Analyzing driving behavior from CAN data using context-specific information

Ashraf Shaikh Mohammed

Follow this and additional works at: https://lib.dr.iastate.edu/creativecomponents

Part of the Digital Communications and Networking Commons

Recommended Citation

This Creative Component is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Creative Components by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
Analyzing driving behavior from CAN data using context-specific information

by

Ashraf Shaikh Mohammed

A creative component report submitted to the graduate faculty in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Computer Engineering

Program of Study Committee:

Dr. Daji Qiao, Major Professor

Iowa State University

Ames, Iowa

2019
Acknowledgement

I would like to thank my major professor, Dr. Daji Qiao for giving me an opportunity to work in this project and providing his guidance and knowledge throughout this project.

Sincere thanks to Andrew of Optical Operations, Ames, Iowa for being a key part of this project design by providing guidance and valuable suggestions and for lending his car for performing experiments and data collection.

I would also like to acknowledge my partners in this research work, Shankar and Archit for teaming up with me during various aspects of this project like data collection and exchanging of ideas for delivering quality results.

Last but not the least, I would like to thank the almighty, my parents and my friends for being a constant support throughout this journey.
## Contents

Acknowledgement .................................................................................................................. 2
List of Figures ............................................................................................................................ 4
List of Tables .............................................................................................................................. 5
Abstract ................................................................................................................................... 6
Introduction ............................................................................................................................... 7
  Controller Area Network (CAN) ............................................................................................ 7
  On-Board Diagnostics port (OBD) ......................................................................................... 7
  Telematics and Usage-based Insurance (UBI) ....................................................................... 8
Related works ............................................................................................................................ 9
  Software Solution for Monitoring and Analyzing Driver Behavior ...................................... 9
  Invisible Sensing of Vehicle Steering with Smartphones ..................................................... 10
  Modeling speed profiles of turning vehicles at signalized intersections .............................. 10
  Verizon Telematics Technical Information Bulletin ............................................................. 11
Implementation ........................................................................................................................ 12
  Objective ............................................................................................................................... 12
  Hardware Setup .................................................................................................................... 12
  Software Setup ..................................................................................................................... 13
  Test Route ............................................................................................................................. 14
Data Logging ............................................................................................................................. 16
  CAN data collection ............................................................................................................. 16
  CAN data post processing .................................................................................................. 16
  GPS data collection ............................................................................................................. 16
  Geofencing .......................................................................................................................... 16
  Automation ........................................................................................................................... 16
  Pre-Grading data processing .............................................................................................. 18
Grading ..................................................................................................................................... 19
  Thresholding ......................................................................................................................... 19
  Scoring ................................................................................................................................. 20
  The Grading Engine ............................................................................................................. 21
  Point Score calculation example ......................................................................................... 21
  Segment score comparison ................................................................................................. 27
  Lap scores comparison ........................................................................................................ 29
Conclusion .................................................................................................................................. 30
  Future work .......................................................................................................................... 30
References .................................................................................................................................. 31
List of Figures

Figure 1: OBD 2 Port Pinout..................................................................................................................7
Figure 2: OBD2 Message format............................................................................................................8
Figure 3: Safe driving results ................................................................................................................9
Figure 4: Aggressive driving results ...................................................................................................9
Figure 5: Gyroscope readings during different turns ............................................................................10
Figure 6: V-Sense state diagram for maneuver classification...............................................................10
Figure 7: 3rd degree polynomial used to fit the speed profile...............................................................11
Figure 8: Speed and acceleration over Time in observed and modeled data ...........................................11
Figure 9: Hardware setup ....................................................................................................................12
Figure 10: List of hardware components .............................................................................................13
Figure 11: Overall setup .......................................................................................................................13
Figure 12: Route tested for driving behavior analysis ..........................................................................14
Figure 13: Left and right turns from stop sign and stop lights ...............................................................14
Figure 14: CAN and GPS data logging .................................................................................................17
Figure 15: Data logging Automation ....................................................................................................17
Figure 16: Acceleration regression for turn/special segments ...............................................................19
Figure 17: Acceleration regression for regular segments .....................................................................19
Figure 18: Deceleration regression for regular segments ......................................................................20
Figure 19: Deceleration regression for turns ........................................................................................20
Figure 20: Segment 0 - Depot to Road ..................................................................................................22
Figure 21: Segment 0 - Jerk Plot ..........................................................................................................22
Figure 22: Segment 0 - Speed Plot .......................................................................................................22
Figure 23: Segment 0 - Acceleration Plot ............................................................................................23
Figure 24: Grading Software console output .........................................................................................24
Figure 25: Grading Software console output showing Lap 1 report ....................................................26
Figure 26: Segment #20 acceleration plots for lap 2 and 4 ..................................................................27
Figure 27: Segment #20 Jerk plots between laps 2 and 4 ....................................................................27
Figure 28: Segment #20 Speed plots between laps 2 and 4 .................................................................28
Figure 29: Bar graph showing scores from laps 1 to 5 .......................................................................29
List of Tables

