Let's Make A Toy Google Map

Xiyang Hu* and Shuang Lin†

*Computer Science & Engineering
Lehigh University
Bethlehem, PA, USA
xih322@lehigh.edu

†Cognitive Science
Lehigh University
Bethlehem, PA, USA
shl224@lehigh.edu

***Abstract*—Personalized travel time, estimations, and map matching are the fundamental functionalities of Google Maps. This report aims to document the creation of a toy model for Google Maps with its functionalities built using current and classic researched methods. The goal is to predict a trip using a given starting location and destination and then estimate personalized travel time. Each vehicle's traces are matched to corresponding road segments for map matching. The traffic speeds for the road segments are predicted based on the map-matched output. The resulting application can output vehicle-specific travel time estimates for any user-created trips.**

I. Introduction

Google Maps is a popular mapping tool using mobility traces from vehicles, to aircraft, to walking. It is able to produce a travel time prediction personalized to the user. Google Maps is essential to daily commuters as well as corporate operations. The same purpose of finding the fastest route in a time-dependent environment is shared among its users: commuting workers, delivery services, emergency vehicles, etc. In times when Google Maps is unavailable, another mapping tool is needed to ensure accurate predictions. Our project aims to create a toy version of Google Maps that will fill its role in predicting an efficient, personalized travel time estimation for a trip between a starting point and a destination. This will be done in a vehicle context, fulfilling the fundamentals of daily usage.

II. Methodology

Our project is split into three stages: Stage 1: Map Matching, Stage 2: Traffic Speed Estimation, and Stage 3: Personalized Travel Time Prediction. Each stage produces an input-output model that can be run in a stage order to produce the desired result.

A. Stage 1: Map Matching

1) Handcraft HMM Map Matching: We use the classic map matching algorithm that implements a Hidden Markov Model (HMM) [4] to achieve map matching. First, a list of candidate roads is found based on the closest distances to the vehicle trace point. Then, the HMM map matching will be performed by calculating three probabilities.

1. The Initial State Probability:

$$\pi_i = p(z_1|r_i) \tag{1}$$

Starting at the first measurement, this returns the probability of the vehicle's first road over all the candidate roads.

2. The Emission Probability:

$$p(z_t|r_i) = \frac{1}{\sqrt{2\pi}\sigma_z} e^{-0.5(\frac{||z_t - x_{t,i}||_{\textit{great circle}}}{\sigma_z})^2} \tag{2}$$

Here, z_t is a GPS point measurement, r_i is the road segment, and σ_z is the standard deviation of GPS measurements. The closest point on the candidate road segment is $x_{t,i}$, and the distance between the GPS point and the candidate road point is measured using great circle distance to account for the surface of the Earth.

3. The Transition Probability:

$$d_t = \mid ||z_t - z_{t+1}||_{\textit{great circle}} - ||x_{t,i} - x_{t+1,j}||_{route} \mid \tag{3}$$

This equation returns the probability of a vehicle moving between candidate road matches at the time between the current GPS point and the next GPS point. Here, z_t is the GPS measurement, and z_{t+1} is the next measurement. The latitude/longitude point of the road segment closest to the GPS point is $x_{t,i}$ while the next road segment point for the next GPS point is $x_{t+1,j}$. The difference between these two points is the route or driving, distance $||x_{t,i} - x_{t+1,j}||_{route}$. The route distance is then compared to the great circle distance between the two GPS measurement points $||z_t - z_{t+1}||_{\textit{great circle}}$.

Lastly, the Viterbi algorithm is used for backward mapping. The algorithm computes the maximum probability and the optimal sequence to get the best path through the HMM lattice. The best path maximizes the product of the measurement and transition probabilities. Each car's GPS point will have a corresponding point on the road network in this best path.

2) Constructed HMM Map Matching: After handcrafting a HMM map matching method, we explored an already constructed map matching method, Leuven Map Matching [3]. This method aligns a trace of GPS measurements to a map or road segments. We realized that this map matching is also based on a Hidden Markov Model (HMM) but with non-emitting states. Hence, the model can deal with missing data.

B. Stage 2: Traffic Speed Estimation

The traffic speed estimation we implemented will be split into three steps. First, we will need more nodes on one trip to create a precise model of speed estimations. Then, we can calculate the speed for each road segment. And at last, we will check all the empty road segments and assign a value to them.

1) Density Function: As a recall, our first stage's HMM map matching result is a predicted matched path composited by a group of nodes from the road network. Therefore, to assign the speed to a corresponding road segment, we need more GPS points for a trip. The density function will take two GPS points and add more points in between them. The in-between nodes will keep the same features as the others, such as time, location, etc. The node splitting is based on 15-second gaps, so every 15s between two nodes will be fulfilled with a node.

