to the physical quantity capturing the vehicle's characteristics and achieve good classification results in practice. For subsequent subsections, we refer to this definition upon mentioning extracting statistical features from time series data.

- Extracting *time-dependent features* from the data. The most notable feature comes from evaluating the spectrogram of the signal. On the flip side, the features obtained from these techniques are not readily explainable, since they are only tangentially associated with the physical quantity. However, they can capture local and rare behavior of the vehicle, making them strong indicators for classification.

- Extracting *event-based features*, for example, hard braking and hard acceleration. These events are often time localized and caused by external sources from the driver road conditions. These features require more engineering and parameter tuning to achieve good discriminative accuracy.

Several features are inspired from modeling vehicle dynamics. Table 2.2 explains the dynamics and associated measurements, where subsequent sections explain intuitively how to extract features. Formal derivations of some of these models are deferred to Appendix A.1.

Vehicle Dynamic Model	**Associated measurements**
Longitudinal Dynamics	a_x, v
Lateral Dynamics	a_y, v
Suspension Response	a_z
Rolling Dynamics	a_y and roll angle

Table 2.2: List of vehicle dynamical models and associated measurements

2.3.2 Suspension response system

The suspension system is designed to reduce the shock coming to the vehicle upon encountering road artifacts, such as potholes. In this problem, we model the suspension as a damped harmonic oscillator that satisfy the following differential equation

$$\frac{d^2z}{dt^2} + 2\zeta\omega_0\frac{dz}{dt} + \omega_0^2 z = 0 \tag{2.1}$$

where ω_0 is the undamped angular frequency of the oscillator, and ζ is the damping ratio. Here $0 < \zeta < 1$ since the damped spring gradually kills oscillations caused by road impacts. With impact value A_0 at time $t = 0$, the damping value follows

$$z(t) = A_0 e^{-\zeta t} \sin(\omega_0 t) \tag{2.2}$$

To learn the parameters ω_0 and ζ, we compute the *autocorrelation* of the vertical acceleration data. Let $v(t)$ be the vertical acceleration at time t. For a lag $s \geq 0$, the autocorrelation corresponding to s is defined by

$$a(s) = \frac{\int v(t)v(t+s)\,dt}{\int |v(t)|^2\,dt} \tag{2.3}$$

with $v(t) = 0$ for values of t outside the domain of interest. Note that the denominator corresponds to the autocorrelation at $s = 0$, so that $a(0) = 1$. The values $a(s)$ correspond to the empirical damping values of the suspension response derived from actual data. The values ω_0 and ζ are chosen to minimize error

$$(\omega_0, \zeta) = \arg \min_{0 \leq \zeta < 1, \omega \geq 0} \int_t (e^{-\zeta t} \sin(\omega_0 t) - a(t))^2\,dt \tag{2.4}$$

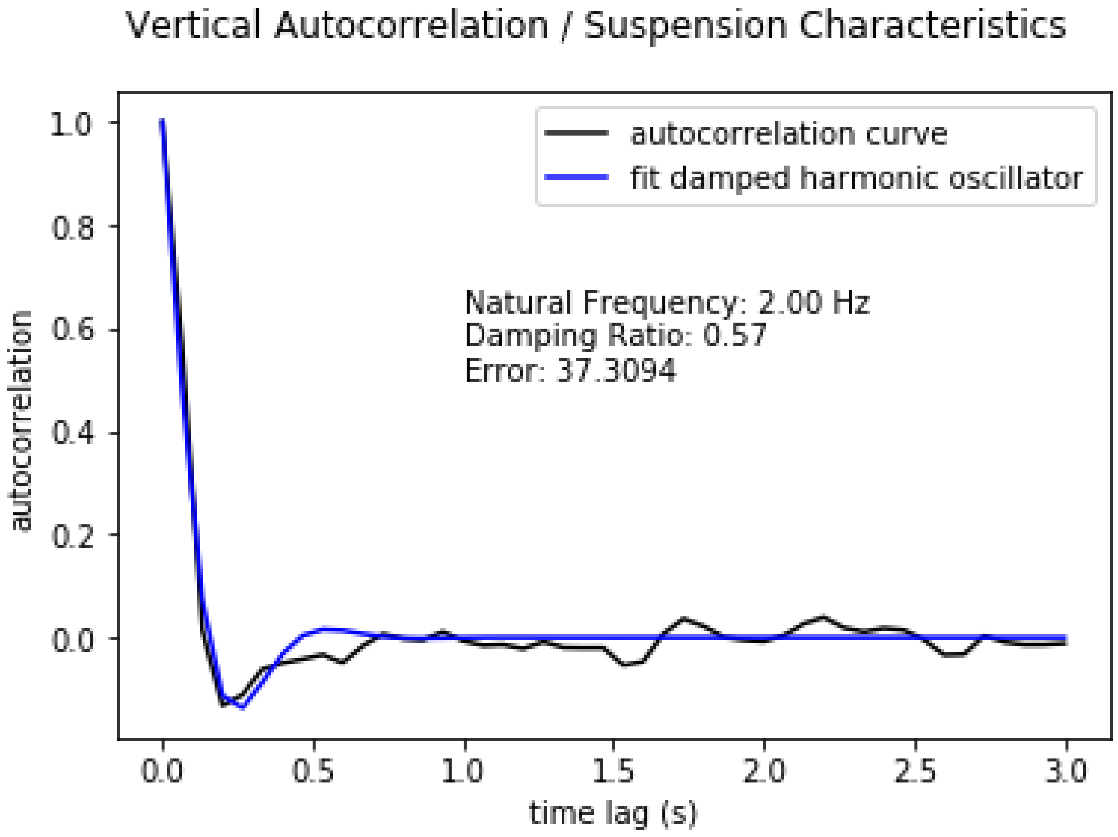

Figure 2-3: Suspension response over time

Since we use empirical data, it is inevitable that there are variations of the returned values accounting for measurement errors. However, there are patterns across the trips. For comfort cars, the damping ratio is typically low (at $0.2 - 0.3$) to maximize user comfort, while for offroad and race cars the damping ratio is higher (typically $0.5 - 0.7$) to quickly smooth the impact.

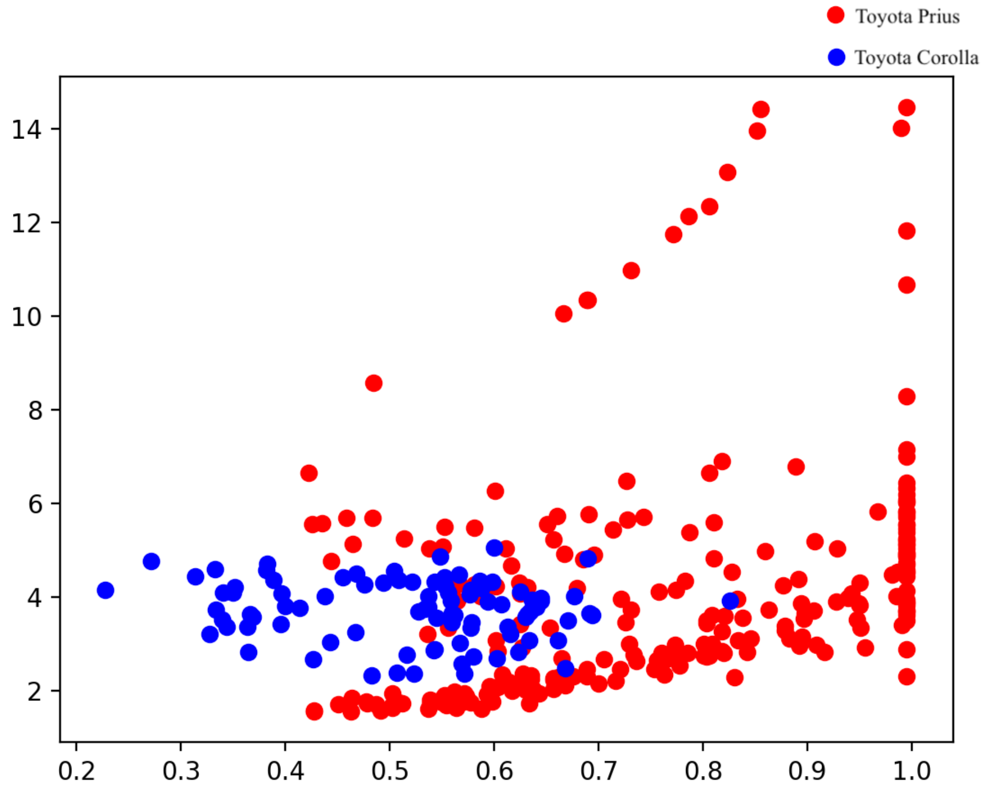

Figure 2-4: Statistical features of vertical acceleration: damping ratio (horizontal) and oscillation frequency (vertical)