Table 1: List of segments with their description ................................................................. 15
Table 2: Segment score calculation .................................................................................... 25
Table 3: Segment scores for Lap 1 ..................................................................................... 26
Table 4: Lap2 vs Lap 4 Score comparison .......................................................................... 28
Table 5: Laps 1-5 Score comparison .................................................................................. 29
Abstract

Typical telematics and fleet-management systems today use embedded systems attached to the vehicle and get their driving data from their diagnostics port to identify the action of driver and grade it to provide feedback based on their quality of driving to efficiently handle the vehicle and also their driving behavior.

Today’s insurance companies provide embedded devices or the customer’s smartphone to analyze basic driving parameters such as speed, rpm, GPS location to understand driver’s braking, acceleration and distance travelled over a period and use it to assess quotes for insurance premium.

But most of the solutions above do not consider context specific information in the cases of fixed-route scenarios whose details can be understood better in the first place and use it to grade the driver’s performance for the trip more efficiently.

In this experiment, a driver’s behavior on a pre-defined route is analyzed on different perspectives by also taking into account of the road context, such as turns, straight road segments, traffic lights, stop signs etc. and graded accordingly and providing a score to reflect their behavior in each segment of the road as well as a complete score for their trip.
Introduction

Vehicle driving behavior analysis involves combination of hardware and software technologies which help us get real-time feedback about the vehicle as well be able to profile a driver’s behavior. Some of the core technologies involved are mentioned below.

Controller Area Network (CAN)
A Controller Area Network (CAN bus) is a robust vehicle bus standard designed to allow microcontrollers and devices to communicate with each other in applications without a host computer. It is a message-based protocol, designed originally for multiplex electrical wiring within automobiles to save on copper, but can also be used in many other contexts.[1]

CAN bus technology has been in development since its conception by Robert Bosch in 1983. The more widely used version of CAN, CAN 2.0 which are further classified into CAN 2.0A and CAN 2.0B based on the identifier width supported.

The latest CAN standard is the CAN-FD or Flexible Data Rate introduced in 2012 which is backward compatible with the CAN 2.0 standard and can also co-exist as a separate bus with the CAN 2.0.

CAN bus is one of five protocols used in the on-board diagnostics (OBD)-II vehicle diagnostics standard. The OBD-II standard has been mandatory for all cars and light trucks sold in the United States since 1996.[1]

On-Board Diagnostics port (OBD)
On-board diagnostics (OBD) is an automotive term referring to a vehicle’s self-diagnostic and reporting capability. OBD systems give the vehicle owner or repair technician access to the status of the various vehicle subsystems.[2]

The OBD-II standard provides list of standardized DTCs. As a result of this standardization, a single device can query the on-board computer(s) for these parameters in any vehicle.

Manufacturers may also add custom data parameters to their specific OBD-II implementation, including real-time data requests as well as trouble codes.

![OBD 2 Port Pinout](image)

Typically, queries with particular PID values to get real-time data is performed to know the vehicles instantaneous speed and other engine parameters
While it did not meet the OBD-II requirements for U.S. vehicles prior to 2003, as of 2008 all vehicles sold in the US are required to implement CAN as one of their signaling protocols.

For OBD2 messages, the identifier is standard 11-bit and used to distinguish between “request messages” (ID 7DF) and “response messages” (ID 7E8 to 7EF).

There are 10 modes as described in the SAE J1979 OBD2 standard. Mode 1 shows Current Data and is e.g. used for looking at real-time vehicle speed, RPM etc. Other modes are used to e.g. show or clear stored diagnostic trouble codes and show freeze frame data.

For each mode, a list of standard OBD2 PIDs exist - e.g. in Mode 01 PID 0D is Vehicle Speed.

**Telematics and Usage-based Insurance (UBI)**

Telematics is a method of monitoring a vehicle. By combining a GPS system with on-board diagnostics it's possible to record – and map – exactly where a car is and how fast it's traveling, and cross reference that with how a car is behaving internally.[4]

Telematics is used in applications such as tracking vehicles, heavy-duty container trucks used for transportation and fleet management.

Fleet management is the management of a company's fleet. Fleet management includes the management of ships and or motor vehicles such as cars, vans and trucks. Fleet (vehicle) Management can include a range of functions, such as vehicle financing, vehicle maintenance, vehicle telematics (tracking and diagnostics), driver management, fuel management, health and safety management and dynamic vehicle scheduling.[5]

Usage-Based Insurance (UBI) is a type of auto insurance that tracks mileage and driving behaviors. UBI is often powered by in-vehicle telecommunication devices (telematics)-technology that is available in a vehicle that is self-installed using a plug in-device or already integrated in original equipment installed by car manufactures. It can also be available through mobile applications.

