Investigating Uber Speed Data: Research Project Report

YIDUO WANG and HAN NGUYEN, Wellesely College, USA

ACM Reference Format:
Yiduo Wang and Han Nguyen. 2024. Investigating Uber Speed Data: Research Project Report. 1, 1 (May 2024), 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The accessibility of complete transportation data in the United States for research purposes has been hindered by barriers such as the costly Google API and the lack or cessation of data sharing by riding apps since 2023. An effective approach to mitigate these challenges involves harnessing open-source map data, such as OpenStreetMap (OSM). Nevertheless, owing to its reliance on crowd-sourced contributions, OSM often suffers from incomplete road segment information, particularly in the US context. To address this deficiency systematically, a two-fold strategy is proposed. Initially, the veracity of available Uber data could be rigorously validated against the comprehensive ground truth provided by the Google API. Following this validation process, the Uber data, authenticated as ground truth, could serve as a foundational dataset for training machine learning models aimed at extrapolating missing road segment data.

Our ultimate objective is to gather comprehensive speed data for every road, enhancing the precision of travel time predictions between various locations, particularly within San Francisco. Our selection of San Francisco as our study area purely stems from the availability of several months' worth of data from Uber Movement, which we secured prior to the cessation of public data access by the platform.

In our research efforts, our initial approach involved applying the shortest path algorithm to OpenStreetMap (OSM) data and juxtaposing the resulting travel times with those derived from Uber's data. However, we encountered disparities due to the absence of certain nodes in OSM that were present in the Uber map, and vice versa. Recognizing this discrepancy, we redirected our focus towards utilizing the Google API. Our aim was twofold: first, to validate the accuracy of Uber data, and second, to employ it as a foundational dataset for training machine learning models tasked with filling in the missing data.

The success of this research holds the potential to bridge the gap between the availability and reliability of transportation data in the United States. By validating the accuracy of Uber data against the established ground truth and successfully training machine learning models to fill in missing road segment data, this research stands to advance transportation science and practice. Ultimately, the culmination of these efforts would enable researchers, policymakers, and practitioners to make more informed decisions and develop more effective transportation solutions.

Authors' Contact Information: Yiduo Wang; Han Nguyen, Wellesely College, Wellesley, Massachusetts, USA.

ACM XXXX-XXXX/2024/5-ART
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 DATA

Our project uses data from three main sources: OSM, Uber Movement, and Google Matrix API.

2.1 OSM Data

OSM is short for OpenStreetMap, the most comprehensive source of geographic map established in 2004. OSM data on San Francisco is retrieved using pyrosm library. There are two separated datasets downloaded: an "edges" dataset with 262768 rows and 36 columns and a "nodes" dataset with 242424 rows and 9 columns.

2.1.1 Edges.

. The 'edges' dataset describes street features such as description of road types, description of conjunction, access permission, etc. For this research project, the 'edges' dataset is used to calculate the average travel time for comparison with the time calculated from Uber dataset. Predominantly, we are only looking at the following relevant columns: 'maxspeed', 'id', 'geometry', 'u', 'v', and 'length'.

The columns 'geometry', 'u', 'v', and 'id' are used as an identifier of routes for comparison with Uber dataset, where 'geometry' is the coordinates, 'id' is a unique identifier' and 'u', 'v' are respectively the start node id and the end node id of a route. The 'length' column measures the length of a route in meters, which is used as the default length for calculating the average travel time for both OSM and Uber data. The column 'maxspeed' measures the maximum speed a vehicle can travel in a designated route.

Since 'maxspeed' stores string values in the following formal "### mph", we first remove its unit. Before calculating the average time traveled in seconds, we convert all the 'maxspeed' values to a numbers. Next, in order to calculate the average travel time in seconds, we have to convert between units since 'maxspeed' is measured in mph while 'length' is measured in meters. The following formula is used:

$$Average\ travel\ time = \frac{length}{maxspeed \times 0.447}$$

2.1.2 Node.

. The "node" dataset contains information about nodes that make up different routes in a map. Ultimately, we are interested in only the 'lat', 'lon', and 'id' columns. The 'id' column is a unique identifier to be compared to identify specific nodes. Information about 'lat' and 'lon' are used to retrieve average time travel from Google Matrix API for comparison with Uber dataset.

2.2 Uber Movement Data

Uber movement data is a collection of aggregated trip data collected from over 10,000 cities all over the world. Data regarding San Francisco neighborhoods at different times throughout all day of March 2020 is downloaded from Uber movement website. Unfortunately, as of October 1st 2023, movement data is no longer available on their website.

The dataset retrieved from Uber movement consists of 13 columns and 10358729 rows. We are interested in only columns 'osm_way_id', 'osm_start_node_id', 'osm_end_node_id', and 'speed_mph_mean'. The 'osm_way_id' is the unique identifier of a route while the 'osm_start_node_id' and 'osm_end_node_id' are respectively the start node id and the end node id of a route. All these three columns are used to merge with OSM dataset.

