Bayesian Estimation of Origin and Destination from Masked Trip Data

Yuneil Yeo[1], Chenming Niu[1], and Maria Laura Delle Monache[1]

Abstract— **This article introduces a statistical method to estimate trips origin and destination locations from a masked trip data set. The estimation method uses trip features, the graph of the network, and publicly accessible external information on the real-time congestion status to find the most probable trips origin and destination based on a Bayesian approach, Markov Chain rule, and rank aggregation method. A case study of Porto, Portugal assesses the performance of the statistical estimation method by comparing the estimated location with the centroids of reported locations and with the actual trip origin and destination. Despite the limitation of the available data, the method provides better estimates of trips origin and destination compared to the centroids of reported locations.**

I. Introduction

With the emerging use of data on real-world problems, more agencies are making data open to the public. Before doing so, agencies mask data to protect citizens' privacy. However, there is a trade-off between privacy protection and the loss of information [14]. Depending on the extensivity of the masking method, the direct analysis not only of individual features but also of aggregate features can be restricted [14]. This limitation has stimulated researchers to explore ways to fully harness the potential of obfuscated data.

One way is to extract the individual features from the masked data. The efforts to extract individual features have been made in multiple fields. Notably, there have been many works related to estimating system components reliability through maximum likelihood estimation based on different lifetime distributions [20], the Bayes approach [20], and the EM algorithm [24]. In [13], authors present a deep neural network framework capable of classifying masked images into categories. In [7], patterned dropout is used to estimate the emissions of undisclosed companies.

Similar works have been conducted in the field of transportation engineering. One example is the re-identification of the freeway bus patterns from publicly open anonymized data on Taiwan electronic toll collection system [9].

In this article, we focus on estimating origin and destination locations from masked data. While direct analysis of masked data is possible, its analysis is less accurate and unreliable as the quality of data resolution is degraded for privacy protection [21]. Estimating origin and destination has the potential to be a great supplementary tool for extracting more accurate travel patterns from masked data.

Estimating origin and destination also has profound potential benefits. Estimation of trip origins and destinations can enhance traffic management and optimize vehicle energy use, facilitating the implementation of intelligent transportation systems [2], [1]. Another benefit is that ride-sharing can be facilitated by determining and grouping the nearby passengers with estimated origin and destination locations [6].

Past works mainly focusing on estimation of trip destination are based on either data-driven methods trained using historical trajectories [11], [12], [18], [8], or statistical methods using temporal-spatial features [15]. External information like driving patterns and driving behaviors are considered to capture route preferences better [11], [12], [8], [19]. Some studies regarding data-driven methods use Bayes Rule, Bayesian Inference, the EM algorithm, or particle filter/Bayesian filter to find the probability of a particular location being the destination based on the trajectories [11], [12] or historical travel patterns and velocities on edges [18]. Others utilize neural networks, including artificial neural networks and convolutional neural networks based on partial trajectories [3], [16] or correlations between locations [26].

To our knowledge, this study is one of the first to estimate both origin and destination locations from the masked trip data. The paper presents a probabilistic method to estimate destinations and origins using a Bayesian approach and external information on the congestion status of the network. The paper does not utilize past trip patterns as prior knowledge of states for estimating origin and destination locations through techniques like the Kalman filter.

While the work presented in the paper estimates the probable locations of the origin and destination, it does not find the exact location of the origin and destination due to the limitation in the details on the publicly available data. The estimation algorithm provides the ranking of origin candidates and destination candidates based on the probability ratio of different pairs of origin candidates and destination candidates. Thus, the presented algorithm preserves privacy while extracting probable origin and destination locations.

The paper is organized as follows. Section II presents the steps of extracting the potential origins, destinations, and paths of each trip. Section III presents the statistical process for the Bayesian Markov analysis of paths. Section IV describes how the rank aggregation method estimates the probable origin and destination locations. Section V demonstrates the case study of estimating the origin and destination of taxi trips using 2015 Porto taxi trip trajectory data from UC Irvine Machine Learning Repository [17].

*Yuneil Yeo is partially supported by the Dwight David Eisenhower Transportation Fellowship Program Under Grant No. 693JJ32445085.

[1]Yuneil Yeo, Chenming Niu, and Maria Laura Delle Monache are with the Department of Civil and Environmental Engineering, University of California, Berkeley, USA yuneily@berkeley.edu, chenming_niu@berkeley.edu, and mldellemonache@berkeley.edu

II. Data Extraction

In this section, we report how to identify origin candidates, destination candidates, and potential paths for each trip.

A. Creation of the Network: Graph Structure

The entire city network is represented through a weighted directed graph consisting of nodes and edges. The nodes of the graph are the intersections of the roads. Edges are the roads between intersections. The direction of the edges represents the direction of travel from one node to another. The weight of an edge is the physical length of a road. OpenStreetMap creates the weighted directed graph of the network by extracting the coordinates of road intersections and the length of roads. Mathematically, we can write a weighted directed graph as $G(V, E)$, where V represents the set of all nodes $v \in V$, and E represents the set of all weighted edges $e \in E$ with weights w_e. The features of nodes and edges on the weighted directed graph are updated with external information on the network congestion status like real-time traffic flows, velocities, or traffic densities.