The basic idea of UBI is that a driver's behavior is monitored directly while the person drives, allowing insurers to more closely align driving behaviors with premium rates.[6]
Related works

Several works have been done before in the field of vehicle driving behavior analysis from just analyzing the driving behavior in general to focusing on certain aspects of driving such as turns, lane shifting etc.

Most of the works use data from various sources such as the data from OBD port, smartphone sensors, external sensors attached to embedded systems placed on the vehicle.

Some of the distinguishing works from various projects and research are highlighted below.

Software Solution for Monitoring and Analyzing Driver Behavior

The motive of [7] is that by giving to the user a reliable driving score, he can relate to it and be aware that there might be some problems in his actions, so he becomes more attentive.

The data is collected using an external device connected to the OBD port of the vehicles. Speed, engine rpm, gear position is collected from the vehicles directly. GPS position and vehicle rotation are collected using sensors from the external device mounted to the vehicle.

The data from all the sensors is mapped to a weighted graph called State Graph. This graph is used to detect anomalies by studying the weights of the edge.

An Arduino based logger, Freematics is used for communicating with the car using OBD2 port.

Each trip receives a total score obtained from four different scores: acceleration, breaking, left corners, right corners.

Figure 3: Safe driving results

Figure 4: Aggressive driving results
Invisible Sensing of Vehicle Steering with Smartphones

In [8], they developed a vehicle steering detection middleware called V-Sense which can run on commodity smartphones without additional sensors or infrastructure support. Instead of using cameras, the core of V-Sense/senses a vehicle’s steering by only utilizing non-vision sensors on the smartphone. Algorithms were designed and evaluated for detecting and differentiating various vehicle maneuvers, including lane-changes, turns, and driving on curvy roads.

V-Sense performance was tested on a Samsung Galaxy S4 with a 1.6GHz quad-core processor running Android 4.4.1 KitKat OS. Over 40 hours of test were conducted and tried to cover different environments both in a parking lot and real roads.

The cars involved in the test were a 2010 Mitsubishi Lancer and a 2006 Mazda 6.

![Gyroscope readings during different turns](image)

*Figure 5: Gyroscope readings during different turns*

![V-Sense state diagram for maneuver classification](image)

*Figure 6: V-Sense state diagram for maneuver classification*

Modeling speed profiles of turning vehicles at signalized intersections

[9] Considers that turning vehicles need special attention in the context of the safety evaluation and improvement of signalized intersections.

By using empirical data of vehicle trajectories collected at signalized intersections in Japan, a model is developed and presented, which provides stochastic speed profiles of free-flowing left- and right turning vehicles. The speed profiles are sensitive to intersection layout and the vehicle speed and position at the beginning and ending of the maneuver.

Based on the observations, a model describing the speed profile of unimpeded (free flowing) turning vehicles was developed. The speed profile is divided into two parts, an inflow part and an outflow part, the boundary defined by the moment the vehicle reaches the minimum speed. The acceleration of both parts was found to follow a parabolic shape.
A polynomial of third degree for the speed as a function of the time as shown in fulfills this requirement. Different coefficients are chosen for the inflow and the outflow.

\[ v = c_1t^3 + c_2t^2 + c_3t + c_4 \]

*Figure 7: 3rd degree polynomial used to fit the speed profile*

For the regression analysis the trajectory data had first to be classified into the inflow and outflow part of the individual speed profiles and cleaned for outliers by visual inspection. The polynomial speed function was then fitted to each trajectory (separately for inflow and outflow).

They found that their coefficients showed an influence of entering of speed of vehicle, approach angle etc. Their speed profiles were typically applicable to free-flowing vehicles, i.e. when not influenced by external disturbances like other vehicles.

**Verizon Telematics Technical Information Bulletin**

In [11], Verizon has mentioned about how they have set thresholds for hard acceleration and braking events for different class of vehicles for their Verizon Telematics 5500 device.

**For Hard braking:**

Light and Medium Duty vehicles: 8.77 MPH/s | 14.11 KPH/s (0.40g)  
Heavy vehicles: 4.82 MPH/s | 7.76 KPH/s (0.22g)  
The maximum threshold for hard braking detection is 21.93 MPH/s | 35.29 KPH/s.

**For Hard acceleration:**

The minimum thresholds for hard acceleration are:

Light and Medium vehicles: 7.90 MPH/s | 12.71 KPH/s (0.36g)  
Heavy vehicles: 4.82 MPH/s | 7.76 KPH/s (0.22g)  
The maximum threshold for hard acceleration detection is 17.55 MPH/s | 28.24 KPH/s.

