Second-Order Destination Inference using Semi-Supervised Self-Training for Entry-Only Passenger Data

Rongye Shi
Department of Electrical and Computer Engineering (ECE)
Carnegie Mellon University
Pittsburgh, PA, USA
rongyeshi@cmu.edu

Peter Steenkiste
Computer Science Department
Department of ECE
Carnegie Mellon University
Pittsburgh, PA, USA
prs@cs.cmu.edu

Manuela Veloso
Machine Learning Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA, USA
mmv@cs.cmu.edu

ABSTRACT

Automated data collection in urban transportation systems produces a large volume of passenger data. However, quite a few of the data are still incomplete, limiting the insight into passenger mobility. The unavailability of destination information in entry-only passenger data is a very common issue. Traditional approaches for estimating passenger destinations rely on heuristics that can recover only some of the missing destinations. To deal with the remaining incomplete data, this paper, for the first time, proposes a second-order inference methodology to leverage semi-supervised self-training to infer the missing destinations. The methodology involves the design of a base learner to predict the missing destinations based on the statistics of a selected similarity-based "training set", and the design of a selection strategy to select new data with high prediction confidence to update the training set. To further improve the inference, we incorporate personal history priors to modify the base learner. We evaluate our designs using two data sources: a real-data inspired traffic-passenger behavior simulation in the city of Porto, Portugal, and the real bus Automated Fare Collection (AFC) data collected from the same city. The experimental results show that compared to baseline methods that do not use self-training, our approach significantly improves the inference performance and achieves notably high accuracies.

CCS CONCEPTS

•**Computing methodologies** → Machine learning → Learning settings → Semi-supervised learning settings; •**Mathematics of computing** → Probability and statistics → Probabilistic inference problems → Computing most probable explanation;

KEYWORDS

Semi-supervised learning, Self-training, Transport, Inference

ACM Reference format:

Rongye Shi, Peter Steenkiste, and Manuela Veloso. 2017. Second-Order Destination Inference using Semi-Supervised Self-Training for Entry-Only Passenger Data. In *Proceedings of the 4th IEEE/ACM International Conference on Big Data Computing, Applications and Technologies, Austin, TX, USA, December 2017 (BDCAT'17)*, 10 pages.
DOI: https://doi.org/10.1145/3148055.3148069

1 INTRODUCTION

The rapid growth in urban populations poses significant challenges to efficiently move city dwellers in a fast, seamless, and convenient manner. To address this challenge, the concept of *Smart City* has been proposed to use urban informatics and technology to improve the quality of urban service. Intelligent transportation system is an important aspect of Smart City. To build this system, city planners require a clear picture of passengers' mobility patterns as a basis for transportation planning. Research that estimates people mobility patterns, especially travel demand patterns, requires access to large-scale and multi-source passenger mobility data. The availability of such data is improving as over the past decades, automated data collection in public transportation systems has become increasingly popular in cities worldwide. Automated data collection infrastructures produce a large volume of data about passengers, providing insights into crowd size, passenger journey time, and spatiotemporal distribution of travel demand.

However, the data collected by those systems are often incomplete, limiting the estimation of the overall demand profile [1]. In particular, the unavailability of passenger destination information is very common in entry-only automated transaction systems. Entry-only systems are popular because they reduce the number of transaction devices that must be supported, and because of the convenience of not having to tap the smart card additionally when alighting. Unfortunately, the passenger transaction records collected by entry-only systems do not directly provide the information necessary for constructing the passenger Origin-Destination (O-D) matrices.

Passenger O-D matrices provide vital information for operation planning and service adjustment [2]. Generally, the O-D matrices of passenger journey can be obtained in three ways.

The first way is to make manual travel surveys which are infrequent, expensive, and prone to response bias [3]. The second way is to install entry-exit systems to all the mass transit vehicles in the city to accurately record boarding and alighting information of each passenger. As mentioned previously, for internal mass transit in a city, there is no simple solution to apply such systems due to hardware cost and passenger inconvenience. The third way is to estimate O-D pairs from either incomplete but direct data or indirect but relevant data, such as cell phone data [4] and parking sensing information data [5]. Due to the practical efficiency, the third path becomes a necessary intermediate step in most travel analysis [6].

Traditional O-D pair estimation methods rely on heuristic logic which is based on the proximity of possible destinations to the next origin. In the case of entry-only passenger data, two assumptions are widely used as the basis of estimation: the most likely destination of a trip is the origin of the next trip, and the most likely final daily destination is the first daily origin [7]. Grounded on the assumptions constrained by time, distance, and other validation rules, disaggregated estimation can apply to infer the destinations of passengers with multiple daily trips.

However, solely relying on heuristic logic is insufficient: a considerable amount of entry-only data remains incomplete, either because we don't have the subsequent trip or because the subsequent trip does not meet the validation rules. In general, the entry-only data with estimated destinations that meet the validation rules account for about 60% of total entry-only data. Here, we refer to the qualified estimated data from traditional methods as *reconstructed data*. Thanks to previous research, the reconstructed data cover the majority of passenger population, of which the destination information is reasonably accurate. We believe that there is a lot of useful information in the reconstructed data for us to look into the remaining data. Thus, it is feasible to make a step forward to conduct extended inference to the destination of the remaining data. We refer to such extended inference using reconstructed data as *second-order inference*.

We realize that the second-order inference of destinations is a semi-supervised learning classification problem. In the problem, reconstructed data and remaining data can be conceptualized as labeled data and unlabeled data, respectively. A classifier is trained to correctly classify each unlabeled data point to a class and label it accordingly. Different from supervised learning methods which need sufficient labeled examples, semi-supervised learning can make use of both labeled and unlabeled data to improve the classification performance.

Self-training is commonly used in semi-supervised learning problems. A self-training method uses its own predictor (termed as base learner) to assign labels to unlabeled data. Then, the newly-labeled data with high confidence are selected to be added to the labeled set for the next iteration. There are two key challenges when leveraging self-training to a real world classification problem. First, the performance of the self-training algorithm strongly depends on the selected newly-labeled data at each iteration [8]. The selection strategy is based on the base learner's confidence of the prediction. Thus, it is vital that the base learner is well designed such that the prediction confidence is correct. Second, in big data scenarios, training effort is a concern. The process of properly selecting a limited number of training data to reduce training effort without seriously reducing the accuracy is vital for the scalability of the approach to large scale dataset.

In this paper, we propose a methodology to leverage semi-supervised self-training to conduct second-order destination inference taking advantage of heuristics-based reconstructed data. The main idea is to apply the explicit destination information in the reconstructed data to infer the remaining missing destinations in an iterative and self-training manner.