B. Data Available

In publicly accessible masked trip data, there is information of a total of I trips. For each trip $i \in \mathcal{I} = \{1, ..., I\}$, we have reported trip distance δ_i, reported trip duration t_i, the exact time the trip started α_i, and the exact time the trip ended ζ_i. Each trip i has a corresponding reported origin region A_i^o and reported destination region A_i^d that are the predefined regions. The set $C_i = \{\delta_i, t_i, \alpha_i, \zeta_i, A_i^o, A_i^d\}$ represents the features of the particular trip i. The set of nodes belonging to A_i^o is defined as $\mathcal{O}_i = \{v \in A_i^o\}$, and the set of nodes in A_i^d is defined as $\mathcal{D}_i = \{v \in A_i^d\}$.

C. Extraction of Origin and Destination Candidates

To extract the trip origin and destination candidates nodes, we first find the actual travel distance of a particular trip $\tilde{\delta}_i$. For $\tilde{\delta}_i$, we need to consider the possible error of the measurement of δ_i by using the tolerance values for on-distance tests for a taximeter listed on [4]. The measured distance of a taximeter must not be over-measured by more than 1% and under-measured by more than 4%. Therefore, the actual trip distance is $0.99\delta_i = l_{\delta_i} \leq \tilde{\delta}_i \leq u_{\delta_i} = 1.04\delta_i$ where the lower bound and the upper bound of the range are l_{δ_i} and u_{δ_i} respectively. With l_{δ_i}, u_{δ_i}, $\mathcal{D}_i$, and $\mathcal{O}_i$, we define the set of origin candidates $O_i = \{v \in \mathcal{O}_i : \|v - w\| \leq u_{\delta_i} \text{ for } w \in \mathcal{D}_i\}$ and the set of destination candidates $D_i = \{v \in \mathcal{D}_i : \|v - w\| \leq u_{\delta_i} \text{ for } w \in \mathcal{O}_i\}$.

D. Extraction of Potential Paths

From D_i and O_i, we compute the set of all feasible paths $\mathcal{S}_i$ of each trip i. A path S_i^n is a sequence of nodes $(v_{a,i}^n)_{a=0}^{m_n}$ with m_n being the total number of nodes in S_i^n and a being the index of the node in S_i^n. S_i^n can also be represented in a sequence of edges $(e_a^n)_{a=1}^{m_n}$ with e_a being the edges connecting $v_{a-1,i}^n$ and $v_{a,i}^n$. The total distance traveled in the path S_i^n must be between l_{δ_i} and u_{δ_i}: $l_{\delta_i} \leq \Sigma_{a=1}^{m_n} w_{e_a^n} \leq u_{\delta_i}$. With this, we can define $\mathcal{S}_i = \{S_i^n : n \in N\}$ with N being the total number of feasible paths for trip i.

III. Bayesian Markov Analysis of Paths

This section describes the steps of the statistical methodology for finding the ratio between the probability of origin candidates and the probability of destination candidates based on the Bayesian approach and the Markov Chain rule. The transitional probability from one node to its adjacent node in the different paths S_i^n is also described.

A. Transitional Probability: Origin to Destination Direction

Given $G(V, E)$ and its adjacency matrix $\mathbf{B}$, the set W_v represents all adjacent nodes directly linked from node v. The transitional probability from node v to node $w \in W_v$ for a particular trip i is found in two steps.

1) The first step is the ”Search Area Algorithm.” The algorithm is based on the assumption that the drivers are more likely to drive in the direction that they want to reach. We define the ”search area” as $H_{v,i}$. If the node $v \notin \mathcal{D}_i$, $H_{v,i}$ is the convex hull of the set of nodes containing v and $\mathcal{D}_i$, Conv$(\{v\} \cup \mathcal{D}_i)$. Otherwise, $H_{v,i}$ is the area created by the line on the node v and the boundary of the destination region. The line, $y_{\perp,v,i}$ is perpendicular to the line from the centroid of A_i^o to the node v, $y_{v,i}$ as shown in Fig 1. The probability of moving to the adjacent node $w \in H_{v,i}$ is p' while the probability of moving to the adjacent node $w \notin H_{v,i}$ is $1 - p'$. Fig 1 shows the example of the ”search area”.

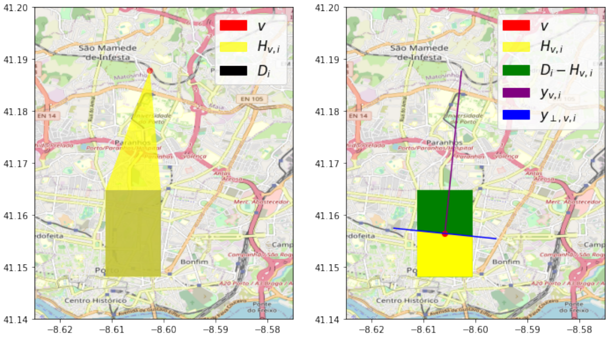


Fig. 1: Example of finding the search area when the node $v \notin \mathcal{D}_i$ (left) and $v \in \mathcal{D}_i$ (right).

