A two-stage stochastic programming approach for dynamic OD estimation using LBSN data

Qing-Long Lu[a], Moeid Qurashi[b] and Constantinos Antoniou[a]

[a]Chair of Transportation Systems Engineering, Technical University of Munich, Munich, Germany
[b]Chair of Transport Modeling and Simulation, Technical University of Dresden, Dresden, Germany

ARTICLE INFO

Keywords:
OD estimation
Demand estimation
Stochastic programming
LBSN data
Activity chain

ABSTRACT

Estimating origin-destination (OD) demand is essential for urban transport management and traffic control systems. With the ubiquity of smartphones, location based social networks (LBSN) data has emerged as a new rich data source with broad urban spatial and temporal coverage highly suitable for OD estimation. Its nature of confirmed trip purpose (activity) and activity chain host makes it more advantageous than other data (e.g., household travel surveys and traffic network detection). On the other hand, LBSN data is a more direct and accurate representation of demand patterns and can remove the significant burden of developing traffic models and estimating simulation-based objective functions. However, thus far, most LBSN-based estimation models only focus on static (day-level) OD estimation, making less use of those characteristics. To this end, this paper establishes a two-stage stochastic programming (TSSP) framework integrating the activity chains to model activity-level dynamic mobility flows using LBSN data. The first-stage model aims to minimize the errors introduced by the inter-zone OD flows alongside the expected errors of the check-in patterns. The second-stage model attempts to minimize the errors produced by the considered check-in pattern scenarios. Markov chain Monte Carlo (MCMC) sampling is used to generate plausible check-in scenarios. A generalized Benders decomposition (GBD) algorithm is presented to solve the two-stage stochastic programming model. We conduct the experiments on the case study of Tokyo, Japan, under the employment of the generalized least squares (GLS) estimator. The results show that algorithm convergence can be guaranteed within several iterations. The approach can provide satisfactory estimations of check-in patterns, zonal production and attraction, and OD flows. Furthermore, multiple objective function states are tested for evaluating the completeness of the proposed framework and exploring its potential for simplification and extension. Incorporating specific penalty terms into the objective function also provides a way to verify the reliability of the two-stage structure and validate the effectiveness of the model. Finally, we discuss the model enhancement from the perspectives of online OD estimation by integrating with LBSN simulations, network-wide OD extrapolation using appropriate scaling methods, and removing the user-side data requirement by leveraging activity chain modeling. The proposed framework provides a novel and effective approach to OD demand estimation using LBSN data.

1. Introduction

An accurate set of dynamic origin-destination (OD) matrices, as a typical representative of mobility demand patterns, is indispensable in urban transportation management and traffic control systems (Ren and Xie, 2017; Xiong et al., 2020). Integrating with a traffic assignment model, they can reproduce the traffic flow and network state in a detailed manner, helping to design and assess practical measures that improve the transport system efficiency (Ben-Akiva et al., 2001; Mahmassani, 2001; Tampere et al., 2010). Given their significance, dynamic OD demand estimation (DODE) has been a long-standing research topic with efforts in the last decade primarily focusing on high-dimensional demand models that pose issues of increasing complexity, indeterminateness, and computational efforts (e.g., Antoniou et al., 2015; Tympakianaki et al., 2015; Qurashi et al., 2019).

Existing OD estimation methods primarily rely on three travel data sources, i.e., traditional household surveys, traffic measurements, and positioning technology-based data (Yang et al., 2015). Among others, traditional household surveys are time-consuming, labor-intensive, and expensive, therefore are normally restrained within a limited area at low frequencies (e.g., once or twice a decade). These surveys, while providing detailed socio-demographical representation of demand, are only viable to develop planning models that depict average network conditions due to

qinglong.lu@tum.de (Q. Lu); moeid.qurashi@tu-dresden.de (M. Qurashi); c.antoniou@tum.de (C. Antoniou)
ORCID(s): 0000-0002-6087-8670 (Q. Lu); 0000-0002-0135-6450 (M. Qurashi); 0000-0003-0203-9542 (C. Antoniou)

their limited frequency. The second data type of traffic measurement relies on fixed detection infrastructure distributed over the network and has been widely used in DODE methods since it provides the required time-varying dynamics of network states for estimation. However, traffic-measurement-based DODE methods, on one hand, structurally suffer from the issue of indeterminateness in estimating realistic OD flows patterns (i.e., multiple sets of varying OD matrix patterns can satisfy the constraints imposed by the traffic measurements and optimize the system objective at the same time) (Cascetta et al., 2013; Antoniou et al., 2016; Qurashi et al., 2022) and on the other require significant computational resources to run dynamic traffic simulations that map estimated demand patterns on network models to attain and match traffic measurements iteratively. Note that both limitations especially increase as the network and OD matrix dimensions increase and therefore are the pivotal focus of recent literature efforts trying to scale DODE for high-dimensional demand models. Moreover, the high costs for installation and maintenance also restrict the number of detectors, so using the traffic information collected at those fixed locations for extrapolating the traffic states of the entire network and estimating the corresponding OD matrices can render biased results.

Methods using the third data source type have attracted much attention in recent years. The ubiquity of smartphones equipped with positioning technologies, such as GPS and Bluetooth, has resulted in the regular real-time generation of large sets of well-distributed data that also provides unprecedented opportunities for the implementation and application of OD estimation methods. One such suitable data type is location-based social networks (LBSN) data that has been used to develop demand models, due to its broad urban spatial and temporal coverage and confirmed trip purposes (Hu and Jin, 2017). LBSN services generate a large amount of anonymous check-in data of venues and users, making it a natural "host" of urban mobility patterns (Jin et al., 2014; Yang et al., 2015). More specifically, the check-in time series of venues record the travel destination distribution in both spatial and temporal dimensions, while the check-in history of users reflects the activity chains of individuals. However, thus far LBSN-based demand estimation approaches are limited to analyzing behavioral characteristics (e.g., Chaniotakis and Antoniou, 2015; Mahajan et al., 2021; Timokhin et al., 2020), estimating static (day-level) demands (e.g., Jin et al., 2014; Yang et al., 2015; Kheiri et al., 2015) and time-of-day dynamic zonal trip arrivals (e.g., Hu and Jin, 2017; Hu et al., 2019). Therefore, there exists a significant scope to explore the usage of LBSN data and develop dynamic OD estimators that can thoroughly utilize the trip purposes and activity chain information, specifically to better address the DODE scalability issues.

Most of the DODE literature that focuses to improve the estimation scalability aims to reduce either problem dimensions or problem non-linearity/complexity by constraining the algorithm search space based on additional model structural or correlation information. For example, a very prominent method is of using Principal Component Analysis (PCA), which extracts variance patterns from historical demand estimates and reduces problem dimensions, i.e., reduces the algorithm search space with the captured variance (Djukic et al., 2012; Qurashi et al., 2019, 2022). Other similar approaches include the use of a correlation assumption, i.e., quasi-dynamic (e.g., Cascetta et al., 2013; Cantelmo et al., 2014b), clustering of homogeneous model parameters (e.g., Tympakianaki et al., 2015), adding structural/correlation information among model parameters and traffic measurements via weight matrices of correlation between ODs and network (e.g., Cantelmo et al., 2014a; Antoniou et al., 2015). While another set of approaches uses response surface methods or (physical) metamodels, which approximate the input/output mapping relationship of demand and traffic measurements using differentiable analytical functions (e.g., Zhang et al., 2017; Osorio, 2019) to directly reduce the computational burden of running dynamic traffic simulations. However, note that all DODE approaches suffer from the requirement of developing and running dynamic traffic simulations due to the use of traffic measurement data and almost none of them can directly generate trip purposes and activity chain information. Whilst, utilizing the LBSN data allows replacing the traffic measurements with the check-in data, which on one hand is a more direct and accurate representation of the demand patterns and on the other removes the significant burden of developing traffic models and estimating simulation-based objective functions.

To tap the potential of using LBSN data for dynamic demand estimation, this paper establishes a two-stage stochastic programming (TSSP) framework integrating the activity chains to model activity-level mobility flows using LBSN data. We assume that similar check-in patterns are generated by the same OD flow pattern, and those check-in patterns are treated as scenarios in the stochastic framework. The first stage minimizes the errors introduced by the inter-zone flows alongside the expected errors of the check-in patterns. The second stage is to minimize the errors produced by each check-in pattern scenario separately. Note that, a scenario is defined as a realization of the second-stage problem state, i.e., the check-in patterns, and the set of realized scenarios provide the required constraints or demand structural information to the first-stage DODE problem formulation. Finally, the proposed two-stage stochastic programming model is addressed by the generalized Benders decomposition (GBD) algorithm. The idea is to construct a master problem and a series of subproblems (one per scenario) with respect to the first-stage and second-stage decision vari-

ables, respectively. These problems are then solved alternately and iteratively until the optimum is found. It is worth mentioning that, stochastic programming has been applied to the OD reconstruction problem based on traffic counts (Jeong and Park, 2021). However, to the best of our knowledge, this is the first effort to apply it to model the dynamic OD estimation problem based on LBSN data. While this approach is used to estimate the OD demand for activities checked-in in LBSN data, the full OD matrix can be extrapolated accordingly by incorporating an appropriate scaling method.

In the remainder of this paper, we review related literature in Section 2. Then, we briefly introduce the LBSN check-in data in Section 3. In Section 4, the mathematical model of the OD estimator based on LBSN data is constructed, followed by the solution algorithms. Later on, case studies are elaborated and model performance is evaluated. In Section 7, potential model enhancements are discussed from multiple aspects. Finally, we draw some conclusions and suggest future directions for research.

2. Related literature

In this section, we summarize the state-of-the-art of using LBSN data for OD pattern estimation and the endeavors that focus on improving the scalability of traffic-measurement-based OD estimators.

2.1. OD estimation based on LBSN data

In contemporary times, social networking sites have adopted the practice of managing access to their data backend through the development of application programming interfaces (APIs). Data retrieval via APIs has emerged as the primary approach for developers and researchers to collect LBSN data (Martí et al., 2019; Liu et al., 2022). Notable examples of LBSN data that can be obtained using APIs or web crawlers include geotagged tweets[1], Foursquare check-ins[2], Instagram posts[3], and Google popularity times[4]. Interested readers can refer to the developer documentation provided by the respective service operators for more details on API usage. Considering the existence of various popular social networking sites worldwide, the collection of substantial amounts of LBSN data for urban analysis is feasible. Nonetheless, the lack of consistency and representativeness has been acknowledged as a common limitation impeding the application of LBSN data (Tasse and Hong, 2014; Hu and Jin, 2017). For example, lower-income urban areas may generate fewer LBSN data due to lower smartphone ownership (Tasse and Hong, 2014). However, some argue that LBSN data can still provide a representative sample of citizen activities attributed to the increasing diversity of user profiles (Pew Research Center, 2023). Moreover, some methods have been proposed to address this problem, such as aggregating different hours of the day for model calibration with the consideration that they share similar sampling bias characteristics (Hu and Jin, 2017). Given the distinct characteristics of different social networking sites, Martí et al. (2019) have also suggested specialized methods to enhance the process of collecting more reliable LBSN data. These methods encompass data retrieval, verification, selection, and filtering, and particularly focus on platforms like Foursquare, Twitter, Google Places, Instagram, and Airbnb.

Leveraging the benefits of high resolution in spatial and temporal dimensions at a low cost, LBSN check-in data has emerged as a reliable secondary data source for travel demand estimation (Hu and Jin, 2017; Silva et al., 2019). The unique feature of self-contained confirmed trip purposes (activities) has given LBSN an unparalleled advantage in OD estimation. LBSN data, a kind of geotagged social media (GTSM) data, is capable of estimating distance and duration distributions, OD matrices, and individual activity-based mobility patterns (Zhou et al., 2018). Furthermore, it is also a supplementary and reliable data source for analyzing urban activities and behavior (Rizwan et al., 2020). Consequently, some LBSN data-based approaches have been proposed to understand and estimate OD demand within urban areas.

Previous studies have shown the feasibility of using LBSN check-in data for demand pattern estimation. For instance, Hu and Jin (2015) applies a simple time-dependent model to estimate trip attraction and validates the feasibility of using LBSN check-in data in demand pattern estimation. Due to the heterogeneity of venue categories in demand patterns, they argued that trip arrival patterns of different venue categories should be considered separately. This treatment is also adopted in our methodology presented in Section 4. Similarly, Yang et al. (2014b) and Yang et al. (2015) apply an integrated model that combines a hierarchical clustering method for venue categorization and a regression

[1] https://developer.twitter.com/en/docs/twitter-api (Accessed on 20.07.2023).
[2] https://foursquare.com/developers (Accessed on 20.07.2023).
[3] https://developers.facebook.com/docs/instagram-api/ (Accessed on 20.07.2023).
[4] https://www.google.com/maps/ (Accessed on 20.07.2023).

model to estimate trip production and attraction. Trips are then distributed according to a singly constrained gravity model. However, these models require calibration with the ground truth OD data provided by the local urban planning agency. Cebelak (2013) and Jin et al. (2014) further improve this approach by replacing the singly constrained gravity model with a doubly constrained one. The experiment results show that the modified method can significantly reduce the OD estimation error caused by the sampling bias imposed by the singly constrained model. Additionally, other conventional trip distribution models for urban travel demand analysis, such as the radiation model, rank-based model, and population-weighted opportunities model, with inputs extracted from LBSN data, have also been tested and compared in Kheiri et al. (2015). However, these models can only provide a static (day-level) solution to the OD estimation problem.

Inspired by the promising performance of the Hawkes process in self-reinforcing behavior modeling in Cho et al. (2014), Hu and Jin (2017) presents a time-of-day zonal arrival estimation model by integrating the Hawkes process and an LBSN check-in observation model into a state-space modeling framework. Such an approach can reduce the sampling bias in OD estimation caused by the difference between social behaviors and real travel patterns. In addition, Hu et al. (2019) incorporates a Latent Dirichlet Allocation (LDA) model for profiling zonal functionality based on the venue categorical distribution and a Pearson Product-Moment Correlation (PPMC) analysis method for measuring pairwise zone correlation. These two methods can help improve the performance of the zonal trip arrival and OD estimation models in the previous model, which is crucial for the performance of LBSN-data-based OD estimators (Jin et al., 2014).

However, none of the existing literature has fully utilized the trip purpose and activity chain information contained in LBSN data. Moreover, most of the aforementioned models were constructed based on conventional gravity models, which require additional information about the population and network. Additionally, these models overlooked the uncertainty factor of user social behaviors.

2.2. Traffic-measurement-based estimators

State-of-the-art traffic measurement systems, such as loop detectors, capture the effect of the mobility demand on the network and not the demand itself (Frederix et al., 2011; Shafiei et al., 2017). Therefore, traffic-measurement-based dynamic OD estimators structurally suffer from the issue of indeterminateness in estimating realistic OD flow patterns and estimate rather the fluctuations with respect to an existing prior OD estimate (from survey/planning models (McNally, 2007)) using time-dependent traffic measurements, instead of the whole demand itself. Furthermore, such methods require mapping the OD estimates into a comparable set of traffic measurements to formulate a traffic-measurement-based objective function. Therefore, an evident distinction in such models is the use of an assignment matrix, where assignment–matrix based algorithms explicitly use an analytical representation of the relationship between demand and traffic counts for DODE (Cascetta and Postorino, 2001; Toledo and Kolechkina, 2012), which, however being computationally inexpensive, is usually assumed to be linear and is not the case in reality, especially for large-scale and complex networks. Thus, recent years saw a shift towards assignment matrix-free methods which allow more accurate modeling of the supply and demand relationship using dynamic traffic assignments (DTA) and can also incorporate other data sources which can not be analytically related to OD flows (e.g. Bluetooth data). However, even the popular assignment matrix-free algorithms, e.g., Simultaneous Perturbation Stochastic Approximation (SPSA) (Balakrishna et al., 2007), fail to estimate large-scale problems, since the gradient approximation gets highly sensitive against increasing problem non-linearity, while the computational requirements (i.e., the number of iterations and running DTA iteratively) also increase exponentially.