In particular, to overcome the aforementioned challenges, we design a base learner to predict the destination based on the statistics of a selected similarity-based training set. The training data are selected according to their similarity to the inputted unlabeled point. After the prediction, we apply a selection strategy to select newly-labeled data with top-ranked prediction confidence to update the labeled set for each iteration of self-training. The proposed design of the base learner and the selection strategy leads to high-accuracy inference of passenger destinations. Besides, the built-in tunable parameters designed in the base learner make it possible to properly limit the training size and eventually reduce the training effort without seriously reducing the inference accuracy. Furthermore, we demonstrate that by incorporating valid priors, in particular, personal history priors to modify alighting distributions, the base learner will produce more reliable destination estimation to achieve higher accuracies.

The main contributions of the paper are as follows:

- To our best knowledge, it is the first attempt to leverage self-training paradigm to passenger destination inference for entry-only passenger data.
- We subtly design a base learner and a selection strategy of newly-labeled data for updating the training set to achieve high inference performance with reasonable training effort.
- We demonstrate that by properly incorporating personal history priors to modify the base learner, further inference improvement can be achieved.
- We evaluate our approach with two data sources: the results of a real-data inspired traffic-passenger behavior simulation in the city of Porto, Portugal, and the real bus AFC data collected from the same city. The experimental results support the validity of our approach. For the simulation data, the exogenous accuracy achieved 83% given 13% wrong labels in the training set. For the real data, the endogenous accuracies achieved 94% and 97% for inference without and with priors, respectively.

Additionally, we discuss the influence of 1) built-in parameters and 2) the wrong labels in the reconstructed data on the overall second-order inference accuracy. The experimental results show that our method is robust against wrong labels.

The rest of the paper is organized as follows. Section 2 discusses related works. Section 3 describes a traditional inference method, and Section 4 describes the detail designs of the self-training methods for the problem of second-order passenger destination inference. Section 5 introduces the data sources used in the experiments. In Section 6, the experiments and results are presented. Lastly, Section 7 concludes our work.

2 RELATED WORK

Over the past two decades, substantial research interest has been placed in the field of O-D matrix estimation from entry-only passenger data. Usually, traditional methodologies estimate the destinations of individual trips, and then aggregate the O-D pairs to construct estimated O-D matrices.

2.1 Destination Estimation using Assumptions

In most traditional methods, the primary assumptions for estimating passenger destinations are that the destinations are close to or exactly at the next recorded location. One well known preliminary work was conducted by Barry et al. [7]. They estimated destinations of entry-only AFC data from the New York City subway system through two assumptions: 1) the most likely destination of the trip is the origin of the next trip, and 2) the most likely final daily destination is the first daily origin. They validated their methodology and assumptions using travel diary surveys. Later on, many other researchers have developed variant procedures to deal with the missing destination information more robustly by taking distance limitation into account and integrating other data resources [9, 10, 11, 12, 13].

In general, the ways in which researchers validate their estimation are bifurcated into *exogenous* validation and *endogenous* validation [14]. The former relies on external datasets, such as ground-truth O-D trips and household surveys that are independent from the reconstructed data. Only a few studies are able to achieve this [1, 10, 12, 15]. As a workaround, endogenous validations are commonly applied to ensure the consistency within the reconstructed data. Most studies apply the endogenous to validate their work [9, 13, 14, 16, 17]. One comparative advantage of our work is that we are able to implement traffic-passenger joint simulation with the help of agent-based traffic simulation tools. Therefore, we can conduct direct exogenous validation that truly evaluates the performance of our approach. On the other hand, as to the real bus data, we validate our inference with labeled data selected from the reconstructed data. Compared to the distance or spatial endogenous validation used in previous work, this is a more reliable validation method.

2.2 Machine Learning for Entry-Only Data

The application of machine learning (ML) to the transportation field is increasing, especially when Automated Vehicle Location (AVL) and Automated Passenger Count (APC) technology became popularized. AVL data are GPS data that keep track of vehicle locations in real time. APC data record the passenger volume in vehicles accurately. The availability of those data enables plenty of machine learning approaches such as supervised learning to achieve significant success in the transportation field (e.g. bus arrival time prediction [18, 19] and vehicle trajectory prediction [20, 21]). In most scenarios, sufficient labeled dataset is a prerequisite.

However, seldom attempts have been made to generalize ML methods to the entry-only data, even this kind of data is usually large in volume and rich in temporal-spatial information. The setbacks are the expensive cost and human effort needed to obtain true destinations. To the best of our knowledge, the first implementation of deep learning for destination prediction using complete entry-exit AFC data was conducted by Jung et al. [22]. In their work, they trained deep artificial neural networks with large amount of true O-D trips provided by an entry-exit bus fare system in Seoul, Korea and predicted destinations given entry information and land-use characteristics. Though this work is a decent starting point, the unavailability of real destinations in other cities limits its feasibility and practicality.

To push forward research in filling the missing destinations of entry-only transactions from which the transportation planners can learn the demand patterns, our work, for the first time, proposes to leverage semi-supervised learning to conduct second-order destination inference. We look into the remaining data with the knowledge obtained from the reconstructed data and iteratively modify the corresponding probabilistic knowledge in a self-training manner. Our work is generic to other types of destination inference problems once fragments of O-D knowledge concerning human mobility are available.

3 FIRST-ORDER INFERENCE

The issue of focus in this paper is as follows: *given the entry-only passenger data of which the destinations are unknown, how can we infer the destinations?*

The process that infers destinations directly from the entry-only data is conceptualized as *first-order inference*. This section introduces a commonly used first-order inference. For general discussion, the entry-only passenger data have the following four attributes:

t	boarding time;
r	route code of the mass transit;
o	boarding stop;
h	passenger ID.

In this section (Section 3), the word “stop” means “stop code”. We also conceptualize:

d	alighting stop;
n^r	number of stops of route r.

First-order inference is primarily conducted based on passenger travels with multiple stages in a single day. This implies that the data are pre-grouped with respect to date before applying this process. Without special note, in this section, the data are assumed to be the same-day data. The objective for estimating the destinations of passenger travels is to determine the alighting stop of each travel stage:

$\hat{d}_{h,s}$	estimated alighting stop of the s-th stage of passenger h.

The first-order inference method applied in this paper is based on two key assumptions that are generalized from previous work [7, 11, 14]: 1) the most likely destination of a travel stage is the downstream route stop nearest to the origin of the next travel stage, and 2) the most likely destination of the last travel stage is the downstream route stop nearest to the origin of the first daily travel stage.

