Going Places: Modeling a Public Transportation System using Reinforcement Learning

Ingrid Fan\textsuperscript{1,*}, Madison Hall\textsuperscript{1,+}, and Anna Yang\textsuperscript{1,||}

\textsuperscript{1}Student, Department of Computer Science, Stanford, CA, 94305
\textsuperscript{*}ifan@stanford.edu
\textsuperscript{+}mhall38@stanford.edu
\textsuperscript{||}ayang7@stanford.edu

ABSTRACT

The goal of our project is to develop a public transportation design algorithm that can model a new public transportation system in San Francisco with the intention to maximize city planning efficiency. This model is built on historical Uber datasets and geographical data of San Francisco from the OpenStreetMap API. We modeled the intersections onto a graph model using OSMnx, and implemented K-Means on the Uber data to determine 10 clusters to represent stations for the public transportation system. We then used Q learning to determine the best policy from any intersection to each particular station. Mapping the results of K-Means to a map, we can see that the 10 clusters determined appear to be well-spread out, with well-placed in the popular districts of San Francisco. The policy outputted from our Q-Learning implementation yielded promising results when tested on smaller subsections of San Francisco.

Keywords: Public Transportation, City Planning, K-Means, Q-Learning, Reinforcement Learning

Introduction

The public transportation system is an integral part of millions of Americans’ daily lives, whether it may be for work or play. With San Francisco having among the nation’s highest public transit usage\textsuperscript{1}, we were inspired to design a model that provides a quick and easy solution to ensure that any future mode of public transportation in San Francisco is up-to-date with commuters’ expectations.

We found that the best public dataset of commuter information in San Francisco comes from rideshare companies, such as Uber. Uber is a ride-hailing giant that allows customers to book drivers using their own cars, fulfilling over 40 million rides monthly. Uber has fundamentally changed the way that cities think about public transportation. Although Uber and other ride-sharing services do promote carpooling and are a step towards alleviating traffic congestion, many commuters are opting towards rideshare instead of using public transportation due to long commute times and inconveniently located stops/stations. According to a study by the UC Davis Institute of Transportation Studies, public transit has seen a 6% decline in use by commuters in major cities after the development of ride-hailing companies such as Uber and Lyft\textsuperscript{2}. Thus, while rideshare services have collected important data on the general flow of commuters, these services are still adding to the problem of traffic congestion.

The model that we intend to create will be a way for city-planners to ensure that they are basing their design on modern data while reducing their unknowns of where the stations should be located and how the stations should be connected. Essentially, we are determining the ideal locations to place new public transit stations as well as the least costly paths from one station to another, to create a more efficient and less costly public transportation system that commuters will be willing to take.

Motivation

We were motivated to tackle this problem due to our own experiences with public transportation in the Bay Area, especially in San Francisco. The current four main modes of public transportation in the Bay – The San Francisco Municipal Railway, The Caltrain, SamTrans, and the Bay Area Rapid Transit (commonly known as BART) – collectively cover significant surface area but are inconvenient in terms of timing and transferring between the four
public transportation companies. This inconvenience is apparent in the numbers; Bay Area Commuter information from 2017 reported a decrease of 12.3 percent in average weekly ridership on all its services and a decrease of 3.7 percent of Caltrain ridership compared to the previous year.3

Furthermore, as rideshare continues to grow, big rideshare companies are introducing models that could quickly price traditional public transportation services out of the industry. For example, Lyft’s recent launch of Lyft Shuttle, in which a carpool shuttle that travels around a designated route during peak hours and makes stops along the path, will likely overtake the market of more affluent commuters, leaving lower-income workers, many of whom are traveling hours every day due to the Bay Area Housing Crisis4 and are reliant on reasonable public transportation prices, to support the public transportation services in place. Thus, we find it in the best interest of city planners to look further into optimizing public transportation, which is beneficial for reducing fuel emissions, traffic congestion, and for increasing the convenience of daily commuters.

**Literature Review**

The concern of the growing pressure on public transportation caused by the entry of on-demand ride-sharing services such as Uber and Lyft has been well-studied in recent years, including in literature by Sadowsky, et. al.5 Sadowsky notes that subway ridership in New York fell in 2016 for the first time since 2009- this is no coincidence considering Uber began offering rides in New York in May 2011. Several explanations have been offered for this phenomenon, including time and cost.

Schwieterman and Michel6 analyzed consumer preferences in their 2016 University of Chicago study, where they compared the price and time of fifty trips in the Chicago area. Each trip started at the same time, but one person used UberPool (Uber’s carpooling service where a passenger will be sharing the trip with other people), and the other used Chicago’s public transit system (bus or subway line). The average time and cost for an UberPool trip was 35:52 minutes and $9.66, respectively. Interestingly, the average time and cost for the same trip on public transportation was 48:29 minutes and $2.29, respectively. Ultimately, they found that consumers didn’t mind paying over 4 times the public transportation cost if it meant a 13 minute deduction in commute.