To address the scalability issues of DODE, recent literature moved towards either reducing the OD dimensions or reducing problem non-linearity (adding structural/correlation information in the objective function). The most prominent scalability approach is the use of Principal Component Analysis (PCA) (Djukic et al., 2012; Qurashi et al., 2019, 2022) that directly reduces OD dimensions and non-linearity transforming OD matrix into lower dimensional orthogonal Principal Components using the extracted variance in historical OD estimates. Although powerful and intuitive, PCA-based methods, strongly rely on the presence and quality of historical estimates to extrapolate estimation patterns; therefore their performance is subjective to it. Next comes the use of quasi-dynamic methods that use a correlation assumption to reduce the DODE estimation variables (Cascetta et al., 2013; Cantelmo et al., 2014b; Bauer et al., 2017). At first, Cascetta et al. (2013) proposed the assumption of keeping OD shares constant with varying generation profiles of each origin, significantly reducing the estimation variables. While later to eliminate the requirement for a historical OD matrix in the original framework, Bauer et al. (2017) combined a GLS estimator for link flows and a maximum entropy term for the non-observable traffic distribution across paths in the modified framework, assuming instead of

constant OD shares, constant path choice proportions over time-of-day intervals for days with similar mobility patterns. Besides them, two other prominent SPSA scalability approaches also exist that explicitly focus to reduce problem non-linearity via either using weight matrices to add ODs flows and network correlation information in the DODE objective function (Antoniou et al., 2015) or clustering homogeneous model parameter (Tympakianaki et al., 2015).

Although all the above-mentioned efforts do significantly improve DODE scalability, they still fundamentally require setting up dynamic traffic models alongside the estimation algorithm setup and running them iteratively (varying by their convergence efficiency). Furthermore, almost all such conventional DODE approaches apart from Cantelmo et al. (2020) do not consider modeling activity chains and trip purpose information during estimation. However, OD flows are an aggregated representation of individuals' activity-travel chains. Such detailed demand modeling is generally modeled through activity-based demand models (ABM) (Bowman and Ben-Akiva, 2001; Viegas de Lima et al., 2018). ABM demand estimation is much more challenging especially in large-scale applications (Flötteröd et al., 2011) due to an exponential increase in modeling data size requiring calibration (since travel behavior is modeled at each individual or household level).

3. LBSN check-in data description

This section provides a brief introduction to the characteristics of LBSN check-in data and the procedure to prepare model inputs for OD estimation using these data.

An LBSN check-in event is automatically recorded when a user posts with geo-location information or visits a venue (a point-of-interest). Each check-in is described by a user ID, a venue ID, and the time of the check-in. Table 1 lists the main fields of a check-in. In this regard, we can treat venues as detectors of check-in events, and users are the objects being detected. Yang et al. (2015) reported that venues and users participate in such services actively as venues can interact with customers in a creative and convenient manner and customers can get awarded (e.g., discounts or "badges") from the social networking sites. As a result, compared to conventional household surveys, LBSN check-in data can be collected with a much higher frequency at a very low cost, and compared to traffic flow measurements, detectors of check-in events (i.e., venues) are "deployed" much denser within the urban area. The high resolution in both spatial and temporal dimensions has made LBSN data a reliable data alternative to model and estimate OD patterns within urban areas.

Table 1
Main fields of check-in events.

Field	Description	Example
Time	Time of the check-in	Apr 03, 2013, 18:17:18
User ID	Unique anonymized ID of the user	4bf58dd8d4898
Venue ID	Unique anonymized ID of the venue	4f0fd5a8e4b
Latitude	Latitude of the venue	35.7051
Longitude	Longitude of the venue	139.6196
Venue category	Category of the venue	Restaurant

By combining with the pre-registered location and category information of venues, check-in data has become a carrier of activity-oriented urban mobility patterns. Such data can be used to model and understand urban travel demand by aggregating the check-ins at venues based on specific categorization methods. Figure 1 depicts the data processing and information extraction procedure for preparing necessary inputs to the proposed OD estimator. Normally, venue-side data and user-side data are distinguished in the site server (Yang et al., 2015). Venue-side data contains the check-in statistics with respect to venues, while user-side data preserves the check-in history of users. As shown in the figure, venue-side data are obtained by aggregating the user-side data at the venue level. It has been recognized that one can aggregate venue-side check-in data based on the categorical hierarchy adopted by the site to model the activity-based mobility flows (Hu et al., 2019). For example, in a three-level categorical hierarchy, Chinese restaurants and buffets belong to the venue type "Restaurant" which is a sub-category node of the root category "Food". Each root category can be viewed as one certain type of activity. The categorical hierarchy thus provides an intuitive way to group venues into activities. While different social networking sites may use different categorization methods, most of them have similar root categories. By comparing the patterns of different activities at different locations, one is able to analyze mobility patterns. On the other hand, the activity share matrix at different times of the day can

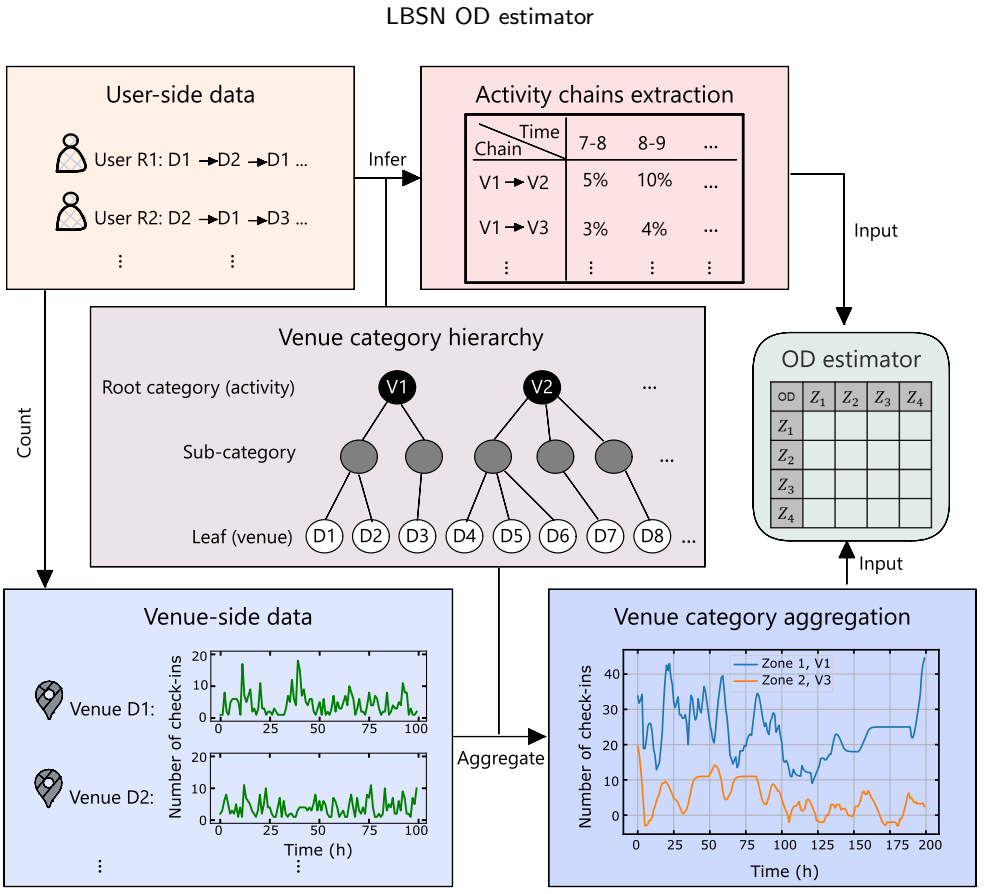


Figure 1: Process of extracting relevant information from LBSN check-in data for OD estimation.

be estimated by aggregating the activity-level movements deduced from the activity chains extracted from user-side data. While activity check-ins reflect the respective productions and attractions, activity share matrices can provide information for estimating the distributions for those productions and attractions. Therefore, one can combine the information contained in LBSN data to estimate and understand the mobility patterns within urban areas. Inspired by this basic idea, in the next section, we develop a mathematical model for OD estimation using LBSN check-in data which integrates the aggregated check-in patterns of venue-side data and the activity chains extracted from user-side data. Noteworthy, the activity chain information is included in the model formulation by specifying activity share matrices in relevant constraints, which can also further reduce the risk of privacy exposure for individuals.

4. Methodology

In this section, we first introduce the generic formulation of the conventional traffic-measurement-based OD estimation problem. Then, we explain the important concepts relevant to LBSN-data-based OD estimation and define the graph model of traffic analysis zones (TAZs). We then develop the mathematical model for the LBSN-data-based OD estimation problem leveraging these concepts and the graph model. Finally, the solution algorithms are provided.

4.1. Generic formulation of traffic-measurement-based OD estimators

By convention, a dynamic OD estimator requires as inputs a prior OD flow estimate, traffic measurements, and a traffic assignment method that maps OD flows to measurements. It is usually expressed as

$$\mathbf{x}^* = \underset{\mathbf{x}\in S_x}{\operatorname{argmin}} \left\{ f_x\left(\mathbf{x}, \mathbf{x}^{(p)}\right) + f_y\left(\hat{Y}, Y\right) \middle| Y = \mathcal{A}(\mathbf{x}) \right\} \tag{1}$$

where S_x is the feasible space of OD flows, $\hat{Y}$ and Y are the vectors of simulated and observed traffic measurements, respectively, $\mathcal{A}$ denotes the assignment method, and f_x and f_y are the goodness-of-fit (GoF) evaluation functions for OD patterns and traffic measurements. However, traffic-measurement-based OD estimators systematically suffer from the so-called high indeterminateness issue due to the imbalance between the number of unknowns (i.e., OD flows) and the number of equations (Antoniou et al., 2016). To this end, quasi-dynamic frameworks (e.g., Cascetta et al., 2013; Cantelmo et al., 2014b) and dimension reduction techniques (e.g., Djukic et al., 2012; Qurashi et al., 2019, 2022) have been widely applied to reduce the OD flow dimensions in the existing literature.

In this study, we leverage the advantages of LBSN data, which offers higher resolution in both spatial and temporal dimensions compared to traditional vehicle traffic measurements. We use this rich LBSN data to develop an OD estimation model aimed at mitigating the limitations inherent in vehicle traffic measurements.

Table 2
Nomenclature

Sets	
$\mathbb{V}_i$	set of activity nodes in the graph model of TAZ i
$\mathbb{Z}$	set of TAZs in the study area
Parameters	
$\boldsymbol{\rho}$	(time-dependent) activity share matrix
$\rho_{uv,i}$	(time-dependent) activity share from activity node u to v in i (normalized by the check-ins at u)
$\mathbf{x}^{(p)}$	(time-dependent) prior OD estimates
$\mathbf{q}$	(time-dependent) observed check-in counts
$q_{v,i}$	(time-dependent) observed check-in counts at activity v within TAZ i
$\mathbf{q}^{(l)}$	(time-dependent) observed check-in counts in the last time interval of the interval of interest
$q_{v,i}^{(l)}$	(time-dependent) observed check-in counts at activity v within TAZ i in the previous time interval
$\hat{P}(\mathbf{x})$	(time-dependent) zonal production derived from $\mathbf{x}$
$\hat{A}(\mathbf{x})$	(time-dependent) zonal attraction derived from $\mathbf{x}$
$P(\mathbf{q})$	(time-dependent) zonal production estimated based on the observed check-in counts $\mathbf{q}$
$A(\mathbf{q})$	(time-dependent) zonal attraction estimated based on the observed check-in counts $\mathbf{q}$
ξ	(time-dependent) problem state at the second stage representing the second-stage scenario
$q_{v,i}(\xi)$	(time-dependent) check-in counts of activity v in TAZ i under scenario ξ
$\Delta_i(\xi)$	(time-dependent) difference of check-in counts in scenario ξ between two successive time intervals
$\hat{\Delta}_i(\mathbf{y}_i)$	(time-dependent) difference of check-in counts derived from the optimized activity flows $\mathbf{y}_i$
Y	observed traffic measurements
$\mathcal{A}$	traffic assignment method
w_x	weighting factor for prior OD estimates in the objective function
w_p	weighting factor for zonal productions in the objective function
w_a	weighting factor for zonal attractions in the objective function
$\underline{\epsilon}_b$	threshold parameter of the lower bound for the posterior OD flows
$\bar{\epsilon}_b$	threshold parameter of the upper bound for the posterior OD flows
$\underline{\epsilon}_a$	threshold parameter of the lower bound for activity shares
$\bar{\epsilon}_a$	threshold parameter of the upper bound for activity shares
$\underline{\epsilon}_s$	threshold parameter of the lower bound for the source imbalance
$\bar{\epsilon}_s$	threshold parameter of the upper bound for the source imbalance
$\underline{\epsilon}_t$	threshold parameter of the lower bound for the sink imbalance
$\bar{\epsilon}_t$	threshold parameter of the upper bound for the sink imbalance
N_s	number of second-stage scenario samples
Decision variables	
x_{ij}	(time-dependent) OD (inter-zone) flow from TAZ i to j
$y_{vu,i}$	(time-dependent) activity flow from activity node v to u within TAZ i

4.2. Traffic analysis zone graph model

To facilitate the subsequent model development, we introduce the graph model of a TAZ as depicted in Figure 2. Specifically, the nodes within the graph model include a group of activity nodes, a virtual source, and a virtual sink. The edges, on the other hand, comprise activity flows that connect the activity nodes and virtual flows that link the virtual source and sink to the activity nodes. It is important to note that inter-zone flows, connecting different TAZs, fall outside the scope of any specific TAZ. Activity nodes and activity flows are defined as follows:

Definition 1 (Activity node)**.** *An activity node is a concentrating representation of the venues belonging to a certain root category inside the modeled traffic analysis zone.*

Definition 2 (Activity flow)**.** *An activity flow indicates the movements of people between two activity nodes within the modeled traffic analysis zone.*

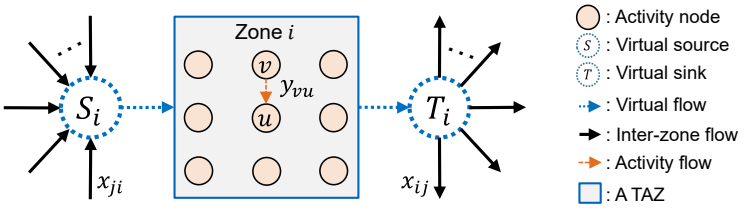


Figure 2: Graph model of a traffic analysis zone.

For example, of the venue category hierarchy illustrated in Figure 1, all venues falling under the category V_1 located within a given TAZ collectively form an activity node. The activity flows represent the movements among these specific venue categories. In addition, we create a virtual source and a virtual sink for each TAZ for aggregating the total incoming and outgoing trips, respectively, and effectively bridging the inter-zone flows and zonal activity flows. Here, inter-zone flows are what we traditionally call OD flows. In the following, we will also refer to them interchangeably as either inter-zone flows or OD flows. Virtual sources and virtual sinks are defined as follows:

Definition 3 (Virtual source)**.** *A virtual source of a traffic analysis zone refers to a node that serves as the central recipient for all inter-zone flows emanating from other traffic analysis zones. The virtual source is responsible for allocating these inflows across all activity nodes within the same zone.*

Definition 4 (Virtual sink)**.** *A virtual sink of a traffic analysis zone refers to a node that serves as the central origin for all inter-zone flows destined for other traffic analysis zones. The virtual sink is responsible for allocating the outflows from the zonal activity nodes to other zones.*

For a TAZ graph model, its virtual source and sink are connected to all of its activity nodes. Virtual flows are used to represent the movement of individuals either from the virtual source to an activity node or from an activity node to the virtual sink. Thus, the sum of flows to the sink and the sum of flows to the source can represent the zonal trip production and attraction, respectively. Later, we will show that inter-zone flows and activity flows are the first-stage and second-stage decision variables in the proposed two-stage stochastic programming model. That is to say, virtual sources and sinks are the bridge that links the first-stage and second-stage decisions. This treatment also enables the utilization of decomposition algorithms to expedite the solution of the OD estimation problem.