Let $dist(stop1, stop2)$ be the Euclidean distance between two stops. Define $A_{h,s} = \{stop_i^{r_{h,s}} | index(o_{h,s}) \leq i \leq n^{r_{h,s}}\}$ as the downstream alighting stop candidate set, where $r_{h,s}$ is the route code of the s-th stage of passenger h, $index(\cdot)$ is the stop index, and $stop_i^{r_{h,s}}$ is the i-th stop of route $r_{h,s}$. The inference method is formulated as:

$$\hat{d}_{h,s} = \underset{d_{h,s} \in A_{h,s}}{argmin}\{dist(o_{h,s+1}, d_{h,s})\}, 0 < s < m_h, \quad (1)$$

$$\hat{d}_{h,s} = \underset{d_{h,s} \in A_{h,s}}{argmin}\{dist(o_{h,1}, d_{h,s})\}, s = m_h, \quad (2)$$

s.t.

$$o_{h,s} \neq d_{h,s},$$
$$dist(o_{h,s+1}, d_{h,s}) < c, \quad 0 < s < m_h,$$
$$dist(o_{h,1}, d_{h,s}) < c, \quad s = m_h,$$

where, m_h is the number of travel stages of passenger h; $o_{h,s}$ and $d_{h,s}$ are the boarding stop and the alighting stop of the s-th stage of passenger h; c is the cut-off distance. The first constraint is to make sure that the origin and destination of a travel stage are two different locations. The second and third constraints are to set a reasonable walking distance threshold which the passenger can transit from one stop to another on foot

Three aspects about the introduced first-order inference need to be emphasized here. The first aspect relates to the process of the final travel stage. If the travel has more than two stages, it is arguably likely that the last journey stage was to reach a destination other than the daily origin. Simply assuming that the latter is true brings great risk of incorrect inference [14]. Therefore, we restricted the use of (2) to two-stage travels ($m_h = 2$), and the final destination of a multiple-stage travel is not assigned even if it meets the constraints. The second aspect is related to using travel history of passengers to infer the destinations of single-stage travels. According to the definition of first-order inference in this paper, we argue that this is an extended process based on reconstructed data, and strictly speaking, this is not a part of first-order inference. Instead, we included this process in second-order inference which is introduced in the next section. The third aspect is about further increasing the sophisticated nature of the method. It is possible to consider time (a combination of vehicle dwell time and passenger boarding time) and vehicle speed in addition to distance to determine the potential destination of a travel stage or to create a constraint according to the number of alighting passengers. But it would be complex and less reliable to deal with the noise caused by temporal traffic jams, traffic light variations, and other uncertain conditions based solely on the passenger data. Thus, we prefer proceeding with a primary method first and gradually increase the subtlety when extra relevant data (such as VGL data, etc.) are available.

In the output of the first-order inference process, the data successfully inferred are called reconstructed data, and the data still unable to be inferred are called remaining data.

4 SECOND-ORDER INFERENCE

To deal with the remaining data, we propose second-order inference to draw probabilistic conclusions about the destinations of remaining data in the presence of destination statistics obtained from the reconstructed data.

4.1 Semi-Supervised Setting

Before introducing technical details of our methodology, we start with converting the second-order inference problem into a semi-supervised learning setting. In this section (Section 4), we change the meaning of o and d to be the boarding stop index and alighting stop index, respectively. The word "stop" means "stop index" in this section. The meaning of other symbols in Section 3 remains unchanged.

In the parlance of data science, the set of reconstructed data is conceptualized as labeled set and the set of remaining data as unlabeled set. Each data point is a vector $x_k = (t_k, r_k, o_k, h_k)$. In our problem, a data point is also called a *transaction*. We have the labeled points $X_l = (x_1, x_2, \dots, x_l)$, of which the labels $Y_l = (d_1, d_2, \dots d_l)$ are provided, and the unlabeled points $X_u = (x_{l+1}, x_{l+2}, \dots, x_{l+u})$ of which the labels are unknown. Note that $d_k \in \{1,2, \dots, n^{r_k}\}$ is the alighting stop of the k-th transaction. We assume that both the labeled data and the unlabeled data are drawn from the same distribution. We can see that l is the number of labeled data, and u is the number of unlabeled data.

The objective of semi-supervised learning is to make use of both labeled data and unlabeled data to label the unlabeled data. In our scenario, we attempt to use this method to fill the missing destinations.

4.2 Baseline Second-Order Inference

The subsection introduces the baseline inference which uses a straightforward way to infer the destination of a transaction by counting on the boarding and alighting statistics in labeled set. That is, we selected proper labeled data to advise: given boarding time t, boarding route r, and boarding stop o, how the probabilities of the destinations are distributed in the rest of the route. Upon this direction, we should be careful about the fact that the downstream alighting distributions could be inhomogeneous and vary in time. However, instead of complete randomness, human trajectories show a high degree of temporal and spatial regularity [23]. Thus, according to the periodic human activity, it is reasonable to assume that the alighting distributions of a given boarding mode (i.e. a fixed route and boarding stop) is stable within a small period of time on a certain weekday. In that sense, we focused on inferring the destinations of transactions on workday basis and constructed O-D matrices for each hour. With the O-D matrices, we could draw the fine-grained alighting distributions.

Specifically, we screened out all labeled transactions of certain weekdays (e.g. Wednesdays) and reorganized them according to route. For each route, we counted the boarding-alighting record hourly and stored the counts in the O-D matrix of that hour. In this way, each route is associated to 24 matrices. For managerial purposes, we saved all the O-D matrices into an O-D matrix dictionary $\Lambda = \{\Lambda_{r,T}\}$ where r indices a route code and T indices a hour. For example, for the route r, in the hour of T, the corresponding O-D matrix $\Lambda_{r,T}$ is in the form of

$$\Lambda_{r,T} = \begin{pmatrix} 0 & \lambda_{1,2} & \lambda_{1,3} & & \lambda_{1,n^r} \\ 0 & 0 & \lambda_{2,3} & \cdots & \lambda_{2,n^r} \\ 0 & 0 & 0 & & \lambda_{3,n^r} \\ & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix}, (3)$$

where

$$\lambda_{i,j} = \Lambda_{r,T}^{(i,j)} = \sum_{k=1}^{l} \mathbf{1}(t_k \in T,\ r_k = r,\ o_k = i,\ d_k = j) \ (4)$$

indicates that in the labeled set, in the time period of T, there are $\lambda_{i,j}$ passengers boarding onto the route r at the stop i and leaving the vehicle at the stop j. Obviously, only upper-right entries of the matrix have non-zero values because under normal conditions, no one can go back to any upstream location or leave the vehicle at the same boarding stop after check-in.