2) The second step is based on the assumption that the drivers are more likely to drive to reach their destination as fast as possible. The second step starts by finding the shortest paths from the node v and its adjacent node w to every possible destination candidate. Then, we compute and compare the travel duration taken by these shortest paths from both node v and its adjacent node w to each destination candidate when moving from node v to node w. From the comparison, we can count $\beta_{v,w,i,-}$, the number of destination candidates of a particular trip i where a travel time from node w to destination is shorter than the travel time from node v to destination. Similarly, we can count $\beta_{v,w,i,+}$, the number of destination candidates of a particular trip i where a travel time from node w to destination is longer than the travel time from node v to destination. p is multiplied to the ratio of $\beta_{v,w,i,-}$ to the total number of destination candidates. In contrast, 1-p is multiplied to the ratio of $\beta_{v,w,i,+}$ to the total number of destination candidates.

The reason for having p and p' is to consider that not all drivers travel in the shortest or fastest paths. Factors like familiarity with routes or the number of turns make drivers not travel in the shortest or fastest paths [23], [25].

The transitional probability from the node v to the adjacent node $w \in W_v$ for a particular trip i is defined as below:

$$P(w|v \wedge C_i) = \begin{cases} p'[\frac{\beta_{v,w,i,-}}{|D_i|}p + \frac{\beta_{v,w,i,+}}{|D_i|}(1-p)], \\ \qquad \text{if } w \in W_v \cap H_{v,i} \\ (1-p')[\frac{\beta_{v,w,i,-}}{|D_i|}p + \frac{\beta_{v,w,i,+}}{|D_i|}(1-p)], \\ \qquad \text{if } w \in W_v \ \& \ w \notin W_v \cap H_{v,i}. \end{cases} \tag{1}$$

The transitional probability in (1) is the weighted sum of $\frac{\beta_{v,w,i,-}}{|D_i|}$ and $\frac{\beta_{v,w,i,+}}{|D_i|}$ by p and $1-p$ respectively. Then, it is scaled by p' or $1-p'$ based on whether the adjacent node is in the search area or not. $P(w|v \wedge C_i)$ is normalized so that the sum of all transitional probabilities from node v is 1.

B. Transitional Probability: Destination to Origin Direction

The transitional probability in the direction from destination to origin is computed similarly with some differences. First, the ”search area”, $H'_{v,i}$, is based on the direction from the destination to the origin. If the node $v \notin \mathcal{O}_i$, $H'_{v,i}$ is the convex hull of the set of nodes containing v and $\mathcal{O}_i$, Conv($\{v\} \cup \mathcal{O}_i$). If the node $v \in \mathcal{O}_i$, then $H_{v,i}$ is the area created by the line on the node v and the boundary of the origin region. The line, $y'_{\perp,v,i}$ is perpendicular to the line from the centroid of A_i^d to the node v, $y'_{v,i}$.

In addition, the transitional probability in the direction from destination to origin is based on $S_i^{n'}$, the sequence of nodes $(v^n_{m_n-a,i})_{a=0}^{m_n}$. Therefore, $S_i^{n'}$ is the reversed order of S_i^n. With this, we can define $\mathcal{S}'_i = \{S_i^{n'} : n \in N\}$.

The set of adjacent nodes in the direction from destination to origin is based on $\mathbf{B}'$, the transpose of adjacency matrix $\mathbf{B}$. Based on $\mathbf{B}'$, the set W'_v represents all adjacent nodes directly linked to a node v.

When moving from node v to the adjacent node $x \in W'_v$, we can count $\eta_{v,x,i,-}$, the number of origin candidates of a particular trip i where the travel time of the shortest path to node x from an origin is shorter than the travel time of the shortest path to node v from an origin. Similarly, we can count $\eta_{v,x,i,+}$, the number of origin candidates of a particular trip i where the travel time of the shortest path to node x from an origin is longer than the travel time of the shortest path to node v from an origin. Equation (2) computes the transitional probability to the adjacent node $x \in W'_v$ from the node v for a particular trip i in the direction from the destination to the origin.

$$P(x|v \wedge C_i) = \begin{cases} p'[\frac{\eta_{v,x,i,-}}{|O_i|}p + \frac{\eta_{v,x,i,+}}{|O_i|}(1-p)], \\ \qquad \text{if } x \in W'_v \cap H'_{v,i} \\ (1-p')[\frac{\eta_{v,x,i,-}}{|O_i|}p + \frac{\eta_{v,x,i,+}}{|O_i|}(1-p)], \\ \qquad \text{if } x \in W'_v \ \& \ x \notin W'_v \cap H'_{v,i}. \end{cases} \tag{2}$$

$P(x|v \wedge C_i)$ is normalized so that the sum of all transitional probabilities from node v is 1.