Verizon’s telematics device follows a single threshold at a time for such events and their severity of detecting these events can be adjusted between this minimum and maximum threshold limits.
Implementation

Objective
Our objective is to create a software which will be able to analyze driving behavior of a person in a fixed route and provide them a driving score for each segment of the road as well as a holistic report which helps to highlight the bad and good driving aspects.

This software is designed by keeping in mind to be used in places where fixed routes are assigned to drivers and multiple trips are made in the same route.

By getting the context-information of the route beforehand, we believe that we gain an advantage over other similar implementations which try to generalize a driver’s environment and provide the same analysis.

Hardware Setup
The Hardware setup consists of the following devices, viz. Raspberry Pi 3+, PiCAN 2 module and a GPS module. Raspberry Pi is the mini-computer responsible for the processing needs.

The PiCAN 2 shield attached to the GPIO pins of the Pi serves as an interface for the Pi to connect to the OBD II port of the car to log CAN data. GPS module is attached to the Pi to obtain the location of the car.

Figure 9: Hardware setup
The vehicle used for testing was a 2011 model Scion xB which supported CAN via its OBD 2 port. The raspberry pi was powered using the Car’s 12 V socket which in turn powered the GPS.

Software Setup
Python version 3.x was used for both data collection using OBD port as well as for the grading engine.

For the raspberry-pi based data logger, following python libraries were used:

- Python-can – for CAN library
- Gps3 – to interface with GPSD services

Further the log files are planned to be stored, graded and retrieved using a cloud service and a web-based GUI is intended to display various results.
Test Route
The test route was chosen by keeping in mind to contain different type of road elements such as stop signs, stop lights, left and right turns, roundabouts and straight segments of road.

The test route is located at Ames, Iowa with a lap of about 8 miles long.

Figure 12: Route tested for driving behavior analysis

Figure 13: Left and right turns from stop sign and stop lights
There are total 36 segments in the test route and they are as described below:

<table>
<thead>
<tr>
<th>Segment number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Depot to road</td>
</tr>
<tr>
<td>1</td>
<td>straight to airport rd. stop</td>
</tr>
<tr>
<td>2</td>
<td>airport rd stop sign</td>
</tr>
<tr>
<td>3</td>
<td>airport rd</td>
</tr>
<tr>
<td>4</td>
<td>airport rd. signal</td>
</tr>
<tr>
<td>5</td>
<td>highway</td>
</tr>
<tr>
<td>6</td>
<td>highway end stop-sign left</td>
</tr>
<tr>
<td>7</td>
<td>straight on bridge</td>
</tr>
<tr>
<td>8</td>
<td>signal on bridge</td>
</tr>
<tr>
<td>9</td>
<td>straight to isaac newton drive</td>
</tr>
<tr>
<td>10</td>
<td>isaac newton drive free right</td>
</tr>
<tr>
<td>11</td>
<td>straight</td>
</tr>
<tr>
<td>12</td>
<td>2nd free right on isaac newton drive</td>
</tr>
<tr>
<td>13</td>
<td>straight</td>
</tr>
<tr>
<td>14</td>
<td>stop sign right to S 16th</td>
</tr>
<tr>
<td>15</td>
<td>straight</td>
</tr>
<tr>
<td>16</td>
<td>Signal</td>
</tr>
<tr>
<td>17</td>
<td>Straight</td>
</tr>
<tr>
<td>18</td>
<td>signal # near old chicago</td>
</tr>
<tr>
<td>19</td>
<td>Straight</td>
</tr>
<tr>
<td>20</td>
<td>signal left</td>
</tr>
<tr>
<td>21</td>
<td>Straight</td>
</tr>
<tr>
<td>22</td>
<td>Signal</td>
</tr>
<tr>
<td>23</td>
<td>Straight</td>
</tr>
<tr>
<td>24</td>
<td>signal</td>
</tr>
<tr>
<td>25</td>
<td>Signal</td>
</tr>
<tr>
<td>26</td>
<td>Straight</td>
</tr>
<tr>
<td>27</td>
<td>roundabout</td>
</tr>
<tr>
<td>28</td>
<td>Straight</td>
</tr>
<tr>
<td>29</td>
<td>roundabout</td>
</tr>
<tr>
<td>30</td>
<td>Straight</td>
</tr>
<tr>
<td>31</td>
<td>roundabout</td>
</tr>
<tr>
<td>32</td>
<td>Straight</td>
</tr>
<tr>
<td>33</td>
<td>right turn to S loop drive</td>
</tr>
<tr>
<td>34</td>
<td>Straight</td>
</tr>
<tr>
<td>35</td>
<td>Stop sign at bus depot</td>
</tr>
</tbody>
</table>

*Table 1: List of segments with their description*
Data Logging

CAN data collection
The first step in order to grade is to obtain CAN data from the OBD2 port in the car. There are various ECUs in the car from which information can be obtained from their corresponding PIDs.