Vertical acceleration manifests from many car-specific features, such as weight [9] and suspension response. Hence in addition to computing the damping coefficient and frequency, we could also compute statistical features of vertical acceleration. However, since vertical acceleration is affected by vehicle speed [10], we need to partition the vertical acceleration values using vehicle speed and collect their features separately.

Another issue is vehicle's weight. In practice, the reading from vertical acceleration comes from vehicle's load, which might include beside curb weight passenger's weight, fuel and extra loads. The last one is especially problematic for estimating parameters of SUV-type vehicle since the vehicle's weight varies significantly between different trips.

2.3.3 Power to weight ratio

By Newton's second law, the power can be represented as

$$P = Fv = ma_x v \tag{2.5}$$

However, using only accelerometer and GPS sensors, there is no obvious way to infer vehicle mass, so we need to settle for the power to weight ratio which is $P/W = a_x v$. Collecting such ratio for each valid sample, we have a timeseries representation on acceleration capacity and engine responsiveness of the vehicle. Since power to weight ratio can capture the instantaneous change of the engine, we consider it a more reliable metric than the conventional metrics, such as braking distance or 0 to 60 mph acceleration time, which require vehicles to run in controlled condition. We collect statistical features from the timeseries.

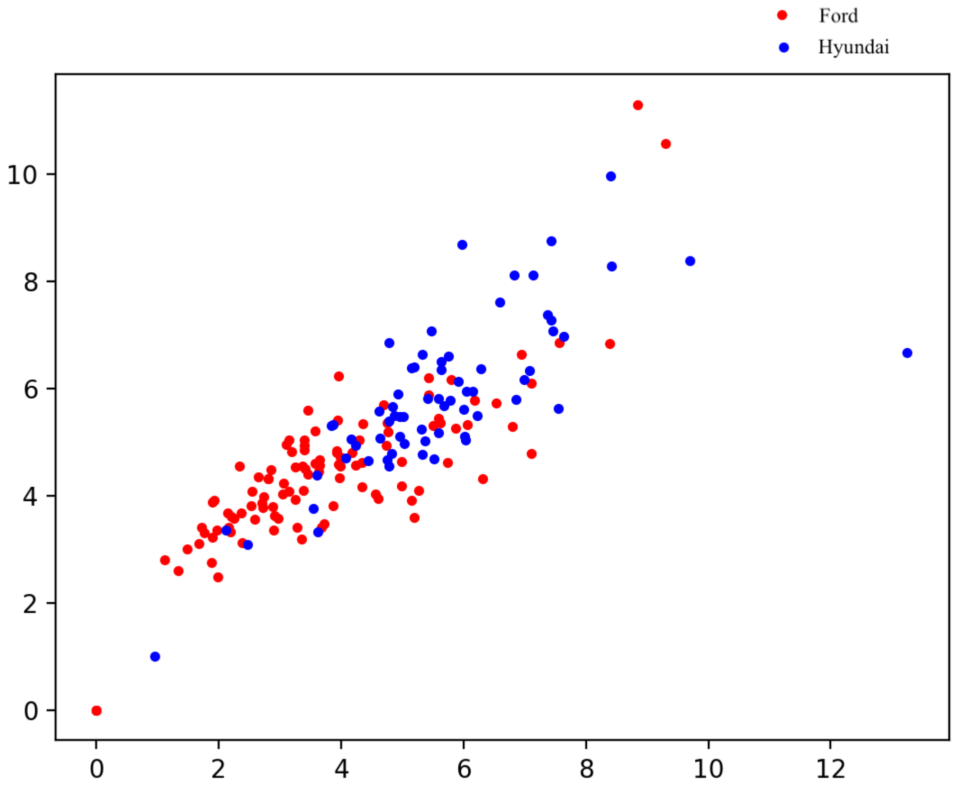

Figure 2-5: Power to weight ratio: mean (horizontal) and standard deviation (vertical).

Note that our empirical power to weight ratio is different from the power to weight ratio quoted from manufacturers, which is often measured at peak engine performance at curb weight (no driver on board). Nevertheless, it is an important measure, since power to weight ratio depends exclusively on engine performance. Comfort and compact cars often have lower power to weight ratio, while sport cars, luxury and SUV have high power to weight ratio to compensate for larger vehicle size.

2.3.4 Aerodynamics and Longitudinal Friction

Vehicle longitudinal dynamics follows the equation

$$F = ma_x = F_T - F_{aero} - F_R \tag{2.6}$$

where F_T is forward tire force, F_{aero} is aerodynamic drag and F_R is longitudinal rolling friction. At high speed, the dominant drag force is aerodynamic drag, which is proportional to the square of vehicle's velocity

$$F_{aero} = \frac{1}{2}\rho C_D A v^2 \tag{2.7}$$

where ρ is atmospheric density, C_D is vehicle's drag coefficient and A is vehicle frontal area. Information about vehicle aerodynamic specification can be found on table A.1. Certain types of vehicle, such as SUV, have higher drag area compared to other types. Therefore they need higher engine power to operate and is less responsive to brake and accelerate compared to other vehicle types. Statistical features of longitudinal acceleration and *square* of velocity would therefore capture the difference between vehicle types.

2.3.5 Lateral dynamics: steering features

Measuring vehicle handling is tricky, because the input impulse coming from steering has small magnitude and occurs in very short period of time. A natural approach would be to measure the *turn radius*, corresponding to how tight a vehicle can make a turn. There are two issues with this approach:

- Noises coming from driving behavior. This is a minor issue since turn radius tends to correlate with how tight a turn a driver will make.
- Noises coming from traffic. This is a big issue since traffic often blocks the vehicle from making small turn as designed. Traffic law also causes drivers to make left turn larger than right turn (assuming the law mandates drivers to drive on the right side of the road).

A better approach is to rely on statistical features from gyroscope sensor, in particular the yaw rate. Recall that the centrifugal acceleration is derived by the equation

$$a = \frac{v^2}{R} \tag{2.8}$$

where a is yaw rate, R is the radius of the turn and v is vehicle's speed. Therefore at any instant, v^2/a characterizes the vehicle's turning capability. Excluding small values of a (indicating vehicle is not turning or ensuring numerical stability), we can collect the statistical features of turn radius.

2.3.6 Autocorrelation coefficients

Previous features ignore the time dependent nature of the time series, which contains many important information about vehicle characteristics. For example, autocorrelation describes the vehicle wheelbase, since when the vehicle is excited by road bumps, the time lag between two consecutive bumps correlates with vehicle's wheelbase length. We compute the autocorrelation coefficients of vertical acceleration following the equation

$$c_d = \frac{\sum_{i=1}^{n} v[i]v[i+d]}{\sum_{i=1}^{n} v[i]^2} \tag{2.9}$$

(here we normalize $c_0 = 1$), and use the first five coefficients as features. Similar definitions can be made for other types of measurements.