Each row in $\Lambda_{r,T}$, after normalization, presents the alighting probability distribution. Thus, given a transaction with the trip starting at the stop o at time t, we constructed the distribution over alighting stop candidates d as follows:

$$p(d|t, r, o; t \in T) = \frac{\Lambda_{r,T}^{(o,d)}}{\sum_{j=1}^{n^r} \Lambda_{r,T}^{(o,j)}}. (5)$$

This is also called alighting empirical distribution, since it is a probability function of destination candidates given statistics in relevant labeled data. Then the inference of the destination can be fulfilled via:

$$\hat{d} = \underset{d}{argmax}\, p(d|t, r, o; t \in T)\ . \ (6)$$

With this framework, the baseline inference can be described as the following procedure:

- Given the route r and the boarding time $t \in T$, find out the O-D matrix $\Lambda_{r,T}$;
- Given the boarding stop o, pick out the o-th row;
- Normalize the row vector to obtain the alighting probability distribution;
- Infer the destination $\hat{d}$ using equation (6).

4.3 Second-Order Inference using Self-Training

Baseline inference is a rough method, and further designs are necessary to achieve higher performance. In the situation where there are considerable amount of unlabeled data, fully supervised training becomes infeasible. Many machine-learning researchers have found that unlabeled data, when used in conjunction with the labeled data, can produce notable improvement in learning accuracy [24]. We found that this property also applies to our passenger destination inference problem - by incorporating the information of unlabeled data to further extrapolate the alighting distribution, the classifier (6) can make more reliable inference. A semi-supervised self-training approach is proposed in this subsection.

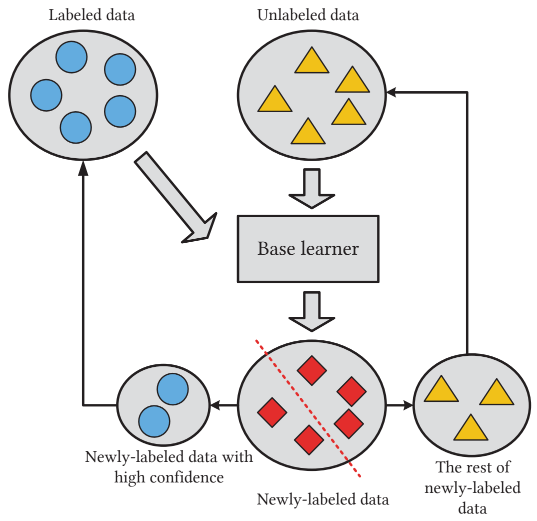
Labeled data
Unlabeled data
Base learner
Newly-labeled data with high confidence
Newly-labeled data
The rest of newly-labeled data

Figure 1: Diagram of semi-supervised self-training.

The framework of semi-supervised self-training is illustrated in Fig. 1. The main idea is that from labeled data, we learn a base learner. The base learner conducts predictions to assign labels to unlabeled data. From the set of newly-labeled data, a selection strategy is applied to select high-confidence predictions and add them to the original labeled set. The unlabeled set is replaced by the rest of newly-labeled data. This process iterates until all unlabeled data are labeled or the iteration exceeds a threshold number. For our passenger destination inference problem, the labeled data are initialized as the reconstructed data and the unlabeled data as the remaining data.

To implement a self-training that suits well to our passenger data, the base learner and selection strategy of newly-labeled data are designed as follows.

4.3.1 Base Learner. The performance of self-training strongly depends on the confidence of the prediction of the base learner. Therefore, to design a base learner that suits well to our passenger data is vital for high inference performance. Though fairly rough, the baseline inference is a good starting point that provides a nice framework upon which we can construct a base learner. The base learner considered in the paper consists of two parts:

- Extrapolate the alighting distribution given the boarding information;
- Draw the probabilistic conclusion about the destination given the extrapolated alighting distribution.

In general, the classifier (6) still applies here as it elects the optimal destinations given the statistics of similar observations. Eventually, the key factor that affects the inference quality lies on the quality of the extrapolation of alighting distributions. Different from the scheme in the baseline that simply relies on the statistics of the labeled data in a fixed hour, a more sophisticated scheme should consider: 1) what kind of labeled data should be selected, and 2) how many of them are needed or *enough* for a good extrapolation.

To this end, we propose to use a time window centered at the boarding time to ensure the involvement of most relevant labeled data for training. We set a tunable bandwidth parameter for time window to achieve flexibility in practice. To formulate this, we defined T^t as the period of time centered at the instant t with a given "bandwidth parameter" Δ (note the bandwidth is 2Δ):

$$T^t = \{\tau | t - \Delta \leq \tau \leq t + \Delta\}. (7)$$

On the other hand, we modified the O-D matrix dictionary $\Lambda = \{\Lambda_{r,T}\}$ by increasing the temporal granularity, e.g., we constructed a base O-D matrix every 6 minutes, and in this case, T (see subsection 4.2) is a predetermined 6-minute interval. Then for a transaction of route r, time t, the O-D matrix is

$$\Lambda_{r,t} = \sum_{\{T | T \cap T^t \neq \emptyset\}} \Lambda_{r,T} \, . (8)$$

This process accumulates the O-D pairs in the period of T^t to construct an O-D matrix given the boarding information. Based on (8), the corresponding alighting distribution is:

$$p(d|t,r,o) = \frac{\Lambda_{r,t}^{(o,d)}}{\sum_{j=1}^{n^r} \Lambda_{r,t}^{(o,j)}} . (9)$$

And thus, we applied the base prediction:

$$\hat{d} = \underset{d}{argmax}\, p(d|t,r,o) \; . \;\; (10)$$

Up until now, the base learner is thoroughly defined. It will also make sense to apply a deep artificial neural network (ANN) as a base learner. However, the training of ANNs is time-consuming. Concerning the algorithm complexity, we did not select ANNs.

4.3.2 Selection Strategy. As to the selection strategy of newly-labeled data, we took advantage of the base prediction in (10), with which a selection strategy of the newly-labeled data based on probability ranking can apply. Specifically, we sorted the newly-labeled data in descending order with respect to inference confidence $p(\hat{d}|t,r,o)$ and selected the first N data along with their labels to add to the original labeled set. The rest of newly-labeled data return and replace the original unlabeled set. The value N is called threshold of selection. The number of the iteration of the entire inference highly depends on N.

Note that the selection of labeled data for the training set in (8) is NOT a part of the selection strategy, but a part of the base learner. Here we need to clarify a point that is related to the trade-offs between training effort and performance. In many cases, the use of the training selection may eventually cause a decrease in inference accuracy. In general, the more relevant labeled data we use for training, the more reliably the base learner will perform. However, in the scenario of large data size, training effort is a critical concern. Li et al. pointed out the phenomenon when processing brain data: using a proper selection metric, which are designed for learning the classifier from a limited number of training data, the training effort could be significantly reduced while keeping competitive accuracies [25]. Similar in our implementation, the use of the training data selection in (8) can optimize the training efficiency of self-training and ultimately achieve maximal scalability and flexibility of our methodology to other practical big data problems. As can be seen later, setting proper time window bandwidth can eventually reduce the training effort without badly deteriorating the inference accuracy.

