Team Control Number

22091106

Problem Chosen

A

2022

IMMC

Summary Sheet

Nowadays, with the rapid development of autonomous driving technology, building a automatic driving system will no longer be a rediculous thought. In this system, connections can be built between the autonomous driving cars, the smart lampposts installed with sensors and communication units and the cloud server. So we need to carry out a plan which will make the transportation system easier and create an indicator system to evaluate the smart lamppost modification plans.

In the first model, we find out the factors that are related to evaluating a lamppost modification plan. We mainly discuss how to calculate the index which is affected by cost, coverage rate of LiDAR and the importance of each road. Afterwards, we make some plans about the measure of coverage rate in different situations.

In the second model, our goal is to establish an overall optimal configuration of all lampposts within a certain area. Instead of simply simulating all measures, we took a more natural q-Learning approach under the thought of greedy algorithms. The machine-learning process, which relied merely on the mathematical index of a given distribution as shown in the first model, gradually narrowed the result till a fixed distribution. In order to confirm the accuracy of the algorithm, we randomly altered certain sets of lampposts and withdrew the consequent changes in variables. After that, we chose the uniformity of lamppost points as the independent variable and conducted a sensitivity analysis, which, after eliminating certain anomalous data, proved that the results were accurate enough. Out of considerations of dual confirmation, we also checked that the results turned out to be optimal.

Last, we evaluated the plan using the model built in the first part. With the help of Python packages *Shapely* and *Matplotlib*, we managed to calculate the coverage area and coverage rate of our plan in the second part. We did a sensitivity analysis on the model, and found some issues where it cannot evaluate correctly. Probable explanations are raised, and we developed a dynamic model to solve the problem.

Smart Lamppost Deployment

Team#22091106

November 21, 2021

Contents

1 Introduction

1.1 Background

Nowadays, with the rapid development of autonomous driving technology, building a automatic driving system will no longer be a rediculous thought. Based on recent technology, autonomous driving technology has begun to move from partial drving at the L2 level to conditional driving automation at the L3 level.

According to the degree of automation of the vehicle[1], when the system cannot withstand the working conditions, the driver needs to take over the malfunctioning vehicle. After activating the automatic driving system, the vehicle itself can complete tasks such as steering, acceleration, deceleration, and reaction. The status detection and reaction are carried out under the operating conditions specified by the automatic driving system.

In order to facilitate vehicles which will better implement the L3 level autonomous driving tasks, using smart lampposts as the main representative of the smart roadside infrastructure. By refitting existing ordinary lampposts into smart lammposts equipped with sensors and communication units. They can collect road data through sensors and upload them onto the cloud server, and then download to the original lamppost or share it with other ones after the server completes the calculations. The autonomous driving cars can obtain the road data by communicating with neighboring smart lampposts.

1.2 Problem Restatement

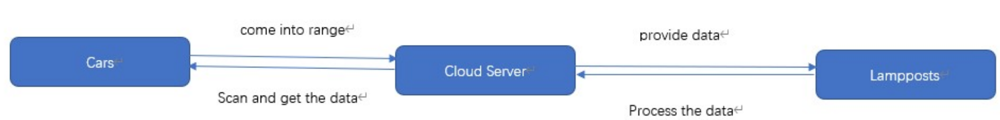

Fig. 1

In this model, we actually need to have some detailed information about the whole transportation system in the area. In addition, we need to know about the driving pattern of the cars to make further dicoveries about the autonomous driving system. In Fig. 1, we can clearly discover the relationship between the car, the lamppost and the cloud server.

So the problem is divided into 3 main parts:

- Build a framework for evaluating the smart lamppost modification plans based on cost and coverage of the road by the sensor and the WiFi communication.

- Give a plan for smart lamppost modification in an area of any city according to the data given.[2]
- Evaluate the lamppost modification plan based on our model and find the strengths and weaknesses of our model.

1.3 General Assumptions

- **All the cars have the same scale.**

  We assume that all the cars can be seen as a car which is 4.8m in length and 1.6m in width. This can flatten the neighborhood which we take into consideration.
- **The impact of buildings is not taken into consideration.**

  Since the distance between two roads is far from the detection zone, the buildings will not affect the detection of the road.
- **Vehicles drive on the left side according to the standard of HongKong.**
- **Neighborhood intersection is not taken into consideration.**
- **Most of the calculation is completed on the client side.**

  There are different types of car on the road. Since the cloud side can't afford such large amount of calculation, we assume that the most of it is completed on the client side.
- **The weather factors can be ignored.**

2 Evaluation Model

2.1 Problem Overview

In this model, we will establish an indicator framwork for the purpose of evaluating smart lamppost modification plans. By comparing the weight of different road types according to the cars' speed, acceleration, deceleration and reaction, we can clearly get the relationships of these areas. Basic road types include normal roads and crossroads. In this model, crossroads is defined as a circlw with a radius of 50m, centering at the middle of the crossing.