4.3. Two-stage stochastic programming model for LBSN-data-based OD estimation

OD flows are generated when people travel for various activities across different locations. Namely, OD patterns are the aggregated result of activity check-in patterns. Since OD patterns of a certain time-of-day interval would not change significantly within a period free of disruptive events (i.e., without critical changes in either supply or demand), the following assumption is plausible and is introduced to drive the modeling of the LBSN-data-based OD estimator.

Assumption 1. *For a certain time-of-day interval, similar activity check-in patterns on different days within a predefined reference period are generated by similar OD patterns.*

Assumption 1 reflects the temporal correlation of urban activity check-in patterns on different days. Similar to traffic measurements, activity check-in patterns can also be utilized to establish constraints for the OD estimator. With this assumption, a certain OD matrix needs to be mapped onto multiple check-in pattern scenarios (defined later in this section). In essence, this assumption allows us to exert additional constraints on the OD estimation problem for a specific time-of-day interval, thereby reducing the search space of the posterior OD estimates. Therefore, the LBSN-data-based OD estimator holds the potential of mitigating the indeterminateness issue existing in those based on traffic measurements.

The previous analysis inspires the development of an LBSN-data-based dynamic OD estimator leveraging a two-stage stochastic programming framework for incorporating multiple check-in pattern scenarios. Stochastic programming is an approach for modeling optimization problems under uncertainty. In the proposed model, check-in patterns represent the uncertain output of a certain OD matrix input to the system. In the two-stage framework, OD flows are the decision variables of the first-stage problem, and activity flows are the decisions of the second-stage problem.

To formulate the problem, we present the following modeling process for a specific time interval τ. For ease of presentation, the superscript τ will be omitted unless specified. With a slight abuse of notation, the variables corresponding to $\tau - 1$ will be indicated with a superscript (l). For example, $\mathbf{q}^{\tau}$ and $\mathbf{q}^{\tau-1}$ will be written as $\mathbf{q}$ and $\mathbf{q}^{(l)}$, respectively. Some important notations with their descriptions are listed in Table 2, in which time-dependent variables are specifically indicated.

4.3.1. *First-stage problem: OD flows estimation*

In the first stage, the OD flows are the decision variables. They are determined within the space restricted by three key considerations. First, akin to traffic-measurement-based OD estimators, taking into account the deviation between the posterior OD estimate and the prior OD estimate in the objective function can align the posterior OD estimate with a known OD pattern. The prior OD flows can be historical values or results from a four-step model that can capture the OD patterns under similar conditions. Second, we can easily deduce the zonal trip production and attraction for every TAZ from the estimated OD flows. Combining these estimated production and attraction values with a relationship model between zonal check-in counts and zonal production/attraction is capable of adjusting the OD demand level based on check-in observations. Third, the inclusion of multiple check-in pattern scenarios in the framework also plays a critical role in constraining the search space of the posterior OD flows.

Therefore, we formulate the generic objective function of the two-stage stochastic programming model for OD estimation using LBSN check-in data as follows:

$$\min_{\mathbf{x}} \quad w_x f_x\left(\mathbf{x}, \mathbf{x}^{(p)}\right) + w_p f_p\left(\hat{P}(\mathbf{x}), P(\mathbf{q})\right) + w_a f_a\left(\hat{A}(\mathbf{x}), A(\mathbf{q})\right) + w_s \mathbb{E}_{\xi}[Q(\mathbf{x}, \xi)] \tag{2}$$

where $\mathbf{x}$ is the decision variable of the first-stage problem, i.e., OD flows, $\mathbf{x}^{(p)}$ is the given prior OD flows. $\mathbf{q}$ is the vector of observed check-in counts, with each element representing the check-in counts at an activity node. $f_x(\cdot)$ is the GoF function measuring the difference between the posterior and prior OD flows. $\hat{P}(\mathbf{x})$ and $\hat{A}(\mathbf{x})$ are the vectors of out-flows (i.e., zonal production) and in-flows (i.e., zonal attraction) of zones, which are obtained by aggregating the posterior OD flows $\mathbf{x}$ correspondingly. $P(\mathbf{q})$ and $A(\mathbf{q})$ are the given production and attraction vectors estimated with the observed check-in counts $\mathbf{q}$. The empirical findings about their relationship are expounded on in Section 5.2. $f_p(\cdot)$ and $f_a(\cdot)$ measure the GoF between the modeled and the measured zonal production and attraction, respectively. w_x, w_p and w_a are weighting factors that quantify the relative reliability of the prior OD estimate, the prior production estimate and the prior attraction estimate. The last term in Equation (2) measures the expected minimum difference between the estimated and observed check-in counts difference given the OD flows $\mathbf{x}$. w_s is a weighting factor that quantifies the trade-off between the optimization of OD flow patterns and check-in patterns.

Besides, we enforce bound constraints on the OD flows to prevent the emergence of unrealistic solutions. These bounds are defined as multiples of the prior OD estimates given by

$$\underline{\epsilon}_b x_{ij}^{(p)} \leqslant x_{ij} \leqslant \overline{\epsilon}_b x_{ij}^{(p)} \qquad \forall i, j \in \mathbb{Z} \tag{3}$$

Where x_{ij} denotes the OD flow from TAZ i to j, $\mathbb{Z}$ denotes the set of TAZs under consideration, $\underline{\epsilon}_b$ (< 1) and $\overline{\epsilon}_b$(> 1) are threshold parameters.

4.3.2. *Second-stage problem: Activity flows estimation*

The further restriction of search space of OD flows is achieved by introducing a set of check-in scenarios of the second-stage problem state, as expressed by the last term in Equation (2), i.e., $w_s \mathbb{E}_{\xi}[Q(\mathbf{x}, \xi)]$. $\mathbb{E}_{\xi}$ calculates the expec-

tation with respect to a random vector ξ, defined on the probability space $(\Omega, \mathcal{F}, \mathcal{P})$, with Ω being the sample space, $\mathcal{F}$ being the event space, and $\mathcal{P}$ being a probability distribution defined on $\mathcal{F}$. ξ is a random variable describing the problem state at the second stage. $Q(\mathbf{x}, \xi)$ is the optimal value of the second-stage problem under scenario ξ.

In this study, a realization of the second-stage problem represents a check-in pattern scenario. Here, check-in patterns represent the changes in check-in counts between two successive time intervals. Considering the activity share difference in different times-of-day intervals, a check-in pattern scenario can then be described as a tuple composed of the check-in counts of the relevant two time intervals and the corresponding activity share matrix, i.e., $\left(\mathbf{q}^{(l)}, \mathbf{q}, \boldsymbol{\rho}\right)$, where $\boldsymbol{\rho}$ indicates the activity share matrix at τ.

The objective of the second-stage problem is to determine the activity flows that minimize the deviation between the estimated check-in counts difference and the observed one (i.e., $\mathbf{q}^{(l)} - \mathbf{q}$). Noting the structure of the proposed TAZ graph model and the definition of check-in pattern scenarios, three conditions must be taken into account and modeled in this stage, including (i) the conservation of check-in counts at activity nodes, (ii) the shares of activity flows, and (iii) the balance of inter-zone OD flows and zonal activity flows at the virtual sources and sinks. For illustration purposes, Figure 3 provides a graphical presentation of these conditions.

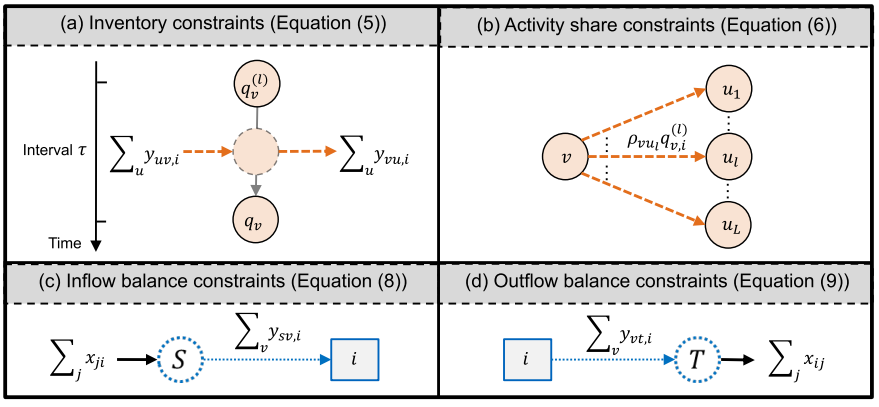


Figure 3: Graphic illustration of the conditions considered in the second-stage problem.

For a given check-in scenario ξ, the objective function of the second-stage problem can be expressed as

$$Q(\mathbf{x}, \xi) = \min_{\mathbf{y}} \quad \sum_{i \in \mathbb{Z}} f_s\left(\hat{\Delta}_i(\mathbf{y}_i), \Delta_i(\xi)\right) \tag{4}$$

where $\Delta_i(\xi) = \mathbf{q}_i^{(l)}(\xi) - \mathbf{q}_i(\xi)$ is the observed check-in counts difference in scenario ξ, $\hat{\Delta}_i(\mathbf{y}_i)$ is the estimated check-in counts difference derived from the optimized activity flows $\mathbf{y}_i$. As can be seen, this objective function is also the term inside the expectation calculator in the last term of the first-stage objective.

For activity node v in TAZ i, we have $\hat{\Delta}_{v,i} = \sum_{u \in (\mathbb{V}_i \setminus \{v\}) \cup \{t\}} y_{vu,i} - \sum_{u \in (\mathbb{V}_i \setminus \{v\}) \cup \{s\}} y_{uv,i}$. $f_s(\cdot)$ is a GoF function measuring the fitting between the estimated and observed check-in counts differences.

The first condition (see Figure 3a) expresses that for an activity node v, the total departing flows cannot be greater than the combined arriving flows and the number of check-ins recorded during the previous interval. It can be written as

$$\sum_{u \in (\mathbb{V}_i \setminus \{v\}) \cup \{t\}} y_{vu,i} - \left(\sum_{u \in (\mathbb{V}_i \setminus \{v\}) \cup \{s\}} y_{uv,i} + q_{v,i}^{(l)}(\xi) \right) \leqslant 0 \qquad \forall v \in \mathbb{V}_i, \forall i \in \mathbb{Z} \tag{5}$$

where $\mathbb{V}_i$ is the set of activity nodes in TAZ i. In practice, only the main venue categories in the TAZ will be considered for the sake of: (i) reducing the noise in the statistics caused by insufficient venues of a specific category; (ii) distinguishing different TAZs with respect to the land-use functionality and characteristics.

We denote $\rho_{vu,i}$ the share of activity flow from activity node v to u within TAZ i during interval τ. We can thus construct activity share constraints for the second condition (see Figure 3b) to restrict the search space of activity

flows by making use of the activity share matrix. This matrix encapsulates the activity chain information derived from user-side data. The constraints are given by

$$(1-\underline{\epsilon}_a)\rho_{vu,i}q_{v,i}^{(l)} \leqslant y_{vu,i} \leqslant (1+\overline{\epsilon}_a)\rho_{vu,i}q_{v,i}^{(l)} \qquad \forall v,u \in \mathbb{V}_i, \forall i \in \mathbb{Z} \tag{6}$$

where $\underline{\epsilon}_a$ and $\overline{\epsilon}_a$ are predefined threshold parameters within the range (0,1). They serve the purpose of mitigating the over-fitting issue during optimization by permitting slight deviations of activity flows from their theoretical values. Note, the activity share matrix can be normalized at either the network level or the TAZ level. For simplification, for a specific time-of-day interval of the day, we consider a common activity share matrix for the entire area, which can be estimated from the historical check-in data. As such, $\rho_{vu,i} = \rho_{vu}, \forall i \in \mathbb{Z}$. Then, Equation 6 can be simplified as

$$(1-\underline{\epsilon}_a)\rho_{vu}q_{v,i}^{(l)} \leqslant y_{vu,i} \leqslant (1+\overline{\epsilon}_a)\rho_{vu}q_{v,i}^{(l)} \qquad \forall v,u \in \mathbb{V}_i, \forall i \in \mathbb{Z} \tag{7}$$

Regarding the third condition, it is natural to impose the inflow and outflow balance constraints on the source and sink nodes, as shown in Figure 3c and 3d, respectively. Importantly, this condition establishes a connection between the decision variables in the first- and second-stage problems. More specifically, the inflow balance indicates that, for a specific TAZ, the aggregate virtual flows from the source node should equal the sum of inter-zone flows to it. Similarly, the outflow balance indicates that, for a specific TAZ, the sum of virtual flows to the sink node should match the sum of inter-zone flows from it. Nevertheless, due to the randomness and incompleteness of the activity information, it is reasonable allow a subtle deviation in both constraints. Consequently, the inflow and outflow balance can be expressed as

$$(1-\underline{\epsilon}_s)\sum_{j\in\{\mathbb{Z}-i\}} x_{ji} \leqslant \sum_{v\in\mathbb{V}_i} y_{sv,i} \leqslant (1+\overline{\epsilon}_s)\sum_{j\in\{\mathbb{Z}-i\}} x_{ji} \qquad \forall i \in \mathbb{Z} \tag{8}$$

$$(1-\underline{\epsilon}_t)\sum_{j\in\{\mathbb{Z}-i\}} x_{ij} \leqslant \sum_{v\in\mathbb{V}_i} y_{vt,i} \leqslant (1+\overline{\epsilon}_t)\sum_{j\in\{\mathbb{Z}-i\}} x_{ij} \qquad \forall i \in \mathbb{Z} \tag{9}$$

where $\underline{\epsilon}_s$, $\overline{\epsilon}_s$, $\underline{\epsilon}_t$ and $\overline{\epsilon}_t$ are predefined threshold parameters in the range (0,1), representing the extent to which the conservation can be violated. $y_{sv,i}$ denotes the virtual flow from the virtual source to activity node v within TAZ i, while $y_{vt,i}$ denotes the virtual flow from v to the virtual sink.

Furthermore, all activity flows and virtual flows should be non-negative as expressed by

$$y_{vu,i} \geqslant 0 \qquad \forall v,u \in \mathbb{V}_i, \forall i \in \mathbb{Z} \tag{10}$$

$$y_{sv,i} \geqslant 0 \qquad \forall v \in \mathbb{V}_i, \forall i \in \mathbb{Z} \tag{11}$$

$$y_{vt,i} \geqslant 0 \qquad \forall v \in \mathbb{V}_i, \forall i \in \mathbb{Z} \tag{12}$$

Note that in the second-stage problem, the optimal activity flows $\mathbf{y}^*$ depend on the first-stage OD flows $\mathbf{x}$ and the second-stage problem state ξ.