2.3.7 Hard acceleration and hard braking

These features are time localized and characterize many of the characteristics of vehicles, as they direct correlate with braking and transmission of a vehicle. We define a hard acceleration as the longitudinal acceleration exceeds $0.5m/s^2$ and an acceleration frame as the consecutive period the hard acceleration exceeds such threshold. For each frame, we compute the duration and mean acceleration in that period, and aggregate over different frames using statistical extraction.

The same idea applies for braking events, using $-0.5m/s^2$ as threshold. Similarly, we can extract features with lateral acceleration and vertical acceleration as input. For lateral acceleration, event corresponds to a sharp left or right turn. For vertical acceleration, event corresponds to vehicle's response when excited by external road event.

2.3.8 Spectral analysis

The spectral content of a time series often contains rich information about time series' characteristics, making it a useful feature to compute. Spectral analysis has been widely applied in a number of domains, including image classification [8] and speech recognition [11]. In vehicles, spectral content comes from engine vibration, either when the vehicle is moving or at idle state. One interesting idea for vehicle model classification is to analyze the sound emitted by the engine as the vehicle moves, emitted via fluctuation of gyroscope. However, the sampling rate of sensors in this problem is too low to capture such information. Therefore we need to settle for lower frequency characteristic, such as idle state vibration which has frequency of $1-2$ Hz. As the vehicle can accumulate certain events, such as accelerating and braking, it is necessary to take Short Time Fourier Transform [11] instead of global Fourier Transform. We partition the time domain signal into short overlapping frames and apply Fourier Transform independently on each frame. Taking overlapping frames is necessary to mitigate the artificial boundary from creating frames.

On each frame, we compute spectral energy, spectral centroid and spectral variance, and aggregate over different frames using statistical extraction. We also compute the spectral flux across the frames, which characterizes the change of spectral content over time. The details on how to compute these features are described in Appendix A.2.

2.4 Some discussions on feature engineering

Although we make the best effort to extract features from trips, the signal of some trips is simply corrupted, rendering them unable to extract features. In such cases, the algorithm discards the entire trip from consideration. Experiments shows that, with the given set of features, only 10 percent of the trips are discarded.

The discrimination accuracy can be improved on some special cases by including metadata features, for example time of day, trip duration or type of road. The intuition is that, within a single driver, there are consistent driving behaviors associated with each vehicle model. However, as the final goal is to build a classifier on vehicle type, utilizing data from all drivers, utilizing these features will result in the classifier overfit toward the specified drivers in the data set. Hence those features are not taken into account when building classifier and are only used per user basis.

2.5 Algorithmic Discussion

2.5.1 Granularity level

A challenge in classification is to decide at which level of granularity the algorithm should work on. Directly using vehicle make and model would be too granular, as there are more than 800 distinct vehicle models, and the usage frequency differs wildly between different models. In addition, with too few drivers driving a certain vehicle model, the classifier risks overfitting for these specific drivers. Likewise, selecting vehicle manufacturer as label is also not a good option, since within the same manufacturer there are multiple types of vehicles, each having very distinct vehicle characteristics.

Instead, we restrict the granularity at *vehicle type*; that is, we classify whether a trip is driven by a compact, sedan or SUV. We manually label some of the popular vehicle models with their corresponding vehicle type and build the corpus using only these vehicle models. On this table and for subsequent chapters, we report only vehicle make and model, ignoring internal variants within vehicle model. This includes year of manufacturing, engine power or number of doors in the vehicle.

Vehicle model	**Vehicle type**
VOLKSWAGEN POLO	sedan
FORD FIESTA	sedan
HYUNDAI I20	sedan
FORD RANGER	SUV
VOLKSWAGEN GOLF	sedan
AUDI A4	compact
BMW 320I	sedan
FORD ECOSPORT	SUV
TOYOTA COROLLA	compact
HONDA JAZZ	sedan
AUDI A3	compact
KIA RIO	compact
FORD FIGO	sedan
LAND ROVER DISCOVERY	SUV
BMW 320D	compact
OPEL CORSA	sedan
FORD FOCUS	compact
HYUNDAI IX35	sedan
TOYOTA FORTUNER	SUV
VOLKSWAGEN TIGUAN	SUV
MERCEDES-BENZ C180	compact
RENAULT CLIO	sedan
TOYOTA YARIS	compact
NISSAN QASHQAI	SUV
KIA PICANTO	SUV

Table 2.3: List of popular vehicle and their type

This list can be potentially expanded, both in term of vehicle make/model and their corresponding label classes with minimal change in the algorithm. Here we choose the partition based on similar vehicle characteristics of the corresponding type. This classification is not perfect; however, as some of the listed vehicle models share characteristics of two different vehicle types.

2.5.2 Classification

Classification is a classic problem in machine learning with many available approaches. For this problem, we build a Random Forest classifier [1] thanks to its ability to process heterogeneous data types. Using the classifier, for each trip we obtain a probability distribution over types of vehicles.

Since the classifier is trained on the generic case, it ignores many user-based information, which we could introduce during the classification step. For example, having knowledge on upper bound of number of vehicles an user has can help restrict the hypothesis space. Suppose we have a classifier, modeled as a function $h : X \times Y \to [0, 1]$ where X is the space of all trip features, and Y is the space of all possible labels. For each $x \in X$, the classifier outputs a probability distribution over Y, that is $\sum_{y \in Y} h(x, y) = 1$, and denote $p(x) := \arg\max_{y \in Y} h(x, y)$. For a driver having trips $x_1, .., x_n$, assuming trips are taken independently, their joint likelihood is

$$\prod_{i=1}^{n} h(x_i, p(x_i)) \tag{2.10}$$

The key observation is that, the set $M = \{p(x_1), .., p(x_n)\}$ corresponds to the vehicles the driver uses, hence its cardinality could not be exceedingly large. A reasonable assumption is to restrict to $|M| \leq k << |Y|$ (if $|Y|$ is large) and reverse the process by searching for all k-subset M of Y and compute the joint probability

$$P(x_1, .., x_n, M) = \prod_{i=1}^{n} \max_{y_i \in M} h(x_i, y_i) \tag{2.11}$$

Choose M_0 that maximizes $P(x_1, .., x_n, M_0)$ and normalize the likelihood of vehicle types of the trip of interest.

2.5.3 Applying heuristic correction

So far, in prediction, we only use telemetry information. This ignores metadata of the trip, such as time of day that the trip takes place, location, duration and distance. Since driver's behavior follows predictable patterns, we could find specific heuristics that, with high confidence, group certain trips into one group sharing the same vehicle. The key is to consider their driving history as a sequence of trips, and find correlations between consecutive trips.

We apply two notable heuristics here:

1. Consecutive matching: if two trips are close in time and the start location of the second trip is close to the end location of the first trip, it is likely the the driver picks up the same vehicle for the later trip, hence two trips come from the same vehicle.

2. Trajectory matching: assuming that some trajectories the driver is likely to repeat over time, we could assign trips having similar trajectories (in either direction) to be driven by the same vehicle. This can be simply implemented at good accuracy by checking several major locations, such as start and end location. To avoid having to search through many trips, we only consider trips within a window of 3 days.

Although the equivalence relation introduced by the two heuristics is not necessarily transitive, we could nevertheless group all such linked trips to the same vehicle. To assign the cluster label for these trips, we calculate the joint probability

$$P(x_1 = c, .., x_n = c) = \prod_{i=1}^{n} h(x_i, c) \tag{2.12}$$

choose label c maximizing the joint probability, and assign all trips in the group with label c.

2.5.4 Other approaches

For comparisons, we also implement alternative algorithms. These approaches also help reveal the nature of dataset and characteristics of discriminative features.

- Raw value: for each trip, create a feature vector consisting of sensor's measurement without any feature engineering. We pick an interval of 2 minute and use three accelerometer sensors, thus having a feature vector of $2 \times 60 \times 15 \times 3 = 5400$ elements. We train a Random Forest classifier based on these features.