4.4 Second-Order Inference using Self-Training with Priors

Beyond the above design, we explored further improvement by using personal history priors. Trépanier et al. introduced the possibility to incorporate passenger daily and weekly travel patterns to complete the missing destinations [16]. This is a feasible insight, since passenger ID is usually available in most entry-only data. Given a passenger ID, the passenger's multiple daily travel stages can be retrieved, and corresponding destinations can be estimated through first-order inference. Our work extends the scope further: once the transactions of the same passenger are recorded continuously over a long time (e.g. several months), the passenger's travel patterns on daily, weekly, and even monthly basis can be estimated by looking into the personal history. As can be imaged, the personal level mobility pattern information can be formulated as priors. It was attempted that the prior could increase the reality level of the alighting distributions of passengers that showed up repetitively.

To flesh this out, we assumed that in the entry-only data, parts of the passenger IDs (attribute h) appear repetitively in different days. With this assumption, we constructed an O-D matrices for each passenger ID and saved them in a dictionary $\Omega = \{\Omega_{r,h}\}$, where the entry is

$$\Omega_{r,h}^{(i,j)} = \sum_{k=1}^{l} \mathbf{1}(\, r_k = r,\; o_k = i,\; d_k = j,\; h_k = h) \, . (11)$$

Each row in $\Omega_{r,h}$, after normalization, is the alighting prior of the person h. Thus, given a transaction with the trip starting at the stop o of route r, the prior of the person h (if the corresponding row has at least one non-zero value) is as follows:

$$q(d|r,o,h) = \frac{\Omega_{r,h}^{(o,d)}}{\sum_{j=1}^{n^r} \Omega_{r,h}^{(o,j)}} . (12)$$

Comparing the prior distribution (12) and the distribution (9), both are probability distributions of destination candidates from which we can draw a destination sample. To make a decision upon choosing a destination using both distributions, we consider a voting-based process. In the process, we have two independent voters. One of them has a voting distribution that follows the prior (12) and the other follows the distribution (9). Each time, each voter chooses a destination independently for the inputted unlabeled data point. Only when the two voters agree with each other, we make use of the chosen destination. Otherwise, the voting repeats until the two voters agree. Then, the modified alighting distribution is shown as follows (the probability of chosing destination d conditioned on agreement):

$$p'(d|t,r,o,h) = \frac{\Lambda_{r,t}^{(o,d)} \times \Omega_{r,h}^{(o,d)}}{\sum_{j=1}^{n^r} \left(\Lambda_{r,t}^{(o,j)} \times \Omega_{r,h}^{(o,j)} \right)} \, . (13)$$

In practice, the row vector $\Lambda_{r,t}^{(o,:)}$ and $\Omega_{r,h}^{(o,:)}$ could be orthogonal to each other (the two voters never agree), and it is invalid to conduct the operation of (13). This issue became nontrivial when we were tuning the bandwidth parameter Δ. A more robust alternative can be achieved using the cumulative distribution function (CDF). Let $P(d|t,r,o)$ and $Q(d|r,o,h)$ be the CDFs of (9) and (12). We propose the following modified alighting distribution:

$$\tilde{p}(d|t,r,o,h) = p \odot q = P(d|t,r,o) \times Q(d|r,o,h) - P(d-1\,|\,t,r,o) \times Q(d-1\,|\,r,o,h). \quad (14)$$

where $\tilde{p}(d=1|t,r,o,h) = P(1|t,r,o) \times Q(1|r,o,h)$. This is equivalent to forming a modified CDF through element-wise multiplication of P and Q. As can be seen in Fig. 2, the $p \odot q$ operation effectively modifies alighting distribution p using prior q. In particular, probabilities at the stops 3, 4, and 5 are properly balanced. In contrast, the calculation of the corresponding distribution is invalid using (13) due to orthogonality.

After modifying the alighting distributions with priors, the rest of the inference procedures are similar to those in subsection 4.3. The main structure of the second-order inference using self-training with priors is presented in Algorithm 1.

Algorithm 1 Outline of the Self-Training algorithm with prior

```
Input:   L, U, N, Iter; L: Labeled set; U: Unlabeled set;
         N: Threshold of selection; Iter: Max iteration number;
         Δ: Bandwidth parameter
Output:  L,U
1.  {$\Lambda_{r,T}$} ←ODMatrixDisctionary(L)
2.  {$\Omega_{r,h}$} ←PersonalODMatrixDisctionary(L)
3.  $n$←1
4.  While (U != empty) and ($n$< Iter) do
5.      S= {}
6.      $n$← $n$+1
7.      For each $x_i$ =( $t_i$, $r_i$, $o_i$, $h_i$) ∈ U do
8.          $p$ ← alightingDistribution({$\Lambda_{r,T}$}, $t_i$, $r_i$, $o_i$, Δ)
9.          $q$ ← personalPrior({$\Omega_{r,h}$}, $r_i$, $o_i$, $h_i$)
10.         If $q$ is valid do
11.             $\tilde{p}$← $p \odot q$
12.             ($d_i$, $Prob\_b_i$) ← Estimator($\tilde{p}$)
13.         Else do
14.             ($d_i$, $Prob\_b_i$) ← Estimator($p$)
15.         Endif
16.         S← S+ ($x_i$, $d_i$, $Prob\_b_i$)
17.     Endfor
18.     $S_{label}$←select $N$ highest $Prob\_b$ points{($x_i$, $d_i$)} from S
19.     $S_{rest}$ ← the remaining less confident points {$x_i$} in S
20.     L← L ∪ $S_{label}$
21.     U← $S_{rest}$
22.     {$\Lambda_{r,T}$} ←ODMatrixDisctionary(L)
23.     {$\Omega_{r,h}$} ←PersonalODMatrixDisctionary(L)
24. Endwhile
25. Return L,U
```

5 DATA AND PRE-PROCESSING

This section describes the data sources that are used for validating our self-training based methodology.

5.1 Passenger Simulation Data of Porto City

Agent-based traffic simulation is a conventional approach in studying urban transportation and passenger activities. Ground truth observations of the passenger flows on each public vehicle provide an independent data source that can be used to evaluate the performance of the inference methods. One commonly used open source traffic simulator is SUMO (Simulation of Urban

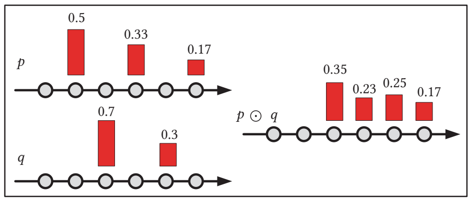

Figure 2: Modification of distribution p using proir q.