4.3.3. *Full two-stage stochastic model*

By integrating the objective functions and constraints of the two stages of problems described above, we obtain the two-stage stochastic programming model of the LBSN-data-based OD estimation problem (TSSP-OD) under uncertainty with multiple check-in pattern scenarios as below.

$$\text{(TSSP-OD)} \quad \min_{\mathbf{x}} \quad w_x f_x\left(\mathbf{x}, \mathbf{x}^{(p)}\right) + w_p f_p\left(\hat{P}(\mathbf{x}), P(\mathbf{q})\right) + w_a f_a\left(\hat{A}(\mathbf{x}), A(\mathbf{q})\right) + w_s \mathbb{E}_\xi[Q(\mathbf{x},\xi)] \tag{13}$$

$$\text{s.t.} \quad \underline{\epsilon}_b x_{ij}^{(p)} \leqslant x_{ij} \leqslant \overline{\epsilon}_b x_{ij}^{(p)} \qquad \forall i,j \in \mathbb{Z} \tag{14}$$

$$\text{where:} \tag{15}$$

$$Q(\mathbf{x},\xi) = \min_{\mathbf{y}} \quad \sum_{i\in\mathbb{Z}} f_s\left(\hat{\Delta}_i(\mathbf{y}_i), \Delta_i(\xi)\right) \tag{16}$$

$$\text{s.t.} \quad \sum_{u\in(\mathbb{V}_i\setminus\{v\})\cup\{t\}} y_{vu,i} - \left(\sum_{u\in(\mathbb{V}_i\setminus\{v\})\cup\{s\}} y_{uv,i} + q_{v,i}^{(l)}(\xi)\right) \leqslant 0 \qquad \forall v \in \mathbb{V}_i, \forall i \in \mathbb{Z} \tag{17}$$

$$(1-\underline{\epsilon}_a)\rho_{vu}q^{(l)}_{v,i}(\xi) \leqslant y_{vu,i} \leqslant (1+\overline{\epsilon}_a)\rho_{vu}q^{(l)}_{v,i}(\xi) \qquad \forall v,u \in \mathbb{V}_i, \forall i \in \mathbb{Z} \tag{18}$$

$$(1-\underline{\epsilon}_s)\sum_{j\in\{\mathbb{Z}-i\}} x_{ji} \leqslant \sum_{v\in\mathbb{V}_i} y_{sv,i} \leqslant (1+\overline{\epsilon}_s)\sum_{j\in\{\mathbb{Z}-i\}} x_{ji} \qquad \forall i \in \mathbb{Z} \tag{19}$$

$$(1-\underline{\epsilon}_t)\sum_{j\in\{\mathbb{Z}-i\}} x_{ij} \leqslant \sum_{v\in\mathbb{V}_i} y_{vt,i} \leqslant (1+\overline{\epsilon}_t)\sum_{j\in\{\mathbb{Z}-i\}} x_{ij} \qquad \forall i \in \mathbb{Z} \tag{20}$$

$$y_{vu,i} \geqslant 0 \qquad \forall v,u \in \mathbb{V}_i, \forall i \in \mathbb{Z} \tag{21}$$

$$y_{sv,i} \geqslant 0 \qquad \forall v \in \mathbb{V}_i, \forall i \in \mathbb{Z} \tag{22}$$

$$y_{vt,i} \geqslant 0 \qquad \forall v \in \mathbb{V}_i, \forall i \in \mathbb{Z} \tag{23}$$

By comparison, the proposed LBSN-data-based OD estimator resembles the generic traffic-measurement-based OD estimator to a certain extent: (i) the proposed model also relies on a prior OD flow estimate; (ii) $f_p(\cdot)$ and $f_a(\cdot)$ in the objective function of the first-stage problem (Equation (13)) and the activity share constraints in Equation (18) play a similar role to a traffic assignment method; (iii) both models need to handle the stochasticity of the observed data.

On the other hand, the difference between the two methods lies in that LBSN data are collected when the travel is finished, but traffic measurements are collected during the travel. In other words, LBSN contains end-to-end information, while traffic measurements record the situation between ends. As a result, the application of LBSN-data-based OD estimators usually demands no network structure but needs information on the activity preferences of travelers. More importantly, different from most traffic-measurement-based OD estimators in the literature, the proposed model can be used for dynamic OD estimation by using only the LBSN data without the need to run computationally expensive dynamic traffic simulations to generate simulated traffic measurements. This puts the proposed methodology at a significant computational advantage against most dynamic OD estimation approaches. Moreover, unlike the previous LBSN-based estimators, essentially, our model derives the OD matrix by reconstructing the activity-based mobility flows, making use of the confirmed trip purpose information.

4.4. Markov chain Monte Carlo sampling and sample average approximation

The proposed model is non-convex as the expectation $\mathbb{E}_\xi$ is usually an integral of a complex function. It is supposed to be difficult to get solved. Accordingly, in practice, we often need to assume ξ has a finite number of possible realizations with a known probability distribution, such that we can estimate $\mathbb{E}_\xi$ by

$$\mathbb{E}_\xi\left[Q(\mathbf{x},\xi)\right] = \sum_n^N p_n f_s\left(\hat{\Delta}_i(\mathbf{y}_i), \Delta_i(\xi_n)\right) \tag{24}$$

where N is the total number of realizations. Applying an appropriate sampling technique and the sample average approximation (SAA) method, the expectation can be approximated as

$$\mathbb{E}_\xi\left[Q(\mathbf{x},\xi)\right] \approx \frac{1}{N_s}\sum_n^{N_s} f_s\left(\hat{\Delta}_i(\mathbf{y}_i), \Delta_i(\xi_n)\right) \tag{25}$$

where N_s is the total number of scenario samples. It is worth mentioning that sampling techniques have to be fed by the check-in pattern scenario of the time interval under estimation (i.e., $(\mathbf{q}^{(l)}, \mathbf{q}, \boldsymbol{\rho})$) in order to generate reliable sampled scenarios. We call the scenario $(\mathbf{q}^{(l)}, \mathbf{q}, \boldsymbol{\rho})$ the reference scenario to distinguish it from the sampled scenarios. The sampled scenarios can be viewed as for the same time-of-day interval over different historical days that share similar context with the reference scenario. In this paper, we apply the Markov chain Monte Carlo (MCMC) sampling method, which has been popularly adopted for scenario generation in stochastic programming. Note, the integration of Monte Carlo (MC) and SAA can reduce the computational costs in generating cuts relative to the discrete-then-solve approach. In Section 4.5, we present our solution algorithm to the problem, Generalized Benders Decomposition (GBD), which is one kind of such approach.

MCMC algorithms refer to a category of MC methods that can generate a sequence of serially correlated samples from a probability distribution known up to a normalizing constant. These correlated samples can form a Markov chain with the stationary distribution as the target density, which also distinguishes MCMC algorithms from other MC methods. The modified Metropolis-Hastings algorithm proposed in Au and Beck (2001) is implemented in this study

as it is easy to implement, involves few parameters, and does not depend on many restrictive assumptions. Such a method can avoid introducing too much noise to the second-stage problem, thereby mitigating the impact of invalid check-in pattern samples on OD estimation. Moreover, the algorithm demonstrates enhanced efficiency in sampling high-dimensional variables, an aspect, where the original Metropolis-Hastings algorithm encounters difficulties.

The modified Metropolis-Hastings algorithm adopts an accept-reject principle. In the nth step, we denote the current state by ξ_n and denote the proposed state by ζ_n. For a specific activity node v, denote the proposal distribution of the number of check-ins at time τ by $\mathcal{S}_v(\cdot|q_v)$. The number of check-ins at the proposed state $q_v(\zeta_n)$ is then sampled from $\mathcal{S}_v(\cdot|q_v(\xi_n))$, i.e., $q_v(\zeta_n) \sim \mathcal{S}_v(\cdot|q_v(\xi_n))$. The acceptance probability can be calculated as

$$a_v(q_v(\xi_n), q_v(\zeta_n)) = \min\left\{1, \frac{\pi_v(q_v(\zeta_n))\mathcal{S}_v(q_v(\xi_n))|q_v(\zeta_n))}{\pi_v(q_v(\xi_n))\mathcal{S}_v(q_v(\zeta_n))|q_v(\xi_n))}\right\} \tag{26}$$

where $\pi_v(\cdot)$ is the known distribution of the number of check-ins of activity node v at time τ, which can be extracted from historical check-in data. Finally, the proposed state ζ_n is accepted or rejected based on the principle below.

$$q_v(\xi_{n+1}) = \begin{cases} q_v(\zeta_n) & \text{if } u_v \leqslant a_v(q_v(\xi_n), q_v(\zeta_n)) \\ q_v(\xi_n) & \text{otherwise} \end{cases} \tag{27}$$

where u_v is a scalar drawn from a standard uniform distribution, i.e., $u_v \sim \mathcal{U}(0, 1)$.

The full description of MCMC for check-in pattern sampling is given in Algorithm 1, with additional implementation details in Section 5.2.

Algorithm 1 Modified metropolis-Hastings algorithm for check-in pattern sampling

Initialize the number of samples to generate N_s.
Initialize the current state $\xi_0 = \xi$ with the check-in patterns of the time interval of interest (i.e., reference scenario).
Initialize $n = 0$.
while $n < N_s$ **do**
  for each $v = 1, 2, \cdots, |\mathbb{V}|$ **do**
    Simulate $q_v(\zeta_n) \sim \mathcal{S}_v(\cdot|q_v(\xi_n))$.
    Compute the acceptance probability:

$$a_v(q_v(\xi_n), q_v(\zeta_n)) = \min\left\{1, \frac{\pi_v(q_v(\zeta_n))\mathcal{S}_v(q_v(\xi_n))|q_v(\zeta_n))}{\pi_v(q_v(\xi_n))\mathcal{S}_v(q_v(\zeta_n))|q_v(\xi_n))}\right\}$$

    Accept or reject:

$$q_v(\xi_{n+1}) = \begin{cases} q_v(\zeta_n) & \text{if } u_v \leqslant a_v(q_v(\xi_n), q_v(\zeta_n)) \\ q_v(\xi_n) & \text{otherwise} \end{cases}$$

  end for
  Construct the $(n+1)$th check-in pattern scenario:

$$\mathbf{q}^{(l)}(\xi_{n+1}) = \mathbf{q}^{(l)}(\xi_0)$$
$$\mathbf{q}(\xi_{n+1}) = \left[q_1(\xi_{n+1}), q_2(\xi_{n+1}), \cdots, q_{|\mathbb{V}|}(\xi_{n+1})\right]^T$$
$$\Delta(\xi_{n+1}) = \mathbf{q}^{(l)}(\xi_{n+1}) - \mathbf{q}(\xi_{n+1})$$

  Set $n := n + 1$.
end while

4.5. Generalized Benders decomposition algorithm

GBD was first proposed in Geoffrion (1972) for addressing the mathematical programming problems with complicating variables (i.e., variables that if fixed to given values render a simple or decomposable problem). In two-stage

stochastic programming, the first-stage decision variables are the complicating variables of the problem. The idea behind GBD is to decompose the original problem into a master problem and a series of subproblems (one per scenario). In the master problem, the first-stage decisions (here, OD flows, $\mathbf{x}$) are optimized. In the subproblems, the second-stage decisions (here, activity flows, $\mathbf{y}$) are optimized. They are solved alternately until convergence. At a specific iteration k, the subproblems are solved first separately resulting in the optimum $\mathbf{y}^k(\xi)$ given $\mathbf{x}^{k-1}$ and scenario ξ. An optimality cut (or feasibility cut) is constructed based on the dual solutions of subproblems (or feasibility problem), which is added to the master problem as a new constraint. Given all cut constraints created through a pass-back mechanism from subproblems in previous iterations, the master problem is solved with respect to $\mathbf{x}$ resulting in $\mathbf{x}^k$. Note, these cuts gradually reduce the feasible space of the complicating variable.

To describe the algorithm, we sequentially provide the formulations of the subproblem (SP), the feasibility problem (FP), and the master problem (MP). At the k-th iteration, for a given scenario ξ_n and $\mathbf{x}^{k-1}$, the SP is formulated as follows:

$$\text{(SP)} \min_{\mathbf{y}} \quad \sum_{i\in\mathbb{Z}} f_s\left(\hat{\Delta}_i(\mathbf{y}_i), \Delta_i(\xi_n)\right) \tag{28}$$

$$\text{s.t.} \quad \text{Constraints (17)-(23)} \tag{29}$$

$$\mathbf{x} = \mathbf{x}^{k-1} : \quad \lambda_n^k \tag{30}$$

Compared to the original second-stage problem, SP has an additional equality constraint, Equation (30), which fixes the complicating variable $\mathbf{x}$ to the optimal value from the previous iteration. The solution of SP provides values to the activity flows $\mathbf{y}^k$ in different scenarios, as well as the corresponding optimal Lagrange multipliers vector associated with Equation (30), i.e., the optimal dual variables, λ^k. However, it is possible that not all SPs are solvable, whilst the feasibility of SPs is directly relative to the construction of Bender's cut at the respective algorithm iteration, i.e., optimality cut or feasibility cut. If SP is feasible, the optimality cut would be established using the terms in the Lagrangian function of SP that are relevant to $\mathbf{x}$. The formulation is given below.

$$\mathcal{L}_o(\mathbf{x}, \mathbf{y}^k(\xi_n), \lambda_n^k) = \sum_{i\in\mathbb{Z}} f_s\left(\hat{\Delta}_i(\mathbf{y}_i^k(\xi_n)), \Delta_i(\xi_n)\right) + (\lambda_n^k)^T(\mathbf{x} - \mathbf{x}^{k-1}) \tag{31}$$

However, if SP is infeasible, the following FP needs to be solved.

$$\text{(FP)} \min_{\mathbf{y},\eta} \quad \eta \tag{32}$$

$$\text{s.t.} \quad \text{Constraints (17)-(23)} \tag{33}$$

$$\eta \geqslant 0 \tag{34}$$

$$\mathbf{x} - \mathbf{x}^{k-1} \leqslant \eta : \quad \mu_n^k \tag{35}$$

Similarly, we can get the Lagrangian multiplier vector μ_n^k for the constraints expressed by Equation (35). The Lagrangian function of FP is given by (the terms irrelevant to $\mathbf{x}$ have been eliminated)

$$\mathcal{L}_f(\mathbf{x}, \mu_n^k) = (\mu_n^k)^T(\mathbf{x} - \mathbf{x}^{k-1} - \eta) \tag{36}$$

In this case, a feasibility cut is created based on the Lagrangian function of FP. Once SPs and/or FPs have been tackled, MP will be updated by adding new constraints of Benders cut in the current iteration. MP is formulated as

$$\text{(MP)} \min_{\mathbf{x},\alpha} \quad w_x f_x\left(\mathbf{x}, \mathbf{x}^{(p)}\right) + w_p f_p\left(\hat{P}(\mathbf{x}), P(\mathbf{q})\right) + w_a f_a\left(\hat{A}(\mathbf{x}), A(\mathbf{q})\right) + \alpha \tag{37}$$

$$\text{s.t.} \quad \underline{\epsilon}_b x_{ij}^{(p)} \leqslant x_{ij} \leqslant \overline{\epsilon}_b x_{ij}^{(p)} \qquad \forall i, j \in \mathbb{Z} \tag{38}$$

$$\frac{w_s}{N_s}\sum_{n=1}^{N_s} \mathcal{L}_o(\mathbf{x}, \mathbf{y}^t(\xi_n), \lambda_n^t) \leqslant \alpha \qquad \forall t \in \mathbb{I}_o \tag{39}$$

$$\mathcal{L}_f(\mathbf{x}, \mu_l^t) \leqslant 0 \qquad \forall l \in \mathbb{S}_f^t, \forall t \in \mathbb{I}_f \tag{40}$$

where $\mathbb{I}_o$ is the set of indices of the iterations at which all SPs are feasible, $\mathbb{I}_f$ is the set of indices of the iterations at which at least one of the SPs is infeasible, and $\mathbb{S}_f^t$ is the set of scenarios whose associated SPs are infeasible at iteration

t. Equations (39) are denominated as optimality cuts, while Equations (40) are feasibility cuts. From the MP, we can get the values of the first-stage decision variables $\mathbf{x}^k$.

It is worth mentioning that if the original objective function is convex on the complicating variable, GBD can guarantee the strong optimality condition, i.e., the optimal solution from the decomposed problems is equivalent to the original problem. For convenience, further details on the procedure of the GBD algorithm are presented in Algorithm 2.

To facilitate understanding, we present an outline of the procedure employed to address the TSSP problem for LBSN-data-based OD estimation in Figure 4. Given the required model inputs (i.e., $\mathbf{q}$, $\mathbf{q}^{(l)}$, $\boldsymbol{\rho}$, and $\mathbf{x}^{(p)}$) pertaining to the time interval of interest τ, MCMC sampling is utilized to generate a batch of similar check-in pattern scenarios based on the reference check-in pattern (i.e., $\mathbf{q}^{(l)}$ and $\mathbf{q}$). By integrating with the given activity shares information (i.e., $\boldsymbol{\rho}$), each individual scenario can be used to configure a corresponding second-stage problem. Combining these second-stage problems with the first-stage problem will result in the TSSP problem, which aims to estimate the OD flows for the given time interval. Subsequently, we employ the GBD algorithm to decompose the problem into a primary master problem and a series of subproblems. By solving these decomposed problems alternately, the optimal OD flows (i.e., $\mathbf{x}$) and activity flows (i.e., $\mathbf{y}$) that correspond to the observed check-in patterns will be obtained. We also give a small example in Appendix A for illustrating these concepts well.