2) Compute Speed: The time difference between a road segment's start and end nodes will be taken in seconds. Then, the great circle distance between the given coordinates of those two nodes will be calculated in meters. The division of the two will result in a speed of meters per second. Then it can be multiplied and converted to miles per hour with a constant of 2.23693629. The road segments and their speeds are recorded to the road network, and the simple moving average (SMA) will be applied to filter a speed trend.

3) Repair unreachable road network: After computing speed for road segments, we realized that many road segments do not have the speed information due to the limited training data. For the roads without training data, the average of the sum of the nearest road speeds can be used as an estimation.

C. Stage 3: Personalized Travel Time Prediction

Personalization equates to vehicle-specificity because we assume the same driver for the exact vehicle. Each taxi vehicle represents one vendor, so the time prediction is personalized to the car. This stage is done in two steps, feature extraction and using prediction. The prediction is done by eXtreme Gradient Boosting (XGBoost), which is an ensemble learning method [2].

1) Feature Extraction: We implement six functions to process the data to extract hidden features. First, we use Principal component analysis (PCA), not to reduce dimensions but to rotate the data. This rotation helps decision tree splits, which is the cornerstone of XGBoost. Second, the Haversine distance is calculated between pickup and dropoff points for passengers in kilometers. This is not the road distance. Third, we compute time series from the initial time feature. Fourth, we extract the average speed at the time series of hours, weekdays, and hours of the week. And we also extract regional average speeds. Fifth, spatial clustering is performed on the pickup and dropoff points with the K-means algorithm [6]. Sixth, the time series and clusters are aggregated to a spatiotemporal duration feature.

2) XGBoost Prediction: The XGBoost model is an ensemble learning method that combines weaker models' predictions to produce a stronger one. First, a model is built from the training data. Then, the second model is built to correct the errors in the first one. This process repeats until the correct prediction threshold or a maximum model threshold is reached. Some features will not be used as they might delude or have no effect on the prediction. Most of the extracted features will be used to train the XGBoost model. The model will predict the travel times of each trip of each vendor.

III. Experiments

A. Dataset description and pre-processing

The road network data is the city map of San Francisco gathered from OSMnx [1]. The map is pre-processed with the nodes' coordinates and IDs. The edges of the map are taken as two separate nodes labeled as the start node and end node. Both start and end nodes' locations are extracted along with their IDs. The mobility trace data is provided by the Cabspotting project from the Exploratorium Museum in San Francisco [5]. The training data consists of 14,792 rows with four features: latitude, longitude, occupancy, and recording time. The occupancy feature is a binary indicator of whether the taxi is occupied or not. The time feature includes both date and time of day. The training data is then pre-processed to extract each trip with features of the previous node's coordinates and the great circle distance between the current and the previous nodes.

IV. Result

A. Stage 1: Map Matching

For stage 1, we will show the results of two methods: our handcraft HMM and Leuven HMM map matching. The result of handcraft is shown in Table I. Here, we list two trip map-matching results. This result is the difference between the trip distance and the predicted distance. Our current result is acceptable because we calculate the path using the straight-line distance between two points, so our accuracy is very low due to the incoherence of GPS points. Then, as shown in Fig 1, we visualize the predicted path for our handcraft HMM map matching method. In this figure, the blue line represents the GPS data, and the green line is our matching path. It is not difficult to see that the results of both Route 1 and Route 2 are gratifying. The Leuven HMM map matching method has a better result, as shown in Fig 2. In this figure, the left side is the map-matching result and the trip corresponding to each GPS point, while the right side is the mapped trajectory presented on the road network. In the right side image, the red point is the GPS point, while the blue line is for the matching path. It is not

TABLE I
THE MODEL ACCURACY.

Trip 1	0.03531
Trip 2	0.21611

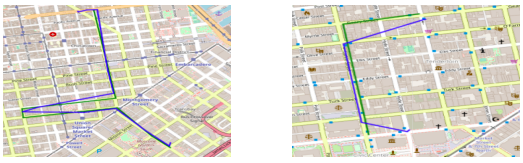

Fig. 1. Visualization Result of Handcraft HMM map matching

difficult to find that the Leuven HMM map matching method is more beautiful and clear. At the same time, it successfully stores the information on each road segment of the matching path, which is convenient for the next stage of use.

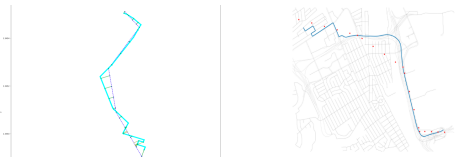

Fig. 2. Visualization Result of Leuven HMM map matching

B. Stage 2: Traffic Speed Estimation

The result of stage 2 is shown in Table II. Here, the result of each road segment is stored in the road network. We combine our own data with the existing road network to obtain our own road network. When the road network is established, some road speed information will be stored in advance, which in our table, we use blue color to indicate. We only need to extract the average value when we store our results, where green speed means the resulting speed after the average. In addition, the advantage of this is that some unpredictable roads will have speed in advance, saving calculation time. At the same time, it has greatly accelerated the speed at which we improve the road network. In the table, the red color speed is the speed we estimated.