Since the goal was to obtain data suitable for grading and general vehicle telematics, we had to obtain the collection of following CAN data:
1. Throttle position (PID: 0x11)
2. Engine RPM (PID: 0x0C)
3. Vehicle Speed (PID: 0x0D)

Data obtained was timestamped to help with further post processing calculations. Such calculations were important for obtaining parameters like acceleration and distance travelled.

The approach used for obtaining CAN data was Stop and Go approach. We ping the CAN Bus for the required PID and wait for it to reply with the message or till timeout condition is met and only then ping it for the next PID. Failure to obtain information on all three PIDs mentioned above results in discarding the obtained information for that collection cycle.

CAN data post processing
The speed and time information obtained was then immediately used to compute acceleration of the car and distance travelled by the car.

GPS data collection
The GPS Latitude and Longitude coordinates were obtained from the GPS module at the rate of 1Hz to yield information on the physical location of the car. This was crucial for geofencing.

Geofencing
The GPS information was used for geofencing, i.e. determining the location of interest points and whether we’ve arrived at them. The idea was to simulate the journey of the bus throughout the day and upload logged data to the database when the bus returns back to the depot.

The trip starts from the depot to the initial bus stop and the data is logged. From then on, we keep doing laps on the test route while logging data for each lap in a separate log file. We stop all logging when we come back to the depot.

Automation
The CAN and GPS data logging and geofencing is completely automated. CAN data and GPS data logging are done in different threads.

The flowcharts in the following page will illustrate the data logging and automation process.
**Figure 14: CAN and GPS data logging**

**Figure 15: Data logging Automation**
Pre-Grading data processing
The data obtained is useful for grading, but we need an extra parameter for grading, viz. jerk, which is the second order rate of change of speed. So, we calculate jerk values for the logged data before we actually send it to the grading engine.

It was also observed that CAN data from the car had a constant delay of 2 seconds. We then removed the lag so that the CAN data matched up with the GPS data. This is crucial to know the exact vehicle parameters at any given location.
Grading

Thresholding
The first step in order to grade is to set the thresholds in which acceleration or deceleration is considered bad to take off penalty.

A bad acceleration or deceleration level can be depended on many criteria such as type of vehicle, human perception, levels which may seem dangerous on roads and can vary according to the environment of the road on which the driver is driving.

For e.g. Verizon in [11] states different thresholds for hard accelerations and hard braking also differentiated for different class of vehicles based on their size as light/medium and heavy for their telematics device.

Adaptive acceleration threshold based on context like vehicle traffic, signal lights are difficult to predict just based on driving parameters unless external sensors are used.

Hence, we decided to follow a regression-based thresholding approach, where we obtain acceleration and deceleration thresholds which we consider bad at a given speed based on the driving data we collected.

The acceleration/decelerations were plotted for different speeds and a closely fitting trendline was obtained with help of Microsoft Excel’s graphing capabilities.

![Acceleration regression for turn/special segments](image1)

![Acceleration regression for regular segments](image2)

<table>
<thead>
<tr>
<th>Speed based acceleration thresholds - applicable to all categories</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Here adaptive speed - based thresholding is derived exponential regression equations</td>
</tr>
<tr>
<td>• Thresholds for turn / straight segments are different. Stricter for turns</td>
</tr>
</tbody>
</table>

Similarly, for deceleration,
Based on this, acceleration or deceleration thresholds are determined dynamically based on the vehicle speed as harsh accelerations are bad at higher speeds than in lower speeds.

For e.g. for heavy vehicles, jackknifing is a common problem at higher speeds which tends to make the vehicle unstable [12]

For turns, it is recommended to keep the turning speed around 15-25 mph [13]

The threshold for speeds is nothing but the speed limit of the road and the threshold for jerks is a constant 2.5 Km/hs\(^2\).

**Scoring**

Across a segment of road, points of raw data are obtained at fixed interval of time of 0.5 seconds. Each point is saved as a row in a CSV file which gives information about the vehicle’s raw data.

**Point Score:**

Each parameter at a point is evaluated using a simple mechanism:

\[
\text{Point Score} = 1 - \frac{\text{excess parameter value at that point}}{\text{parameter threshold for that point}}
\]

The parameter threshold is the speed limit, acceleration limit or jerk limit.