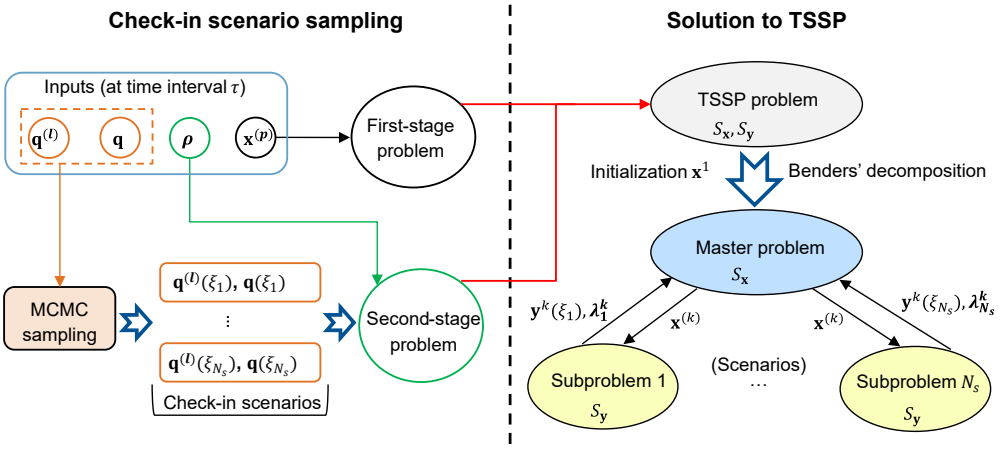


Figure 4: Solution procedure of the proposed LBSN-data-based OD estimator.

5. Experimental design

In this study, we test the proposed OD estimator using the Foursquare check-in data. Foursquare was launched in 2009 and has provided the leading LBSN service for more than a decade. As of 2019, it has included 105 million POIs in 190 countries and regions and received 1 billion check-ins annually. Foursquare data thus has broad spatial coverage and can somewhat capture human behavior in urban areas.

5.1. Case study setup

The Foursquare check-in data of Tokyo city, Japan, from April 2012 to February 2013[5], are used in the following experiments. Figure 5a shows the map of the study area and the delineation of TAZs. The study area (1,302 km^2) is divided into 17 TAZs. Figure 5b exhibits a heatmap of 10,000 check-in records randomly sampled from the entire dataset which contains 573,703 records. The heatmap has a clear center and the color intensity gradually fades from the center outward. Note, TAZs are devised based on the density of check-ins for the sake of statistical significance, i.e., denser areas have more TAZs.

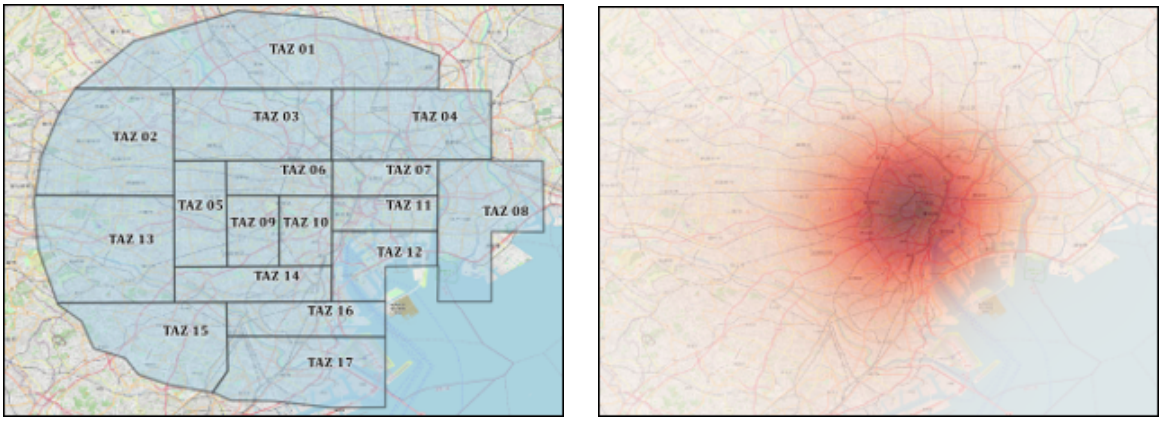
Due to the lack of venue-side data, we aggregate user-side data hourly for each root venue category[6], each TAZ, and each day to reconstruct the venue-side dataset. Categories with fewer than five check-ins are not further defined

[5] We refer the reader to Yang et al. (2014a) for a more detailed description of the dataset.

[6] Foursquare has its own proprietary taxonomy of more than 1000 categories. According to the hierarchical taxonomy of categories (version 2012), ten parent categories are defined, including Arts & Entertainment, College & University, Event, Food, Nightlife Spot, Outdoors & Recreation, Professional & Other Places, Residence, Shops & Service, Travel & Transport.

Algorithm 2 Generalized Benders decomposition algorithm for OD estimation

Initialize the OD flows $\mathbf{x}_0$.
Initialize the iteration index $k = 1$, complicating variables $\mathbf{x}^k = \mathbf{x}_0$, error tolerance ϵ, maximum number of iterations M.
Set the lower bound of the objective function $\underline{z}^k = 0$, and the upper bound $\overline{z}^k = \infty$.
while $|\overline{z}^k - \underline{z}^k|/|\underline{z}^k| \geqslant \epsilon$ and $k < M$ **do**
  Set $k := k + 1$.
  Solve the subproblems by fixing $\mathbf{x}$ as $\mathbf{x}^{k-1}$.
  if all subproblems are feasible **then**
    Obtain solution $\mathbf{y}^k$ and the dual variables of those constraints that fix the complicating variables to given values λ^k.
    Calculate:
    $z = w_x f_x(\mathbf{x}^{k-1}, \mathbf{x}^{(p)}) + w_p f_p(\hat{P}(\mathbf{x}^{k-1}), P) + w_a f_a(\hat{A}(\mathbf{x}^{k-1}), A) + \frac{w_s}{N_s} \sum_n^{N_s} \sum_z f_s \left(\hat{\Delta}_i(\mathbf{y}_i), \Delta_i(\xi_n)\right)$.
    Update the upper bound $\overline{z}^k = \min\{\overline{z}^{k-1}, z\}$.
    Set $\mathbb{I}_o := \mathbb{I}_o \cup \{k\}$.
  else
    Solve the feasibility problems associated with the infeasible subproblems.
    Obtain solution $\mathbf{y}^k$, dual variable μ^k and the set of infeasible subproblems $\mathbb{S}_f^k$.
    Set $\mathbb{I}_f := \mathbb{I}_f \cup \{k\}$.
  end if
  Add the new optimality cut (or feasibility cuts) to the master problem.
  Solve the master problem to get $\mathbf{x}^k$ and α^k.
  Update the lower bound $\underline{z}^k = f_x(\mathbf{x}^k, \mathbf{x}^{(p)}) + w_p f_p(\hat{P}(\mathbf{x}^k), P) + w_a f_a(\hat{A}(\mathbf{x}^k), A) + \alpha^k$.
end while



(a) Layout of TAZs (b) Heatmap of check-ins

Figure 5: Study area: City of Tokyo.

as activity nodes of the TAZ. This can help identify the functionality of TAZs and the source of their production and attraction over time. For instance, the TAZ with a huge number of "College & University" check-ins is more likely an area at which higher-educational institutions are located. It is worth mentioning that even though "home" and "work" activities are not covered adequately in LBSN data, the demand for them is still partially included in "Residential" and "Professional" activities, respectively.

Furthermore, we apply the moving average (seven days) technique to cancel some randomness of check-in behavior. In terms of the user-side dataset, we first extract the activity chain of each user. An activity share matrix can then be derived by counting the transfers between every two activities followed by normalization.

5.2. Algorithm setup

In the following experiments, we set the threshold parameters, $\{\underline{\epsilon}_s, \overline{\epsilon}_s, \underline{\epsilon}_t, \overline{\epsilon}_t, \underline{\epsilon}_a, \overline{\epsilon}_a\}$, as 0.2. The bound constraint parameters, $\{\underline{\epsilon}_b, \overline{\epsilon}_b\}$, are 0.2 and 5, respectively. We set the weighting factors $\{w_x, w_p, w_a, w_s\}$ as $\{1, 10, 0, 1\}$ unless otherwise specified.

For the MCMC sampling, the number of second-stage realizations N_s is set to 5. We assume a Gaussian distribution for the proposal distribution of an activity node v at time τ denoted by $S_v(\cdot|q_v(\xi_n))$. Specifically, $S_v(\cdot|q_v(\xi_n)) = \mathcal{N}(q_v(\xi_n), 0.1q_v(\xi_n))$, where $\mathcal{N}(\mu, \sigma)$ represents a Gaussian distribution with μ mean and σ^2 variance. Since Gaussian distributions are symmetric, $S_v(q_v(\xi_n))|q_v(\zeta_n)) = S_v(q_v(\zeta_n))|q_v(\xi_n))$, therefore Equation (26) reduces to

$$a_v(q_v(\xi_n), q_v(\zeta_n)) = \min\left\{1, \frac{\pi_v(q_v(\zeta_n))}{\pi_v(q_v(\xi_n))}\right\} \tag{41}$$

Regarding the GBD algorithm, the convergence threshold is $\epsilon = 0.05$. Without loss of generality, we apply the generalized linear squares (GLS) estimator in the goodness-of-fit functions, $f_x(\cdot)$, $f_p(\cdot)$, $f_a(\cdot)$ and $f_s(\cdot)$, resulting in a convex optimization problem with complicating variables. In this case, the GBD algorithm can guarantee the same global optimum solution as the original problem. Mathematically, $f_x(\cdot)$, $f_p(\cdot)$, $f_a(\cdot)$ and $f_s(\cdot)$ are given as follows:

$$f_x(\mathbf{x}, \tilde{\mathbf{x}}^\tau) = (\mathbf{x} - \tilde{\mathbf{x}}^\tau)^T \Lambda_x (\mathbf{x} - \tilde{\mathbf{x}}^\tau) \tag{42}$$

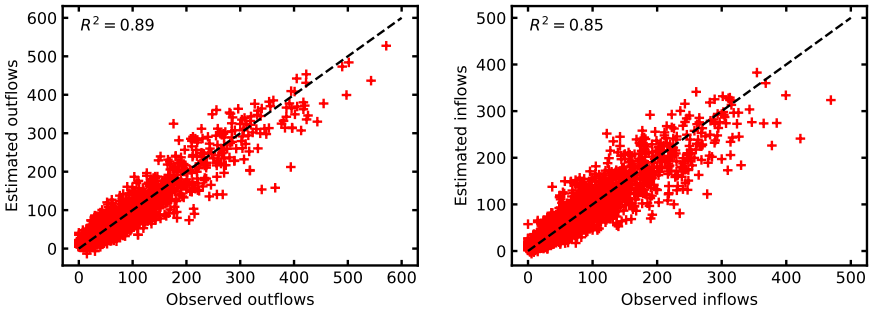
$$f_p\left(\hat{P}(\mathbf{x}), P(\mathbf{q})\right) = \left(\hat{P}(\mathbf{x}) - P(\mathbf{q})\right)^T \Lambda_p \left(\hat{P}(\mathbf{x}) - P(\mathbf{q})\right) \tag{43}$$

$$f_a\left(\hat{A}(\mathbf{x}), A(\mathbf{q})\right) = \left(\hat{A}(\mathbf{x}) - A(\mathbf{q})\right)^T \Lambda_a \left(\hat{A}(\mathbf{x}) - A(\mathbf{q})\right) \tag{44}$$

$$f_s\left(\hat{\Delta}_i(\mathbf{y}_i), \Delta_i(\xi_n)\right) = \left(\hat{\Delta}_i(\mathbf{y}_i) - \Delta_i(\xi_n)\right)^T \Lambda_s \left(\hat{\Delta}_i(\mathbf{y}_i) - \Delta_i(\xi_n)\right) \tag{45}$$

where $\Lambda_x, \Lambda_p, \Lambda_a$ and Λ_s are the dispersion matrices of the prior OD estimates, out-flows distribution, in-flows distribution, and check-in pattern, respectively. For simplicity, we set $\Lambda_x = \Lambda_p = \Lambda_a = \Lambda_s = \text{diag}(\mathbf{1})$.

Moreover, considering the linear relationship between the out-flows and the number of check-ins within the same TAZ observed from empirical data (as shown in Figure 6a), we calculate $P(\mathbf{q}) = \hat{\theta}^T \mathbf{q}$, where $\hat{\theta} = (C^T C)^{-1} C^T P_0$, C is the matrix of the number of check-ins aggregated by TAZs, and P_0 is the vector of historical observed outflow patterns, as the reference to penalize the demand level deduced from the estimated OD flows. Figure 6a compares the observed out-flows and the out-flows estimated by a simple linear regression model based on the number of check-ins. The R-square is about 0.89. It is worth mentioning that the entire check-in dataset is used for estimating $\hat{\theta}$. Similarly, we can get the relationship model between in-flows and the number of check-ins $A(\mathbf{q})$ with the same method.



(a) Out-flows comparison (b) In-flows comparison

Figure 6: Linear relationship between out-flows (left), in-flows (right), and the number of check-ins.

5.3. Demand scenario setup

We conduct experiments on the morning peak (7 am - 10 am) of February 1st, 2013. The morning peak is divided into three estimation time intervals each for one hour. A one-hour time interval is set due to the limitation on the check-in data density of this dataset. However, the proposed methodology is applicable to smaller time intervals in

the context of sufficient data, such as 15 minutes or half an hour. Regarding demand scenario generation, Antoniou et al. (2016) points out that the quality of the prior OD estimate, in terms of both demand level and patterns, is a key element that affects the performance of the OD estimator. Following the suggestions therein, we perturb the true OD flows to derive the historical OD flow estimates. More specifically, we create the prior OD estimate using the following formulation:

$$\tilde{\mathbf{x}}^{\tau} = (\phi_{rd} + \phi_{rm}\delta)\mathbf{x} \tag{46}$$

where ϕ_{rd} and ϕ_{rm} are the reduction and randomization parameters for perturbation, respectively, and δ is the random perturbation vector following Gaussian distribution. In the following experiments, we apply $\phi_{rd} = 0.7$, $\phi_{rm} = 0.3$, and $\delta \sim \mathcal{N}(0, 1/3)$ (99.7% of values located in [-1,1]), implying that the prior OD estimate is an out-of-date low-demand scenario.

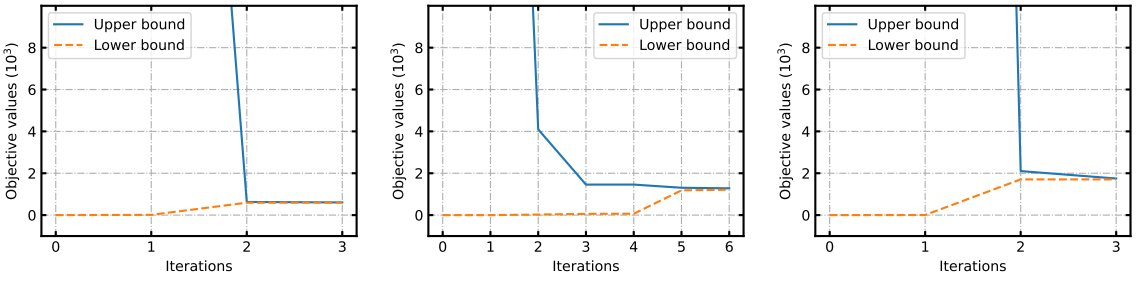
6. Results

In this section, first, we analyze the convergence performance of the GBD algorithm. Then, the fit of the estimated OD flows to the target OD demand, to the check-in pattern, and to the activity share is presented. To explore the effect of the quality and quantity of second-stage scenarios, we then compare the model performance under different numbers of scenario samples. We also test the models equipped with different formulations of the objective function (different combinations of GoF components, i.e., prior OD component, production component, attraction component, and check-in pattern component) for the purpose of validating the proposed framework.