C. Probabilities of Sequences

To find the probability of a sequence $(v^n_{a,i})_{a=1}^{m_n-1}$ given an origin $v^n_{0,i}$, we use (3). Equation (3) is based on the chain rule of the probabilities and (1):

$$P((v^n_{a,i})_{a=1}^{m_n-1}|v^n_{0,i} \wedge C_i) = \prod_{a=0}^{m_n-2} P(v^n_{a+1,i}|v^n_{a,i} \wedge C_i) \tag{3}$$

In addition, we compute the probability of a destination $v^n_{m_n,i}$ considering the previous node sequence $(v^n_{a,i})_{a=1}^{m_n-1}$ based on the Markov Chain rule through (4). Markov Chain rule is satisfied in the study as a driver's decision to go to a particular node from the current node in trip i only depends on the current node given C_i:

$$P(v^n_{m_n,i}|(v^n_{a,i})_{a=1}^{m_n-1} \wedge C_i) = P(v^n_{m_n,i}|v^n_{m_n-1,i} \wedge C_i). \tag{4}$$

Similarly, (5) makes use of the chain rule and (2) to find the probability of $(v^n_{a,i})_{a=1}^{m_n-1}$ given a destination $v^n_{m_n,i}$:

$$P((v^n_{a,i})_{a=1}^{m_n-1}|v^n_{m_n,i} \wedge C_i) = \prod_{a=2}^{m_n} P(v^n_{a-1,i}|v^n_{a,i} \wedge C_i) \tag{5}$$

Markov Chain rule once again is used for (6) to find the probability of an origin given the rest of the node sequences $(v^n_{a,i})_{a=1}^{m_n-1}$:

$$P(v^n_{0,i}|(v^n_{a,i})_{a=1}^{m_n-1} \wedge C_i) = P(v^n_{0,i}|v^n_{1,i} \wedge C_i). \tag{6}$$

D. Ratio of Probability of Origin and Probability of Destination

The direct calculation of the probability of origin and destination is unattainable. Therefore, we utilize the Bayes theorem to estimate the probability of node sequence $(v^n_{a,i})_{a=1}^{m_n-1}$ given the condition C_i in two ways based on (3), (4), (5), and (6) as shown in (7) and (8):

$$\begin{aligned} &P((v^n_{a,i})_{a=1}^{m_n-1}|C_i) \\ &= \frac{P((v^n_{a,i})_{a=1}^{m_n-1}|v^n_{m_n,i} \wedge C_i)}{P(v^n_{m_n,i}|(v^n_{a,i})_{a=1}^{m_n-1} \wedge C_i)} \cdot P(v^n_{m_n,i}|C_i) \\ &= \frac{P((v^n_{a,i})_{a=1}^{m_n-1}|v^n_{m_n,i} \wedge C_i)}{P(v^n_{m_n,i}|v^n_{m_n-1,i} \wedge C_i)} \cdot P(v^n_{m_n,i}|C_i) \end{aligned} \tag{7}$$

$$\begin{aligned} &P((v^n_{a,i})_{a=1}^{m_n-1}|C_i) \\ &= \frac{P((v^n_{a,i})_{a=1}^{m_n-1}|v^n_{0,i} \wedge C_i)}{P(v^n_{0,i}|(v^n_{a,i})_{a=1}^{m_n-1} \wedge C_i)} \cdot P(v^n_{0,i}|C_i) \\ &= \frac{P((v^n_{a,i})_{a=1}^{m_n-1}|v^n_{0,i} \wedge C_i)}{P(v^n_{0,i}|v^n_{1,i} \wedge C_i)} \cdot P(v^n_{0,i}|C_i). \end{aligned} \tag{8}$$

As both (7) and (8) is solved for $P((v^n_{a,i})_{a=1}^{m_n-1}|C_i)$, we can set (7) and (8) equal. Then, we can solve for the ratio between probabilities of a particular origin $o_{i,j} \in O_i$ and the probability of a particular destination $d_{i,h} \in D_i$ as shown in (9). $j \in J$ = {1, ..., $|O_i|$} represents the index of a particular origin in O_i. Likewise, $h \in H$ = {1, ..., $|D_i|$} represents the index of a particular destination in D_i.