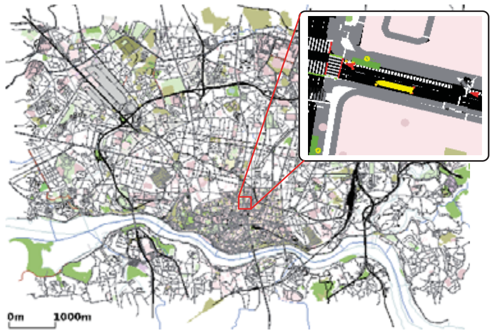

Figure 3: Simulation of Porto City in SUMO.

Mobility) - an agent-based microscopic traffic simulation package designed to handle large scale road traffic [26]. With this tool, it is convenient to simulate the behaviors of buses, trains, cars, and passengers at the city scale. To validate our research, we conducted city-wide bus passenger behavior simulation in SUMO for a case study of Porto, Portugal.

For traffic simulation, we applied SUMO interfaces to automatically extract and convert Porto road infrastructure information from public resources (OpenStreetMap, etc.) and imported virtual city traffic networks and demand. The main city bus operator STCP provides on its website detailed information about the bus routes, stops (along with geographical locations), and weekly schedule timetables. With the information, we established the entire bus transportation system within the area of interest as shown in Fig. 3. The established bus transportation network includes 136 bus routes, 855 bus stops and 5,723 buses running in a normal workday. Simulation tests have confirmed that the bus behaviors in SUMO match well to the real bus system, i.e., each bus departs at scheduled time, proceeds along the designated route, and pulls at designated stops accordingly.

For the bus passenger simulation, once the travel demand of the person is defined (travel time, travel origin, and destination), the platform will generate a travel plan and simulate the passenger's behaviors in terms of waiting at the bus stop, boarding to the planned route, alighting, and transiting to another bus until the passenger arrives at the final destination. The modeling and generation of traffic demand and passenger travel demand follow the spatial temporal distribution model derived and generalized from passenger trajectory data [27]. We focused on the model learned to reproduce average workday passenger travel demand over all Wednesdays in the year of 2013-2014. With this setting, we simulated 10 workdays with 32k passengers on each.

The bus passenger simulation records are included in the simulation output log. The raw data examples are presented in

Table 1. In order to test our inference approach, we extracted relevant passenger attributes: passenger ID, route, boarding time (in second), boarding stop, and alighting stop of each transaction. The total passenger transaction records in those 10 days are 635,111. We concealed the true destinations and applied our inference methodology to those data.

5.2 AFC Bus Passenger Transaction Data of Porto City

The second dataset is the set of transaction records in the whole three months of January, April and May 2010 in STCP buses in Porto City. The transaction data were collected by an entry-only AFC system called Andante. When a passenger taps a travel card on the fare reader, a transaction record will be produced. Each transaction record contains several attributes among which we are interested in: 1) travel card serial number; 2) transaction timestamp; 3) bus stop where the transaction took place; 4) route number; 5) route direction; and 6) vehicle trip starting time.

In data pre-processing, we fused the Andante AFC system data with the public data source including the list of 2,374 bus stops (with stop code, zone, and geographical locations), and the structure of 66 bus routes (with the bus stops and their sequence for each direction). The raw data have about 3% problematic samples, of which some important attributes such as the bus code, timestamp, or route number are either missing or illogical. After correctness, 1% of them remain unsolved, and we removed those records.

After pre-processing, we reorganized the data according to the need of our study. Specifically, the transaction records in all Wednesday of the three months were selected for validating our methodology. There are 12 Wednesdays totaling 2,422,079 transactions in the areas of interest. Those Wednesdays are normal weekdays and avoid any localized special holidays. Within each day, the transactions were grouped on passenger basis. The reorganized samples of a passenger multiple stage travel on May 19th are presented in Table 2.

Table 1: Simulation Data Examples

Passenger ID	Boarding Code	Alighting code	Route	Boarding time (s)	Alighting time (s)
119959	CSXS2	CVI2	602_rev	60132	60322
119959	NAT2	ACN1	204_rev	60502	60719
119959	AQL1	SPTO1	203	61430	61532

Table 2: Porto AFC Data Examples

Passenger ID: 20029975520					
Route	Boarding Code	Dirt.	Date	Vehicle starting time (s)	Boarding time (s)
801	ATSR2	2	19/5	30547	30977
801	CB1	1	19/5	31174	32849
801	ATSR1	1	19/5	44252	46210
801	SPC	2	19/5	49739	49793

6 EXPERIMENTS

We conducted experiments with both simulation data and real AFC data. The performances of different approaches were compared. This section details the experiment design and results.

6.1 Experimental Setup and Evaluation Metrics

6.1.1 Preparation of Reconstructed Set. We applied first-order inference described in Section 3 to obtain the reconstructed set and remaining set. To favor accuracy over the percentage of estimated destinations, we controlled the cut-off distance to a conservative level.

For the simulation data, the cut-off distance c was set to be 500 meters (m). Then, of all 635,111 transactions, 353,123 of them were inferable with the remaining 281,988 data points being placed in the remaining set. Thus, the first-order inference rate is 55.60%. Because we had ground-truth destinations for the reconstructed set, we could calculate the exogenous accuracy which is 86.23% (the number of correct labels divided by the size of reconstructed set).

For the AFC data, the cut-off distance c was set to be 600 m. First-order inference reconstructed about 61.58% of the total transactions, and 1,491,416 data were labeled. Because the true destinations were not available, endogenous validation was applied. Specifically, we selected out the labeled data of three Wednesdays (20/1, 14/4, 12/5) and used them as the test set (379,676 data points). The other part of labeled data (1,111,740 data points) became the reconstructed set.

6.1.2 Parameters of Interest. Four different types of second-order inference methods were implemented. The Random inference method simply assigned a destination from the downstream route to the unlabeled data point. The baseline method, self-training (ST) method, and self-training with priors (STP) method conducted inference in the way described in Section 4. The proposed ST/STP methods were designed with tunable parameters. We were particularly interested in studying the impact of time window bandwidth 2Δ and the threshold of selection N on the inference accuracy.

Since different remaining sets had different sizes, we needed a unified indicator to represent the threshold. We introduced a normalized selection threshold $N_k = N/u$, where u is the size of the remaining set. In practice, we used its reciprocal which is called the Selection Threshold Coefficient N_k^{-1} . N_k^{-1} is informative since it implies the approximate number of iteration times the algorithm executes to complete the self-training as well as the "granularity" of inference. The larger the N_k^{-1} is, the more fine-grained inference will be performed.