Q-Learning, a popular reinforcement learning (RL) technique, has been shown to perform well in path-finding problems because it does not require a concrete model of the environment and can handle problems with stochastic transitions and rewards.7 Konar et. al.8 demonstrate how Q-Learning can be used for path planning of a mobile robot, which is similar to our task of interest: given Uber drop-off and pick-up locations, find best locations and best possible paths between these locations. Additionally, Konar et. al describe one of their main objectives to be to use Q-Learning to determine an obstacle-free trajectory of optimal path length and energy. Their positive results are promising for our analogous objective of computing the best path given the ”obstacle” of intersections.

**Data Analysis & Visualization**

We pulled data and APIs from a few different sources to make this project possible. First, we extracted data from Uber Movement, Uber’s city-based public datasets. Their San Francisco specific data set gave us information on where Uber trips frequently begin or end in the city, thus giving us a scale of which sectors of San Francisco have the highest demand for transportation services. This data is the bulk of what we used to determine the best locations for stations in our public transportation model.

<table>
<thead>
<tr>
<th>sourceid</th>
<th>dstid</th>
<th>hod</th>
<th>mean_travel_time</th>
<th>standard_deviation_travel_time</th>
<th>geometric_mean_travel_time</th>
<th>geometric_standard_deviation_travel_time</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>5</td>
<td>17</td>
<td>2917.18</td>
<td>790.28</td>
<td>2810.98</td>
<td>1.32</td>
</tr>
</tbody>
</table>

**Figure 1.** Example of a line of data from Uber Movement. Includes an Uber trip’s pick up location, drop off location, hour of day, and travel time.

The APIs we utilized include GoogleV3’s geolocator API and OpenStreetMap API. We used GoogleV3’s
geolocator API to convert the Uber Movement data’s positional IDs respective addresses to applicable latitude and longitude coordinate pairs, as our algorithm requires continuous data for the best results. Additionally, we extracted the coordinates of all intersections in San Francisco from the OpenStreetMap API. GraphServer and OSMnx were also used to make a visual representation of the intersections given by OpenStreetMap.

Modeling

We modeled San Francisco as a graph using (latitude, longitude) coordinates of all possible driving intersections. Each intersection corresponds with one node in the graph, and the set of edges for a particular node corresponded with the possible driving paths from that node.

![Image](image.png)

**Figure 2.** All intersections and driving paths within the San Francisco city borders

Implementation

K-Means

To accurately assess the landscape of the most popular drop off and pick up locations in San Francisco, we decided to implement K-Means. K-Means is a common clustering algorithm that clusters data by trying to separate samples into $n$ groups of equal variance. Additionally, we chose K-Means as K-means is known to converge quickly to a local optimum and uses an iterative refinement technique, which is useful for capturing variance in data. These characteristics of K-Means apply well to our goal of finding which points would be best for new transit stations.

We utilized Uber Movement data, which has data from over two billion trips\(^9\). The data contained drop-off and pick-up locations to the nearest block with each location anonymized using a "movement ID".

We converted our pick-up and drop-off addresses to latitude and longitude, which are continuous values, and could more accurately represent our San Francisco as a "city grid". Next, to achieve our goal of using clusters to demonstrate best new transit points, which we experimented with different $n$ values. Ultimately, after observing the eight BART stations and two Caltrain stations currently in San Francisco, we settled on training our model with $n = 10$, which we find to be the most representative of the current transportation system in San Francisco. We chose to have the option of setting the Caltrain station as our eleventh node for the purposes of increasing accessibility. For the next phase of our project, Q-Learning, we used the 10 cluster points found, along with the Caltrain Station, as end states for the algorithm.
Q-Learning

We chose to use Q-Learning to find the optimal path between each stop as Q-Learning is an off policy method and is guaranteed to converge at the optimal strategy.

We defined the states as the coordinates of every street intersection in the city of San Francisco, with its adjacent intersections’ coordinates as its next states. The next states were found by looking at each nodes’ edges in the graph, and finding the end node of each edge. The actions are deterministic to the next state. The rewards for each (state, next state) pair was the negated sum of a Euclidean distance function and the price estimate from the next state to the end node. Specifically, to calculate our price estimate, we utilized Uber Developers’ Price Estimate API. All (state, action) pairs were seen in the data and thus no Q values approximations were necessary to build the policy. The problem was modeled with a discount rate of 0.95 to ensure that we are taking account of rewards in the future, which is necessary to best avoid dead-ends and unnecessary looping or detouring.