- Feature engineering-based algorithms, but with some components removed. We implement two cases, one with only statistical features, and another combining statistical features and event-based features (but without spectrogram features).

- 1-dimensional Convolutional Neural Network (1D-CNN). This approach has achieved success in classifying trips by driving style [2]. In deep learning-based algorithms, instead of doing extensive hand-crafted feature engineering, one can instead implement a neural network that implicitly learns such features during training, automatically choosing the right features depending on specific applications.

  In this problem, we select a 2-minute segment of the trip, which is further divided into frames of 2 second long with 1 second overlapping between consecutive frames. In each frame, we compute statistical features of the measurements and arrange the features to form a statistical feature matrix. We apply convolution and max pooling across frames only in time domain. The results after convolution and pooling is connected to fully connected layers and subsequently the output layer.

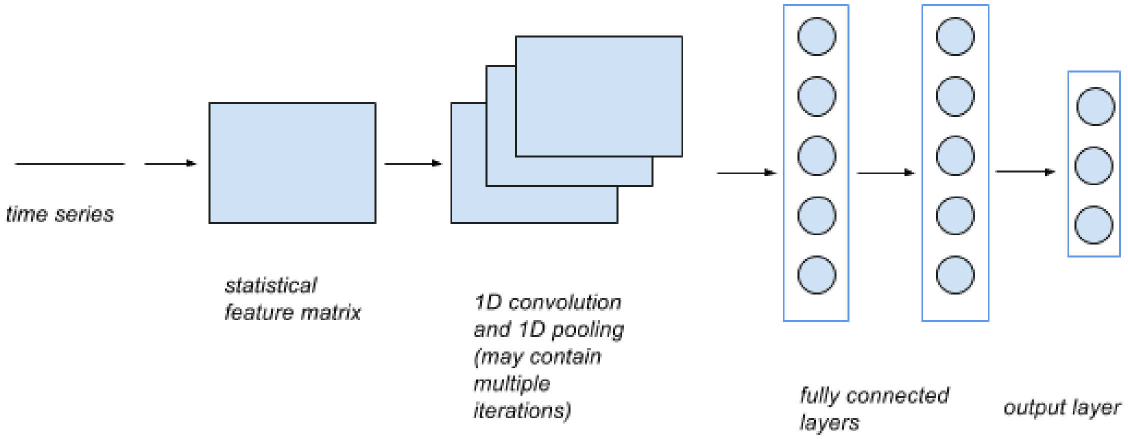
time series
statistical feature matrix
1D convolution and 1D pooling (may contain multiple iterations)
fully connected layers
output layer

Figure 2-6: Architecture of 1D Convolutional Neural Network

2.6 Influence of Driver on Vehicle Identification

Throughout Section 2.3, we implicitly extract features containing driver input, despite performing necessary engineering techniques to reduce their influence. However; since driver input is a significant part of telemetry signal, the natural question arises: how big is their influence into vehicle identification? There are two cases, trips containing only a single driver, and trips coming from multiple drivers.

We argue that, if restricting to the same driver, a supervised method would still give good classification results. The reason is that driving style is consistent on a driver, and by conditioning on driver the remaining signal manifests the difference between vehicle models.

On the other hand, if the dataset contains trips from multiple users, classification becomes significantly harder. Different drivers own different variants of the same vehicle model, and even on the same vehicle model their driving style is unique and is not easy to factor out. Therefore, in addition to building the classifier, choosing the right granularity is also crucial for applying to user vehicle identification. Experiments in Chapter 3 will justify these claims.

Chapter 3

Results

3.1 Overview of experiment setup

As discussed in the Chapter 2, a classification or clustering algorithm needs to be robust in various conditions. We also assess that driving style might be a major factor affecting the classification accuracy. Therefore, we design a suite of tests covering the following scenarios:

1. Same driver test, with the same driver driving multiple vehicle models. The classifier is expected to classify trips based on vehicle models.

2. Driving style test, where trip history comes from multiple drivers, labeled by the driver. The classifier is expected to classify trips by their corresponding drivers.

3. Vehicle model test, where trip history comes from several predetermined vehicle models, each is driven by many drivers. The classifier is expected to classify trips by their corresponding vehicle models.

4. Vehicle type test, where trip history comes from many vehicle models, each is labeled by its vehicle type. The classifier is expected to classify trips by their corresponding vehicle type.

Finally, we apply the classifier obtained in step 4, combined with additional heuristics for user vehicle identification. We report the clustering accuracy without and with heuristics.

For experiments, we typically restrict a smaller data set due to computational constraints. On each test, we collect data conforming to the testing scheme described, split into training and testing data and report accuracy at 10-fold cross validation (CV). The accuracy here indicates the percentage of trips classified with their correct label. We find that the accuracy plateaus with sufficient data. All the analysis are done using Amazon AWS c4.x8large instance.

3.2 Classification

3.2.1 Same driver test

We run multiple tests, on each test we select a driver driving regularly at least two vehicle models (that each vehicle model represents at least 10 percent the total number of trips). We select two most popular models per user and balance their vehicle representativeness in data. The classifier is trained using Random Forest with all the features described in Chapter 2. The following accuracy is reported per pair of vehicles, driven by the same user.

As shown here, conditioned on the same driver, the classifier is able to differentiate vehicle models at high accuracy. Although all tests are designed with only two vehicle models, it is trivial to extend to multiple vehicle models, accepting a marginal drop of accuracy. Hence the problem can be solved efficiently if for *each* driver there is sufficient labeled data about trip history per vehicle model (about 20 trips per vehicle). One can build a classifier per user and apply that on user vehicle identification.

What remains a hard question is to identify vehicle models on users without any

Vehicle Model 1	Vehicle Model 2	Accuracy (10-fold CV)
HONDA CIVIC	MITSUBISHI PAJERO	79.8
TOYOYA CAMRY	HONDA JAZZ	84.2
BMW 435I	BMW 550I	87.0
VOLKSWAGEN AMAROK	MERCEDES-BENZ C200	79.3
HYUNDAI SANTE	FIAT BRAVO	84.8
FORD FIGO	KIA RIO	67.2
KIA SEDONA	PEUGEOT 107	87.8
BMW 320D	TOYOTA RUNX	87.2

Table 3.1: Classification results of same driver test

labeled data.

3.2.2 Driving style test

We collect trip history of several drivers, labeling trip by the driver regardless of the vehicle model they are using. We select 100 trips per driver, running Random Forest classifier and report the accuracy measured by 10-fold CV. As shown here, the method reports good accuracy on classifying driving style.

Number of drivers	Accuracy (10-fold CV)
2	95.3
5	77.1
10	57.5

Table 3.2: Classification results of driving style test

3.2.3 Vehicle model test

We run the experiment with multiple pairs of vehicles. In each test, we collect 2000 trips per vehicle model, subject to no more than 30 trips coming from the same driver. We train the classifier using Random Forest classifier.

The accuracy drop compared to the same driver test suggests that the proposed fea-

Vehicle Model 1	Vehicle Model 2	Accuracy (10-fold CV)
BMW 320D	NISSDAN TIIDA	77.5
FORD FIESTA	MAZDA CX-3	52.1
KIA RIO	ISUZU KB250	71.2
HYUNDAI SANTE	KIA SOUL	67.3
AUDI A3	BMW Z4	75.6
HONDA JAZZ	MERCEDES-BENZ SLK	70.4
HYUNDAI I20	LAND ROVER RANGE	77.0
AUDI A4	HONDA CIVIC	59.8

Table 3.3: Classification results of vehicle model test (many drivers)

ture engineering approach does take driver characteristic into account, which accounts for more variance among drives in the same class. The result also shows that the classification accuracy is higher on pairs of vehicles of different types, suggesting that a classifier by vehicle type, albeit noisy, could still serve as a good indicator for user vehicle identification problem.