6.1.3 Evaluation Metrics. The performance of different inference methods were compared based on two evaluation metrics: the mean square error (MSE) and the inference count accuracy. The definition of MSE is: $MSE = \frac{\sum_{i=1}^{u}(d_i^{true} - \hat{d}_i)^2}{u}$, where d_i^{true} is the true stop index of test set, $\hat{d}_i$ is the inferred stop index. The inference count accuracy denotes the following value: $accuracy = \frac{\sum_{i=1}^{u} \mathbf{1}(d_i^{true} = \hat{d}_i)}{u}$. In addition, in this paper, the terms "inference accuracy" and "accuracy" all refer to this value. It is necessary to note that the accuracy for simulation data is the exogenous (ground-truth) accuracy since the true destinations are available. On the other hand, the accuracy of AFC data inference is the endogenous accuracy because the labels are provided by traditional methods.

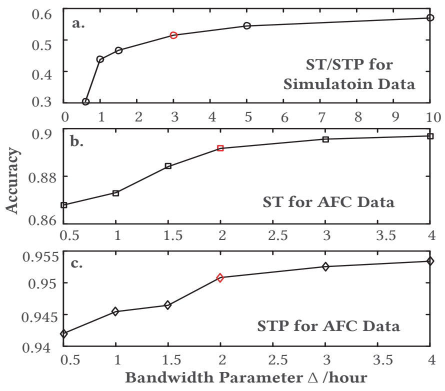

Figure 4: Impact of the bandwidth on accuracy (N_k^{-1}=10).

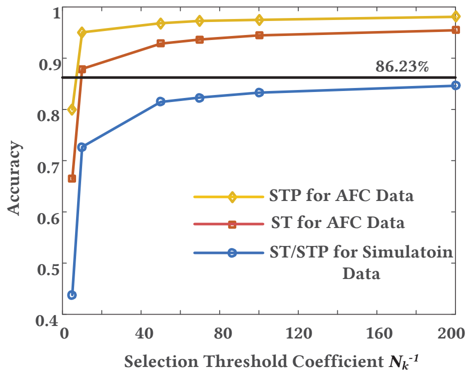

Figure 5: Impact of the selection threshold on accuracy (Δ=2 hours for AFC data, Δ=3 hours for simulation data).

Table 3: Second-Order Inference Performance of Different Methods

	Random		Baseline		ST		STP	
	MSE	Accuracy	MSE	Accuracy	MSE	Accuracy	MSE	Accuracy
Simulation Data	248.09	3.12%	25.89	17.76%	2.21	**83.27%**	2.21	**83.27%**
AFC Data	241.46	2.98%	63.27	26.15%	3.758	**94.46%**	1.964	**97.49%**

ST/STP parameters: N_k^{-1} = 100 for all; Δ=2 hours for AFC data, Δ=3 hours for simulation data.

6.2 Results of Parameter Sensitivity

The influence of the time window bandwidth and the selection threshold on accuracy is measured.

Fig. 4 illustrates that wider bandwidths bring improvements in accuracy given a fixed Selection Threshold Coefficient. Both ST and STP perform more reliable destination inference when more relevant labeled data are used for learning a base learner. A more interesting property is that there is an elbow region beyond which the accuracy is noticeably less sensitive to the bandwidth. This implies that by selecting a bandwidth in elbow region, we may reduce the training set without significantly deteriorating the accuracy. For instance, according to Fig. 4a, the accuracy at the bandwidth of 6 hours (Δ=3 hours) is reasonably close to that of 20 hours (Δ=10 hours), but the training effort of pro-cessing labeled data of 20 hours is way heavier than the former. In experiments, the algorithm running time of 10-hour Δ is reduced to less than 43% when using 3-hour Δ. The red markers are suggested elbow bandwidth points which provide reasonable trade-offs between accuracy and training efficiency.

Fig. 5 reports the performance of ST/STP on both datasets as the Selection Threshold Coefficient increases given certain bandwidths. The increase in the value of Selection Threshold Coefficient N_k^{-1} indicates that a smaller number of newly-labeled data are selected to update the labeled set. Thus, more iteration is executed before the inference completes. This is beneficial since with more iteration, the improvement of base learner is more fine-grained. Thus, we will have more experienced base learners to deal with very low confident unlabeled data. This figure also illustrates that the STP approach can effectively leverage prior information (personal travel information) to further improve the self-training performance. For AFC data, STP is general superior to ST under the same conditions, and it achieves higher accuracy. However, this improvement does not appear for simulation data. The reason is that the simulator generates and simulates passengers on a daily basis, and the passengers simulated in different days are completely independent. As a result, no personal travel information can be extracted from the simulation data. Fortunately, most real datasets contain records of the same passengers in different days, and the practical value of STP in real world is high.

One more discussion about Fig. 5 concerns the influence of wrong labels of reconstructed set on the overall accuracy. Since we have ground truth for simulation data, this issue can be studied. The benchmark line indicates the proportion of correct labels in the reconstructed set. This value implies the upper bound of ST/STP inference accuracy which is about 86.23%. By selecting proper parameters and conducting inference through more iteration, our self-learning method is shown to approach to that limit and achieve competitive accuracies (83.27%) effectively. This result demonstrates the robustness of the proposed self-training method against wrong labels in the reconstructed set.

6.3 Comparison between Methods

Table 3 provides the performance of different second-order inference methods. The baseline method diminishes the inference MSE distinctly compared to the random method. However, overall precise inference is lacking. This is because, in baseline method, the base learner makes inference only once based on the statistics of labeled set, and the conclusions drawn from those very uncertain alighting distributions usually lead to mistakes. In contrast, the ST method significantly reduces MSE and achieves the accuracies of about 83% for simulation data and 94 % for AFC data. Furthermore, the STP method takes advantage of prior knowledge to achieve a compelling accuracy of 97%. The reason for the improvements by using self-training is that by increasing labeled set with selective high-confident newly-labeled data, the original highly certain alighting distributions will be reinforced. The reinforcement in highly certain distributions will gradually influence and reshape the "surrounding" distributions that are originally very uncertain. As learning proceeds, the uncertain alighting distributions may become certain, and the chance we draw correct destination inference from those updated distributions improves.

Finally, the inference completion rate of ST/STP is 91.85% for simulation data and 99.96% for AFC data, respectively. Some data points are not inferred because in our implementation, we set an acceptance threshold, and the inference confidence must exceed this threshold to be accepted. This acceptance threshold can prevent some incorrect newly-labeled outlier data from being included into the labeled set.

7 CONCLUSIONS

In this paper, we proposed and implemented a semi-supervised self-training method to fulfil second-order inference. We showed that with proper design of base learner to extrapolate the alighting distributions from the reconstructed data in combination with priors, significant inference accuracy can be achieved. Furthermore, the training effort can be effectively reduced by tuning built-in parameters. We conducted experiments with two entry-only datasets. Experimental results demonstrated that our approach performs better than the compared methods and achieved very high inference accuracy.