Equation (9) finds the ratio between the probability of $o_{i,j}$ and probability of $d_{i,h}$ with two considerations. First, (9) considers the possibility where there might be multiple paths starting from $o_{i,j}$ and ending at $d_{i,h}$ for a particular trip i. Also, (9) integrates the principle of mutual exclusivity, as a person can only drive one path for a trip:

$$\frac{P(o_{i,j}|C_i)}{P(d_{i,h}|C_i)} = \sum_{\substack{n \in N: \\ (v_{0,i}^n = o_{i,j}) \wedge (v_{m_n,i}^n = d_{i,h})}} \left(\frac{P(v_{0,i}^n|C_i)}{P(v_{m_n,i}^n|C_i)} \right)$$
$$= \sum_{\substack{n \in N: \\ (v_{0,i}^n = o_{i,j}) \wedge (v_{m_n,i}^n = d_{i,h})}} \frac{P((v_{a,i}^n)_{a=1}^{m_n-1}|v_{m_n,i}^n \wedge C_i) P(v_{0,i}^n|v_{1,i}^n \wedge C_i)}{P(v_{m_n,i}^n|v_{m_n-1,i}^n \wedge C_i) P((v_{a,i}^n)_{a=1}^{m_n-1}|v_{0,i}^n \wedge C_i)}. \tag{9}$$

IV. Rank Aggregation

Using the ratio of probabilities for all possible origin $o_{i,j}$ and destination $d_{i,h}$ pairs of a particular trip i from (9), we rank origins for each destination and the destinations for each origin. Then, the rank aggregation method finds the most probable location(s) of the origin and destination for a particular trip i based on the normalized overall rankings.

A. Ranking of Origins

We first compute $\mathbf{r_{O_i,d_{i,h}}}$, the ranking of the origins $o_{i,j} \in O_i$ for each destination candidate $d_{i,h}$ based on (9). $\jmath \in \mathcal{J}$ = $\{1,\ ...,\ |O_i|\}$ is the ranked index of a particular origin based on $\mathbf{r_{O_i,d_{i,h}}} = [o_{i,1}, o_{i,2}, ..., o_{i,\jmath}, ..., o_{i,|O_i|}]^T$ s.t. $\frac{P(o_{i,1}|C_i)}{P(d_{i,h}|C_i)} \geq \ldots \geq \frac{P(o_{i,|O_i|}|C_i)}{P(d_{i,h}|C_i)}$.

We repeat the process for every destination candidate. Afterward, matrix $\mathbf{R_{O_i}} = [\mathbf{r_{O_i,d_{i,1}}}, \mathbf{r_{O_i,d_{i,2}}}, \cdots, \mathbf{r_{O_i,|D_i|}}]$ is formed where its columns are the ranking of origins for a particular destination candidate $d_{i,h} \in D_i$. In addition, we construct the vector of importance weights $\omega_{\mathbf{D_i}}$ = $\{\frac{1}{|D_i|},\ ...,\ \frac{1}{|D_i|}\}$ with length of $|D_i|$. Each destination candidate has an equal weight, meaning that there is no preference for certain destinations.

Among the two rank aggregation methods presented in [5], we choose the fuzzy preference relation approach that uses $\mathbf{R_{O_i}}$ and $\omega_{\mathbf{D_i}}$ to find the overall ranking of origins, $\mathbf{r}^*_{O_i}$.

B. Ranking of Destinations

The ranking of destinations and the most probable destination are computed similarly. We compute $\mathbf{r_{D_i,o_{i,j}}}$, the ranking of the destinations $d_{i,\hbar} \in D_i$ for each origin candidate $o_{i,j}$ based on (9). $\hbar \in \mathcal{H}$ = $\{1,\ ...,\ |D_i|\}$ is the ranked index of a particular destination based on $\mathbf{r_{D_i,o_{i,j}}} = [d_{i,1}, d_{i,2}, ..., d_{i,\hbar}, ..., d_{i,|D_i|}]^T$ s.t. $\frac{P(o_{i,j}|C_i)}{P(d_{i,1}|C_i)} \leq \ldots \leq \frac{P(o_{i,j}|C_i)}{P(d_{i,|D_i|}|C_i)}$.

After finding $\mathbf{r_{D_i,o_{i,j}}}$ for every origin candidate, we create the matrix $\mathbf{R_{D_i}} = [\mathbf{r_{D_i,o_{i,1}}}, \mathbf{r_{D_i,o_{i,2}}}, \cdots, \mathbf{r_{D_i,o_{i,|O_i|}}}]$ where its columns are the ranking of destinations for a particular origin candidate. Similar to $\omega_{\mathbf{D_i}}$, we construct the vector of importance weights $\omega_{\mathbf{O_i}}$ = = $\{\frac{1}{|O_i|}, ..., \frac{1}{|O_i|}\}$, where each origin candidate has equal weight as there is no preference for certain origins. The fuzzy preference relation approach in [5] uses $\mathbf{R_{D_i}}$ and $\omega_{\mathbf{O_i}}$ to find the overall ranking of destination candidates, $\mathbf{r}^*_{D_i}$.

C. Most Probable Origin-Destination Pair

The set of most probable origin-destination pair(s) for the particular trip i is based $\mathbf{r}^*_{O_i}$, $\mathbf{r}^*_{D_i}$, and $\mathcal{S}_i$. From $\mathcal{S}_i$, the feasible origin-destination pairs of a trip i are found. Then, we find the overall ranking of the origin from $\mathbf{r}^*_{O_i}$ and the overall ranking of the destination from $\mathbf{r}^*_{D_i}$ for each origin-destination pair. The overall ranking of the origin and the destination of each S_i^n is normalized by dividing its rank by the total length of $\mathbf{r}^*_{O_i}$ and the total length of $\mathbf{r}^*_{D_i}$ respectively.