3.2.4 Vehicle type test

In this experiment, we sample 20000 trips from each type of vehicle, using only vehicle models listed on Table 2.3 and conditioned that no driver has more than 30 trips in the dataset. We then build a classifier on vehicle type. Here, there are three different vehicle types: SUV, compact and sedan. The result is listed as the percentage of trips having vehicle type classified correctly.

Algorithm	Accuracy (10-fold CV)
Raw value	33.5
1D-CNN	35.0
Basic + events	40.5
Basic + events + spectrogram	45.0

Table 3.4: Classification results of vehicle type test

In table 3.4, we use the following shorthand notation:

- Basic: indicate all features collected via statistical extraction methods and time-dependent features, mainly vehicle dynamics features, but excluding spectral features.
- Events: indicate event-based features, such as hard acceleration and braking.
- Spectrogram: indicate features obtained from computing spectrogram.

As shown here, directly using raw value does not give any better predictive ability than random guessing. While CNN and basic features help obtaining some discriminate accuracy, the significant contribution comes from using vehicle's short time response, manifested through spectral features.

3.3 Clustering

We apply the classifier in 3.2.4 into clustering problem. To evaluate the results, we need to distinguish between users having one vehicle and users having two or more vehicles, since evaluation metric differs.

For users having only one vehicle, the metric the ratio between the size of the largest cluster and total number of trips. In this case, without heuristic the average ratio is 0.75 and with heuristic the average ratio is 0.9, implying the classifier approach does recognize there is only one cluster.

For users having two or more vehicles, we need to compare obtained clusters with ground truth data, subject to permutations of labels. By constructing the confusion matrix and sum over permutation having the largest size, divided by total number of trips, we find that without heuristic the average ratio is 0.55 and with heuristic the average ratio is 0.60. In this case, the classifier recognizes different vehicles to some extent.

The result shows that the classifier tends to assign trips by the same vehicle to different clusters, hence the heuristic can correct to some extent. A more robust classifier would likely to improve the identification accuracy. In conclusion, there is a limiting factor on accuracy obtained with multiple vehicles, and a supervised approach as described in section 3.2.1 would yield a better result.

Chapter 4

Discussions and Extensions

The thesis has described an algorithm for vehicle model classification. The approach only requires data collected from smartphone sensors with simple set up, enabling its scalability and ubiquity in various environments. In addition to traditional techniques in many other machine learning problems, in this particular problem, the success of the algorithm combines both study of vehicle dynamics and understanding of driver's usage pattern, the later is to compensate for difficulties of implementing a "pure" machine learning algorithm. A simple extension of the algorithm allows for classification of *transportation mode*, such as train, bike or walking.

It is interesting to see the variations considering different phone positions (for example, hand or pocket) and different smartphone models (for example, Android versus iPhone). While the basic measurements are the same, different smartphone models also apply different algorithms for motion detection or filtering noise. While this problem did not take into account difference between smartphone models, it remains an interesting question to distinguish the difference on data quality collected by different smartphone models and how it affects classification results.

In practice, an user-input trip may alternate between different modes of transportation (such as car to bus or train). Even when using only a single vehicle in a trip, not all collected data comes exclusively from driving; for example, an user can stop the vehicle at gas station, refuel and come back. *Trip segmentation*, which separates different modes of transportation interleaving in a given trip, would improve the analysis accuracy and give more insights on users' driving behavior.

Our technique on time series analysis often extracts the features from a single time series one at a time. Vectorized approach, which extracts features of multiple time series could provide further insights and relations between different measurements of the vehicle. Likewise, the features obtained during extraction step only loosely depends on vehicle dynamics. A more systematic approach would be constructing a vehicle dynamical model, and infer underlying parameters.

In addition to classifying vehicle types, we can apply the same idea to estimate vehicle's parameters, such as curb weight, dimensions and aerodynamics coefficients. We did not do these estimations in the thesis due to its difficulty to obtain ground truth data, since there are multiple conflicting information sources and for many vehicle models all the parameters are not available.

Although certain aspects of user behavior are considered to aid classification, these properties are often case-specific and heuristic. Having a systematic approach in studying user behavior would be useful in implementing more robust vehicle identification models and help unveil the way drivers use their vehicles.

Bibliography

[1] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[2] Weishan Dong, Jian Li, Renjie Yao, Changsheng Li, Ting Yuan, and Lanjun Wang. Characterizing driving styles with deep learning. *arXiv preprint arXiv:1607.03611*, 2016.

[3] J Ferreira and E Minike. Pay-as-you-drive auto insurance in massachusetts: A risk assessment and report on consumer, industry and environmental benefits. department of urban studies and planning, massachusetts institute of technology. *Massachusetts Institute of Technology (http://dusp. mit. edu/) for the Conservation Law Foundation, http://www. clf. org/, http://www. clf. org/our-work/healthy-communities/modernizing-transportation/pay-as-you-drive-auto-insurance-payd*, 2010.

[4] National Research Council (US). Committee for the Study of a Motor Vehicle Rollover Rating System. *The National Highway Traffic Safety Administration's Rating System for Rollover Resistance: An Assessment*, volume 265. Transportation Research Board, 2002.

[5] Giancarlo Genta. *Motor vehicle dynamics: modeling and simulation*, volume 43. World Scientific, 1997.

[6] Samuli Hemminki, Petteri Nurmi, and Sasu Tarkoma. Accelerometer-based transportation mode detection on smartphones. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, page 13. ACM, 2013.

[7] Bret Hull, Vladimir Bychkovsky, Yang Zhang, Kevin Chen, Michel Goraczko, Allen Miu, Eugene Shih, Hari Balakrishnan, and Samuel Madden. Cartel: a distributed mobile sensor computing system. In *Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 125–138. ACM, 2006.

[8] Dengsheng Lu and Qihao Weng. A survey of image classification methods and techniques for improving classification performance. *International journal of Remote sensing*, 28(5):823–870, 2007.

[9] Phong X Nguyen, Takayuki Akiyama, Hiroki Ohashi, Masaaki Yamamoto, and Akiko Sato. Vehicle's weight estimation using smartphone's acceleration data to control overloading. *International Journal of Intelligent Transportation Systems Research*, pages 1–12, 2015.

[10] Hiroki Ohashi, Takayuki Akiyama, Masaaki Yamamoto, and Akiko Sato. Modality classification method based on the model of vibration generation while vehicles are running. In *Proceedings of the Sixth ACM SIGSPATIAL International Workshop on Computational Transportation Science*, page 37. ACM, 2013.

[11] Geoffroy Peeters. A large set of audio features for sound description (similarity and classification) in the cuidado project. 2004.

[12] Rajesh Rajamani. *Vehicle dynamics and control.* Springer Science & Business Media, 2011.

[13] Marie C Walz. Trends in the static stability factor of passenger cars, light trucks, and vans. Technical report, 2005.

[14] Robert C Weast et al. Handbook of physics and chemistry. *CRC Press, Boca Raton, 1983–1984*, 1986.

[15] Robert A White and Helmut Hans Korst. The determination of vehicle drag contributions from coast-down tests. Technical report, SAE Technical Paper, 1972.

Appendix A

Formal Derivations

This section presents formally the necessary mechanical and mathematical knowledge for models used in the problem.

A.1 Vehicle Dynamics

Several formula is derived from [12], which explains vehicle dynamics in great detail.

A.1.1 Longitudinal Vehicle Dynamics

Longitudinal dynamics quantifies characteristics of vehicle engine power and longitudinal acceleration. In this model, we assume the quarter car model. Assuming road pitch is zero, the longitudinal forces acting on a vehicle can be described by the equation

$$F = ma_x = F_T - F_{aero} - F_R \tag{A.1}$$

where

- F_T is tire force.
- F_{aero} is the aerodynamic drag.
- F_R is rolling friction.
- m is vehicle's mass.