2.2 Assumption

- **All the destinations can be detected and can be abstracted as a dot.**
- **The car will update its newest driving data at any time oof the driving process.**

2.3 Variables and Constants

Symbol	Definition
S	Evaluating Index
P	Overall Cost
a_1	The Coverage Rate of Sensors on Crossroads
a_2	The Coverage Rate of Sensors on Normal Roads
b_1	The Coverage Rate of WiFi Communication on Crossroads
b_2	The Coverage Rate of WiFi Communication on Normal Roads
k_1	Weight of Crossroads
k_2	Weight of Normal Roads
b	The rate of the most cars that can be connected to WiFi
k_3	Weight of b

2.4 Determining the Factors

For we need to think more about the speed and accelerations when a car is passing crosswords, we consider the crossroads more important than normal roads. So we can infer that: $k_1 > k_2$

So by defining the weight of coverage rate in different areas, we find out a simple formula describing the effectiveness of the system:

$$S = \frac{k_1(a_1 + b_1)^3 + k_2(a_2 + b_2) + k_3 b}{P}$$

$$(k_1 > k_3 > k_2)$$

In this formula, $a_1$is identified as $\frac{S_{coverage}}{S_{all}}$, and in the same way we can easily get the formula describing a_2,b_1,b_2.

3 Modification Plan

3.1 Problem Overview

In this model, we will focus on the smart lamppost modification plan based on the transportation system in HongKong.[2](The map is attached at the end of this essay.) We can abstract this map into a rectangular coordinate system.

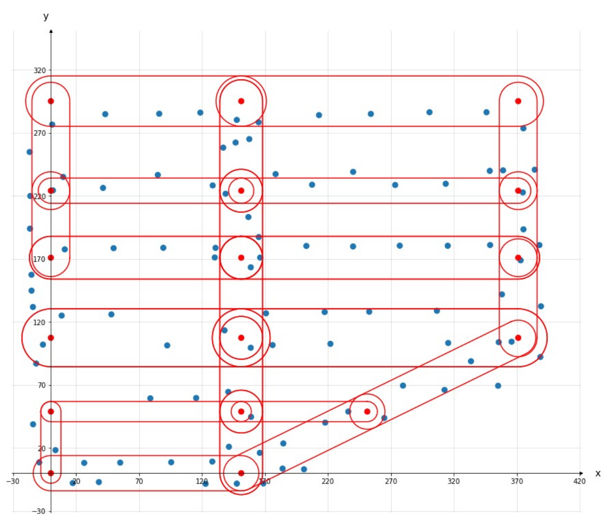

Fig. 2

After searching the traffic rules in this area, we find that Gaoshida Evenue has a spped limit of 70km/h(19.5m/s) and other roads have a speed limit of 50km/h(14.0m/s). So speed needs to be taken into consideration.

3.2 Definition

We assume that an autonomous driving need no time to turn left and 5 seconds to turn right. The number comes from the calculation according to the average width of roads and the highest speed to change direction safely in normal conditions. This means that when a car turn right, it will disappear immediately and appear at the end of the line in this road 5 seconds later. In this process, the car will exert no impact neither on other cars nor on the queuing line.

So now let we assume that we are in an autonomous driving car. If the car is running normally, it will drive at a constant speed when on normal roads, change there position when changing its direction and deceleraing when getting the information that it can't pass the crossroad.

By understanding these proecess, we can make us have a better understanding of the modification plan. Moreover, there are some rules that we need to obey in our plan.

- **Pedestrians are ignored in this model.**
- **All the crossroads should be covered with LiDAR.**

  Through our calculation, the stopping distance turns out to be at least 16m, so we need to make sure that LiDAR can cover all the crossroads.
- **The cars which are not connected to LiDAR drive at a constant speed.**

3.3 Model Calculation

To get further understanding of the transportation system in this area, we use the Q-learning Algorithm to calculate the best modification plan.