6.1. Algorithm analysis

6.1.1. Convergence

Figure 7 shows the convergence results of the proposed estimation model. Note that since the initial upper bound is an infinity (a very large value in numerical computing), the upper bound at the first two iterations are invisible in the figures. Recalled that the upper bound is updated after solving subproblems, whilst the lower bound is updated after solving the relaxed master problem. At each iteration, the algorithm needs to solve N_s subproblems in parallel first, and then solve the relaxed master problem once. It is worth mentioning that the relaxed master problem can be solved very efficiently (in seconds) as it has a limited number of constraints and all constraints have the same format. In addition, we also found that, in our case, subproblems are always feasible if $\mathbf{x}_0$ is feasible, which means all Benders type cuts are optimality cuts, and therefore no feasibility problems needed to solve and no feasibility cuts are inserted to the master problem. In consequence, these experiments can be solved at a rather cheap computational cost, which indicates the proposed modeling framework has the potential to estimate dynamic OD matrices for large-scale networks. As expected, the algorithm converges within only several iterations for all study intervals, due to the property of the GBD algorithm and the convexity of the problem (GLS estimator). Besides, we found that the convergence of the upper bound can also lead to similar solutions to the ones obtained after the gap between the upper and lower bounds converges. Thus, in practice, one can use solely the upper bound to define the algorithm termination criterion. Considering that the lower bound may sometimes frustrate (which we observed in some experiments), this criterion can be a useful substitute.



(a) 7 am - 8 am (b) 8 am - 9 am (c) 9 am - 10 am

Figure 7: Convergence performance of the GBD algorithm.

6.1.2. Estimation quality evaluation

Figure 8 illustrates the quality of estimation with respect to the check-in patterns by comparing the empirical and estimated check-in pattern using 45° plots. We can see that all data points are aligned close to the "$y = x$" line in the three experiments. It means the proposed model is capable of recreating the check-in patterns. Recall that the check-in pattern is defined as the vector of the difference in check-in counts between successive time stamps. In other words, the good fitting of check-in patterns implies our model can also be used to predict the check-in statistics of activities (venue types). Actually, activity flows are the decision variables of the second-stage problem and check-in patterns represent the merging residuals of activity flows at activity nodes, so it is natural and plausible to reach such a conclusion.

Similarly, Figure 9 visualizes the quality of the estimated OD flows by comparing it with the target OD flows using 45° plots. The R^2 of estimated versus target OD flows in three experiments are 0.91, 0.89 and 0.93, respectively. Overall, the model reaches an acceptable estimate with a slight underestimation in the high-demand OD pairs. We note that the prior OD estimates underestimate the target values by 30% on average. While our model still exhibits underestimation (especially in the high-demand OD pairs), it can, to a certain extent, improve the situation, attributed to the inclusion of $f_p(\cdot)$ and the batch of check-in pattern scenarios. However, the accuracy still cannot meet the requirements of some transportation applications. As can be seen from Figure 9, most OD pairs observe a demand of fewer than 100 with many extremely small values. These OD pairs, on one hand, impede the convergence of the algorithm, and on the other reduce the accuracy of the GLS estimator. Thus, three methods can be used to improve the OD flow estimation, including (i) removing the OD pairs with small demand from the model, (ii) applying a reliable weighting matrix in the GLS estimator, instead of using diag($\mathbf{1}$) like in this paper, and (iii) replacing the GLS estimator with other estimators, such as maximum entropy.

Further, Figure 10 compares the theoretic activity shares and the estimated activity shares. Due to the large number of points, we add the heatmap effect in the figure to represent the density of points. Brighter colors mean greater density and vice versa. Overall, there are more points in the range of smaller values. Furthermore, in the figure, we can observe heteroscedasticity among the data points, indicating that the variance of the estimated activity shares increases with higher values of the true activity share. This phenomenon arises due to the activity share constraints expressed by Equation (18). These constraints restrict the feasible space of activity flows to be in proportion to the activity share matrix extracted from the activity chain information. As a consequence, the estimation of activity shares is affected by this constraint, leading to varying levels of variability in the data points based on the true activity shares. In addition, we also note that more active movements (larger activity shares present) can be observed in intervals 8 am - 9 am and 9 am - 10 am. Together with the label of these activity flows, for example, from "Residence" to "Professional & Other Places", Figure 10 can provide useful auxiliary information for dynamic traffic management, and help venues design working schedules.

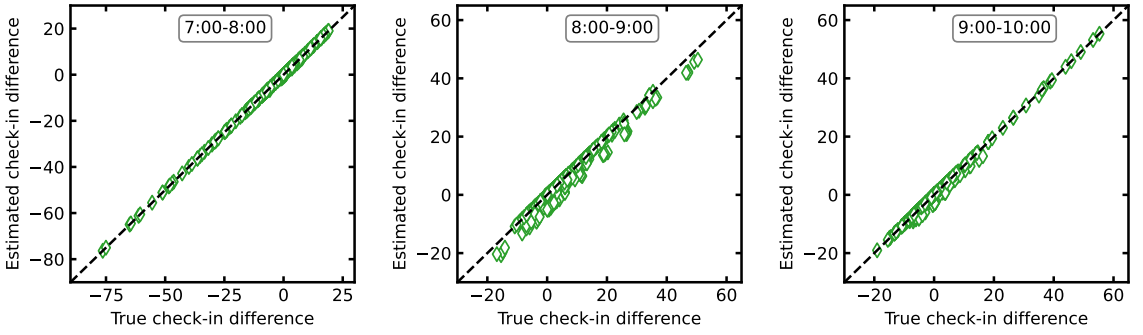


Figure 8: Comparison of true and estimated check-in patterns (performance of the second-stage problem).

6.1.3. Sensitivity against the number of second-stage scenarios

The model attempts to eliminate the common indeterminateness issue of traffic-measurement-based OD estimators by incorporating a batch of check-in scenarios that would impose constraints on the search space of the OD matrix with the help of the two-stage stochastic programming framework. The number of scenarios is critical to the effectiveness

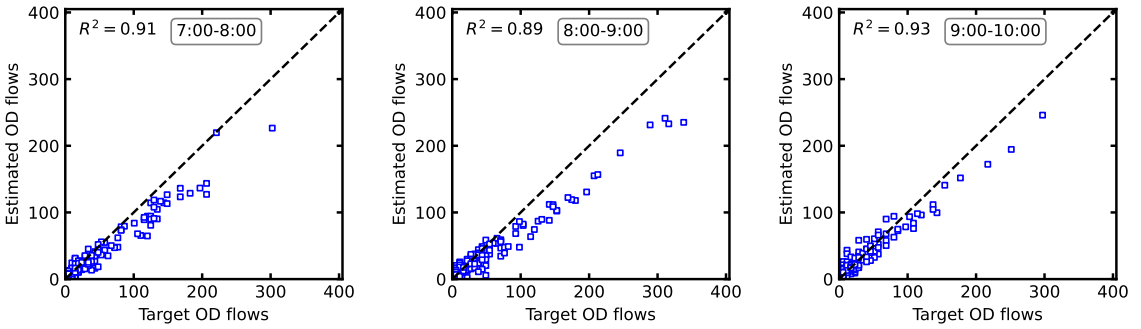


Figure 9: Comparison of target and estimated OD flows (performance of the first-stage problem).

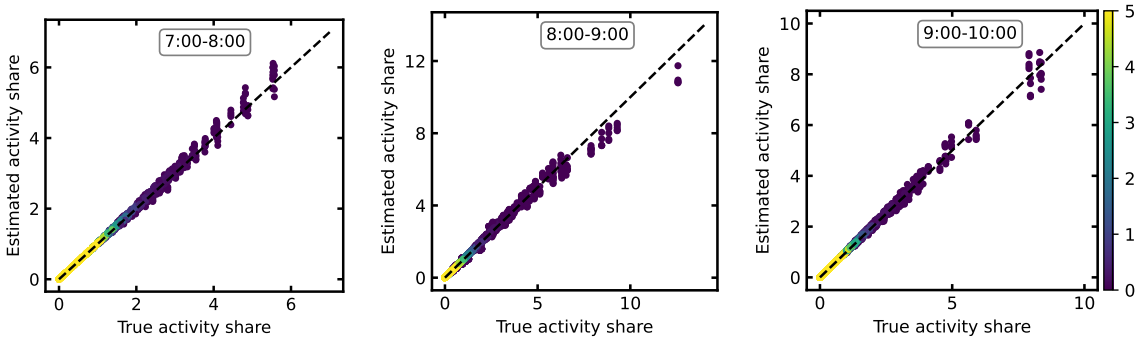


Figure 10: Comparison of true and estimated activity share (constraint violation).

of the method and the improvement in computational cost (compared to simulation-based approaches). Here, we apply four values of N_s (5, 10, 50, and 100) to examine the model and plot their results in Figure 11. Four important variables, i.e., OD flows, zonal production, zonal attraction, and check-in patterns, are included in the comparison, and the comparison is conducted with R^2 of the estimated versus target values (with the intercept fixed at 0). Among others, OD flows and check-in patterns are the decision variables of the model for capturing the demand pattern, while zonal production and zonal attraction are counted in the objective function with the purpose of increasing the accuracy of the demand level. For each N_s, considering the randomness of check-in pattern sampling, results from 10 replications are used to create the box plots.

Clearly, $N_s = 10$ outperforms $N_s = 5$ in the estimation of all variables, represented by better average R^2s. It implies that 5 check-in scenarios are insufficient to effectively restrict the search space of the OD matrix, while the new constraints introduced by adding new scenarios can still make a difference. On the contrary, considering too many check-in scenarios also has the possibility of resulting in sub-optimal solutions, especially because of the inclusion of biased scenarios, and adding more complexity. This is evidenced by the worse performance (except for attraction estimation) of the $N_s = 50$ model compared to that of $N_s = 10$. $N_s = 50$ even leads to a smaller R^2 than that of $N_s = 5$ in terms of OD flow and check-in pattern estimation. On the other hand, the $N_s = 50$ model outperforms the models with fewer scenarios in terms of zonal attraction estimation. However, we found that increasing N_s from 50 to 100 would reduce the model performance in all four indicators. This implies that adding too many check-in scenarios into consideration can no longer improve the model performance and might even lead to a performance decline.

Furthermore, we note that the deviation of R^2 in $N_s = 10$ is greater than the other scenarios. It reflects the significance of the quality of check-in scenarios in the proposed model, which is in line with our analysis in Section 4.4. Despite variations in performance among the four model scenarios, all of them can output estimations of sufficient quality, thereby indicating the robustness of the proposed approach.

It is noteworthy that the experiments and comparisons presented here were conducted in an objective state without the presence of zonal attraction ($\{w_x, w_p, w_a, w_s\} = \{1, 10, 0, 1\}$). Thus, zonal attraction serves as a validation metric to evaluate the effectiveness of the model and algorithm. The good fit of zonal attraction in all four scenarios indicates the soundness of the approach. Notably, the zonal attraction scores even outperform those of zonal production, further reinforcing the validity of the model and algorithm.

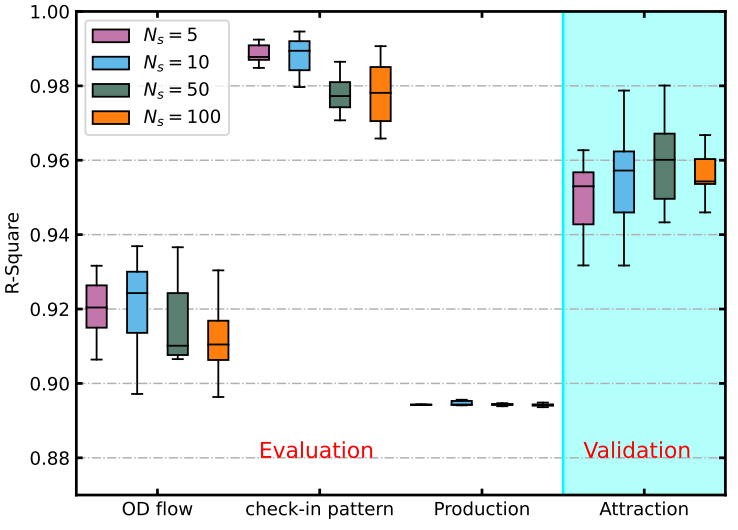


Figure 11: Boxplot for R-squares with different numbers of second-stage scenarios.

6.2. Objective function analysis

Referring back to Section 4.3, the objective function consists of three main types of error terms, i.e., $\mathbb{E}_\xi$, $\{f_p, f_a\}$, and f_x, where $\mathbb{E}_\xi$ is the base model formulated as the two-stage problem, while f_p and f_a can be triggered to add constraints via the trip generation/attraction relationship with check-in patterns subject to the city data, and f_x is a conventional error term widely used in DODE literature to keep the estimated OD patterns close to the prior OD flows. Therefore, the objective function can take multiple states depending on the availability and quality of the data. It is important to analyze the performance of the proposed model under all possible combinations. In this section, we compare the model performance under five different objective function states. Table 3 lists the weights of four error terms of these objective states. More specifically, WS tests the model without any restrictions on the OD flows and zonal production and attraction but solely with the check-in pattern optimization. WX0 is used to examine the importance of the prior OD flows. WP0 and WA0 check the necessity of restricting zonal production and attraction, respectively. Finally, state ALL represents the complete form of the objective function.

Figure 12 compares the model performance under these objective function states in terms of zonal production and attraction and OD flows. As can be seen from Figure 12, WS obtains a very biased estimation in all variables. This verifies the necessity and reliability of the proposed two-stage stochastic programming framework from the opposite side since if only considering the expectation term in the objective function the model would degrade to a purely single-stage problem. It means despite check-in patterns to a certain extent can be an information carrier of OD patterns, the auxiliary information about the demand level is required for a satisfactory estimation. Adding restrictions on zonal production and attraction as in WX0 can significantly improve the situation. In addition to a good fit in zonal production and attraction, WX0 also provides a fair estimation of OD flows even without feeding any information about OD patterns. Surprisingly, WP0 performs nicely in all three estimation tasks even only having penalties for posterior OD estimates and zonal attraction estimates. Nonetheless, it is also understandable with the consideration that the OD matrix is the mapping result between zonal production and attraction so determining the OD matrix and zonal attraction is capable of reproducing the zonal production as well. Similarly, we can observe comparable results in WA0. Furthermore, as asserted in Section 6.1.3, the error terms that are excluded from the objective function, such as zonal production in WP0 and zonal attraction in WA0, can be employed for validation purposes.

Overall, this proves the superiority of the proposed model in the estimation tasks of zonal production and attraction. Moreover, it also implies that with further integration of trip distribution information (e.g., the observed trip length frequency distributions from LSBN activity data), the model can robustly find correct OD matrix patterns, purely relying on LBSN data, without any prior OD estimate information (part focused for future research). In fact, such potential has been presented in WX0 in which only production and attraction data are available. As expected, ALL also leads to promising results. However, its performance cannot explicitly exceed that of WP0 and WA0.

Figure 13 shows a clearer comparison of the model under different objective functions by using R^2 as measurement. There is no surprise that WS performs worst in all three tasks. Note, in this study R^2 is calculated by fitting the corresponding data to a linear regression model constrained so that the intercept must equal 0. Under this constraint, the model may obtain a negative R^2 when the model cannot capture the trend of the data, thereby performing worse than the mean estimate (i.e., $\hat{y}_i = 1/n \sum_i^n y_i, \forall i$). This happens in WS, where the fits of production and attraction result in a R^2 of -1.10 and -0.45, respectively. On the contrary, the rest models achieve a score greater than 0.9 in all tasks, except WX0 gets around 0.6 in OD flow estimation. These results conform to the scatter plots in Figure 12.

Table 3
Weights of error terms in different objective function states.