- a_x is vehicle's longitudinal acceleration.

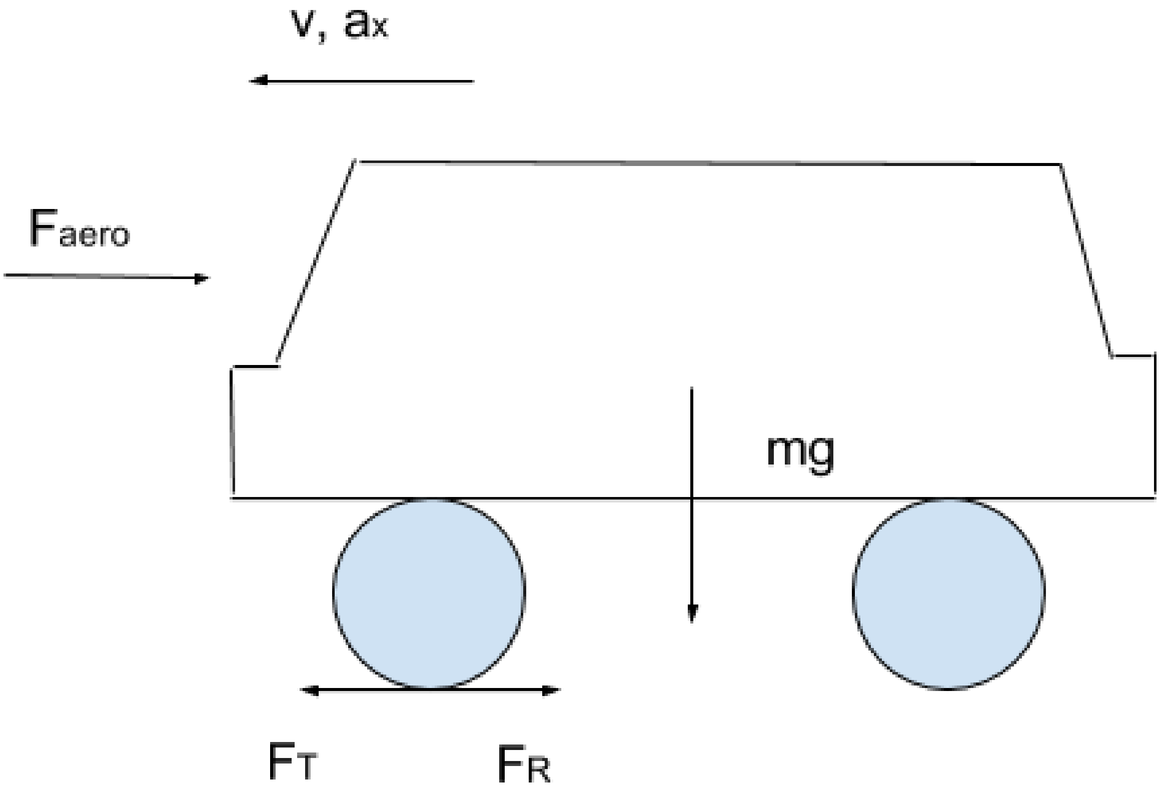

Figure A-1: Longitudinal dynamics equation

(Note that in the diagram, we combine forces acting on front and rear tires for simplification).

The rolling friction occurs due to friction between the tire and the road and is proportional to the normal force acting on vehicle, hence can be described as

$$F_R = -c_R mg \tag{A.2}$$

where c_R is rolling resistance coefficient.

The aerodynamic drag is proportional to the *square* of velocity

$$F_d = -\frac{1}{2}\rho(v + v_{wind})^2 C_D A \tag{A.3}$$

Here ρ is air density, v_{wind} is wind's velocity (with positive value indicates wind direction against vehicle's motion) C_D is vehicle's drag coefficient and A is vehicle's frontal area. The quantity C_DA can be empirically determined by a coast down test [15]. Typical experiments assume ρ is constant value at $101.325kPa$, measured at sea level and temperature of 15 degree Celcius. [14]

In both equations, the minus sign indicates the force act against vehicle motion.

Experiments show that tire force is generated by slip force, which comes as difference between tire rotational velocity and longitudinal velocity of vehicle's axle. The difference is $r\omega - v$ where r is tire's radius and ω is tire's angular velocity. Longitudinal slip ratio is then defined

$$\sigma = \frac{r\omega - v}{v} \text{ if the vehicle is braking} \tag{A.4}$$

$$\sigma = \frac{r\omega - v}{r\omega} \text{ if the vehicle is accelerating} \tag{A.5}$$

The tire force is then calculated

$$F_T = C_\sigma \sigma \tag{A.6}$$

where C_σ is longitudinal tire stiffness.

A.1.2 Designs of Passive Suspension

When a vehicle travels on the road, it is subject to perturbation due to road input. The goal of suspension is to absorb such perturbation, which makes a ride more comfortable and ensures vehicle control. Ride qualify can be quantified by measurements of vertical acceleration.

A *passive suspension* can be modeled as a spring-mass system. While a passive

Vehicle model	**Drag Coefficient**	**Frontal area** (m^2)	**CdA** (m^2)
VOLKSWAGEN POLO	0.32	2.04	0.65
FORD FIESTA	0.32	2.15	0.69
HYUNDAI I20	0.30	2.55	0.76
FORD RANGER	0.49	2.40	0.96
AUDI A4	0.27	2.20	0.59
BMW 320I	0.28	2.20	0.62
FORD ECOSPORT	0.37	2.90	1.07
TOYOTA COROLLA	0.29	2.09	0.60
AUDI A3	0.31	2.08	0.64
LAND ROVER DISCOVERY	0.36	3.84	1.38
BMW 320D	0.31	2.06	0.64
OPEL CORSA	0.32	1.96	0.62
FORD FOCUS	0.32	2.11	0.67
TOYOTA FORTUNER	0.38	3.40	1.29
VOLKSWAGEN TIGUAN	0.37	2.54	0.94
MERCEDES-BENZ C180	0.30	2.05	0.61
RENAULT CLIO	0.33	1.86	0.61
TOYOTA YARIS	0.29	2.14	0.62
NISSAN QASHQAI	0.33	2.88	0.95
KIA PICANTO	0.34	1.98	0.67

Table A.1: List of vehicles and their aerodynamic information

suspension purely absorbs road perturbation, an *active suspension* could induce actuator to damp external force by electronic control. In this section, we only consider passive suspension. Using a quarter car model, its parameters represent

- m_s is equivalent to vehicle body mass.
- m_u is axle mass.
- k_s is coefficient of suspension.
- k_u is stiffness of tire.
- b_s is damping factor.

Alternative to the quarter car model is the half car model, which includes both front and rear suspension. As shown in Chapter 2, the latency between acceleration of front versus rear suspension can be used to estimate vehicle's wheelbase.

The parameters in the half car model includes

- k_{t1}, k_{t2} are stiffness of front and rear tire.
- m_{u1}, m_{u2} are front and rear axle mass.
- k_1, k_2 are coefficient of front and rear suspension.
- m is vehicle body mass.
- ℓ_f, ℓ_r are distance of front and rear suspension to center of mass. Consequently, $\ell_f + \ell_r$ corresponds to vehicle's wheelbase.