While jerk limit is fixed based on experimental data at 2.5 Km/hs\(^2\), speed limit is determined by the speed limit assigned to the road segment, acceleration threshold is determined as discussed above and varies based on the segment being special or a straight one, and whether the vehicle is accelerating or decelerating.

If the parameter value doesn’t exceed the threshold value, then the parameter score is 100%.

The total score for that point is equally shared by the three grading parameters, viz. speed, acceleration and jerk.
**Segment and Lap scores:**

A segment contains multiple points as described above. A lap comprises multiple segments which are special or straight as discussed before. We use weighted sum based on distance for calculating the scores. The reason distance is chosen as a deciding factor is because a person is a threat on the road for only as long as the distance driven poorly by him.

A segment score is calculated as:

\[
\text{segment score} = \frac{\sum (\text{PointScore} \times \text{distance travelled in that point})}{\text{total distance travelled in the segment}}
\]

Finally, a lap score is calculated as:

\[
\text{lap score} = \frac{\sum (\text{segmentScore} \times \text{segmentDistance} \times \text{importanceWeight})}{\sum (\text{segmentDistance} \times \text{importanceWeight})}
\]

For lap score, the importance weight of straight segments is 1 whereas that of special segment are twice of it i.e. 2.

This method helps us provide more weightage to the score of special segments which includes important road segments such as turns, stop signs, etc. These segments are more prone to accidents.

**The Grading Engine**

The grading engine comprises of two main components, viz. trip_grader and events_grader.

The trip_grader is responsible for loading the csv log file, parsing it, populating parameter lists for each segment, passing it to events_grader and calculating final score for the lap.

The events_grader takes all the necessary parameters such as speed, distance, accelerations, passed by value from trip_grader and uses those points to calculate point-based score and finally returns a score percentage for the segment.

When calculating scores for stop sign zones, we consider two additional factors, viz. whether it was a rolling or a complete stop and how many times the driver hesitated at the stop sign.

The complete stop score is calculated as:

\[
\text{complete stop score} = 100 - (\text{minimum speed at the stop sign})^3
\]

The hesitation score is calculated as:

\[
\text{hesitation score} = 100 - (\text{number of hesitations at the stop sign})^3
\]

These scores are combined with stop sign segment score with equal weightage.

**Point Score calculation example**

Let’s take Lap 1’s segment 0 out of 36 segments in total. Segment 0 is the part from the bus depot’s Stop sign to joining the South Loop Drive road.
The following parameter plots were generated for this segment for Lap 1:

Figure 21: Segment 0 - Jerk Plot

Figure 22: Segment 0 - Speed Plot
If seen closely, the acceleration plot shows two points above acceleration threshold. Also, the jerk plot shows some jerky behavior in which some of them cross the set threshold of 3.5 Km/hs\(^2\). But the speeds have been within the limit throughout the segment. So, with the above formula for point score, we get for the point 2 in acceleration plot as below:

- Speed limit score = 100
- Acceleration score:
  \[ 1 - \left( \frac{7.397515 - 6.350087030672567}{6.350087030672567} \right) = 1 - 0.164947025807 = 0.835052974193 \]
- Jerk Score:
  \[ 1 - \left( \frac{7.397515 - 3.5}{3.5} \right) = 1.1135 = -0.11357 \]

So, combining acceleration, jerk and speed limit scores (in %) with 33% each, we get:

\[ 0.33 \times 100 + 0.34 \times 83.5052974193 + 0.33 \times (-11.357) \approx 57.6438 \%

The below figure shows console output from the python program reflecting the score above:
Similarly, we get a score for the point 14 in acceleration plot:

**Segment score calculation example:**

All the points in this segment are combined using their distance weights using the formula mentioned before. The table below shows all the point scores for Segment 0 for this lap.

<table>
<thead>
<tr>
<th>Point#</th>
<th>Speed score</th>
<th>Accn score</th>
<th>Jerk score</th>
<th>Point score</th>
<th>Distance weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0.16376</td>
</tr>
<tr>
<td>13</td>
<td>100</td>
<td>83.5053</td>
<td>-11.3576</td>
<td>57.6438</td>
<td>0.354314</td>
</tr>
<tr>
<td>14</td>
<td>100</td>
<td>100</td>
<td>95.13106</td>
<td>98.39325</td>
<td>0.875897</td>
</tr>
<tr>
<td>15</td>
<td>100</td>
<td>100</td>
<td>95.1132</td>
<td>98.38736</td>
<td>1.165162</td>
</tr>
<tr>
<td>16</td>
<td>100</td>
<td>100</td>
<td>86.58777</td>
<td>95.57396</td>
<td>1.242332</td>
</tr>
</tbody>
</table>

*Figure 24: Grading Software console output*
### Table 2: Segment score calculation

