CSCI 720: Big Data Analytics
GPS Project
Group 3

Team Members

- Swapnil Shah : sss4174
- Srujan Shetty : sgs2892

1. **Abstract**

The objective of this project is to read the gps data recorded by the tracker, then process the data and return the fastest path from Professor Kinsman's house to RIT. In this project, we are trying to reduce the cost functions by minimizing three attributes:

1→ Total time spent on stop signs, traffic lite and car parked other than at destination.
2→ Total time spent when speed is over 60 mph.
3→ Total number of turns i.e. left or right turns on the overall route.

Below are brief idea on our approach to achieve above mentioned goals.

1→ We are monitoring the speed recorded by GPS tracker and a flag is raise whenever the speed falls below a mentioned threshold. Then the time is recorded until the speed rises over a pickup threshold. Depending on the time spent at a spot, we determine if the car is at a stop sign, traffic signal, or is parked.
2→In this case, instead of performing the laborious work of looking after every speed and keeping track of every time speed went over 60 mph, we approximate all this by only recording the highest spped achieved in a trip and dividing the value by 60.
3→ For the sole purpose of execution of this project, here we assume that before every left or right turn, the car will be slowed down to a certain speed. Here we are keeping the threshold similar to the one kept for stop signs above. Then at these points we find the bearing angles between two points and determine if the car has taken a turn or not.

2. **Overview**

In this section we are describing what are the tasks of the methods that we have included in the project

$get\_coordinates()$

This method converts the latitude and longitude from $GPRMC$ line of gps data and converts it into a format that can be stored in kml format file and read by Google Earth. Direction south and west can be differentiated by the negative sign in front of the co-ordinate value.

make_Modifications()

The method helps us to figure out whether the car took a turn or not between the start position and end position. We find the bearing angle between the two co-ordinates. We are making use of the standard formula that we have read for calculating the bearing angle. The concept for figuring out the direction where the car is exactly moving towards is used from the code we found as a reference. Reference is mentioned near the part which is been used.

calculateAngle()

The method helps us to figure out whether the car took a turn or not between the start position and end position. We find the bearing angle between the two co-ordinates. We are making use of the standard formula that we have read for calculating the bearing angle. The concept for figuring out the direction where the car is exactly moving towards is used from the code we found as a reference. Reference is "https://stackoverflow.com/questions/3209899/determine-compass-direction-from-one-lat-lon-to-the-other".

get_StopSigns()

This is considered as the most important method of the entire project. Here we read the spped from gps file and then perform the following operation based on their values. Initially we have mentioned a initialPickUpSpeed which is to iterate through all the co-ordinates at start of trip There is a upperBound mentioned which raises a flag if the speed falls below this speed. Until the time, the car is in the range of upperBound and lowerBound, time is recorded. Noe once the car goes over the pickUp speed, we know that car is no longer in stop state. Based on the time calculated above, we predict where the car was at a stop sign or traffic light or was parked. For every stop sign and traffic signal, the condition is checked if the car took a turn from this point or not.

calculateCost()

This is the method where we calculate the cost. The overall travel time is divided by 30 minutes i.e. average time of a journey Maximum speed in the journey is divided by 52.13 knot i.e. average spped in a journey Total number of times a car is stopped or took a turn is divided by 5 i.e. average time the situation arises in a

3. **Team Composition**

   Figuring stop signs, traffic lights

   As a team, Swapnil and I decided to each do one important task required for the project. We divided the task to figure out the stop signs, traffic lights as one task and the second task was to find the turns taken by the car. I took up the work for stop signs as I had read through the gps text file and was having a idea of how the speed was varying over the course of the journey. This analysis helped to figure out the conditions that would be satisfied when a car reaches a stop sign or when it is parked.

   Calculating car moving direction

   On the other hand, Swapnil worked through the approach to find how can be identify that whether a car turned or not. Then we figured out about the calculation of bearing angle and an approach to make it work. Bearing angle was giving us the direction in degrees, but we were not sure how to make use of it to check the condition for change of direction. Then we came across a page at StackOverFlow which solved our problem and it states which direction the car is moving towards at a given degree.

4. **GPS to KML**

   Issues while converting GPS to KML

   For the checkpoint 1 we made use of simplekml package which it worked well to save all the co-ordinates. But now in this final submission we had to add the pin points to specify the stop signs. So then we decided to create out own method $outputKML()$ which writes the coordinates and pin points into the kml file.

   Data cleaning and anomaly detection

   While going through the entire gps text file, it was noted that there were multiple lines which had missing data and few lines where multiple coordinates were recorded. So in order to handle this situation conditions are checked before each GPRMC line is processed.