State	w_x	w_p	w_a	w_s
WS	0	0	0	1
WX0	0	5	5	1
WP0	1	0	5	1
WA0	1	5	0	1
ALL	1	5	5	1

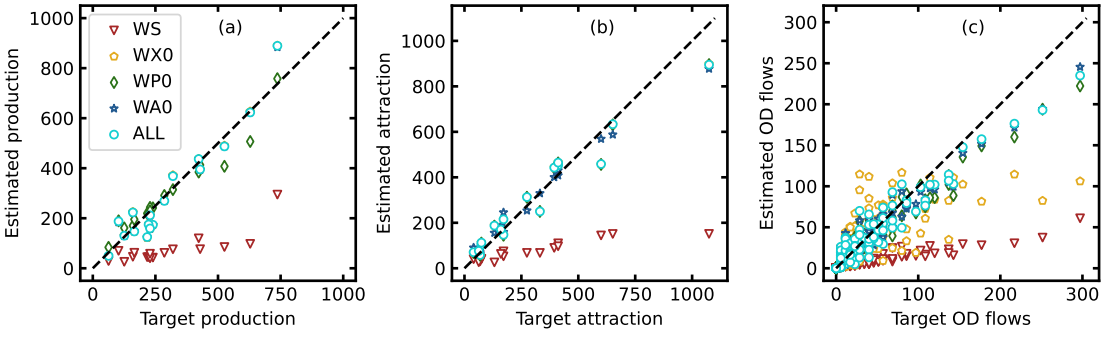


Figure 12: Comparison among different objective formulations.

7. Discussion

The effectiveness of the approach present in this study has been verified in Section 6. In this section, we discuss its enhancement from three perspectives, including (i) the integration with large-scale LBSN simulation models, (ii) the method for scaling the LBSN OD matrix to network-wide OD, and (iii) the substitutes for the user-side LBSN data.

7.1. Integrating large-scale LBSN simulation models against data sparsity

Utilizing real-world LBSN datasets easily suffers from the issues of data sparsity, small datasets, privacy concerns, and a lack of ground truth (Kim et al., 2020). Hence, many studies have been proposed to simulate synthetic LBSN-data-based on human patterns of life. For example, Kim et al. (2019) and Kim et al. (2020) develop an agent-based location-based social simulation framework by leveraging an open-source simulation toolkit, Multi-Agent Simulation of Neighborhoods (MASON). Each agent is defined on spatial networks composed of locations and social networks of users, in which the agent model logic is driven by Maslow's Hierarchy of Needs. Well-established models can provide synthetic but realistic LBSN data that meets the data requirement of the model proposed in this study. Integrating

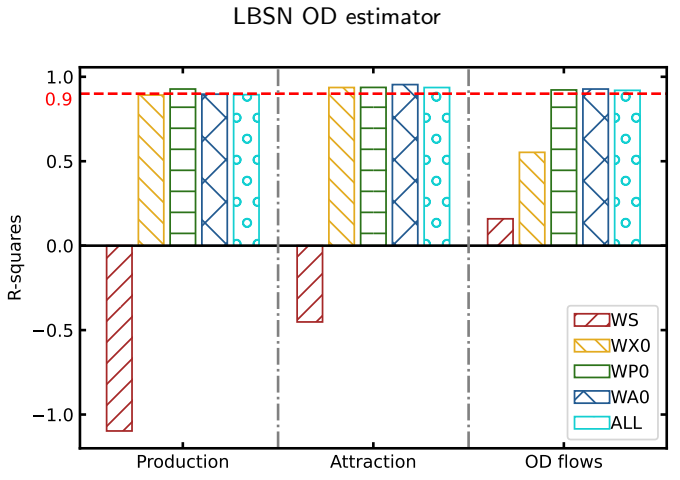


Figure 13: R^2 of zonal production, attraction and OD flows using different objective formulations.

our model with an LBSN simulation model can thus provide a complete solution to the urban OD matrix estimation problem. The integrated model can be used for either offline or online estimation. For the former, two models need to be run separately and sequentially: the LBSN dataset created by the LBSN simulation model is input to the two-stage stochastic framework to estimate the OD patterns in a short period. For the latter case, two models are run alternately with a feed-forward control algorithm to correct the LBSN simulation using the OD matrix estimated by the proposed model.

7.2. Scaling towards network OD flows

The OD flows mentioned in the previous sections only represent a part of the overall network OD flows that are observed in the LBSN check-in dataset (refer to LBSN OD matrix/flows hereafter). Whilst, we usually need the actual network OD matrix in practical applications instead. It is a critical input to traffic simulation models for evaluating traffic management and policy measures (Antoniou et al., 2016). We argue that due to the random nature of check-in behaviors, the LBSN OD matrix follows the same structural pattern as the network OD matrix, except for the deviation in the magnitude of the demand level. Figure 14 compares the TomTom OD shares, LBSN OD shares, and the estimated OD shares, all of which are normalized with the respective zonal production. Due to the lack of required data, here the average of TomTom OD flows of all Wednesdays in January 2021 is used to represent the "true" network OD flows. TomTom is a Dutch corporation launched in 1991 that specializes in the production of car navigation systems of all types. Many enterprises, such as Microsoft and Uber, use TomTom's positioning technologies, due to their high precision. The TomTom data are collected from the probe vehicles which are equipped with TomTom's positioning devices. Given the high penetration rate of TomTom positioning devices, we believe that TomTom data can capture the real traffic state to a large extent. Note, seven TAZs with missing data in TomTom are excluded from the comparison. Figure 14 shows that the LBSN OD share matrix has a similar pattern with the network OD shares, and most travel demands happened between the OD pairs within the dashed orange area. It also verifies our assumption that the general OD patterns will not change significantly. Since the estimated OD flows fit the LBSN OD flows well as demonstrated in Section 6, the estimated OD shares can also be a good estimate of the network OD shares. However, for example, the OD pairs inside the dotted purple area also indicate the bias between network OD flows and LBSN OD flows. This may be caused by (i) the long distance in time that the two datasets are collected, (ii) the bias in data collection methods, and (iii) the removal of seven TAZs affects the accuracy of normalization. Nevertheless, Figure 14 confirms that the LBSN OD flows estimated by the proposed model can approximate the real network OD flows after appropriate scaling. Theoretically, the scaling matrix can be either used on the estimation results or directly integrated into the estimation framework like $\Phi_p(\mathbf{q})$ and $\Phi_a(\mathbf{q})$. Methods to estimate the scaling matrix can include, (i) attaching the estimated demand to a traffic assignment model and utilizing conventional traffic measurement data sources to approximate the scaling matrix, (ii) utilizing other open-source network demographic data (i.e., population, workplaces) to directly scale up the productions and attractions for representing the network-level travel demand. Note that exploring more precise scaling methods to improve the network OD estimation is part of our future work.

On the other hand, noting the possibility of extending the model with the integration with an LBSN simulation

model (e.g., Kim et al., 2019, 2020), it would become unnecessary to estimate the scaling matrices anymore as one can infer the full knowledge of check-in of the whole population using the simulated data from LBSN simulations. In this regard, LBSN data is completely equivalent to traffic measurements and also has the advantage of additional information about travel purposes. Besides, the absence of dynamic traffic simulation is computationally expensive in the model, which also makes LBSN-based OD estimation models have promising prospects in online implementation, especially after noting that the LBSN simulation can be run very efficiently (Kim et al., 2020).

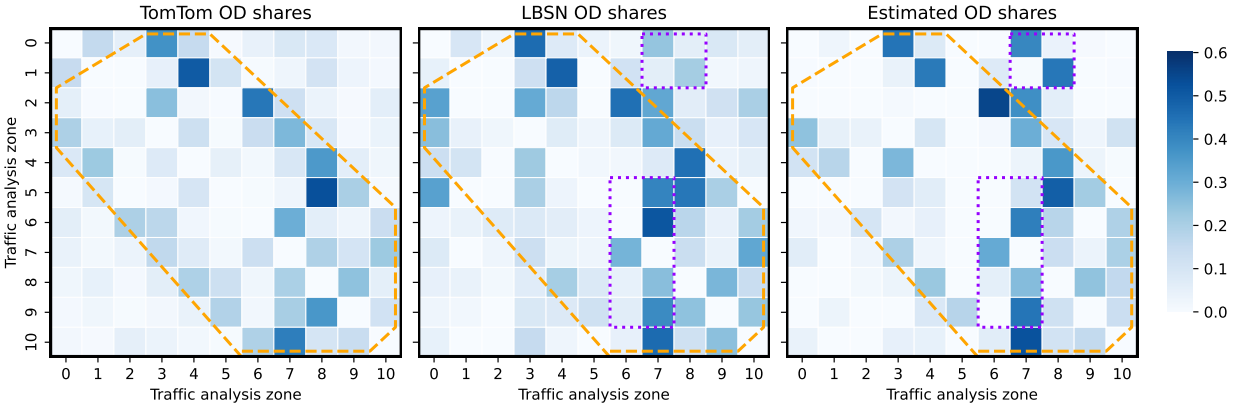


Figure 14: Comparing TomTom OD shares, LBSN OD shares, and estimated OD shares. All values are normalized by the zonal production of the respective origin.

7.3. Substitutes for user-side LBSN data

The collection and use of location-based social networking check-in data have raised an uproar of concern for user privacy (Chaniotakis and Antoniou, 2015), as it may reveal sensitive information about users' whereabouts and activities. Although data anonymization and other information protection measures have been adopted, the public remains cautious about the confidentiality of personal information, as such data may also be susceptible to misuse, unauthorized access, or security breaches. In this respect, the availability of user-side data may impede the application of the proposed model. Nevertheless, we argue that the user-side data is not necessary for the use of the model. Recall that the user-side data is only used to extract the activity chain information for constraining the second-stage decision variables (i.e., activity flows). As a result, an activity chain model can be substituted perfectly for the user-side data. Similar to LBSN simulation models and OD scaling methods, the integration with an activity chain model can also be realized in two ways, namely, (i) solely adopting the activity share matrix (as the activity chains extraction process shown in Figure 1) estimated by the activity chain model, and (ii) directly assembling the activity chain model into the two-stage framework. Integrating such approaches is part of our future work.

8. Conclusion

Origin-destination (OD) estimation has traditionally relied on two primary data sources, i.e., household travel surveys and traffic network detection. However, these methods suffer from various limitations that restrict their applicability in practice. Household travel surveys, despite providing detailed socio-demographic information, are known to be time-consuming, labor-intensive, and expensive, restraining their coverage and frequency. Consequently, they are typically limited to use in urban planning models that usually require only average network conditions, rather than up-to-date fine-grained mobility information. On the other hand, the high cost of installation and maintenance of traffic detection infrastructure restricts the density of detectors across urban areas, leading to issues of indeterminateness in OD estimation methods. The effectiveness of traffic-measurement-based estimators is also highly dependent on the extent to which the fixed detectors can represent the traffic state of the entire network (which is a strong assumption). As a result, OD estimation methods that utilize inexpensive and widespread data sources to overcome these limitations are in demand.

Considering the stochastic nature of human behaviors and transportation systems, we propose a dynamic OD estimator by utilizing the scenario-based two-stage stochastic programming framework, which integrates the activity

chains extracted from location based social networks (LBSN) check-in data to model activity-level mobility flows. Given that OD flows are the results of aggregated activity flows, the OD matrix can thus be derived from activity flows easily. More specifically, utilizing the framework, our approach aims to minimize the errors introduced by the inter-zone OD flows and the expected errors of the check-in patterns in the first stage, while minimizing the errors produced by the check-in pattern scenarios in the second stage. Markov Chain Monte Carlo (MCMC) sampling is implemented to generate a batch of statistically significant check-in scenarios. Ultimately, a generalized Benders decomposition (GBD) algorithm is applied to solve the two-stage stochastic programming problem, in which the optimal solution is obtained by solving a relaxed master problem and a series of subproblems (one per scenario) alternately.

To evaluate the effectiveness of the proposed approach, we conduct the case study in Tokyo city, Japan, and employ the generalized least squares (GLS) estimator to measure model performance in estimations of check-in and OD patterns, which on the other guarantees the problem convexity. The experiment results show that the convergence of the GBD algorithm can be attained within several iterations. More Importantly, in regard to estimation quality, the model also observes a good fit for check-in patterns, zonal production, zonal attraction, and OD flows. The robustness with respect to the check-in scenarios is also examined and evidenced by incorporating different numbers of check-in scenarios in the second-stage problem. We also explore the objective function formulation space by adopting multiple objective function states with different combinations of error terms. We find that the interdependence of zonal production, zonal attraction, and OD matrix makes the inclusion of all error terms in the objective unnecessary. In particular, considering two of them can already produce estimations as good as considering all. It intuitively follows that the variable excluded from the objective function can serve as the validation metric.

In addition to the advantages of being publicly available with broad coverage, LBSN data holds the nature of self-contained confirmed trip purposes. Unlike traffic measurements that capture the network dynamics between ends (of activity chains), LBSN data is benefiting from end-to-end information on individuals' daily mobility. Hence, our model is simulation-free and independent of network structure, therefore the required computational effort is almost the bare minimum compared to traffic-measurement-based estimators (which run dynamic traffic assignment iteratively) given that the model convergence requires several iterations in addition to the manual efforts to set up the simulation model itself. While the computational time of traffic-measurement-based estimators increases with the complexity and scale of network structure, the LBSN-based estimator presented in this study only depends on the dimension of the OD matrix and the complexity of the graph model (e.g., the number of activity nodes considered). Furthermore, the inclusion of trip purposes information in LBSN data allows for a better understanding of the attractiveness and popularity of activities at different times of day, which is missing in most existing OD estimators.

However, the proposed methodology also exhibits the following limitations: (i) In cases of LBSN data lacking venue category information, the methodology may not be viable due to the inherent challenge of differentiating between POIs situated within the same building. (ii) Furthermore, the variability in the representativeness of LBSN check-in data across diverse activities, stemming from potential sampling bias, might affect the reliability of the estimation results. In regards to further enhancements and future work, the methods can extend the proposed model from three different aspects. First, integrating an LBSN simulation model, on the one hand, can create a complete solution for using LBSN data to estimate demand patterns either offline or online, benefiting from the efficiency of large-scale LBSN simulation. On the other hand, it also avoids the usage of personal activity data that may cause privacy risks even though these data have been anonymized before release. Second, we show that one can easily scale up the LBSN OD matrix to approximate the real network OD matrix, which can spark the implementation of the proposed model in practical applications. Similar OD shares observed between LBSN data and floating car data further bolster the application potential of LBSN-based OD estimators. Embedding a suitable scaling method into the model will also be a promising direction for enhancing the practical applicability of the approach. Third, given that user-side data are only used for extracting activity chain information, any activity chain models that can provide similar information can be perfect substitutes for the user-side LBSN data when it is unavailable. These model enhancements are part of our future work. Furthermore, considering that the proposed model is an activity-level flow estimation model, it will also be interesting to study the relationship between the graph model and zonal functionalities and conduct more exploration of trip purpose information contained in the LBSN data.

Acknowledgements

This work was supported by the European Interest Group CONCERT-Japan DARUMA project (Grant Number: 01DR21010) funded by the German Federal Ministry of Education and Research (BMBF). This research was also

partially funded by the German Research Foundation DFG (TRAMPA Project, Grant 415208373) and the BMBF funded cluster MCube (COLTOC project, Germany, funding number: 03ZU1105KA). We would also like to thank TomTom N.V. for providing the network OD data.

Appendix A A small illustration example

In this Appendix, we provide a small illustration example of creating check-in pattern scenarios for the implementation of the proposed LBSN-data-based OD estimator. The example area consists of two TAZs, z_1 and z_2, as shown in Figure 15. z_1 has activity nodes v_1 and v_2, while z_2 has v_1, v_2 and v_3. To clarify the concept of check-in scenarios, we consider an OD estimation problem of two time intervals τ and $\tau+1$.