The self-training inference method is extendable. The framework of self-training with priors can incorporate any kind of pre-knowledge including human mobility patterns, weather, road conditions, and so forth. Also, our work is generic to other types of destination inference problems once fragments of O-D knowledge concerning human mobility are available.

One limitation of our second order inference is that our method is based on the assumption that both the reconstructed data and the remaining data are from the same distribution. In reality, there might not be a perfect match. To deal with the mismatch, further priors need to be included to the algorithm to ensure good accuracy. This will be included in the future work.

ACKNOWLEDGMENTS

The authors would like to thank Teresa G. Dias, António A. Nunes, and João F. Cunha for providing the AFC data, and Shenghua Liu for helpful discussions. This work is supported by the FCT under the Carnegie Mellon-Portugal ERI S2MovingCities project. We also thank the other project members, including Susana Sargento and Ana Aguiar. The views and conclusions contained in this document are those of the authors only.

REFERENCES

[1] L. Moreira-Matias, and O. Cats. 2016. Toward a Demand Estimation Model Based on Automated Vehicle Location. *Transportation Research Record: Journal of the Transportation Research Board* 2544 (2016), 141-149.

[2] M. P. Pelletier, M. Trépanier, and C. Morency. 2011. Smart card data use in public transit: A literature review. *Transportation Research Part C: Emerging Technologies* 19, 4 (2011), 557-568.

[3] M. Yin, M. Sheehan, S. Feygin, JF. Paiement, and A. Pozdnoukhov. 2017. A generative model of urban activities from cellular Data. *IEEE Transactions on Intelligent Transportation Systems.* Online: DOI: 10.1109/TITS.2017.2695438.

[4] S. Isaacman, et al. 2012. Human mobility modeling at metropolitan scales. In *Proceedings of the 10th international conference on Mobile systems, applications, and services* (MobiSys'12). ACM, New York, NY, 239-252.

[5] X. Chen, et al. 2016. Parking Sensing and Information System: Sensors, Deployment, and Evaluation. *Transportation Research Record: Journal of the Transportation Research Board* 2559 (2016), 81-89.

[6] M. Carey, C. Hendrickson, and K. Siddharthan. 1981. A method for direct estimation of origin/destination trip matrices. *Transportation Science* 15, 1 (1981), 32-49.

[7] J. Barry, R. Newhouser, A. Rahbee, and S. Sayeda. 2002. Origin and destination estimation in New York City with automated fare system data. *Transportation Research Record: Journal of the Transportation Research Board* 1817 (2002), 183-187.

[8] J. Tanha, M. van Someren, and H. Afsarmanesh. 2017. Semi-supervised self-training for decision tree classifiers. *International Journal of Machine Learning and Cybernetics* 8, 1 (2017), 355-370.

[9] J. Zhao, A. Rahbee, and NH. Wilson. 2007. Estimating a Rail Passenger Trip Origin-Destination Matrix Using Automatic Data Collection Systems. *Computer-Aided Civil and Infrastructure Engineering* 22, 5 (2007), 376-387.

[10] J. Farzin. 2008. Constructing an automated bus origin-destination matrix using farecard and global positioning system data in Sao Paulo, Brazil. *Transportation Research Record: Journal of the Transportation Research Board* 2072 (2008), 30-37.

[11] J. Barry, R. Freimer, and H. Slavin. 2009. Use of entry-only automatic fare collection data to estimate linked transit trips in New York City. *Transportation Research Record: Journal of the Transportation Research Board* 2112 (2009), 53-61.

[12] W. Wang, JP. Attanucci, and NH. Wilson. 2011. Bus passenger origin-destination estimation and related analyses using automated data collection systems. *Journal of Public Transportation* 14, 4 (2011), 131-150.

[13] D. Li, Y. Lin, X. Zhao, H. Song, and N. Zou. 2011. Estimating a transit passenger trip origin-destination matrix using automatic fare collection system. In *Proceedings of the 16th International Conference on Database Systems for Advanced Applications* (DASFAA'11). Springer, Berlin, Heidelberg 502–513.

[14] AA. Nunes, TG. Dias, and JF. e Cunha. 2016. Passenger journey destination estimation from automated fare collection system data using spatial validation. *IEEE Transactions on Intelligent Transportation Systems* 17, 1 (2016), 133-142.

[15] M. Munizaga, F. Devillaine, C. Navarrete, and D. Silva. 2014. Validating travel behavior estimated from smartcard data. *Transportation Research Part C: Emerging Technologies* 44 (2014), 70-79.

[16] M. Trépanier, N. Tranchant, and R. Chapleau. 2007. Individual trip destination estimation in a transit smart card automated fare collection system. *Journal of Intelligent Transportation Systems* 11,1 (2007), 1-4.

[17] L. Zhang, S. Zhao, Y. Zhu, and Z. Zhu. 2007. Study on the method of constructing bus stops OD matrix based on IC card data. In *Proceedings of the 3rd International Conference on Wireless Communications, Networking and Mobile Computing* (WiCOM'07). IEEE, 3147-3150.

[18] R. Jeong, and R. Rilett. 2004. Bus arrival time prediction using artificial neural network model. In *Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems* (ITSC'04). IEEE, 988-993.

[19] M. Chen, X. Liu, J. Xia, and SI. Chien. 2004. A Dynamic Bus-Arrival Time Prediction Model Based on APC Data. *Computer-Aided Civil and Infrastructure Engineering* 19, 5 (2004), 364-376.

[20] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas. 2013. Predicting taxi–passenger demand using streaming data. *IEEE Transactions on Intelligent Transportation Systems* 14, 3 (2013), 1393-1402.

[21] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas. 2016. Time-evolving OD matrix estimation using high-speed GPS data streams. *Expert systems with Applications* 44 (2016), 275-288.

[22] J. Jung J, and K. Sohn. 2017. Deep-learning architecture to forecast destinations of bus passengers from entry-only smart-card data. *IET Intelligent Transport Systems* 11, 6 (2017), 334 - 339.

[23] M. C. Gonzalez, C. A. Hidalgo, and A. L. Barabási. 2008. Understanding individual human mobility patterns. *Nature* 453 (2008), 779-782.

[24] X. Zhu, "Semi-Supervised Learning Literature Survey. 2006. Technical Report 1530, Univ. of Wisconsin-Madison.

[25] Y. Li, C. Guan, H. Li, and Z. Chin. 2008. A self-training semi-supervised SVM algorithm and its application in an EEG-based brain computer interface speller system. *Pattern Recognition Letters* 29, 9 (2008), 1285-1294.

[26] R. Blokpoel, et al. 2016. SUMO 2016 - Traffic, Mobility, and Logistics. DLR.

[27] Trajectory - Prediction Challenge Dataset, ECML/PKDD 2015. Available: http://www.geolink.pt/ecmlpkdd2015-challenge/dataset.html