Q-learning[4] is an off policy reinforcement learning algorithm that seeks to find the best action to take given the current state. It doesn't need any policies because the q-learning function learns from actions that are outside the current policy by taking different actions randomly. More specifically, Q-learning seeks to learn a policy that maximizes the total reward. Therefore, we can use Q-learning to find the policy that best fits the condition in the selected area.

Algorithm 1 Find the best modification plan.

Cost = 5000 · (Lampposts with only LiDAR)
+3000 · (Lampposts with only WiFi)
+10000 · (Lampposts with both)
Formula of calculating coverage: $\frac{S_{\text{road\_covered}}}{S_{\text{roads}}}$

$$\sum_{\substack{\mathrm{roadtype} \overset{\mathrm{def}}{=\!=} \mathrm{r} \in \{\mathrm{Intersection}, \mathrm{Normal}\} \\ \mathrm{device} \overset{\mathrm{def}}{=\!=} \mathrm{d} \in \{\mathrm{LiDAR}, \mathrm{WiFi}\}}} \frac{\mathrm{r} \cdot \mathrm{d}}{\mathrm{Cost}} = \frac{k_1 (a_1 + b_1) + k_2 (a_2 + b_2)}{P}$$

return State
L = Set of all SmartLamps
done = **False**
initialize q_table, LEARNING_RATE and DISCOUN
while not done: **do**
  update q_table
end while
for lamp in L: **do**
  while in attempt to change the configuration of lamp: **do**
    if Get_State(new_distribution) > Get_State(current_distribution): **then**
      change distribution
    else
      remain
    end if
  end while
end for
previous loop → find q_table[max_distribution] and corresponding action
current_distribution = q_table[current_state + action]
new_distribution = (1 - LEARNING_RATE) * current_distribution +
LEARNING_RATE× (ΔGet_{State} + DISCOUNT * max_distribution)
q_table[current_state + action] = new_distribution
conduct the loop continuously until the previous loop doesn't alter anything:
done = **True**

Algorithm 2 Simulation of the driving condition.

```
for j in range(33): do
    if roads[i] crosses with roads[j]: then
        for k in range(1, num_of_cars + 1_): do
            if car[k] is on roads[i]: then
                if car[k] is accelerating: then
                    add the speed of car[k]
                end if
                if car[k] is decelerating: then
                    minus the speed of car[k]
                end if
                if car[k] is at the crossroads (and is about to turn): then
                    find the number of crossroad and the status of car[k]
                    choose a random direction to turn to
                end if
                set the turning time of car[k] according to the direction
            else {# on the road but not crossroad}
                judge whether this car can pass the crossroad (by calculating the least time it
                would take get to the crossroad)
                if can_pass crossroad: then
                    drive in max_speed
                else
                    if distance_to_crossroad <= min_decelerate_dis: then
                        set the deceleration rate as max_decelerate_rate (a constant)
                    end if
                end if
            end if
        end for
    end if
end for
```

Algorithm 3 Match lampposts with WiFi

```
vector<int> lines[1009] # lines[i] refers to the spots that can be linked to the car No. i
used = [] # judging whether a WiFi spot can be used
match = [] # record the matching car of each WiFi spot
for i in lines[car]: # all the spots that can connect with this car do
    if used[car][i] == 0: then
        used[car][i] = 1
        if match[line[car][i]] == 0 or find(match[line[car][i]]) == 1:# this spot is not connected or this position can be spared then
            match[line[car][i]] = car # record the pair
            return 1# this car can be connected
        end if
    end if
end for
return 0 # all the available spots cannot connect the car
while updating the information of the cars: do
    set 4 spots for each WiFi lamp (each spot can link at most one car)
    for k in cars: do
        for i in WiFi lamps: do
            if the distance between WiFi lamp i and car k is no more than 100m: then
                link an edge between car k and the 4 spots of WiFi lamp i
            end if
        end for
    end for
end while
```

3.4 Result

According to these rules and the calculation ideas we have put foward, we finally drew a complete figure of our plan, which is shown in the Fig. 3 below.