Similarly, the 'speed_mph_mean' column is used to calculate the average travel time in seconds of a route. Since it is measured in mph, unit conversion is necessary before calculating the average

travel time. The same formula used on OSM data is used on Uber data:

$$Average\ travel\ time = \frac{length}{maxspeed \times 0.447}$$

2.3 Google Data

At first, we used Google Distance Matrix API because it is most relevant to our end goal, allowing users to identify the most efficient travel routes between multiple possible origins and destinations. However, for reasons not identified, it worked with the example nodes on its website but did not work with most of our nodes when we fed in their coordinates or place id. Therefore, we switched to Google Directions API, which accepts the place id of our nodes and return the directions of most efficient routes when calculating directions and the duration. The API request took the following form: https://maps.googleapis.com/maps/api/directions/outputFormat?parameters. The result returned by Google Distance Matrix API is formatted as a JSON file and returns comprehensive information regarding the nodes, distance, duration, and other relevant details of the specified route or routes between the given origin and destination points.

The provided Python script employs functions to extract geographic coordinates and retrieve travel information between specified locations. Through integration with Google's Geocoding API, the script converts latitude and longitude coordinates into unique place identifiers (IDs). Subsequently, it utilizes the Google Directions API to compute travel distances and durations between pairs of locations specified by their respective place IDs. The script's implementation includes error handling mechanisms to ensure robustness and minimize redundant API calls, thereby enhancing efficiency.

3 SHORTEST PATH ALGORITHM

The rows in the merged data are then categorized based on which dataset(s) have their speed data. Each road is classified as either "Both", "OSM", "Uber", or "Neither", depending on whether it has speed data from both Uber and OSM, only from OSM, only from Uber, or from neither, respectively.

An undirected graph is then created using networkx, which is populated nodes, edges, and the travel time calculated from the Uber data and OSM data. To compute the shortest weighted time traveled between two different nodes, we use Dijkstra's shortest path algorithm.

Initially, 1000 random combinations of nodes are generated from the merged dataset. Since the order of start and end nodes are shuffled, these pairs of nodes are completely new. Dijkstra's method is then used on these 1000 pairs to compute the average travel time using Uber data. However, given repeated sampling, less than 100 pairs of these nodes form a route that exist in both the Uber dataset and OSM dataset.

Our team transitions to choosing 1000 random pairs of nodes from the existing pairs. We then run the Dijkstra's algorithm on these 1000 pairs of nodes and obtain the average travel time generated from Uber dataset.

4 CONTRIBUTION

After cleaning OSM and Uber datasets, we construct 2 graphs comparing the speed between OSM and Uber and the average time traveled between OSM and Uber. However, there seems to be no apparent trend between OSM data and Uber data.

Initially, we hope to use the Dijkstra's shortest path algorithm on both the Uber and OSM datasets. However, there are less than 100 pairs of nodes where a path exists on both OSM and Uber datasets. We then only use Dijkstra's algorithm to obtain 2027 pairs of unique nodes and its corresponding average time traveled using Uber data.

Using geocode, we obtain 100 pairs of nodes with its corresponding travel time calculated from Uber data and obtained from Google data. However, the travel time between 2 datasets seem to be very different from each other.

Fig. 1. Comparison of speed between OSM and Uber

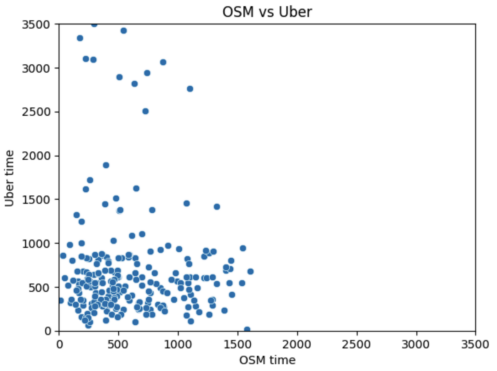

Fig. 2. Comparison of time travelled between OSM and Uber

5 DISCUSSION

One challenge we encounter when utilizing the Google Directions API lies in the format of the time data it returns, which is presented in a human-readable format. This contrasts with our requirement for time data in seconds, particularly for seamless integration with Uber's time-related functionalities. To address this issue, our future plans entail implementing a solution to parse the time data returned by the Google Directions API by time units ("day", "hour"," second"). This parsed data will then undergo conversion into seconds, aligning it with the format required for compatibility with Uber's time-based calculations and operations. By implementing this strategy, we aim to ensure seamless interoperability between the Google Directions API and Uber's time specifications, facilitating smoother integration and enhancing the functionality and efficiency of our application.

Nevertheless, from the first 100 pair comparisons we've observed between Uber travel time and Google travel time, they are all measured in minutes. Therefore, we separated the string by space and converted it to seconds for more precise analysis. Notably, there appears to be a substantial variance between the two datasets, prompting us to question whether there's a significant distinction in how the duration is calculated. One possibility is a variance in mode; for instance, Google might include walking time in its estimates, whereas Uber focuses solely on driving time.

To delve deeper into this comparison, I conducted a correlation analysis using STATA, resulting in a correlation coefficient of -0.6957. This negative correlation suggests a moderately strong inverse relationship between Uber and Google travel times within our sample. Further investigation into the methodology and variables used by each platform may elucidate the factors contributing to this disparity and correlation.