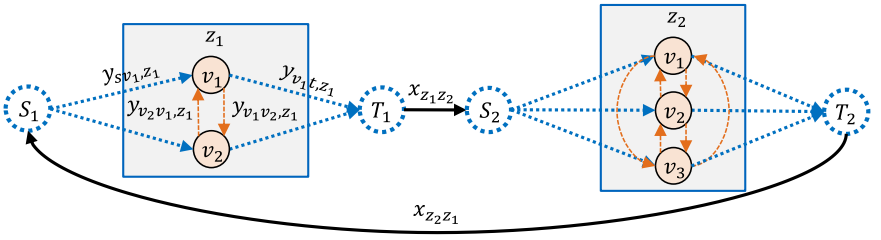


Figure 15: A small example area with two TAZs.

Solving this problem requires as inputs the check-in counts in $\tau-1$, τ and $\tau+1$, and the activity share matrices of τ and $\tau+1$. Recall that the activity share can be aggregated at either the network level or the TAZ level. Here we apply the same treatment of calculating the activity shares at the network level. The outputs of the estimator are the OD flows. Specifically, these inputs and outputs are given as

$$\mathbf{q}^{\tau-1} = \left[q^{\tau-1}_{v_1,z_1}, q^{\tau-1}_{v_2,z_1}, q^{\tau-1}_{v_1,z_2}, q^{\tau-1}_{v_2,z_2}, q^{\tau-1}_{v_3,z_2}\right]^T \qquad \boldsymbol{\rho}^{\tau} = \begin{bmatrix} \rho^{\tau}_{v_1v_1} & \rho^{\tau}_{v_1v_2} & \rho^{\tau}_{v_1v_3} \\ \rho^{\tau}_{v_2v_1} & \rho^{\tau}_{v_2v_2} & \rho^{\tau}_{v_2v_3} \\ \rho^{\tau}_{v_3v_1} & \rho^{\tau}_{v_3v_2} & \rho^{\tau}_{v_3v_3} \end{bmatrix} \qquad \boldsymbol{\rho}^{\tau+1} = \begin{bmatrix} \rho^{\tau+1}_{v_1v_1} & \rho^{\tau+1}_{v_1v_2} & \rho^{\tau+1}_{v_1v_3} \\ \rho^{\tau+1}_{v_2v_1} & \rho^{\tau+1}_{v_2v_2} & \rho^{\tau+1}_{v_2v_3} \\ \rho^{\tau+1}_{v_3v_1} & \rho^{\tau+1}_{v_3v_2} & \rho^{\tau+1}_{v_3v_3} \end{bmatrix}$$

$$\mathbf{q}^{\tau} = \left[q^{\tau}_{v_1,z_1}, q^{\tau}_{v_2,z_1}, q^{\tau}_{v_1,z_2}, q^{\tau}_{v_2,z_2}, q^{\tau}_{v_3,z_2}\right]^T$$

$$\mathbf{q}^{\tau+1} = \left[q^{\tau+1}_{v_1,z_1}, q^{\tau+1}_{v_2,z_1}, q^{\tau+1}_{v_1,z_2}, q^{\tau+1}_{v_2,z_2}, q^{\tau+1}_{v_3,z_2}\right]^T \qquad \mathbf{x}^{\tau} = \left[x^{\tau}_{z_1z_2}, x^{\tau}_{z_2z_1}\right]^T \qquad \mathbf{x}^{\tau+1} = \left[x^{\tau+1}_{z_1z_2}, x^{\tau+1}_{z_2z_1}\right]^T$$

Using the inputs, we can construct two reference check-in pattern scenarios $\left(\mathbf{q}^{\tau-1}, \mathbf{q}^{\tau}, \boldsymbol{\rho}^{\tau}\right)$ and $\left(\mathbf{q}^{\tau}, \mathbf{q}^{\tau+1}, \boldsymbol{\rho}^{\tau+1}\right)$. MCMC fed by the reference scenarios is then used to generate the respective sampled scenarios following Algorithm 1. Taking $N_s = 2$ as an example, Figure 16 shows the process of generating sampled scenarios. $\mathbb{S}^{\tau}$ and $\mathbb{S}^{\tau+1}$ denote the set of sampled scenarios for τ and $\tau+1$, respectively. For a specific time interval, its sampled scenarios will be used to construct the second-stage problems. For instance, $\left(\mathbf{q}^{\tau-1}(\xi^{\tau}_1), \mathbf{q}^{\tau}(\xi^{\tau}_1), \boldsymbol{\rho}^{\tau}\right)$ and $\left(\mathbf{q}^{\tau-1}(\xi^{\tau}_2), \mathbf{q}^{\tau}(\xi^{\tau}_2), \boldsymbol{\rho}^{\tau}\right)$ will build up two second-stage problems for estimating the OD flows in τ. At iteration k of Algorithm 2, given $\mathbf{x}^{(k)} = \left[x^{\tau,(k)}_{z_1z_2}, x^{\tau,(k)}_{z_2z_1}\right]^T$, the subproblems (SP, Equation (28)-(30)) corresponding to these two second-stage problems will be solved to update the upper bound of the objective function (Equation (13)) to facilitate the algorithm towards convergence.

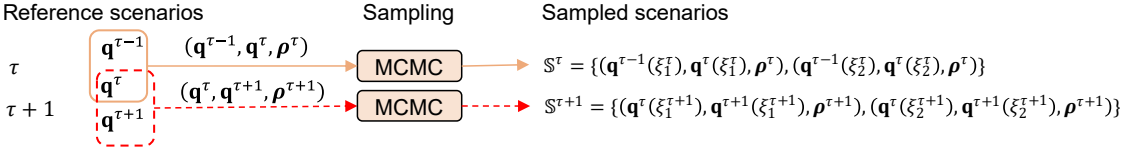


Figure 16: Check-in pattern scenarios sampling.

References

Antoniou, C., Azevedo, C.L., Lu, L., Pereira, F., Ben-Akiva, M., 2015. W-SPSA in practice: Approximation of weight matrices and calibration of traffic simulation models. Transportation Research Part C: Emerging Technologies 59, 129–146.

Antoniou, C., Barceló, J., Breen, M., Bullejos, M., Casas, J., Cipriani, E., Ciuffo, B., Djukic, T., Hoogendoorn, S., Marzano, V., Montero, L., Nigro, M., Perarnau, J., Punzo, V., Toledo, T., van Lint, H., 2016. Towards a generic benchmarking platform for origin–destination flows estimation/updating algorithms: Design, demonstration and validation. Transportation Research Part C: Emerging Technologies 66, 79–98.

Au, S.K., Beck, J.L., 2001. Estimation of small failure probabilities in high dimensions by subset simulation. Probabilistic engineering mechanics 16, 263–277.

Balakrishna, R., Antoniou, C., Ben-Akiva, M., Koutsopoulos, H.N., Wen, Y., 2007. Calibration of microscopic traffic simulation models: Methods and application. Transportation Research Record 1999, 198–207.

Bauer, D., Richter, G., Asamer, J., Heilmann, B., Lenz, G., Kölbl, R., 2017. Quasi-dynamic estimation of OD flows from traffic counts without prior od matrix. IEEE Transactions on Intelligent Transportation Systems 19, 2025–2034.

Ben-Akiva, M., Bierlaire, M., Burton, D., Koutsopoulos, H.N., Mishalani, R., 2001. Network state estimation and prediction for real-time traffic management. Networks and Spatial Economics 1, 293–318.

Bowman, J.L., Ben-Akiva, M.E., 2001. Activity-based disaggregate travel demand model system with activity schedules. Transportation Research Part A: Policy and Practice 35, 1–28.

Cantelmo, G., Cipriani, E., Gemma, A., Nigro, M., 2014a. An adaptive bi-level gradient procedure for the estimation of dynamic traffic demand. IEEE Transactions on Intelligent Transportation Systems 15, 1348–1361.

Cantelmo, G., Qurashi, M., Prakash, A.A., Antoniou, C., Viti, F., 2020. Incorporating trip chaining within online demand estimation. Transportation Research Part B: Methodological 132, 171–187.

Cantelmo, G., Viti, F., Tampère, C.M., Cipriani, E., Nigro, M., 2014b. Two-step approach for correction of seed matrix in dynamic demand estimation. Transportation Research Record 2466, 125–133.

Cascetta, E., Papola, A., Marzano, V., Simonelli, F., Vitiello, I., 2013. Quasi-dynamic estimation of o–d flows from traffic counts: Formulation, statistical validation and performance analysis on real data. Transportation Research Part B: Methodological 55, 171–187.

Cascetta, E., Postorino, M.N., 2001. Fixed point approaches to the estimation of O/D matrices using traffic counts on congested networks. Transportation Science 35, 134–147.

Cebelak, M.K., 2013. Location-based social networking data: doubly-constrained gravity model origin-destination estimation of the urban travel demand for austin, TX .

Chaniotakis, E., Antoniou, C., 2015. Use of geotagged social media in urban settings: Empirical evidence on its potential from twitter, in: 2015 IEEE 18th International Conference on Intelligent Transportation Systems, IEEE. pp. 214–219.

Cho, Y.S., Ver Steeg, G., Galstyan, A., 2014. Where and why users "check in", in: Proceedings of the AAAI Conference on Artificial Intelligence.

Djukic, T., Van Lint, J., Hoogendoorn, S., 2012. Application of principal component analysis to predict dynamic origin–destination matrices. Transportation Research Record 2283, 81–89.

Flötteröd, G., Bierlaire, M., Nagel, K., 2011. Bayesian demand calibration for dynamic traffic simulations. Transportation Science 45, 541–561.

Frederix, R., Viti, F., Corthout, R., Tampère, C.M., 2011. New gradient approximation method for dynamic origin–destination matrix estimation on congested networks. Transportation Research Record 2263, 19–25.

Geoffrion, A.M., 1972. Generalized benders decomposition. Journal of Optimization Theory and Applications 10, 237–260.

Hu, N.W., Jin, P.J., 2015. Dynamic trip attraction estimation with location based social network data balancing between time of day variations and zonal differences. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences 2, 193.

Hu, W., Jin, P.J., 2017. An adaptive hawkes process formulation for estimating time-of-day zonal trip arrivals with location-based social networking check-in data. Transportation Research Part C: Emerging Technologies 79, 136–155.

Hu, W., Yao, Z., Yang, S., Chen, S., Jin, P.J., 2019. Discovering urban travel demands through dynamic zone correlation in location-based social networks, in: Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Springer. pp. 88–104.

Jeong, I.J., Park, D., 2021. Stochastic programming approach for static origin–destination matrix reconstruction problem. Computers & Industrial Engineering 157, 107373.

Jin, P.J., Cebelak, M., Yang, F., Zhang, J., Walton, C.M., Ran, B., 2014. Location-based social networking data: exploration into use of doubly constrained gravity model for origin–destination estimation. Transportation Research Record 2430, 72–82.

Kheiri, A., Karimipour, F., Forghani, M., 2015. Intra-urban movement flow estimation using location based social networking data. The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 40, 781.

Kim, J.S., Jin, H., Kavak, H., Rouly, O.C., Crooks, A., Pfoser, D., Wenk, C., Züfle, A., 2020. Location-based social network data generation based on patterns of life, in: 21st IEEE International Conference on Mobile Data Management, IEEE. pp. 158–167.

Kim, J.S., Kavak, H., Manzoor, U., Crooks, A., Pfoser, D., Wenk, C., Züfle, A., 2019. Simulating urban patterns of life: A geo-social data generation framework, in: Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 576–579.

Viegas de Lima, I., Danaf, M., Akkinepally, A., De Azevedo, C.L., Ben-Akiva, M., 2018. Modeling framework and implementation of activity-and agent-based simulation: an application to the greater boston area. Transportation Research Record 2672, 146–157.

Liu, Z., Wang, A., Weber, K., Chan, E., 2022. Categorisation of cultural tourism attractions by tourist preference using location-based social network data: The case of Central, Hong Kong. Tourism Management 90.

Mahajan, V., Cantelmo, G., Antoniou, C., 2021. Explaining demand patterns during covid-19 using opportunistic data: a case study of the city of munich. European Transport Research Review 13, 1–14.

Mahmassani, H.S., 2001. Dynamic network traffic assignment and simulation methodology for advanced system management applications. Networks and Spatial Economics 1, 267–292.

Martí, P., Serrano-Estrada, L., Nolasco-Cirugeda, A., 2019. Social Media data: Challenges, opportunities and limitations in urban studies. Com-

puters, Environment and Urban Systems 74, 161–174.
McNally, M.G., 2007. The four-step model, in: Handbook of transport modelling. Emerald Group Publishing Limited.
Osorio, C., 2019. High-dimensional offline origin-destination (OD) demand calibration for stochastic traffic simulators of large-scale road networks. Transportation Research Part B: Methodological 124, 18–43.
Pew Research Center, 2023. Social media fact sheet. URL: https://www.pewresearch.org/internet/fact-sheet/social-media/. Accessed on 20.07.2023.
Qurashi, M., Lu, Q.L., Cantelmo, G., Antoniou, C., 2022. Dynamic demand estimation on large scale networks using principal component analysis: The case of non-existent or irrelevant historical estimates. Transportation Research Part C: Emerging Technologies 136, 103504.
Qurashi, M., Ma, T., Chaniotakis, E., Antoniou, C., 2019. PC–SPSA: employing dimensionality reduction to limit spsa search noise in dta model calibration. IEEE Transactions on Intelligent Transportation Systems 21, 1635–1645.
Ren, J., Xie, Q., 2017. Efficient OD trip matrix prediction based on tensor decomposition, in: Proceedings of the 18th IEEE International Conference on Mobile Data Management, IEEE. pp. 180–185.
Rizwan, M., Wan, W., Gwiazdzinski, L., 2020. Visualization, spatiotemporal patterns, and directional analysis of urban activities using geolocation data extracted from LBSN. ISPRS International Journal of Geo-Information 9, 137.
Shafiei, S., Saberi, M., Zockaie, A., Sarvi, M., 2017. Sensitivity-based linear approximation method to estimate time-dependent origin–destination demand in congested networks. Transportation Research Record 2669, 72–79.
Silva, T.H., Viana, A.C., Benevenuto, F., Villas, L., Salles, J., Loureiro, A., Quercia, D., 2019. Urban computing leveraging location-based social network data: a survey. ACM Computing Surveys (CSUR) 52, 1–39.
Tampere, C.M., Viti, F., Immers, L.B., Elgar, E., 2010. New Developments in Transport Planning. volume 15. Edward Elgar Publishing.
Tasse, D., Hong, J.I., 2014. Using social media data to understand cities .
Timokhin, S., Sadrani, M., Antoniou, C., 2020. Predicting venue popularity using crowd-sourced and passive sensor data. Smart Cities 3, 42.
Toledo, T., Kolechkina, T., 2012. Estimation of dynamic origin–destination matrices using linear assignment matrix approximations. IEEE Transactions on Intelligent Transportation Systems 14, 618–626.
Tympakianaki, A., Koutsopoulos, H.N., Jenelius, E., 2015. c-SPSA: Cluster-wise simultaneous perturbation stochastic approximation algorithm and its application to dynamic origin–destination matrix estimation. Transportation Research Part C: Emerging Technologies 55, 231–245.
Xiong, X., Ozbay, K., Jin, L., Feng, C., 2020. Dynamic origin–destination matrix prediction with line graph neural networks and kalman filter. Transportation Research Record 2674, 491–503.
Yang, D., Zhang, D., Zheng, V.W., Yu, Z., 2014a. Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. IEEE Transactions on Systems, Man, and Cybernetics: Systems 45, 129–142.
Yang, F., Jin, P.J., Cheng, Y., Zhang, J., Ran, B., 2015. Origin-destination estimation for non-commuting trips using location-based social networking data. International Journal of Sustainable Transportation 9, 551–564.
Yang, F., Jin, P.J., Wan, X., Li, R., Ran, B., 2014b. Dynamic origin-destination travel demand estimation using location based social networking data, in: Transportation Research Board 93th Annual Meeting.
Zhang, C., Osorio, C., Flötteröd, G., 2017. Efficient calibration techniques for large-scale traffic simulators. Transportation Research Part B: Methodological 97, 214–239.
Zhou, Y., Lau, B.P.L., Yuen, C., Tunçer, B., Wilhelm, E., 2018. Understanding urban human mobility through crowdsensed data. IEEE Communications Magazine 56, 52–59.