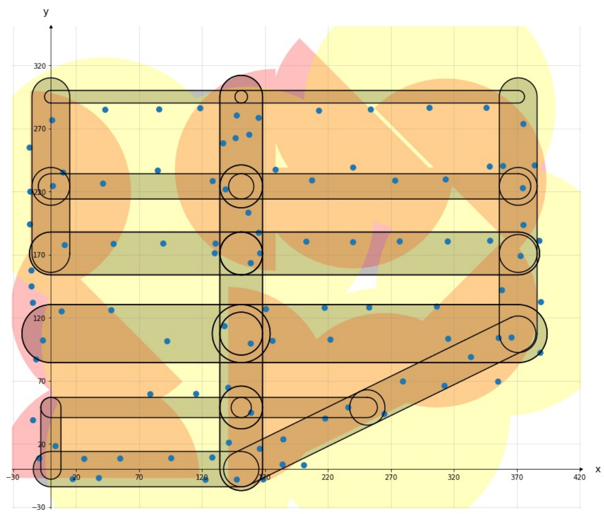

Fig. 3

In this picture, the yellow part refers to the WiFi-covered area while the red part refers to the LiDAR-covered part. We can see that most of the space in this area is now covered by Wifi, so cars driving in this area can get connected to the cloud server and get the data provided to make better driving plans. Because WiFi technology has a wider range of coverage than LiDAR technology, there are parts which are WiFi-covered but not LiDAR-covered. But since we have assumed that cars drive at a constant speed when they're not LiDAR-covered part, the impact can be ignored.

In all we need to disribute 16 smart lampposts. There are 9 lampposts which only need to be installed with LiDAR sensors, 6 lampposts which only need to be installed with WiFi access points and only 1 lamppost to be installed with both of them. There postition coordinates are shown in Fig. 4 and Fig. 5.

WiFi Lampposts

#	xWiFi	yWiFi
1	41.427	226.357
2	92.3732	101.392
3	95.542	8.66328
4	156.851	203.248
5	264.77	43.785
6	300.676	286.392
7	358.18	141.86

Fig. 4

LiDAR Lampposts

#	xLiDAR	yLiDAR	Angle
1	48.0341	126.007	135
2	-16.3686	219.959	270
3	140.853	64.5057	270
4	178.457	237.312	90
5	240.017	239.083	180
6	313.536	229.637	315
7	306.539	128.857	225
8	38.1314	-7.04044	0
9	254.081	285.061	135
10	264.77	43.785	26

Fig. 5

After calculating the cost, we find that we only need to spend $73000 on our smart lamppost modification plan.

3.5 Sensitivity Analysis

In this part, we will discuss the sensitivity of our plan and find out whether our plan is useful enough for the city and everyday transportation.

As mentioned in the modification plan, we get a detailed picture of the area we study and the position of each lamppost we need to distribute. By changing the position coordinates of LiDAR sensors and WiFi Access Point randomly, we can see the change of virables mentioned in Evaluation Model.

WiFi Points

#	xWiFi Original	yWiFi Original	xWiFi Edited	yWiFi Edites	S Before	S After	A Distribution	B Distribution Before	B Distribution After
1	41.427	226.357	-	-	0.7233	0.6360	0.7622	0.9336	0.9212
2	264.77	43.785	-	-	0.7233	0.6571	0.7622	0.9336	0.8604
3	95.542	8.66328	-	-	0.7233	0.5601	0.7622	0.9336	1.0207
	156.851	203.248	-	-					
4	300.676	286.392	-	-	0.7233	0.5390	0.7622	0.9336	0.7645
	358.18	141.86	-	-					
5	92.3732	101.392	-	-	0.7233	0.6036	0.7622	0.9336	1.0218
	264.77	43.785	-	-					
6	95.542	8.66328	86.0436	285.186	0.7233	0.6506	0.7622	0.9336	0.8559
7	156.851	203.248	130.134	171.142	0.7233	0.6822	0.7622	0.9336	0.8785
8	264.77	43.785	207.411	228.876	0.7233	0.6450	0.7622	0.9336	0.7967
9	41.427	226.357	1.63815	224.409	0.7233	0.6514	0.7622	0.9336	0.9159
	92.3732	101.392	277.036	180.506					
10	264.77	43.785	147.635	280.308	0.7233	0.6245	0.7622	0.9336	0.7538
	300.676	286.392	374.797	222.762					
11	-	-	1.63815	224.409	0.7233	0.7005	0.7622	0.9336	0.8984
12	-	-	130.777	178.793	0.7233	0.7000	0.7622	0.9336	0.7803
13	-	-	48.0341	126.007	0.7233	0.6934	0.7622	0.9336	0.8300
14	-	-	147.635	280.308	0.7233	0.6839	0.7622	0.9336	0.8155
	-	-	212.927	284.101					
15	-	-	-16.3686	219.959	0.7233	0.6631	0.7622	0.9336	0.8818
	-	-	375.278	193.511					