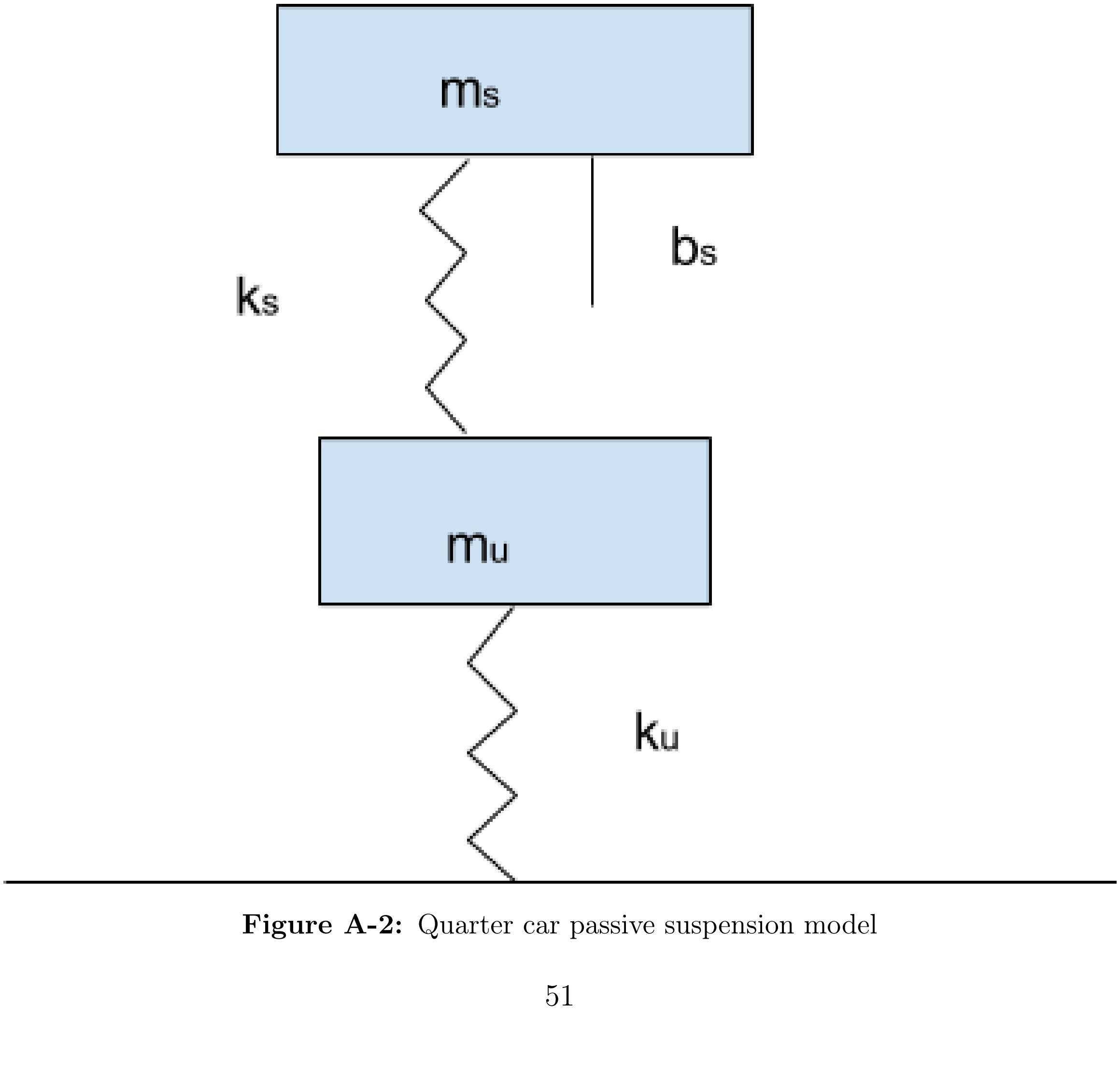

Figure A-2: Quarter car passive suspension model

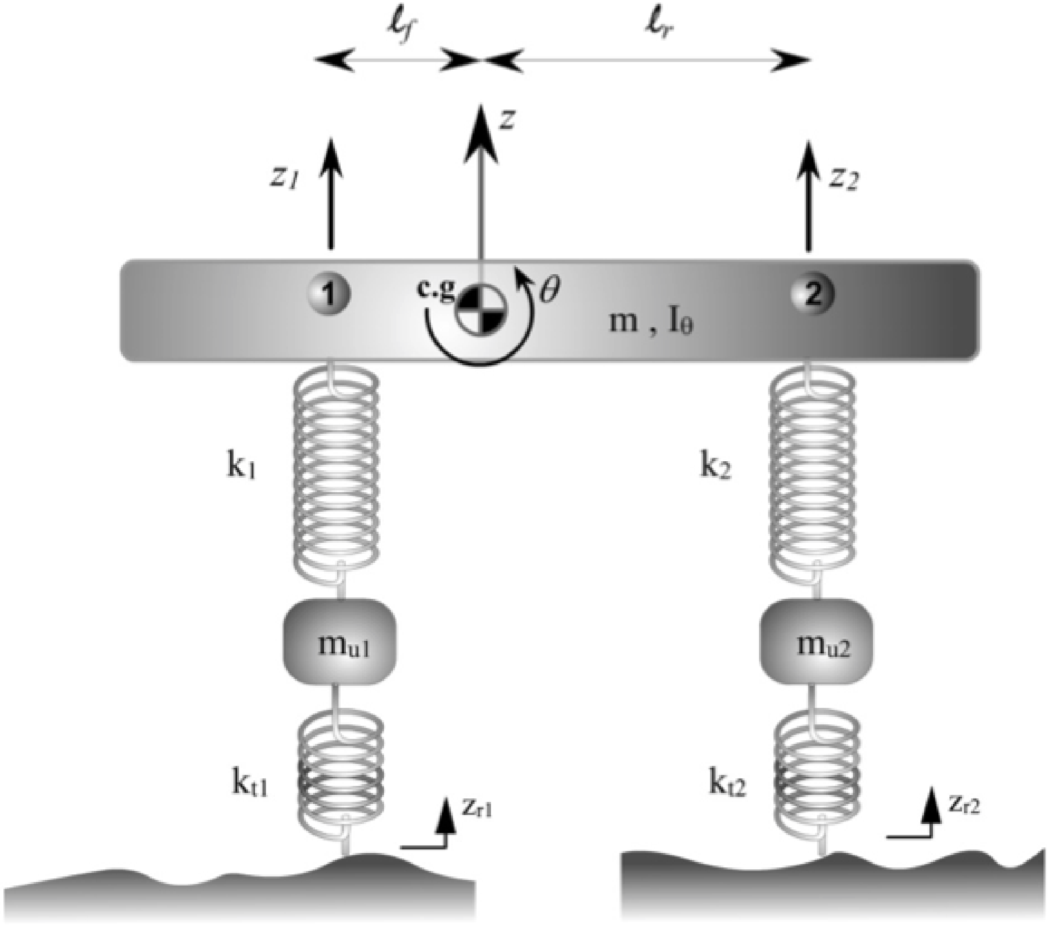

Figure A-3: Half car passive suspension model [12]

A.1.3 Roll Dynamics

Rolling is one of the major cause for fatal accidents. Roll occurs when vehicle can no longer keep balance along the axis along vehicle's body. Controlling vehicle roll is crucial for traction and vehicle stability.