For each origin-destination pair candidate, we then compute the combined normalized ranking of the pair by summing the normalized ranking of the origin and destination of the pair. The origin-destination pair with the highest combined normalized rank is the most probable pair(s) of the particular trip i. Multiple origin-destination pairs can have the same combined normalized rank, so there may be multiple most probable origin-destination pairs. The most probable origin-destination pair(s) are the estimates of the origin(s) and destination(s) of a particular trip i.

V. Case Study: Porto, Portugal

To evaluate the performance of the origin and destination estimation method, the city of Porto in Portugal is used as a case study. The assessment is done by comparing the actual origin and destination locations of taxi trips in Porto to their estimated origin and destination locations. We assume p and p' to be 93.2% and 70%, respectively, based on [8] and [22]. The next sections describe the case study in detail.

A. Porto Data

Different Porto data sets are utilized. The information on districts of Porto is from [10]. From the UC Irvine Machine Learning Repository, the data on the trajectories of taxis operated in Porto are used [17]. In addition to these data sets, the graph G of Porto is created using OpenStreetMap. Fig 2 shows the graph of the network of Porto, Portugal.

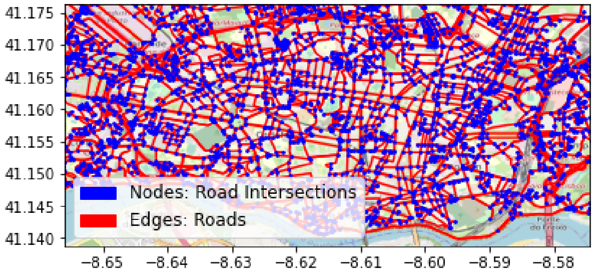


Fig. 2: The network of Porto, Portugal. There are 10,565 edges (red) and 5,029 nodes (blue) in Porto, Portugal.

1) Districts in Porto Data: Districts in Porto, Portugal are used as predefined regions for reported origin regions and reported destination regions where the pick-up and drop-off of taxi trips have happened. 18 districts in Porto, Portugal from [10] are shown in Fig 3.

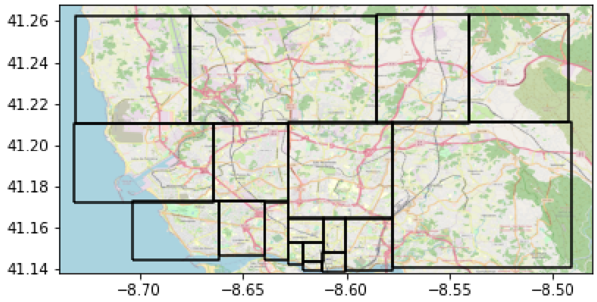


Fig. 3: Plot of Districts in Porto, Portugal

2) *Taxi Trip Data:* Among other publicly available taxi trip data, Porto 2015 taxi trip data from [17] contains actual coordinates of origin and destination in addition to coordinates in each trajectory. While the data set contains various features, only pick-up time, drop-off time, pick-up coordinates, and drop-off coordinates of each trip are used.

From the trajectories of taxi trips, the travel distance and travel time can be derived as GPS coordinates are recorded every 15 seconds. The trajectories of taxi trips are also utilized to estimate the mean velocity of links at a particular time period. For performance evaluation purposes, the case study utilizes the taxi trips that happened from 10:30AM to 10:45AM on multiple days.

VI. Results

This section presents the results of the origin and destination estimation. Fig 4 shows the location of estimation, centroid locations, and actual locations of a trip.

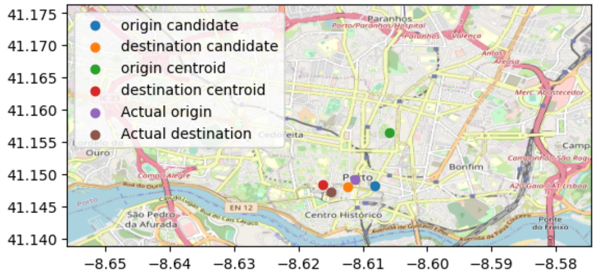


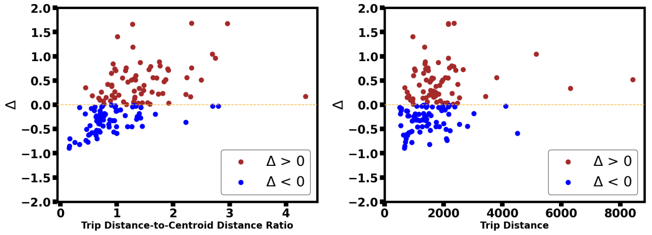
Fig. 4: A Map with 1. Estimated origin and destination, 2. Centroid location of origin area and destination area, and 3. actual origin and destination locations

The performance assessment of the algorithm is done through 1. mean average error (MAE), 2. plot of the percent difference between estimate error and centroid error, and 3. box plots. We first compute the mean average error to compare the distance from the estimated to the actual location, $\Delta_{actual,estimates}$, and the distance from the centroid of the reported region to the actual location, $\Delta_{actual,centroid}$. For the case study, $\Delta_{actual,estimates}$ is 497.66 m while $\Delta_{actual,centroid}$ is 566.08 m. The MAE of the estimation algorithm is 12.09% smaller than MAE between the centroid locations of the reported region to the actual location. Note that the estimated location is still within the reported region. From MAE, we can see an improvement.