LiDAR Points

#	xLiDAR Original	yLiDAR Original	Angle Original	xLiDAR Edited	yLiDAR Edited	Amgle Edited	S Before	S After	A Distribution Before	A Distribution After	B Distribution
1	306.5390	128.8570	225.0000	-	-	-	0.7233	0.6896	0.7622	0.7408	0.9336
2	48.0341	126.0070	135.0000	-	-	-	0.7233	0.7071	0.7622	0.7145	0.9336
3	254.0810	285.0610	135.0000	-	-	-	0.7233	0.7001	0.7622	0.7974	0.9336
	140.8530	64.5057	270.0000	-	-	-					
4	313.5360	229.6370	315.0000	-	-	-	0.7233	0.7049	0.7622	0.7679	0.9336
	264.7700	43.7850	26.0000	-	-	-					
5	313.5360	229.6370	315.0000	-	-	-	0.7233	0.6297	0.7622	0.6907	0.9336
	306.5390	128.8570	225.0000	-	-	-					
	264.7700	43.7850	26.0000	-	-	-					
6	254.0810	285.0610	135.0000	254.0810	285.0610	250.0000	0.7233	0.7139	0.7622	0.7622	0.9336
7	140.8530	64.5057	270.0000	140.8530	64.5057	120.0000	0.7233	0.6925	0.7622	0.7622	0.9336
8	240.0170	239.0830	180.0000	86.0436	285.186	180.0000	0.7233	0.6819	0.7622	0.9038	0.9336
9	178.4570	237.3120	90.0000	178.4570	237.3120	270.0000	0.7233	0.6316	0.7622	0.7622	0.9336
	254.0810	285.0610	135.0000	254.0810	285.0610	280.0000					
10	240.0170	239.0830	180.0000	1.63815	224.409	180.0000	0.7233	0.6299	0.7622	0.7919	0.9336
	313.5360	229.6370	315.0000	128.194	9.22972	315.0000					
11	-	-	-	1.0461	276.598	270.0000	0.7233	0.6013	0.7622	0.7436	0.9336
12	-	-	-	221.966	102.653	20.0000	0.7233	0.6444	0.7622	0.7551	0.9336
13	-	-	-	41.4270	226.3570	135.0000	0.7233	0.5889	0.7622	0.7129	0.9336
14	-	-	-	165.918	16.1489	235.0000	0.7233	0.6125	0.7622	0.7741	0.9336
	-	-	-	375.232	273.629	128.0000					
15	-	-	-	165.918	16.1489	235.0000	0.7233	0.5563	0.7622	0.6595	0.9336
	-	-	-	41.4270	226.3570	135.0000					

Fig. describing patterns of change

From these two charts, we get the formula describing the sensitivity of our model.

$$I_{sensitivity} = \frac{\frac{\Delta \mathrm{aD}}{\mathrm{aD}}}{\frac{\Delta S}{S}} = 4.0065$$

This shows that our model is robust and reasonable. It can apply to various changes and can make better distribution plans.

4 Evaluation of Our Own Plan

4.1 Problem Overview

In the last section, we have analyzed the sensitivity of our plan. In this part, we will calculate the coverage area and the coverage rate of our plan accurately. Later, we use our index to evaluate our plan. By getting the number and comparing it with other plans in which some viriables are changed randomly, we will find the strengths and weaknesses of our plan. Later we will show what improvements we decide to make to our model and what we can do with the development of the automatic driving system. A dynamic model is developed to solve the problem of unaccurate evaluation.

4.2 Result Analysis

Strength

-

Weaknesses and Expectation

- **The evaluation model is static.**

  In the first part, we ignore the movement of cars, this will reduce the accuracy of this model.

- **There is difference between our model and reality.**

  Nowadays, the WiFi technology is not so effecient and the calculation on the client side doesn't have such a speed as assumed in our model. Our model should make improvements. In addition, we should always pay attention to new technology while the technology is developing.

Fig. The Map of Selected City Areas

References

[1] https://www.sae.org/standards/content/j3016_201806/

[2] https://www.map.gov.hk/gm/map/

[3] https://www.mathworks.com/matlabcentral/answers/180860-check-if-2d-points-are-evenly-distributed

[4] Andre Violante, Simple Reinforcement Learning: Q-learning,https://towardsdatascience.com/simple-reinforcement-learning-q-learning-fcddc4b6fe56