TABLE II
THE ROAD NETWORK AFTER STAGE 2.

Name	highway	oneway	length	maxspeed(mph)
Howard Street	secondary	True	277.735	25
Howard Street	secondary	True	89.640	25
Howard Street	secondary	True	49.601	25
Howard Street	secondary	True	136.867	25
Howard Street	secondary	True	277.309	20
4th Street	secondary	True	188.240	16.75
4th Street	secondary	True	2.675	7.625
4th Street	secondary	True	57.863	14.75

C. Stage 3: Personalized Travel Time Prediction

The stage 3 result will be split into feature extraction and XGBoost Prediction. The feature extraction shows how we discover hidden features in the training data. The XGBoost prediction shows the MAE result of our predicted result.

1) Feature Extraction: The extraction process produced 38 new hidden features for the training of the XGBoost model.

As shown in Fig 3, the PCA rotates the left image into the right image.

As shown in Fig 4, we observe that the speed and time of the day have a huge relation.

As shown in Fig 5, the average speed is split on a regional level with the highest speeds in the green zones.

As shown in Fig 6, we can see that we successfully cluster all the regions based on the dropoff locations.

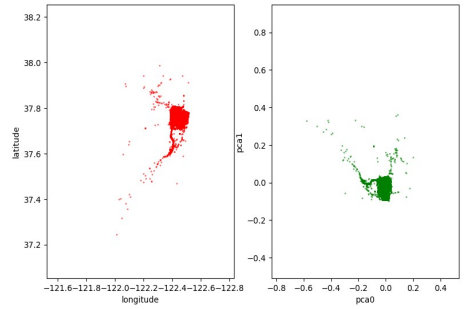

Fig. 3. Visualization Result of PCA transformation

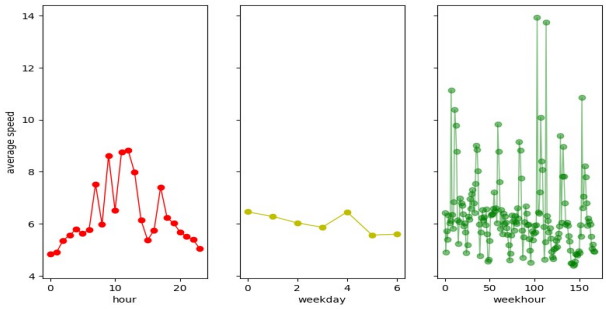

Fig. 4. Visualization Result of Average Speeds at Time Series

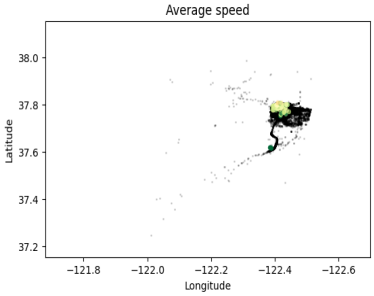

Fig. 5. Visualization Result of Average Speeds at Regions

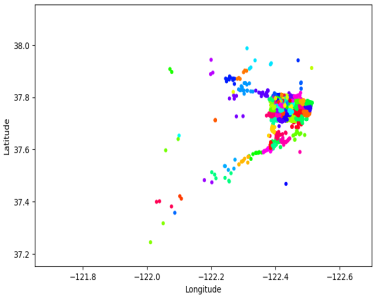

Fig. 6. Visualization Result of Spatial Clustering

2) XGBoost Prediction: The performance evaluation is based on the mean absolute error (MAE). Our personalized travel time prediction model resulted in an MAE of 199.499189 seconds.

V. Conclusion

In this paper, we implement a toy Google Map which can map the given trip into the real network, estimate each road segment's speed, and finally estimate the time for each trip. In the future, we can improve our model by more hyperparameter tuning of the XGBoost model. We can also implement a deep learning model that will outperform the XGBoost model since XGBoost sits in between the spectrum of machine learning and deep learning.

VI. Reference

References

[1] Geoff Boeing. Osmnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*, 65:126–139, 2017.

[2] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM.

[3] Wannes Meert and Mathias Verbeke. Hmm with non-emitting states for map matching. In *European Conference on Data Analysis (ECDA), Date: 2018/07/04-2018/07/06, Location: Paderborn, Germany*, 2018.

[4] Paul Newson and John Krumm. Hidden markov map matching through noise and sparseness. In *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, pages 336–343, 2009.

[5] Michal Piorkowski, Natasa Sarafijanovoc-Djukic, and Matthias Grossglauser. A Parsimonious Model of Mobile Partitioned Networks with Clustering. In *The First International Conference on COMmunication Systems and NETworkS (COMSNETS)*, January 2009.

[6] Junjie Wu. *Advances in K-means clustering: a data mining thinking*. Springer Science & Business Media, 2012.