Fig 5 compares the performance of the estimates to the centroid with respect to (a) the trip distance-to-centroid distance ratio and (b) trip distance. Centroid distance is defined as the haversine distance between the centroid of the pickup area and the centroid of the dropoff area. Since the centroid distance reflects the closeness of the pickup area and the dropoff area, the trip distance-to-centroid distance ratio shows the trip distance relative to the areas closeness. The percent difference between estimate error and centroid error is computed as follows:

$$\Delta = \frac{\Delta_{actual,estimates} - \Delta_{actual,centroid}}{\Delta_{actual,centroid}}. \qquad (10)$$

From (10), negative Δ shows that the estimate is closer to the actual location than the centroid as the distance between the estimate to the actual location is shorter than the distance between the centroid to the actual location for a trip.



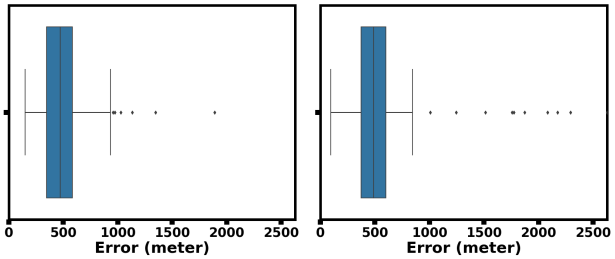
(a) Trip Distance-to-Centroid Distance Ratio

(b) Trip Distance

Fig. 5: Plots of Δ with respect to (a) Trip Distance-to-Centroid Distance Ratio and (b) Trip Distance. Brown Points represent trips with Δ greater than 0. Blue Points represent trips with Δ less than 0.

From Fig 5, the percent difference generally increases as the trip distance-to-centroid distance ratio increases. This is because the number of origin candidates and destination candidates is more likely to increase as the travel distance-to-centroid distance ratio increases, which indicates that considering more origin and destination candidates in the estimation process may likely reduce the estimation's accuracy.

Fig 6 shows the box plots of the errors for both cases to see the spread of the errors.



(a) Error Between Estimates and Actual

(b) Error Between Centroid and Actual

Fig. 6: Box Plots of Errors of (a) Estimates and (b) Centroids.

The first-quartile (Q1), median (Q2), and third-quartile (Q3) of the errors between the estimate and actual locations are 345.13 m, 469.96 m, and 584.54 m, respectively. Meanwhile, Q1, Q2, and Q3 of the errors between the centroid locations of the reported region and actual locations are 372.27 m, 489.56 m, and 598.87 m, respectively. Fig 6 additionally shows that the error between estimates and actual locations contains the outliers.

We additionally examine the effects of p and p' on the accuracy of the estimates.

			p	
		0.5	0.7	0.9
p'	0.5	46.17 m	235.70 m	235.70 m
	0.7	2083.97 m	131.58 m	1814.87 m
	0.9	2083.97 m	131.58 m	131.58 m

TABLE I: One Example of MAE with Different p and p' combinations of trip 3498

From Table I, the MAE between estimates and actual ranges from 46.17 m to 2083.97 m based on the choice of p and p'. The MAE changes with p and p' since it determines the transitional probability.

VII. Discussion/Future Works

While the estimates performed better than the centroids, the error between actual and estimates is still large. This can be due to several reasons. First, the speed of links is estimated either from the trajectories data or is approximated based on the nearby roads which introduces some error. Second, the choice of p and p' affects the performance of the algorithm as seen in Table I. Third, the larger the size of predefined regions, the more origin and destination candidates, which lead to larger errors, as seen in Fig 5.

Considering possible reasons, the future work is to 1) perform a detailed analysis of the errors with simulation in different road networks, 2) conduct the analysis on p and p' to choose the best p and p' for the estimation, 3) test the algorithm in a different area where both real-time traffic data with different traffic flow parameters and the actual trip data are available.

References

[1] Pietro Casabianca, Yu Zhang, Miguel Martínez-García, and Jiafu Wan. Vehicle destination prediction using bidirectional lstm with attention mechanism. *Sensors*, 21(24):8443, 2021.

[2] Rui Chen, Mingjian Chen, Wanli Li, and Naikun Guo. Predicting future locations of moving objects by recurrent mixture density network. *ISPRS International Journal of Geo-Information*, 9(2), 2020.

[3] Alexandre De Brébisson, Étienne Simon, Alex Auvolat, Pascal Vincent, and Yoshua Bengio. Artificial neural networks applied to taxi destination prediction. *arXiv preprint arXiv:1508.00021*, 2015.