**Lap 1 Score Calculation**

With the scores of 36 segments, the scores are combined using respective segment distances and their importance weight. Straight segments have importance weight of 1 while special segments have weight of 2. The segment scores for 36 segments in lap 1 are as shown below:

<table>
<thead>
<tr>
<th>segment#</th>
<th>segment score</th>
<th>importance weight</th>
<th>distance weight</th>
<th>segment weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>95.86899634</td>
<td>2</td>
<td>18.569215</td>
<td>37.13843</td>
</tr>
<tr>
<td>1</td>
<td>95.99294125</td>
<td>1</td>
<td>50.267164</td>
<td>50.26716</td>
</tr>
<tr>
<td>2</td>
<td>80.19201824</td>
<td>2</td>
<td>78.511683</td>
<td>157.0234</td>
</tr>
<tr>
<td>3</td>
<td>92.81578278</td>
<td>1</td>
<td>1651.564997</td>
<td>1651.565</td>
</tr>
<tr>
<td>4</td>
<td>88.865298</td>
<td>2</td>
<td>397.941844</td>
<td>795.8837</td>
</tr>
<tr>
<td>5</td>
<td>91.86340516</td>
<td>1</td>
<td>1996.798726</td>
<td>1996.799</td>
</tr>
<tr>
<td>6</td>
<td>94.6431517</td>
<td>2</td>
<td>179.109355</td>
<td>358.2187</td>
</tr>
<tr>
<td>7</td>
<td>95.73650256</td>
<td>1</td>
<td>121.568581</td>
<td>121.5686</td>
</tr>
<tr>
<td>8</td>
<td>98.3646994</td>
<td>2</td>
<td>171.293566</td>
<td>342.5871</td>
</tr>
<tr>
<td>9</td>
<td>99.87183987</td>
<td>1</td>
<td>29.03914</td>
<td>29.03914</td>
</tr>
<tr>
<td>10</td>
<td>88.00776585</td>
<td>2</td>
<td>123.354061</td>
<td>246.7081</td>
</tr>
<tr>
<td>11</td>
<td>100</td>
<td>1</td>
<td>60.968979</td>
<td>60.96898</td>
</tr>
<tr>
<td>12</td>
<td>97.92623617</td>
<td>2</td>
<td>69.15631</td>
<td>138.3126</td>
</tr>
<tr>
<td>13</td>
<td>99.0862523</td>
<td>1</td>
<td>28.680298</td>
<td>28.6803</td>
</tr>
<tr>
<td>14</td>
<td>95.95853701</td>
<td>2</td>
<td>120.025341</td>
<td>240.0507</td>
</tr>
<tr>
<td>15</td>
<td>98.38307091</td>
<td>1</td>
<td>52.847854</td>
<td>52.84785</td>
</tr>
</tbody>
</table>
Table 3: Segment scores for Lap 1

<table>
<thead>
<tr>
<th>Segment</th>
<th>Time (s)</th>
<th>Lap</th>
<th>Distance (m)</th>
<th>Total Distance (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>85.1848029</td>
<td>2</td>
<td>247.514617</td>
<td>495.0292</td>
</tr>
<tr>
<td>17</td>
<td>90.61772262</td>
<td>1</td>
<td>1600.250704</td>
<td>1600.251</td>
</tr>
<tr>
<td>18</td>
<td>95.13503923</td>
<td>2</td>
<td>245.931994</td>
<td>491.864</td>
</tr>
<tr>
<td>19</td>
<td>92.39971041</td>
<td>1</td>
<td>1850.918215</td>
<td>1850.918</td>
</tr>
<tr>
<td>20</td>
<td>95.94085331</td>
<td>2</td>
<td>223.869136</td>
<td>447.7383</td>
</tr>
<tr>
<td>21</td>
<td>94.32372641</td>
<td>1</td>
<td>376.356358</td>
<td>376.3564</td>
</tr>
<tr>
<td>22</td>
<td>88.22481761</td>
<td>2</td>
<td>235.670757</td>
<td>471.3415</td>
</tr>
<tr>
<td>23</td>
<td>93.49905298</td>
<td>1</td>
<td>260.86261</td>
<td>260.8626</td>
</tr>
<tr>
<td>24</td>
<td>93.72311335</td>
<td>2</td>
<td>234.08957</td>
<td>468.1791</td>
</tr>
<tr>
<td>25</td>
<td>92.51846522</td>
<td>2</td>
<td>174.410318</td>
<td>348.8206</td>
</tr>
<tr>
<td>26</td>
<td>94.3438066</td>
<td>1</td>
<td>346.392768</td>
<td>346.3928</td>
</tr>
<tr>
<td>27</td>
<td>82.75899711</td>
<td>2</td>
<td>122.672478</td>
<td>245.345</td>
</tr>
<tr>
<td>28</td>
<td>99.40442204</td>
<td>1</td>
<td>303.614267</td>
<td>303.6143</td>
</tr>
<tr>
<td>29</td>
<td>98.34386837</td>
<td>2</td>
<td>135.561223</td>
<td>271.1224</td>
</tr>
<tr>
<td>30</td>
<td>97.22435297</td>
<td>1</td>
<td>316.565371</td>
<td>316.5654</td>
</tr>
<tr>
<td>31</td>
<td>85.39592184</td>
<td>2</td>
<td>133.8084</td>
<td>267.6168</td>
</tr>
<tr>
<td>32</td>
<td>94.04027603</td>
<td>1</td>
<td>57.487399</td>
<td>57.4874</td>
</tr>
<tr>
<td>33</td>
<td>89.22279845</td>
<td>2</td>
<td>81.900026</td>
<td>163.8001</td>
</tr>
<tr>
<td>34</td>
<td>96.23759836</td>
<td>1</td>
<td>550.484203</td>
<td>550.4842</td>
</tr>
<tr>
<td>35</td>
<td>99.96234519</td>
<td>2</td>
<td>54.737831</td>
<td>109.4757</td>
</tr>
</tbody>
</table>