<table>
<thead>
<tr>
<th>s</th>
<th>a</th>
<th>r</th>
<th>sp</th>
</tr>
</thead>
<tbody>
<tr>
<td>(-122.4954804, 37.7705085)</td>
<td>(-122.4953209, 37.7701508)</td>
<td>-0.063165843</td>
<td>(-122.4953209, 37.7701508)</td>
</tr>
<tr>
<td>(-122.4954804, 37.7705085)</td>
<td>(-122.4955809, 37.7704608)</td>
<td>-0.063498071</td>
<td>(-122.4955809, 37.7704608)</td>
</tr>
<tr>
<td>(-122.4954804, 37.7705085)</td>
<td>(-122.4956671, 37.7707439)</td>
<td>-0.063656478</td>
<td>(-122.4956671, 37.7707439)</td>
</tr>
<tr>
<td>(-122.4729944, 37.7878482)</td>
<td>(-122.4736948, 37.7877336)</td>
<td>-0.052038582</td>
<td>(-122.4736948, 37.7877336)</td>
</tr>
<tr>
<td>(-122.4729944, 37.7878482)</td>
<td>(-122.4730422, 37.7878459)</td>
<td>-0.051621026</td>
<td>(-122.4730422, 37.7878459)</td>
</tr>
</tbody>
</table>

Figure 3. Subset of our States, Actions, Rewards, and New States dataset prepared for Q-Learning.

Figure 4. graphical baseline representation of best paths calculated using A* between clusters calculated from our K-Means implementation

(a) Shortest route from Caltrain Station to other stations  (b) All shortest routes from each station to all other stations
Results & Analysis

Our K-Means algorithms outputted these 10 locations, which we are using as our public transit station locations. We manually set an 11th location as the Caltrain station (700 4th St, San Francisco, CA 94107) to increase the accessibility of our model.

POLICY TO STATION 7
(-122.4954804, 37.7705085) to (-122.4953209, 37.7701508)
(-122.4729944, 37.7878482) to (-122.4729994, 37.7878927)
(-122.4522533, 37.7628529) to (-122.4508093, 37.7630256)
(-122.4104937, 37.8103442) to (-122.4106063, 37.8103015)
(-122.4589088, 37.7673959) to (-122.4589051, 37.7674464)

Figure 6. Subset of our policy for a path to our 7th station at coordinates (-122.4343379, 37.7536888)

For each station, we calculated a policy that takes in the station as the End State. Our Q-Learning algorithm returned a policy that gives the best action at any intersection in San Francisco. Thus, from each policy, we are able to extract the calculated route to each station from any intersection in the city. Setting our start state at any other station location and following a station’s resulting policy yields the best route between the two stations.

Error Analysis

Visually, we can see that our stations are nicely spread throughout the city and thus covers a large surface area, with some stations centrally located in popular areas such as the Mission District and the Financial District. Possible errors include locational bias due to the Uber Movement data that our K-Means algorithm ran on. For privacy purposes, Uber cannot release exact addresses of each pick-up and drop-off location but instead generalize all pick-up and drop-offs into an address that is representative of an entire sector.

We tested our Q-Learning algorithm using a small subset of intersections, and analyzed the policy outputted for the subset. We manually mapped out the best path based on the optimal policy to a particular end node in the subset given to us by Q-learning.
The path outputted given the policy appears to be the optimal (lowest cost) path possible from the particular start node to the end node. The cost of a path is dependent on the distance as well as the price estimate of a path - we can see that the path given was in fact, one of the paths with the shortest distance taken from start to end.

**Challenges**

While our algorithm was tested and analyzed on a smaller subset of our data, we found expanding our error analysis to analyze our complete model to be tedious and time consuming. Further work includes creating a more qualitative analysis function for our model.

With more information on the costs and utility of increasing or decreasing the number of stations, we could better determine the optimal number of stations to base our model on. Our algorithm is flexible to changing this value, however, and therefore this aspect could be easily implemented at any point that more information on this matter is obtained.

Our cost function was determined using a price estimate as well as distance, however, commuters may not weigh these equally. With additional information about how much a commuter may weigh the price estimate of the ride versus how far away the end location is, we could better determine the reward function that best represents the costs and rewards in the real world.

As stated earlier, our extracted Uber data splits San Francisco into a few hundred different sectors and snaps all addresses within each sector to a specific address. Consequently, the locations of the stations are biased to the addresses that Uber chose for each sector. Greater accuracy could be obtained with more specific commuter data.

**References**

4. Carter, S. M. This 30-year-old commutes 4 hours, and 140 miles, every day so he doesn’t have to pay $4,500-a-month San Francisco rent. *CNBC Make It* (2018).


**Author contributions statement**

I.F. found and cleaned Uber data, cleaned and calculated SARSP data (including implementing reward function for SARSP data), extracted and cleaned geographical intersections data, designed and implemented Q-Learning on our model, modeled policy on small graph, analyzed results, wrote report.

M.H. Cleaned Uber data, converted the movement IDs to real street addresses, designed and implemented K-Means, organized cluster data, implemented Uber Price API for rewards, conducted literature review, wrote report.

A.Y. Cleaned Uber data, researched API’s for lat-long conversion, converted real street addresses to lat-longs using Geocoding API, researched graphing API’s and OpenStreetMap, created graph model using OSMnx, found states/nodes and edges and next states for Q-Learning, created graph figures for report, analyzed results, wrote report.