[4] Division of Measurement Standards, California Department of Food and Agriculture. California code of regulations title 4, division 9, 2017. accessed 2023-07-03.

[5] E. Dopazo and M.L. Martínez-Céspedes. Rank aggregation methods dealing with incomplete information applied to smart cities. In *2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–7, 2015.

[6] Patrick Ebel, Ibrahim Emre Göl, Christoph Lingenfelder, and Andreas Vogelsang. Destination prediction based on partial trajectory data. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 1149–1155. IEEE, 2020.

[7] You Han, Achintya Gopal, Liwen Ouyang, and Aaron Key. Estimation of corporate greenhouse gas emissions via machine learning. *arXiv preprint arXiv:2109.04318*, 2021.

[8] Eric Horvitz and John Krumm. Some help on the way: Opportunistic routing under uncertainty. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, page 371–380, New York, NY, USA, 2012. Association for Computing Machinery.

[9] Hsieh-Hong Huang, Jian-Wei Lin, and Chia-Hsuan Lin. Data re-identification—a case of retrieving masked data from electronic toll collection. *Symmetry*, 11(4):550, 2019.

[10] Rami Ibrahim and M Omair Shafiq. Detecting taxi movements using random swap clustering and sequential pattern mining. *Journal of Big Data*, 6:1–26, 2019.

[11] John Krumm. Real time destination prediction based on efficient routes. 2006.

[12] John Krumm and Eric Horvitz. Predestination: Inferring destinations from partial trajectories. In Paul Dourish and Adrian Friday, editors, *UbiComp 2006: Ubiquitous Computing*, pages 243–260, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[13] Kamila Lis, Mateusz Koryciński, and Konrad A Ciecierski. Classification of masked image data. *Plos one*, 16(7):e0254181, 2021.

[14] Roderick JA Little et al. Statistical analysis of masked data. *JOURNAL OF OFFICIAL STATISTICS-STOCKHOLM-*, 9:407–407, 1993.

[15] Shudong Liu, Liaoyuan Zhang, and Xu Chen. Travel destination prediction based on origin-destination data. In Leonard Barolli, Aneta Poniszewska-Maranda, and Tomoya Enokido, editors, *Complex, Intelligent and Software Intensive Systems*, pages 315–325, Cham, 2021. Springer International Publishing.

[16] Jianming Lv, Qing Li, and Xintong Wang. T-conv: A convolutional neural network for multi-scale taxi trajectory prediction, 2017.

[17] Ferreira Michel Mendes-Moreira Joao L. L. Moreira-Matias, Luis and J. J. Taxi Service Trajectory - Prediction Challenge, ECML PKDD 2015. UCI Machine Learning Repository, 2015. DOI: https://doi.org/10.24432/C55W25.

[18] Donald J. Patterson, Lin Liao, Dieter Fox, and Henry Kautz. Inferring high-level behavior from low-level sensors. In Anind K. Dey, Albrecht Schmidt, and Joseph F. McCarthy, editors, *UbiComp 2003: Ubiquitous Computing*, pages 73–89, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[19] Alberto Rossi, Gianni Barlacchi, Monica Bianchini, and Bruno Lepri. Modelling taxi drivers' behaviour for the next destination prediction. *IEEE Transactions on Intelligent Transportation Systems*, 21(7):2980–2989, 2020.

[20] Ammar M Sarhan. Estimations of parameters in pareto reliability model in the presence of masked data. *Reliability Engineering & system safety*, 82(1):75–83, 2003.

[21] Dara E Seidl, Piotr Jankowski, and Ming-Hsiang Tsou. Privacy and spatial pattern preservation in masked gps trajectory data. *International Journal of Geographical Information Science*, 30(4):785–800, 2016.

[22] Aly M. Tawfik, Hesham A. Rakha, and Shadeequa D. Miller. An experimental exploration of route choice: Identifying drivers choices and choice patterns, and capturing network evolution. In *13th International IEEE Conference on Intelligent Transportation Systems*, pages 1005–1012, 2010.

[23] Mohan Venigalla, Xi Zhou, and Shanjiang Zhu. Psychology of route choice in familiar networks: Minimizing turns and embracing signals. *Journal of Urban Planning and Development*, 143(2):04016030, 2017.

[24] Jianfeng Yang, Jing Chen, and Xibin Wang. Em algorithm for estimating reliability of multi-release open source software based on general masked data. *IEEE Access*, 9:18890–18903, 2021.

[25] En Jian Yao, Long Pan, Yang Yang, and Yong Sheng Zhang. Taxi driver's route choice behavior analysis based on floating car data. *Applied Mechanics and Materials*, 361:2036–2039, 2013.

[26] Jing Zhao, Jiajie Xu, Rui Zhou, Pengpeng Zhao, Chengfei Liu, and Feng Zhu. On prediction of user destination by sub-trajectory understanding: A deep learning based approach. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 1413–1422, New York, NY, USA, 2018. Association for Computing Machinery.