   Issues in the file

   There were multiple lines which had missing data and few lines where multiple coordinates were recorded

5. **Cost Function**

   For the cost function we made use of the function provided by professor i.e. making use of total travel time and maximum speed of the journey. To this function we added the parameter of total number of times car was required to stop and take a turn. This number is then divided by 5 which we consider as the average number of turns that are required in every journey.

6. **Stop Detection**

   As it is mentioned in the overview section above, we made use of the speed of car to figure out the probability of car coming to a stop. It can stop either at a stop sign, or traffic light or when the car is parked.

   Whenever a car approaches to a halt, it has to come below an upper bound speed so that it work as a flag to check the possibility of car coming to stop after covering some distance. This is the logic that we are making use of. Whenever the car goes below our upper bound we keep a track of the speed there after. From here on if the speed reaches almost zero, we know that the car has come to a halt.

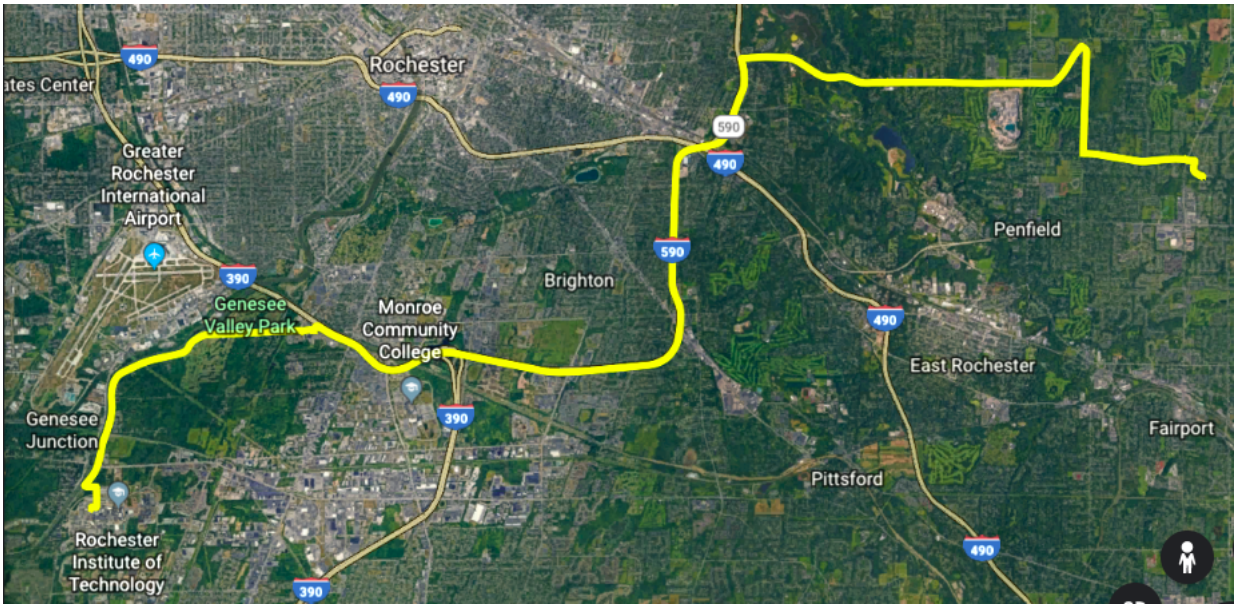
   Later when the car speed goes over the upper bound speed, we know the car began to move from the spot where it came to halt. We check the total time in seconds for which the car was stopped at a particular spot. Depending on the time spent at a spot, we decide whether the car was at a stop sign or traffic light or was it parked.

7. **Detecting Turns**

   We are assuming that throughout the entire course of travel, whenever the car needs to take a turn, it needs to bring the car to a halt. Using this assumption, we are checking turns from the points that we obtained using the Stop Detection approach. From each of these positions, we move 10 points ahead and check the direction of where the car is heading. If the car direction varies for initial and the final positions, then we consider that the car takes a turn from this point.