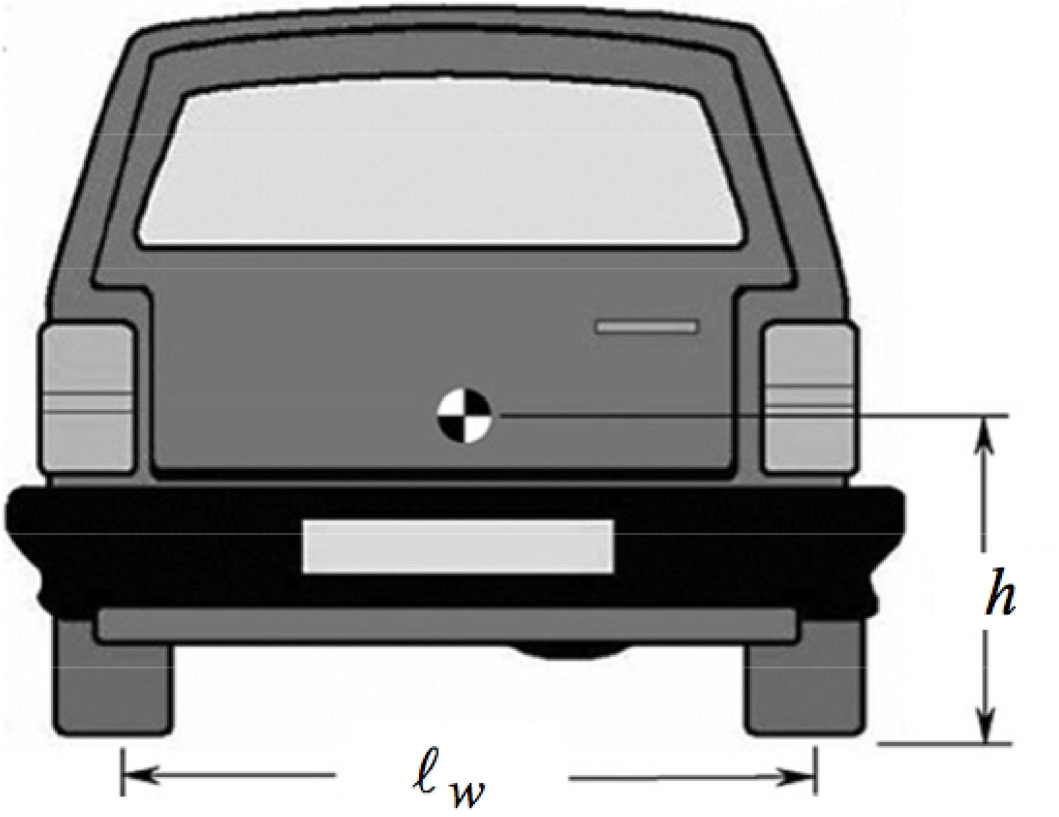

Figure A-4: Basic quantities for rolling dynamics [4]

Intuitively, a vehicle is less likely to roll with wider track width and lower height. Formally, roll stability is quantified by *static stability factor* (SSF), defined as

$$SSF = \frac{\ell_w}{2h} \tag{A.7}$$

where

- ℓ_w is vehicle's track width.
- h is the height of vehicle's center of gravity.

SSF consequently defines lift off acceleration, or the threshold of lateral acceleration in which a rollover occurs.

$$a_{y-lift-off} = SSF \cdot g = \frac{\ell_w}{2h} g \tag{A.8}$$

Note that the above quantity is purely based on geometrical shape of the vehicle and ignores roll preventive mechanisms, such as electronic stability control.

Rollover risk	Crash likelihood	SSF
5 stars	less than 10 percent	> 1.44
4 stars	10-20 percent	$1.25 - 1.44$
3 stars	20-30 percent	$1.13 - 1.24$
2 stars	30-40 percent	$1.04 - 1.12$
1 star	more than 40 percent	< 1.04

Table A.2: SSF and rollover rating (static test) [4]

Vehicle type	SSF
Passenger Cars	1.41
SUVs	1.17
Pickup Trucks	1.18
Mini Vans	1.24
Full Vans	1.12

Table A.3: Average SSF by vehicle type, model year 2003 [13]

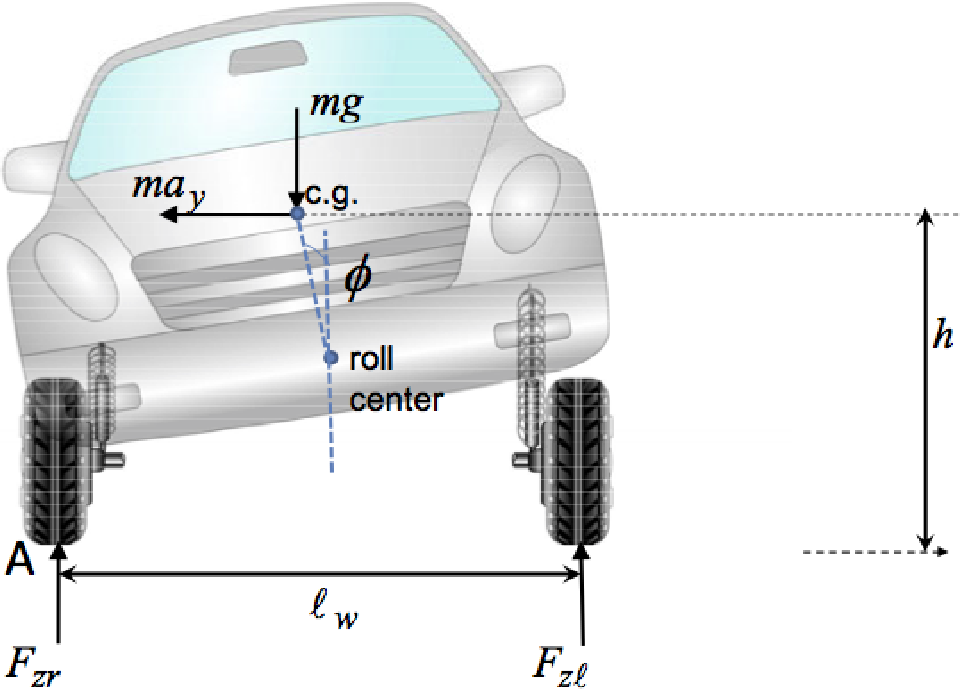

Figure A-5: Balance equation describing roll moment [12]

SSF can be derived from the roll moment balance equation. Here, the vehicle has lateral acceleration a_y, and the load on the inner and outer tire is F_{zl} and F_{zr}, respectively. The moment equation at the bottom of the outer tire is

$$ma_y h + F_{zl}\ell_w - mg\frac{\ell_w}{2} = 0 \tag{A.9}$$

hence the force on the inner tire is

$$F_{zl} = \frac{mg\frac{\ell_w}{2} - ma_y h}{\ell_w} \tag{A.10}$$

Setting $F_{zl} = 0$, it follows the threshold acceleration causing rollover to occur is $a_y = \frac{\ell_w}{2h}g$.

A.2 Short time Fourier Transform

For a time series T, partition it into possibly overlapping short frames $c_1, .., c_k$. On each frame, apply Fourier Transform and take the absolute value of the coefficients. Denote the transformed frames as $d_1, .., d_k$, where the coefficients for frame i is $d_{i1}, .., d_{im}$ with m is the number of coefficients. We apply the following feature extractions. To simplify the notation, for all following features (except spectral flux), we consider a single frame with coefficients $d_1, .., d_m$, and the values are aggregated across frames by statistical extractions.

- Spectral centroid, which computes the weighted mean

$$\mu = \frac{\sum_{j=1}^{m} j \cdot d_j}{\sum_{j=1}^{m} d_j} \tag{A.11}$$

- Spectral energy, which is the average sum of square of the coefficients in the frame:

$$R = \frac{1}{m}\sum_{j=1}^{m} d_j^2 \tag{A.12}$$

- Spectral spread, which is equivalent to standard deviation.

$$\sigma^2 = \sum_{j=1}^{m} (j - \mu)^2 d_j \tag{A.13}$$

- Spectral skew, which measures the skewness of the dataset. We first compute the third moment

$$m_3 = \sum_{j=1}^{m} (j - \mu)^3 d_j \tag{A.14}$$

  and then divide by third power of spectral spread: $\gamma_3 = \frac{m_3}{\sigma^3}$.

- Spectral kurtosis: we first compute the fourth moment

$$m_4 = \sum_{j=1}^{m} (j - \mu)^4 d_j \tag{A.15}$$

  and then divide by fourth power of spectral spread: $\gamma_4 = \frac{m_4}{\gamma_4}$

- Spectral flux, which characterizes the change of spectral content. For this feature, we consider all frames $c_1, c_2, ..$ in succession and compute

$$\frac{1}{k-1} \sum_{i=2}^{k} ||d_i - d_{i-1}||_2^2 \tag{A.16}$$