The lap score is calculated as:

\[
\text{Lap score} = \frac{1455979}{15750.92} = 92.43 \%
\]

Lap 1 Report:
regular distance = 9654.667633999999, total distance = 12702.22500000, total weight = 15750.923084, segment_scores = 1455978.7120828105
best segments = [11], worst segments = [2], worst segment score = 80.19201824372307
final trip score is 92.43767519643426 out of 100

Figure 25: Grading Software console output showing Lap 1 report
Segment score comparison

To give an idea about how driving is graded for the same segment, let us focus on segment #20 from lap 2 and lap 4 in our test route, which is a special segment with a stop light followed by a left turn.

Figure 26: Segment #20 acceleration plots for lap 2 and 4

Figure 27: Segment #20 Jerk plots between laps 2 and 4
As we can see from the above plots, especially that of acceleration and jerk for segment #20 for laps 2 and 4.

Both seem to have stopped at the signal, but a hard braking was done in lap 2 near the stop line than lap 4.

Also, accelerations during the turn were above the threshold a lot of times as well as jerky in nature in lap 2 compared to lap 4.

The lap 4 had good deceleration near the stop line and overshot acceleration threshold couple of times during the turn and was slightly jerky.

Hence based on the above performance, the grading software provided the following scores:

<table>
<thead>
<tr>
<th>Lap</th>
<th>Distance (m)</th>
<th>Score (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>247.78</td>
<td>61.16215</td>
</tr>
<tr>
<td>4</td>
<td>245.22</td>
<td>89.77</td>
</tr>
</tbody>
</table>

*Table 4: Lap2 vs Lap 4 Score comparison*

The lap 2 showed relatively poor performance than lap 4 and hence got a score of 61.16% whereas lap 4 with its flaws was awarded 89% for this segment.
Lap scores comparison

<table>
<thead>
<tr>
<th>Lap</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>92.43</td>
</tr>
<tr>
<td>2</td>
<td>91.81</td>
</tr>
<tr>
<td>3</td>
<td>91.37</td>
</tr>
<tr>
<td>4</td>
<td>90.60</td>
</tr>
<tr>
<td>5</td>
<td>93.15</td>
</tr>
</tbody>
</table>

*Table 5: Laps 1-5 Score comparison*

- The lap 5 was targeted to be driven in the best behavior as possible and its score being the highest among these laps, reflects that. The lap 4 had lot of bad segments compared to other laps and thus it got the least score.
- In other laps, certain segments were observed bad behaviors and their segment scores clearly reflected it, like the example above.
- The overall score is weighed by their score, their distances and their importance. Hence the scores are relatively close.

*Figure 29: Bar graph showing scores from laps 1 to 5*
Conclusion

From the above setup, we were successfully able to analyze driving behavior in a known environment which helped us to grade a driver better.

A robust grading algorithm was devised to consider the distance travelled and importance of that segment along with various driving parameters like speed, acceleration.

This detailed segment level analysis threw some light on understanding segments of different importance and help analyzing the driving score and report better to improve a driver’s performance.

Future work

- Combining driving parameters for a point score can be improved based on data available.
- Additional sensors such as Accelerometers can be used to improve grading of turns while considering more context information such as turn radius, turn length, turn angle etc.
- A higher frequency GPS will help geo-location based bounding much better.
- The grading engine can be improved to detect catching of traffic lights
- If a vehicle can be modified, accessing internal CAN bus for steering wheel data, braking data can provide more valuable information.
References