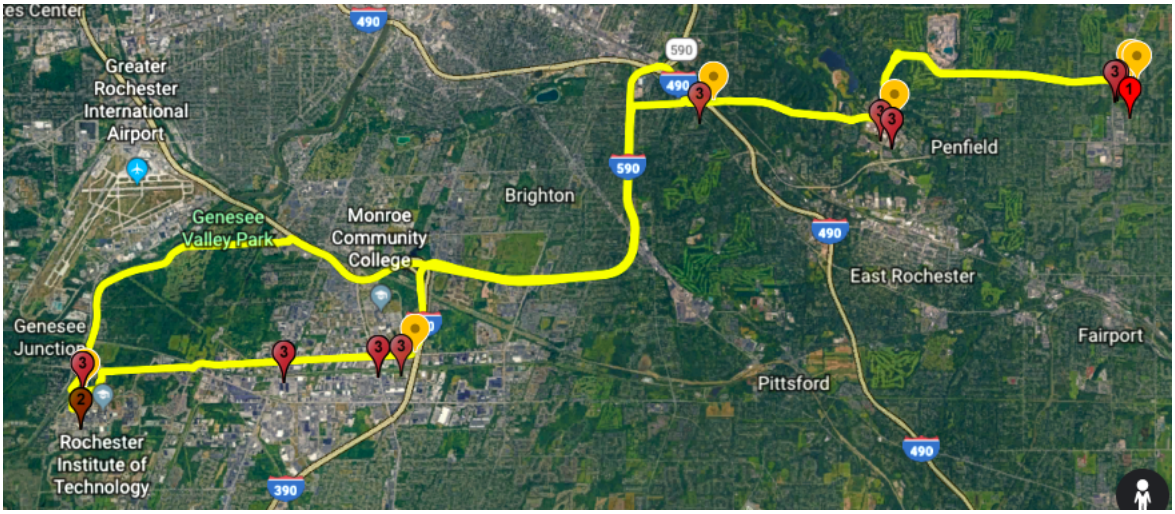
8. **Results of path file**

```
<?xml version="1.0" encoding="UTF-8"?>
 <kml xmlns="http://www.opengis.net/kml/2.2" xmlns:gx="http://www.google.com/kml/ext/2.2">
 <Document id="1">
 <Style id="4">
 <LineStyle id="5">
 <color>ff00ffff</color>
 <colorMode>normal</colorMode>
 <width>6</width>
</LineStyle>
</Style>
<Placemark id="3">
<name>GPS_Project_Group3</name>
<description>Speed in Knots, instead of altitude.</description>
<styleUrl>#4</styleUrl>
<LineString id="2">
<coordinates>
-77.680163,43.085848,0.01
-77.680163,43.085848,0.01
-77.680163,43.085848,0.01
-77.680163,43.085848,0.01
-77.680163,43.085848,0.0
-77.680163,43.085848,0.0
-77.680163,43.085848,0.0
-77.680163,43.085848,0.0
-77.680163,43.085848,0.0
-77.680163,43.085848,0.0
-77.680163,43.085848,0.0
-77.680163,43.085848,0.0
-77.680163,43.085848,0.01
-77.680163,43.085848,0.0
-77.680163,43.085848,0.01
-77.680163,43.085848,0.01
-77.680163,43.085848,0.01
```



9. **Results of hazard file**

```
<Placemark>
<description>Stop Light</description>
<Point>
<coordinates>
-77.441143,43.141115,0.0
</coordinates>
</Point>
</Placemark>
<Placemark>
<description>Red PIN for A Stop</description>
<Style id="normalPlacemark">
<IconStyle>
<color>ff0000ff</color>
<Icon>
<href>http://maps.google.com/mapfiles/kml/paddle/1.png</href>
</Icon>
</IconStyle>
</Style>
<Point>
<coordinates>-77.631858,43.114145,0.0
</coordinates>
</Point>
</Placemark>
<Placemark>
<description>Red PIN for A Stop</description>
<Style id="normalPlacemark">
<IconStyle>
<color>ff0000ff</color>
<Icon>
<href>http://maps.google.com/mapfiles/kml/paddle/1.png</href>
</Icon>
</IconStyle>
```



10. **Discussion Section**

Issues with the approach

The entire GPS route processing depends on the data gathered from the GPS device, but the data recorded is not clean. There are multiple lines with duplicate data or missing data. Also there are many parameters that we are not even making use of.

Noise Removal and Signal Processing

As mentioned above there were multiple records with either duplicate or missing data, so before performing any further operations we made sure that the data are in proper format and the total length of GPRMC data is correct.

11. **Summary**

We found this entire project to be one of the unique ones that we have worked on till this date. To start with, we got to learn how exactly a kml file looks like. It was a great experience to work with the kml file and exciting to see how the Google Earth processes the file to display the entire route.

Since the GPS data consisted of some noise data, we did the pre processing step which we feel will be of great assistance to us in any future projects in which we will be working with data.

Trying to determine the cost function has improved our understanding of the regularization for a cost function. We have gained knowledge and experience to work with GPS data that we can now perform many operations over it.

Usefulness

As UPS makes great use of the GPS to avoid taking any left turns, just like this application when we figure out whether a car is taking a left or right turn, we can find a route with close to no left turns.

This can any day serve as the approach to find the fastest or the shortest route to reach from a source to a destination.